Physics-Enhanced Machine Learning for Chemical Kinetics

Physikalische Maschinenlernverfahren für die Modellierung chemischer Kinetik
Zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.)
Genehmigte Dissertation von Felix Antonidas Döppel aus Bensheim
Tag der Einreichung: 26. März 2024, Tag der Prüfung: 13. Mai 2024

1. Gutachten: Prof. Dr. Martin Votsmeier
2. Gutachten: Prof. Dr. Oliver Weeger
3. Gutachten: Prof. Dr. Olaf Deutschmann

Darmstadt, Technische Universität Darmstadt

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Chemistry Department
Ernst-Berl-Institute
Votsmeier Group

Physics-Enhanced Machine Learning for Chemical Kinetics
Physikalische Maschinenlernverfahren für die Modellierung chemischer Kinetik

Accepted doctoral thesis by Felix Antonidas Döppel

Date of submission: 26. März 2024
Date of thesis defense: 13. Mai 2024

Darmstadt, Technische Universität Darmstadt

Publications

Parts of this work have been previously published or presented at international conferences:

Research articles contained within this work

1. F. A. Döppel and M. Votsmeier
Efficient machine learning based surrogate models for surface kinetics by approximating the rates of the rate-determining steps.
Chemical Engineering Science, **2022**, *262*, 117964.
DOI: 10.1016/j.ces.2022.117964

2. F. A. Döppel and M. Votsmeier
Efficient Neural Network Models of Chemical Kinetics Using a Latent asinh Rate Transformation.
Reaction Chemistry & Engineering, **2023**, *8*, 2620-2631.
DOI: 10.1039/D3RE00212H

3. F. A. Döppel, T. Wenzel, R. Herkert, B. Haasdonk and M. Votsmeier
Goal-Oriented Two-Layered Kernel Models as Automated Surrogates for Surface Kinetics in Reactor Simulations.
Chemie Ingenieur Technik, **2024**, *96*, No. 5, 1-11.
DOI: 10.1002/cite.202300178

4. F. A. Döppel and M. Votsmeier
Robust Mechanism Discovery with Atom Conserving Chemical Reaction Neural Networks.
ChemRxiv, **2023**.
DOI: 10.26434/chemrxiv-2023-1r389

Research articles not contained within this work

5. T. Kircher, F. A. Döppel and M. Votsmeier
Global reaction neural networks with embedded stoichiometry and thermodynamics for learning kinetics from reactor data.
Chemical Engineering Journal, **2024**, *485*, 149863.
DOI: 10.1016/j.cej.2024.149863

6. T. Kircher, F. A. Döppel and M. Votsmeier
Embedding Physics into Neural ODEs to learn Kinetics from Integral Reactors.
Computer Aided Chemical Engineering, **2024**, accepted.
DOI: 10.26434/chemrxiv-2024-10xzj-v2

Talks

1. F. A. Döppel and M. Votsmeier
   Representing detailed surface kinetics by neural networks
   26th International Symposium on Chemical Reaction Engineering and the 9th Asia-Pacific Chemical Reaction Engineering Symposium. 05. - 08. December 2021; New Delhi, India - Online event

2. K. Wilhelm, F. A. Döppel and M. Votsmeier
   Physics-Informed Neural Networks for Reactor Simulations
   Annual Meeting on Reaction Engineering 2023. 15. - 17. May 2023; Frankfurt am Main, Germany

3. F. A. Döppel and M. Votsmeier
   Efficient Implementation of Detailed Surface Kinetics by Neural Network Representations of the Rate-determining Step
   27th International Symposium for Chemical Reaction Engineering. 11. - 14. June 2023; Quebec City, Canada

4. F. A. Döppel, T. Kircher and M. Votsmeier
   Physics-Embedded Neural Networks for Chemical Kinetics
   28th North American Catalysis Society Meeting. 18. - 23. June 2023; Providence, USA

5. F. A. Döppel and M. Votsmeier
   Efficient Implementation of Detailed Surface Kinetics by Neural Network Representations of the Rate-determining Step
   GDCh Science Forum Chemistry 2023. 04. - 06. September 2023; Leipzig, Germany

6. F. A. Döppel, T. Kircher and M. Votsmeier
   Neural Networks for Detailed Surface Kinetics
   14th European Congress of Chemical Engineering and 7th European Congress of Applied Biotechnology. 17. - 21. September 2023; Berlin, Germany

Posters

1. F. A. Döppel and M. Votsmeier
   Representing detailed surface kinetics by neural networks
   Annual Meeting on Reaction Engineering and ProcessNet Subject Division Heat & Mass Transfer 2022. 18. - 20. July 2022; Würzburg, Germany

2. F. A. Döppel and M. Votsmeier
   Representing detailed surface kinetics by neural networks
   7th International Symposium on Modeling of Exhaust-Gas After-Treatment. 11. - 13. September 2022; Bad Herrenalb/Karlsruhe, Germany

3. F. A. Döppel and M. Votsmeier
   Efficient Implementation of Detailed Surface Kinetics by Neural Network Representations of the Rate-determining Steps
   Annual Meeting on Reaction Engineering 2023. 15. - 17. May 2023; Frankfurt am Main, Germany

4. F. A. Döppel, T. Wenzel, R. Herkert, B. Haasdonk and M. Votsmeier
   Two-layered Vectorial Kernel Models for Detailed Surface Kinetics using a Goal-Oriented Approach
   Annual Meeting on Reaction Engineering 2023. 15. - 17. May 2023; Frankfurt am Main, Germany

5. T. Kircher, F. A. Döppel and M. Votsmeier
Learning Kinetic Models from Experimental Data using Neural ODEs
Annual Meeting on Reaction Engineering 2023. 15. - 17. May 2023; Frankfurt am Main, Germany

6. T. Kircher, F. A. Döppel and M. Votsmeier
Learning Kinetic Models Directly from Experimental Data by Physics Embedded Neural ODEs
14th European Congress of Chemical Engineering and 7th European Congress of Applied Biotechnology. 17. - 21. September 2023; Berlin, Germany

Abstract

The energy transition and the transformation of the chemical industry are major efforts in addressing the challenges of climate change. Both require the development of new and optimized catalytic devices. The rational design of these devices depends on a thorough understanding of the underlying chemical kinetics. As kinetic model development gets outpaced by the ever-increasing availability of kinetic data, there is a growing demand for automated mechanism discovery. But even when detailed kinetic models are available, their use for the knowledge-based design of catalytic devices is limited by their high computational cost. The objective of this work is to design physically plausible machine learning models that facilitate the discovery of chemical reaction mechanisms and enable their efficient implementation in simulations of catalytic devices.

The cost of reactive flow simulations is often reduced by replacing the computationally intensive evaluation of chemical kinetics with a numerically less demanding, so-called surrogate model. However, surrogates have not yet been systematically applied to surface kinetics of industrially relevant complexity because they rely on logarithmic data normalization. The logarithm, however, is incompatible with systems that contain intermediate species or operate close to the chemical equilibrium, as the modelled source terms change sign and thereby leave the domain of the logarithm. This work provides two methods that extend the scope of surface kinetic surrogate modelling: 1. Modelling the rates of the rate-determining steps instead of species source terms re-enables logarithmic normalization because the considered elementary rates are strictly positive. 2. Replacing the logarithm with specialized logarithm-like functions extends the domain of allowed source terms to positive as well as negative numbers. Further, this work introduces neural networks to surface kinetic modelling and demonstrates their superiority over splines, the former state of the art, in terms of accuracy, prediction time, and the required amount of storage space. The proposed latent data transformation technique makes use of the high structural flexibility of neural network models to embed the principles of atom conservation, the Arrhenius law, and the law of mass action directly into the model. The resulting surrogates accurately predict the chemical kinetics of industrially relevant systems, as exemplarily shown for the preferential oxidation of CO, which is relevant to hydrogen production for fuel cell applications, and the ammonia oxidation under industrially relevant conditions of the Ostwald process. Reactor simulations of these systems are accelerated by four to five orders of magnitude.

Increasingly detailed kinetics, such as first principles kinetic Monte Carlo simulations, are becoming the gold standard in chemical engineering. Their solution is computationally so expensive that even the collection of a sufficient amount of data to train a surrogate model becomes infeasible. This work proposes a grid-free training set design scheme that evaluates only those data that significantly contribute to the accuracy of the surrogate. Applied to the preferential oxidation of CO, this leads to a 14-fold reduction in the amount of required training data. Uncertainty estimation is performed by two-layered kernel models and further employed to track the accuracy of surrogates during their use in reactor simulations. This allows to refine the model on-the-fly and thereby ensures reliable simulation results, even outside the original training range of the model.

The recently developed chemical reaction neural network is a digital twin of the classic microkinetic mechanism that has found widespread application in many fields, such as (bio-)chemical engineering and combustion. While it encodes some fundamental physical laws, mass and atom conservation are still violated. Here, atom conservation is enforced by adding a dedicated neural network layer which can be interpreted as constraining the model to physically realizable stoichiometries. As the resulting models are physically consistent, they are more robust to limited data availability, noisy data, and systematic measurement errors.

Overall, this work improves the physical interpretability and extrapolation capabilities of machine learning models for chemical kinetics. In particular, it presents physics-enhanced neural network architectures, that contain the fundamental physical laws of atom conservation and thermodynamics as well as the more specific Arrhenius law and the law of mass action. Together with accurate uncertainty quantification, this yields fast and reliable implementations of chemical kinetics into reactive flow simulations, allowing their systematic evaluation for the knowledge-based design of catalytic devices. Further, these models perform automated mechanism discovery, pushing the boundary of kinetic insights.

Zusammenfassung

Die Energiewende und die Transformation der chemischen Industrie sind zentrale Maßnahmen, um den Herausforderungen von Umweltzerstörung und Klimawandel entgegenzutreten. Beide erfordern jedoch die Entwicklung neuer und optimierter katalytischer Reaktoren. Deren Auslegung verlangt wiederum ein weitreichendes Verständnis der zugrunde liegenden chemischen Kinetik. Da die Entwicklung kinetischer Modelle nicht mit der zunehmenden Verfügbarkeit kinetischer Daten Schritt halten kann, steigt nun der Bedarf an automatisierten Methoden zur Bestimmung chemischer Reaktionsmechanismen. Doch selbst wenn detaillierte kinetische Modelle verfügbar sind, verhindert deren hoher Rechenbedarf ihre Nutzung zum rationalen Design katalytischer Reaktoren. Ziel der vorliegenden Arbeit ist es, physikalisch plausible Maschinenlernverfahren zu entwickeln, die die Entdeckung chemischer Reaktionsmechanismen erleichtern und deren effiziente Implementierung in Simulationen chemischer Reaktoren ermöglichen.

Der Rechenbedarf solcher Simulationen wird häufig dadurch reduziert, dass die physikalischen Modelle zur Bestimmung von Reaktionsraten durch numerisch weniger anspruchsvolle, sogenannte Ersatzmodelle ausgetauscht werden. Im Fall heterogen katalysierter Reaktionen gilt es, die Modelle der Oberflächenreaktionen zu ersetzen. Das gelingt bisher allerdings nur in einfachen Fällen, da die nötigen Ersatzmodelle ausschließlich in Verbindung mit einer logarithmische Datennormalisierung funktionieren. Der Logarithmus ist jedoch nur für positive Werte definiert. Somit können Spezies mit Quelltermen wechselnden Vorzeichens nicht modelliert werden. Das betrifft insbesondere Systeme, in denen Zwischenprodukte eine Rolle spielen und solche, die in der Nähe des chemischen Gleichgewichts betrieben werden. Die vorliegende Arbeit bietet zwei Methoden, um Ersatzmodelle dennoch auf Oberflächenreaktionen anzuwenden. 1. Werden die Raten des geschwindigkeitsbestimmenden Schritts anstelle der Nettobildungsraten modelliert, lässt sich die logarithmische Normierung wie gewohnt anwenden, da Elementarraten stets positive Werte annehmen. 2. Alternativ kann der Logarithmus durch spezielle, nah verwandte Funktionen ersetzt werden, deren Definitionsbereich alle reellen Zahlen umfasst und daher auch negative Quellterme zulässt. Des Weiteren werden in dieser Arbeit erstmals künstliche neuronale Netze zur Modellierung von Oberflächenraten eingesetzt. Diese sind dem bisherigen Stand der Forschung, den sogenannten Splines, in Bezug auf Genauigkeit und Geschwindigkeit von Vorhersagen sowie Speicherbedarf weit überlegen. Die vorgestellte Methode der latenten Datentransformation nutzt die Flexibilität künstlicher neuronaler Netze, um die physikalischen Prinzipien von Massenerhaltung, Arrheniusgesetz und Massenwirkungsgesetz direkt in der Modellstruktur zu codieren. Die resultierenden Ersatzmodelle geben präzise Schätzungen der Reaktionsraten von industriell relevanten Systemen. Exemplarisch wird das anhand der bevorzugten Oxidation von CO in wasserstoffhaltigen Gasen demonstriert. Dieser Prozess findet Anwendung, um Wasserstoff aus der Dampfreformierung für die Nutzung in Brennstoffzellen aufzubereiten. Als weiteres Beispiel wird die Ammoniakoxidation im Ostwaldverfahren unter industriell relevanten Bedingungen herangezogen. Reaktorsimulationen beider Systeme werden um vier bis fünf Größenordnungen beschleunigt.

In der chemischen Verfahrenstechnik ist ein Trend hin zu immer detaillierteren Kinetiken zu beobachten. Mittlerweile haben sich dort kinetische Monte-Carlo Simulationen als Goldstandard zur Beschreibung heterogen katalysierter Systeme etabliert. Diese sind jedoch so rechenintensiv, dass oft nicht ausreichend viele Daten zum Training von Ersatzmodellen angesammelt werden können. In der vorliegenden Arbeit wird ein Ansatz zum rasterfreien Trainingsset-Design entwickelt. Nach diesem werden Trainingsdaten nur dann ermittelt, wenn sie signifikant zur Genauigkeit des Ersatzmodells beitragen. Das geschieht mithilfe zweischichtiger Kernmodelle, die es erlauben, die Modellgenauigkeit zu schätzen. Infolgedessen werden die Menge benötigter Trainingsdaten und der damit verbundene Rechenaufwand für die präferentielle Oxidation von CO um Faktor 14 reduziert. Zudem erlaubt das, Ersatzmodelle noch während Reaktorsimulationen ständig zu verbessern, sodass sie bei Bedarf auch außerhalb des ursprünglichen Trainingsbereichs verlässliche Ergebnisse liefern.

Das kürzlich entwickelte *Chemical Reaction Neural Network* ist ein digitaler Zwilling des klassischen mikrokinetischen Mechanismus und hat bereits in vielen Bereichen Anwendung gefunden. Obwohl es einige grundlegende physikalische Gesetze codiert, wird die Atombilanz noch immer verletzt. Die vorliegende Arbeit sichert den Atomerhalt durch Hinzufügen einer eigens dafür ausgelegten Schicht im künstlichen neuronalen Netz, welche das Modell auf physikalisch realisierbare Kombinationen stöchiometrischer Koeffizienten beschränkt. Da die resultierenden Modelle physikalisch konsistent sind, sind sie robuster gegenüber begrenzter Datenverfügbarkeit, Messabweichungen und systematischen Messfehlern.

Insgesamt verbessert die vorliegende Arbeit die physikalische Plausibilität sowie die Verlässlichkeit von Maschinenlernverfahren im Bereich der chemischen Kinetik. Insbesondere werden künstliche neuronale Netze strukturell so angepasst, dass sie grundlegende physikalische Gesetze wie Atomerhaltung und Thermodynamik beachten. In Verbindung mit akkuraten Fehlerschätzern lässt sich chemische Kinetik somit zuverlässig und numerisch effizient in Reaktorsimulationen implementieren. Das erlaubt schlussendlich ein rationales Design katalytischer Reaktoren in vertretbarer Zeit. Außerdem automatisierte solche Modelle die Entdeckung chemischer Reaktionsmechanismen und tragen damit zum Verständnis bisher unbekannter Reaktionspfade bei.

Contents

1 Introduction

Catalytic reactions have long played a central role in our daily lives: They are used to clean the air of harmful pollutants arising from combustion engines, ensure food security by providing fertilizers, and produce plastics. Now, climate change calls for new and optimized concepts of catalytic devices that enable the energy transition and the transformation of the chemical industry, e.g. by reducing CO_2 emissions [1] and producing chemicals from renewable energies [2]. Model-based engineering lays the foundation for a rational design of these devices [1]. However, they show highly complex behaviours dominated by the intricate coupling of physical processes occurring on multiple time and length scales (Figure 1.1). The multi-scale models required to describe this behaviour are computationally expensive, and their accuracy is highly dependent on a thorough understanding of the underlying chemical kinetics [1, 3–5].

Recently, machine learning led to major breakthroughs in many fields such as the prediction of protein folding [6], weather forecasting [7], computer vision [8–10], and natural language processing [11, 12]. Now, it presents unique opportunities for chemical engineering but also a set of new challenges [13]:

1. The training of accurate machine learning models requires huge amounts of high-quality data. In contrast to classical big data applications, relevant experimental data are much more sparse and expensive.

2. There is low trust in machine learning models because they lack explainability and have limited extrapolation capabilities. To be used in chemical engineering, they have to fulfil high safety standards.

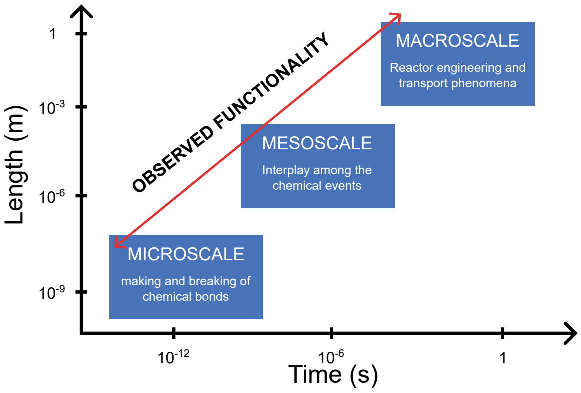


Figure 1.1: Schematic representation of the time and length scales involved in chemical processes. Adapted from [4].

Hybrid modelling is the combination of data-driven models and physical knowledge and is widely seen as a promising approach to ensuring interpretability, extrapolation capability, reliability, and trust of machine learning models [5, 13, 14]. The most famous and widely applied example is physics-informed neural networks, which incorporate prior knowledge as a penalty term during training to encourage physically consistent solutions [15, 16]. Even more reliable models can be achieved by hard constraints, i.e. embedding physical laws directly into the model structure. In this case, physically plausible solutions are guaranteed.

This work develops such physics-enhanced machine learning models in the context of chemical kinetics. Specifically, it presents tailored neural network structures to build numerically efficient surrogates [17, 18], which are urgently needed to enable multi-scale simulations on an industrially relevant scale [1, 3–5]. Model reliability and their extrapolation capability are addressed by uncertainty quantification during the whole process from data generation, to model training and their use in reactive flow simulations [19]. Finally, the physical plausibility of the state-of-the-art machine learning tool for automated mechanism discovery [20] is enhanced by enforcing the fundamental law of mass and atom conservation [21].

2 Theoretical background

2.1 Modeling Reactive Systems

This section describes the basics of modelling surface reactive systems that are required to follow the results presented in this work.

2.1.1 Kinetic Models

This section describes the basics of surface kinetic modelling and is mainly based on the textbooks of Wedler and Freund [22] as well as Jess and Wasserscheid [23]. Reaction kinetics describe how the rate r of a reaction depends on the reaction conditions, which are usually given by the temperature T, species activities a_i, as well as the type and concentration of a catalyst, if present. Activities are typically approximated with the concentration c_i in the liquid phase, and partial pressures p_i in the gas phase. Reaction equations are written in the form

$$|\nu_{\mathrm{A}}|\mathrm{A} + |\nu_{\mathrm{B}}|\mathrm{B} \longrightarrow |\nu_{\mathrm{C}}|\mathrm{C} + |\nu_{\mathrm{D}}|\mathrm{D} \tag{2.1}$$

where the stoichiometric coefficients ν_i describe how many molecules of a species i are involved in the reaction. By convention, ν is negative for reactants (left side of Eq. 2.1) and positive for products (right side of Eq. 2.1).

A reaction is called elementary reaction if it is performed within a single step. In this case, the reaction rate is given by the law of mass action (Eq. 2.2). The temperature dependence of the reaction constant k is given by the Arrhenius law (Eq. 2.3) with the pre-exponential factor k_0, the activation energy E_A, and the universal gas constant R.

$$r = k \cdot \prod_i a_i^{\nu_i} \tag{2.2}$$

$$k = k_0 \cdot \exp\left(-\frac{E_A}{R \cdot T}\right) \tag{2.3}$$

All chemical reactions are reversible. The De Donder relation [24] expresses the effective rate in terms of the forward rate $\vec{r}$, activities and thermodynamics

$$r = \vec{r} \cdot \left(1 - \frac{Q}{K}\right) \tag{2.4}$$

where

$$Q = \prod_i a_i^{\nu_i} \tag{2.5}$$

and

$$K = \exp\left(\frac{-\Delta_\mathrm{R} G}{RT}\right) \tag{2.6}$$

with the Gibbs free enthalpy $\Delta_\mathrm{R} G$ of the reaction. If $-\Delta_\mathrm{R} G \gg RT$ the reverse reaction is usually neglected and the reaction is assumed to be irreversible. In chemical equilibrium $Q = K$ and the effective rate is 0.

The source term $\dot{s}_i$ describes the concentration change of species i over time t caused by the sum of all chemical reactions j occurring in the system.

$$\dot{s}_i = \frac{\mathrm{d}c_i}{\mathrm{d}t} = \sum_j \nu_{i,j} \cdot r_j \tag{2.7}$$

In some cases, this differential equation can be solved analytically to express the concentration as a function of time. For example, for an irreversible elementary reaction

$$\mathrm{A} \longrightarrow \mathrm{B} \tag{2.8}$$

follows

$$\frac{\mathrm{d}c_\mathrm{A}}{\mathrm{d}t} = -k \cdot c_\mathrm{A}, \tag{2.9}$$

which can be solved to

$$c_\mathrm{A}(t) = c_\mathrm{A}(t=0) \cdot e^{-kt}. \tag{2.10}$$

In most cases, however, an analytical solution does not exist and equation 2.7 has to be integrated numerically.

Most industrially relevant processes involve solid catalysts [25]. In this case, reactions occur at the catalyst surface. This process generally involves five steps

1. Diffusion of the reactants to the surface
2. Adsorption on the surface
3. Reactions on the surface
4. Desorption of the products
5. Diffusion of the products into the bulk phase

Due to steps 1. and 5. the local concentration above the surface is different from the concentration measured in the bulk phase. Steps 2. and 4. are included into the overall reaction network. For example, a simple surface catalysed reaction

$$\mathrm{A} \xrightarrow{\text{cat.}} \mathrm{B}$$

involves at least the following three reactions

$$\mathrm{A} + {}^* \longrightarrow \mathrm{A}^* \tag{2.11}$$

$$\mathrm{A}^* \longrightarrow \mathrm{B}^* \tag{2.12}$$

$$\mathrm{B}^* \longrightarrow \mathrm{B} + {}^* \tag{2.13}$$

where * denotes the empty adsorption site and A*, B* are the corresponding adsorbed species. Commonly, mean-field kinetics are assumed and the activity of surface species is expressed with the surface coverage ϑ_i, which amounts to the fraction of surface sites occupied by species i[1].

More advanced descriptions of surface reaction rates acknowledge that the reactivity of a surface species depends on its local environment. First principles kinetic Monte Carlo simulations (Fig. 2.1) are becoming the gold standard to provide accurate insights into reactive surfaces, but they are computationally much more demanding than mean-field kinetics [26].

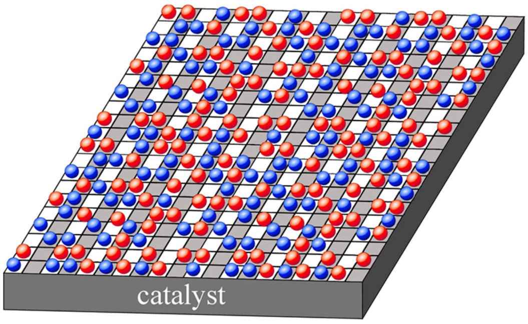


Figure 2.1: Prototypical surface configuration for the kinetic Monte Carlo simulation of CO oxidation on $RuO_2(110)$. White and grey squares denote different adsorption sites. Adsorbed oxygen atoms are shown as red balls and adsorbed CO molecules as blue balls. Adapted with permission from [27]. Copyright 2019 American Chemical Society.

The time scales of surface reactions are commonly much shorter than those of gas phase processes. Therefore, in most systems, steady state of the surface coverages can be assumed, even for transient descriptions of the gas phase [28]. To solve the surface for steady state, coverages are numerically integrated over time until their derivatives are sufficiently close to zero (Eq. 2.14).

$$\frac{\mathrm{d}\vartheta_i}{\mathrm{d}t} = 0 \tag{2.14}$$

This amounts to solving a system of differential algebraic equations (DAEs) which can be computationally

[1] In this case, the total concentration of active sites is included within the pre-exponential factor

very demanding because DAEs are a form of infinite stiffness [29, 30]. Indeed, solving for detailed surface kinetics is frequently reported to be the computationally most demanding step of high-fidelity reactor simulations [4].

2.1.2 Reactor Models

This section describes the basics of reactor modelling and is mainly based on the textbook of Baerns *et al.* [31]. The key part of every chemical plant is the reactor, an enclosed vessel where the chemical reactions take place. Modelling the reaction progress in a reactor requires equations for the rate of the reaction as well as the transport of mass and heat. Here, the idealized case of isothermal reactors is considered, i.e. assuming a uniform temperature which is constant over time and space. The material balance can generally be expressed as a differential equation considering the accumulation of substance i in a control volume V by balancing the incoming and outgoing flows as well as the generation or consumption inside the control volume by reactions (Eq. 2.15).

$$\underbrace{\frac{\mathrm{d}\dot{n}_i}{\mathrm{d}t}}_{\text{Accumulation}} = \underbrace{\dot{n}_{i,\text{in}}}_{\text{Inward flow}} - \underbrace{\dot{n}_{i,\text{out}}}_{\text{Outward flow}} + \underbrace{V \cdot \dot{s}}_{\text{Generation or consumption}} \tag{2.15}$$

Frequently, a set of key assumptions is used to provide simplified descriptions of three common reactor types, namely batch reactor, plug-flow-reactor (PFR), and continuously stirred tank reactor (CSTR). By assuming ideal mixing within the control volume, the amount of substance n can be expressed in terms of the concentration c

$$n = c \cdot V. \tag{2.16}$$

Similarly, molar fluxes $\dot{n}$ can be expressed in terms of the volumetric flow rate $\dot{V}$

$$\dot{n} = c \cdot \dot{V}. \tag{2.17}$$

Further, the concentration of the outgoing flow is identical to the concentration within the control volume, yielding the alternative description of the material balance

$$V \cdot \frac{\mathrm{d}c_i}{\mathrm{d}t} = \dot{V}_{\text{in}} \cdot c_{i,\text{in}} - \dot{V}_{\text{out}} \cdot c_i + V \cdot \dot{s}. \tag{2.18}$$

In the batch reactor there are no incoming or outgoing flows, leading to the simplified material balance

$$\frac{\mathrm{d}c_i}{\mathrm{d}t} = \dot{s}. \tag{2.19}$$

The CSTR is operated in steady state, so there is no accumulation and incoming and outgoing volume flows are identical. Introducing the average residence time

$$\tau = \frac{\dot{V}}{V} \tag{2.20}$$

leads to the analytical expression of the CSTR material balance

$$0 = \frac{c_{i,\mathrm{in}} - c_i}{\tau} + \dot{s}. \tag{2.21}$$

In the stationary PFR, ideal mixing is assumed within an infinitesimally small control volume at each axial position z. There are no radial concentration gradients and dispersion leading to mixing between the control volumes is neglected. Volume (Eq. 2.22) and volumetric flow rate (Eq. 2.23) are expressed in terms of the cross sectional area A, the width of the control volume dz and the velocity u to obtain the PFR material balance (Eq. 2.24) [31].

$$V = A \cdot \mathrm{d}z \tag{2.22}$$

$$\dot{V} = A \cdot u \tag{2.23}$$

$$0 = -\frac{\mathrm{d}\left(c_i \cdot u\right)}{\mathrm{d}z} + \dot{s} \tag{2.24}$$

In this work, PFR models are used to validate the accuracy of surrogate models of steady state surface kinetics, and batch models are used to perform automated mechanism discovery from simulated reactor outlet data. For more advanced models, including the description of non-isothermal reactors considering dispersion, the reader is referred to elsewhere [23, 31]. Overall, the most detailed description of catalytic devices is obtained through multi-scale models that spatially resolve the exact reactor geometry and consider all relevant time and length scales [3, 4].

2.2 Machine Learning

> "*[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.*"
>
> - Arthur Samuel, 1959[32]

In other words, machine learning is the process in which a machine improves its ability in a defined task by evaluating data instead of following prescribed (behavioural) rules in the form of code [33]. It is a very lively field of research. Alone for the year 2023 *Google*'s search engine for literature research *Google Scholar* shows more than $300\,000$ hits for the keyword "machine learning" [34]. Machine learning applications, such as spam filters [35] and chatbots, are increasingly prevalent in everyday life. They may also have a considerable social impact, particularly in medical diagnosis [36, 37], drug discovery [38], or the analysis of satellite data, for example in response to natural disasters [39]. Further, there are

- numerous software libraries that are specialized for machine learning, including Pytorch[40], Tensorflow[41] and JAX[42] in Python, and others in MATLAB[43], R[44] and Julia[45],

- internet forums to help understand machine learning [46],

- YouTube channels with more than a million subscribers, presenting scientific advances on a weekly basis [47], and

- comprehensive textbooks on this topic [33, 48–51].

This work therefore only presents the basics relevant for the results section.

2.2.1 Neural Networks

Artificial neural networks (hereinafter referred to as neural networks or NNs for short) are biologically inspired systems that were described in their simplest form as early as 1943 [52]. While research into these systems has ebbed and flowed, they now benefit from the large amount of data available, the massive increase in computing power over the last few decades, and the ability to parallelize computations, for example through the use of graphics cards [33].

Depending on the specific use case, NNs often have a characteristic architecture, such as Generative Adversarial Networks for image generation [53], Graph Neural Networks for weather forecasting [7], or Transformers for natural language processing [11]. The following section describes the *feedforward neural network* architecture (Fig. 2.2) commonly used for regression.

The basic building blocks of a neural network are called neurons. Input neurons (blue) feed the input values, commonly called *features*, into the network in the form of an input vector **x**. Their value is passed on to the neurons of the first *hidden layer* (green) via the connections (black lines). The section on the top right shows what happens within such a neuron. It computes the scalar product of the incoming values and its weight vector **w** and the *bias* b is added before an *activation function* σ is applied, for example, the sigmoidal function shown in Figure 2.2. Overall, the result h_i of the mathematical operations in neuron i can be described as

$$h_i = \sigma_i \left(\mathbf{x} \cdot \mathbf{w}_i + b_i \right) \tag{2.25}$$

and is passed to every neuron in the following layer until the output layer (orange) of the NN is reached. The output neurons usually have a different activation function, for example, the identity $f(x) = x$, to avoid limiting the value range of the NN prediction $h\left(\mathbf{x}\right)$. The term feedforward neural network derives from the unidirectional flow of signals from input to output [33].

The high potential of NNs for regression lies in their ability to approximate any multivariate function with arbitrary precision but comes at the expense of the computational complexity in the training and prediction phase [54]. It requires a non-linear activation function in the hidden layer(s) and either an arbitrary number of neurons in a single *layer* or an arbitrary number of *layers* with a fixed number of neurons [55, 56].

Training a neural network means searching for a combination of weights and biases that offers the best results in approximating a target function. The quality of the approximation is described by the loss

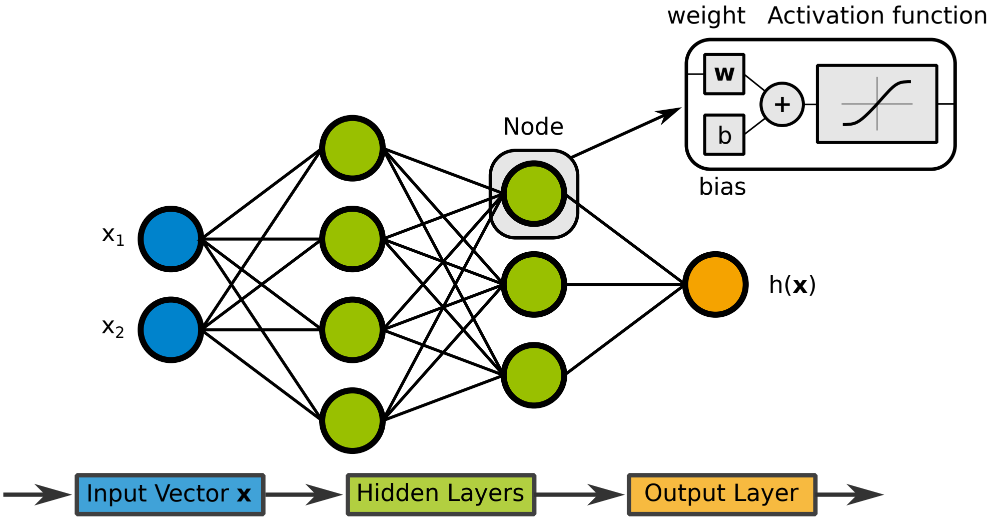


Figure 2.2: Schematic illustration of a feedforward neural network. Here, the input vector $\mathbf{x}$ has two components: the features x_1 and x_2. Their numeric value is propagated to all four nodes in the first hidden layer. Each hidden node computes a weighted sum of all its input values by multiplication with the weight vector $\mathbf{w}$ before adding the bias b. The result of this operation yields a scalar value, which is passed through to a non-linear activation function, e.g. the sigmoid function shown in the top right corner. The resulting value is passed on to all nodes of the next layer until the output layer is reached. Here, the neural network output h is provided, which can be a vector or a scalar. Overall, the output is a function of the input vector $\mathbf{x}$ and additionally depends on all weight vectors, all bias values, the choice of activation function, and the neural network architecture, i.e. the number of nodes and hidden layers.

function, usually the mean square error

$$\mathrm{MSE}(\mathbf{X}, h) = \frac{1}{m} \sum_{i=1}^{m} \left(h\left(\mathbf{x}^{(i)}\right) - \mathbf{y}^{(i)} \right)^2 \tag{2.26}$$

with the data matrix $\mathbf{X}$ containing all m instances of feature $\mathbf{x}^{(i)}$ and target $\mathbf{y}^{(i)}$ values, and the predictive model function h. The *backpropagation* algorithm [57] computes the derivative of the loss function with respect to the model parameters using analytical derivatives and the chain rule. First [58] or second [59] order optimizers use this information to adjust the model parameters. The initial weight and bias values are randomly generated to prevent symmetry [33], resulting in different solutions even when utilizing the same data and training algorithm. Therefore, this work trains multiple models using the same set of *hyperparameters*[2] and reports only the average performance.

A common problem in machine learning is *overfitting*. It occurs when there is insufficient data available

[2] Hyperparameters are variables that a neural network cannot learn, as they are determined before the learning process. Nevertheless, they influence the prediction of the network. Examples of important hyperparameters are architecture, activation function, choice of the optimizer, and scaling of the feature- and target values.

for a model or when too many parameters are used to describe a dataset. If there are too many degrees of freedom available, the model may learn characteristics of the data set that do not accurately represent the measured variable, such as (systematic) measurement errors. This principle is best illustrated by fitting a data set with conventional polynomials. Figure 2.3 shows measurements that are scattered around a linear function. The black curve shows the ground truth, i.e. the true values without measurement errors. A high-order polynomial fit to the data is represented by the blue curve. This polynomial has significantly more degrees of freedom than necessary. Consequently, the model is heavily influenced by the measurement errors and does not generalise well to unseen conditions.

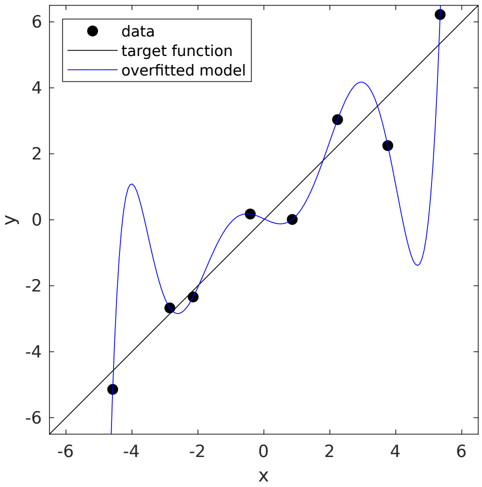


Figure 2.3: Noisy data is fitted with a high-order polynomial. Although all known data points are predicted with zero deviation, the model does not generalize well to unseen conditions. It is said to be overfitting. A linear fit to the data has higher prediction errors but provides more reasonable predictions for unseen conditions in this range.

Overfitting can be avoided by assessing the generalization capabilities of a model using a separate validation dataset that was not used during training. If the model's performance on the validation set is noticeably worse than on the training set, or if it decreases during training, the model is considered to be overfitting. The training process is usually stopped at this point. One way to actively avoid overfitting is regularization. One example is extending the loss function to penalize excessively high values in weights and biases. Another approach is to constrain the model to physically plausible solutions.

2.2.2 Physics-informed Machine Learning

Established machine learning methods like neural networks are considered black-box models. While they have proven useful in many scientific fields such as heterogeneous catalysis [60, 61], they lack interpretability, and have limited extrapolation capabilities [13]. Therefore, they cannot be fully relied upon in high-stakes settings such as healthcare [62, 63]. Additionally, their performance critically relies on the amount and quality of training data. In natural sciences and engineering, however, data is usually

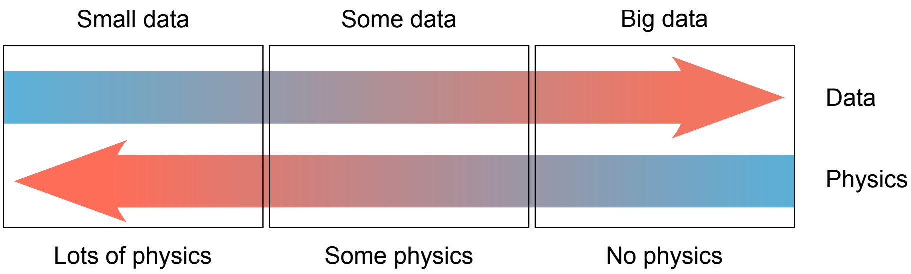


Figure 2.4: If a system is physically well understood, its behaviour can be accurately predicted, even if only small amounts of data are available. Typical machine learning methods work well for big data problems, where the lack of physical understanding is compensated by data availability. Physics-informed machine learning combines both approaches to describe systems with limited data availability by using physical insights. Adapted with permission from [16].

not readily available [64]. Nonetheless, physical systems have been observed and modelled for centuries with limited amounts of data, and resulting knowledge has been compiled into widely accessible equations. So, when machine learning is used to describe physical systems, it almost seems wasteful to disregard the physical understanding and exclusively rely on data-driven methods. Hybrid modelling is the combination of data-driven models and physical knowledge and is widely seen as a promising approach to ensure interpretability, extrapolation, reliability, and trust of machine learning models [5, 13, 14]. It operates on problems where limited data availability can be made up for by physical insights (Fig. 2.4). The following list contains the five stages of machine learning [65], and gives examples of enriching them with physical information.

1. **Formulating a problem to model**
   Rather than replacing the whole process of physical modelling with machine learning, it is often beneficial to replace only small parts of it. Typically the computationally most expensive parts are replaced, and the overall framework of the model is maintained. For example, it is an established method to replace quantum chemical calculations in molecular dynamics simulations with machine learning forcefields [66].

2. **Curating data**
   Collecting data from physical models or experiments already exerts a (weak) physical bias to the model. Further, data augmentation is routinely used to bias models towards symmetry [67].

3. **Choosing an architecture**
   The most reliable way to enforce physically plausible solutions is to embed physical knowledge directly into the model structure: Convolutional neural network respect symmetries found in natural images [68], Lagrangian neural networks conserve energy [69, 70], and modern turbulence models respect Galilean invariance [71]. Depending on the exact context, this family of models is known as *physic-augmented* [72], *physic-constrained* [73], or *physic-enhanced* [74] neural networks.

4. **Designing a loss function**
   The recently developed *physics-informed* neural networks [15] penalize deviations from physical laws by adding a corresponding term to the loss function. This penalty is considered a soft constraint, i.e. it encourages physically plausible solutions but does not guarantee them.

5. **Selecting an optimization algorithm**
   Methods like constrained least squares constrain the space of possible solutions and can be used to avoid physically implausible behaviour.

More details and examples of physics-informed machine learning can be found in recent reviews [16, 75–77].

2.2.3 Neural ordinary differential equations

Neural ordinary differential equations (Neural ODEs) are a recently proposed family of deep neural networks which parametrizes derivatives [78]. Here, the concept will be explained briefly using chemical kinetics as an example. Suppose the reactant concentration c_{in} of a chemical species in the feed of a reactor is known and the outlet concentration of this species c_{out} shall be predicted. The simplest way to approach this task is to train a model, e.g. a standard feedforward neural network f_{NN}, that predicts the outlet as a function of the feed concentration (Fig. 2.5a):

$$c_{\text{out}} = f_{\text{NN}}(c_{\text{in}}) \tag{2.27}$$

Alternatively, a recurrent neural network f_{RNN} can be used to predict only the concentration change Δc within the reactor (Fig. 2.5b):

$$c_{\text{out}} = c_{\text{in}} + \underbrace{f_{\text{RNN}}(c_{\text{in}})}_{\Delta c} \tag{2.28}$$

It has been found, that accuracy is improved when multiple incremental changes are predicted along the reactor coordinate z (Fig. 2.5c) [79]:

$$c(z_{i+1}) = c(z_i) + \underbrace{f_{\text{dRNN}}(c(z_i))}_{\Delta c_i} \tag{2.29}$$

This corresponds to a multi-step numerical Euler integration. In the limit of infinitesimally small steps, Neural ODEs f_{NODE} provide the derivative of the concentration with respect to the axial coordinate, i.e. the source term (Fig. 2.5d). They are coupled with ODE solvers to provide the concentration at the outlet or any other position by integration:

$$c_{\text{out}} = c_{\text{in}} + \int_0^L \underbrace{f_{\text{NODE}}(c(z))}_{\frac{\mathrm{d}c}{\mathrm{d}z}}\, dz \tag{2.30}$$

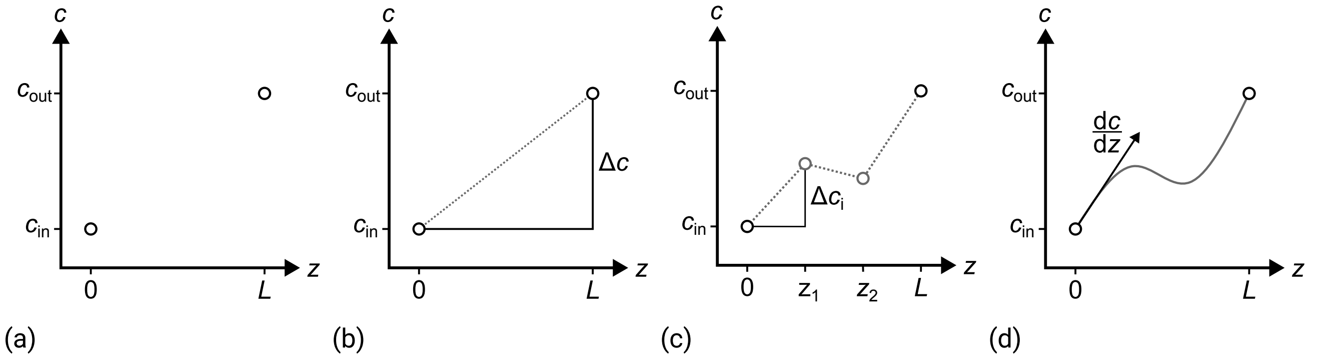


Figure 2.5: Schematic illustration of different strategies to predict the outlet concentration c_{out} at $z = L$ of a chemical reactor as a function of the feed concentration c_{out} at $z = 0$. (a) A feedforward neural network model directly predicts c_{out} as a function of c_{in}. (b) A recurrent neural network model predicts the concentration change Δc within the reactor. (c) A deep recurrent neural network iteratively predicts small incremental changes within the reactor, producing intermediate solutions. (d) A Neural ODE predicts the derivative $\frac{\mathrm{d}c}{\mathrm{d}z}$ of the concentration with respect to the reactor coordinate. The full concentration profile is obtained via numerical integration.

Neural ODE parameters are optimized by back propagation, which works best with fixed step ODE solvers or using the adjoint method, which is compatible with any black box ODE solver, see [78] for details. Overall, the Neural ODE technique allows to model chemical source terms using concentration measurements from an integral reactor.

3 State of the Art

This section presents the current state of the art in machine learning-based mechanism discovery and numerical acceleration of reactive flow simulations through surface kinetic surrogates. It compiles the current research needs in this area in the form of five challenges.

3.1 Surrogate models for surface kinetics

The solution of chemical kinetics is typically responsible for the vast majority of the computational cost associated with simulations of both, gas-phase [80, 81] and surface-reactive systems [82]. This limits the model-based design of large and challenging systems [4]. Surrogate models replace computationally expensive operations with numerically less demanding models and are widely used in chemical engineering [83]. In the context of reactive flow simulations, they provide efficient predictions of chemical source terms as a function of the reaction conditions, e.g. temperature and partial pressures. This is an example of a physics-informed problem formulation. This chapter provides an overview of the state-of-the-art surrogate modelling techniques for surface reactive systems and puts a focus on strategies to deal with the highly non-linear input-output relation typical for chemical kinetics.

3.1.1 Interpolation methods

Multivariate splines [84] were used to model precomputed steady state source terms from a detailed surface mechanism for CH_4 oxidation on platinum [85], automotive ammonia slip catalysts [86, 87], ethylene and methanol partial oxidation [88, 89], as well as selective catalytic reduction [90], resulting in speed-ups of two to three orders of magnitude. Further, variants of the Shepard interpolation were used to model surface reaction rates obtained by first principles kinetic Monte Carlo (1p-kMC) simulations of CO oxidation on Pd and RuO_2 [91–93].

All of these works report that the best interpolation performance is achieved, if the logarithm of source terms is mapped as a function of the logarithm of partial pressures and the inverse temperature (Eq. 3.1).

$$\ln \dot{s} = f_{\text{surrogate}} \left(\ln \boldsymbol{p}, T^{-1} \right) \tag{3.1}$$

In hindsight, this procedure can be justified physically. Chemical source terms are governed by highly non-linear equations (Eq. 2.2 and Eq. 2.3), leading to values spanning multiple orders of magnitude. Typical surrogate models are not designed to cover such strong non-linearities. The logarithm can be used to express reaction rates as a linear function of the inverse temperature, logarithmic partial pressures and logarithmic surface coverages (Eq. 3.2). In steady state, the latter are a function of the former two, so that the logarithm of rates can be expressed as a function of only inverse temperature and logarithmic partial pressures (Eq. 3.3).

$$\ln r_j = \ln k_{0,j} - \frac{E_A}{R} \cdot \frac{1}{T} + \sum_i \nu_{i,j} \cdot \ln p_i + \sum_\ell \nu_{\ell,j} \cdot \ln \vartheta_\ell \tag{3.2}$$

$$\overset{\text{s.s.}}{=} f\left(\ln \boldsymbol{p}, T^{-1}\right) \tag{3.3}$$

This linearization approach makes interpolation methods more likely to produce accurate predictions. Finally, it is transferred to source terms, which are linear combinations of the rates (Eq. 2.7). Overall, this is an example of physics-informed data curation.

As the logarithm is only defined for positive inputs[1], accurate surrogates can only be obtained if all source term values in the training data set are strictly positive (modelling as $\ln(\dot{s})$) or strictly negative (modelling as $\ln(-\dot{s})$). This prohibits the acceleration of 1. systems that include intermediate species, which are first formed (positive source terms) and later consumed again (negative source terms) and 2. systems close to thermodynamic equilibrium.

CHALLENGE 1: Extend the scope of kinetic surrogate modelling

For most industrially relevant systems, model-based design cannot benefit from accelerated simulations through kinetic surrogates because they rely on logarithmic data normalization, which is incompatible with intermediate species and conditions close to chemical equilibrium.

Interpolation functions rely on a dense sampling of known solutions to interpolate between. Due to the so-called *curse of dimensionality*, these methods become ineffective for high-dimensional problems. Multivariate splines, for example, require the input values of the precomputed solutions to lie on a regular grid [94]. Consequently, the number of solutions to compute before the model is generated grows exponentially with the number of input dimensions (Fig.3.1). But even if the required number of solutions can be computed, the curse of dimensionality also affects the number of parameters to be stored in memory, ultimately limiting the use of splines [90].

[1] The logarithm can be generalized to negative numbers by adding the complex number i: $\log -x = i\pi + \log x$. Predicting such data would require an additional classification algorithm and does not necessarily solve the problem of data normalization.

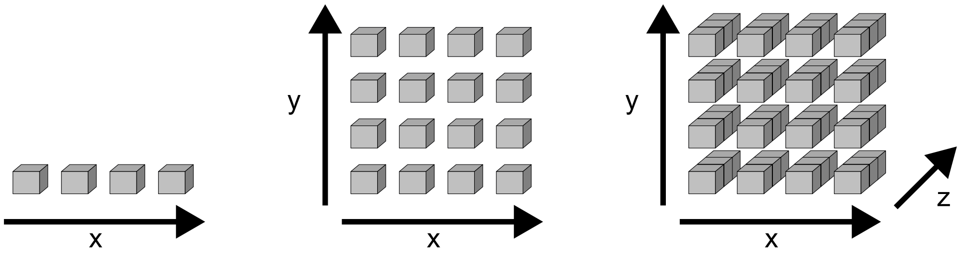


Figure 3.1: The number of points on a regular grid grows exponentially with the number of dimensions.

Challenge 2: Overcome the curse of dimensionality

Surface kinetic surrogate modelling with state-of-the-art interpolation methods is currently restricted to small and medium-sized systems with fewer than ten gas species because memory demand and the required amount of training data increase exponentially with the number of variables. Machine learning methods are needed to overcome this so-called curse of dimensionality and enable the efficient implementation of more detailed kinetic mechanisms.

3.1.2 Random Forests

Decision trees are a widely-used machine learning method that can be employed for regression and classification tasks. They iteratively subdivide the training range into smaller intervals and assign them a constant value, for example, the average value of training data found in this interval (Fig. 3.2). Consequently, the output of a decision tree is not continuous and looks like a stepwise function. Random forests [95] are an ensemble learning technique which averages the predictions of multiple uncorrelated decision trees to reduce the risk of overfitting.

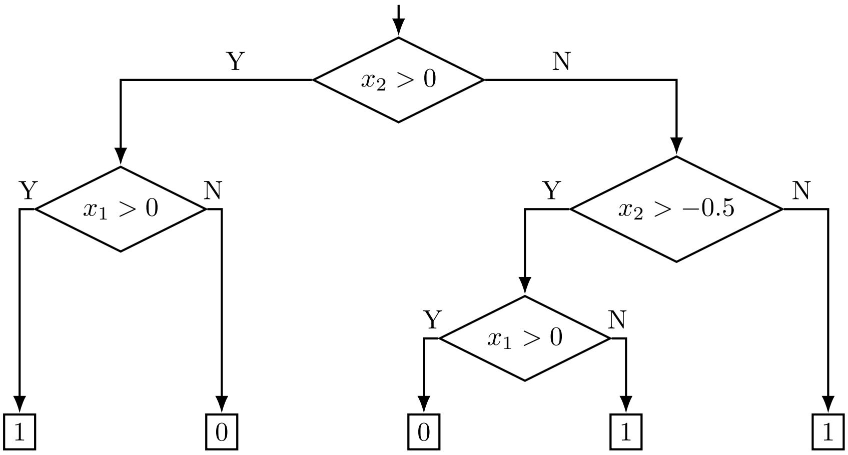


Figure 3.2: Schematic representation of a decision tree. It predicts different discrete output values depending on the input vector (x_1, x_2).

Partopour *et al.* used random forests for the efficient implementation of ethylene oxidation kinetics into a particle-resolved CFD simulation and reported a speed-up factor of over 500 towards the solution of the full microkinetic mechanism [96]. Further, they employed the readily available variable importance measure of the random forest technique [48] analogous to the degree of rate control method [97] to identify the importance of reactions and species and discuss its potential for model reduction purposes.

Bracconi and Maestri used random forests for the efficient implementation of a five-dimensional mean-field water-gas shift mechanism and a two-dimensional first principles kinetic Monte Carlo model of CO oxidation on RuO_2 [98]. Kinetic Monte Carlo simulations are computationally much more expensive than evaluating mean-field kinetics, so the authors proposed a training set design scheme which iteratively increases the amount of precomputed (1p-kMC) solutions until the surrogate model reaches a user-defined target accuracy (Fig. 3.3). This led to a 60-80% reduction in the amount of training data required [98]. In this scheme, the position of new points is chosen according to the slope of the function, which is an indicator for the training data density required by a random forest.Training data points are placed on an orthogonal grid to allow convenient numerical differentiation. Although only the most important variables are considered in this procedure, as every grid-based scheme it will ultimately suffer from the curse of dimensionality for high-dimensional problems.

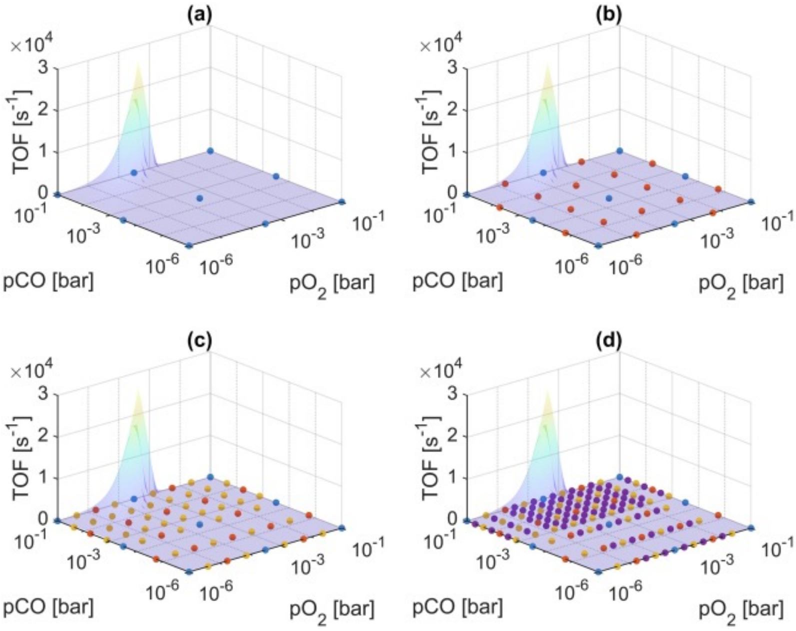


Figure 3.3: Evolution of the training data set in the iterative training set design scheme proposed by Bracconi and Maestri [98]. CO oxidation source terms are modelled as a function of O_2 and CO partial pressures. The initial grid of three equally spaced points in each direction is iteratively refined (a)-(d) based on the estimated gradient of the function. Reprinted from [98].

CHALLENGE 3: Reduce the cost of data collection

Grid-free training set design schemes are needed to reduce the cost of training data collection and thereby enable surface kinetic surrogates of industrially relevant systems with a 1p-kMC level of detail.

The discontinuous nature of tree-based models brings several disadvantages for modelling source term data. For example, they are scale-invariant and therefore cannot benefit from logarithmic data normalization. Further, they require a large amount of training data to achieve a high relative accuracy of values spanning multiple orders of magnitude (Fig. 3.4). Therefore, this work will explore neural network-based surrogates rather than tree-based models.

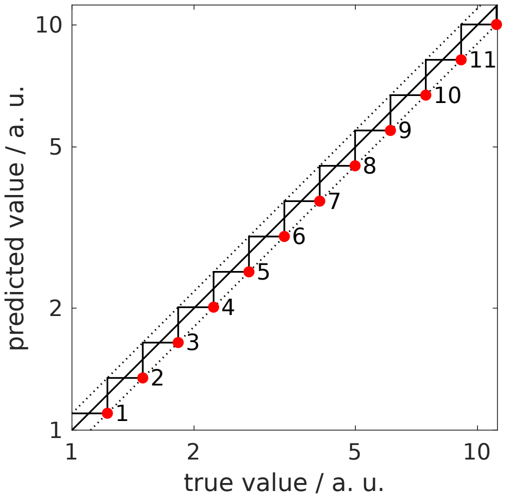


Figure 3.4: Decision trees (black stairs) require more than ten discrete outputs to predict any continuous function within 10% relative accuracy (dotted lines) over a range of one full order of magnitude.

3.1.3 Reliability

Machine learning models are generally considered as black-boxes which lack explainability and physical interpretability. Therefore, model reliability is a major concern, especially in extrapolation settings [64]. By their very nature interpolation methods are also not suited for extrapolation tasks. Importantly, learning in high dimensions always amounts to extrapolation [99]. So, while being a critical factor for the overall accuracy of reactor simulations [3], the quality of the kinetic surrogate model can never be blindly relied upon, especially when unexpected or extreme conditions occur.

CHALLENGE 4: Increase safety and trust

To increase the reliability of machine learning models, it is necessary to provide uncertainty quantification and physical interpretability. Without these, the models may not find application in the model-based design of catalytic devices.

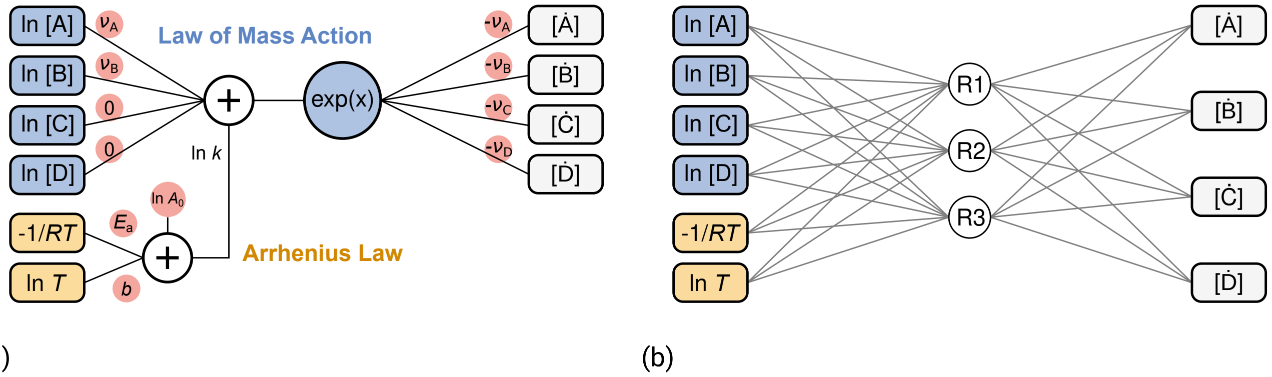


Figure 3.5: Schematic illustration of the CRNN for four species (A-D) and three reactions (R1-R3). All trainable parameters (marked red) have a physical interpretation. Adapted from [20].
(a) Every node in the CRNN encodes the Arrhenius law and the law of mass action to map the rates of a single reaction as a function of concentrations and temperature.
(b) Multiple reaction nodes are stacked in a single hidden layer to represent a complete reaction network.

3.2 Autonomous mechanism discovery

Autonomous discovery plays an increasingly important role in chemical sciences [100, 101]. The discovery of reaction pathways is a prime candidate for automation because even for species of just a few atoms millions of elementary reactions are possible [102]. This vast number makes it difficult to explore the reactions manually in an unbiased way [100].

The Reaction Mechanism Generator [103, 104] is a well-established tool used for generating detailed microkinetic mechanisms. It carries out first principle calculations to select only the kinetically likely pathways from the entire chemical reaction space of a system. It has been applied to gas phase [105] and surface reactive systems [106]. Parallel to this theory-driven approach, the chemical reaction neural network has recently emerged as a data-driven approach to automated mechanism discovery. It is a digital twin of the classical chemical reaction network [20] that structurally embeds the Arrhenius equation (Eq. 2.3) and the law of mass action (Eq. 2.2) into a neural network (Fig. 3.5). This is an example of a physics-informed architecture. To this end, the logarithmic neural network approach [107] is adopted to express the multiplication and power function operations found in those equations as addition and multiplication operations[2]:

$$k \cdot \prod_i c_i^{\nu_i} = \exp\left(\ln k + \sum_i \nu_i \cdot \ln c_i\right) \tag{3.4}$$

All parameters of the CRNN can directly be interpreted as kinetic parameters. Together with the Neural ODE technique [78], this interpretation has been used to parametrize microkinetic mechanisms

[2] This works because concentrations and pre-exponential factors are non-negative. Concentrations near zero are commonly set to a small threshold value to avoid numerical problems.

from integral reactor data. Some examples include chemical and biochemical engineering systems [20, 108, 109], biomass pyrolysis [110], stability analyses of battery materials [111], and the decomposition of high energy density materials [112–114]. A similar approach has further been applied to optimize the parameters of existing detailed fuel combustion mechanisms [115, 116], discover hydrogen as well as methane combustion mechanisms [117], and accelerate large combustion mechanisms for high-fidelity simulations of turbulent flames [118].

Physical consistency of discovered mechanisms

A common concern about CRNN-based mechanisms is that they generally violate the atom balance. This happens because stoichiometric coefficients are treated as any other model parameter without physical constraints. As a result, the model assumes reactions that do not preserve the number of atoms per element. While several attempts have been reported to enforce atom conservation in neural network-based kinetic models, none of them guarantee physically consistent models for mechanism discovery:

- atom conservation can be encouraged by physics-informed loss functions [119–121], which penalize deviations from the atom balance to guide model training towards physical consistency. However, this is only a soft constraint and does not guarantee fully consistent models.
- model predictions can be corrected in post-processing steps [122], but this does not guarantee consistent mechanistic parameters.
- the overall mass balance can be structurally embedded [110], but this does not guarantee the more rigorous atom conservation.
- the atom balance can be structurally embedded based on a given stoichiometric matrix [74, 123–126], but this is not suited for mechanism discovery, where the stoichiometric coefficients are generally unknown and subject to optimization.

Challenge 5: Discover physically consistent reaction pathways

Currently, there is no way to ensure atom conservation during autonomous mechanism discovery with chemical reaction neural networks. Physical consistency has to be maintained to make the CRNN a reliable tool for kinetic modelling.

4 Results

The scientific results obtained in this work have already been published and are presented in this manner below. Where present, the supporting information documents are provided. The following overview highlights each publication's contributions to overcome the research challenges derived from the relevant literature in the previous chapter.

The first publication **"Efficient machine learning based surrogate models for surface kinetics by approximating the rates of the rate-determining steps"** [17] makes use of mechanistic insights to extend the scope of kinetic surrogate modelling to systems with intermediate species and those that operate close to chemical equilibrium (Challenge 1). It further introduces neural networks as surrogates for surface kinetics and compares them to splines, the former state of the art, to address Challenge 2. The approach is showcased using the preferential oxidation of CO, which is relevant to hydrogen production for fuel cell applications [127]. Reactor simulations of this system are accelerated by a factor of over 10 000.

The second publication **"Efficient neural network models of chemical kinetics using a latent asinh rate transformation"** [18] proposes the use of logarithm-like transformation functions as an alternative way of addressing Challenge 1 that does not require mechanistic insights and thereby further extends the scope from modelling known kinetic mechanisms to discovering kinetics from differential reactor data. More generally, this work introduces latent data transformation, a way to embed physical knowledge (Challenge 4) into neural network structures. Besides the preferential oxidation, this is shown for ammonia oxidation under industrially relevant conditions of the Ostwald process.

The third publication **"Goal-Oriented Two-Layered Kernel Models as Automated Surrogates for Surface Kinetics in Reactor Simulations"** [19] explores the benefits of two-layered kernel models, a novel machine learning method. It concludes that uncertainty quantification is a key tool to increase data efficiency and model reliability during training and application of kinetic models. In particular, it presents a grid-free training set design scheme that reduces the cost of data collection by more than one order of magnitude (Challenge 3) and quantifies the accuracy of the resulting kernel-based surrogates during their use in simulations (Challenge 4).

The fourth publication **"Robust Mechanism Discovery with Atom Conserving Chemical Reaction Neural Networks"** [21] addresses the physical plausibility of data-driven mechanism discovery techniques (Challenge 5). The proposed atom conservation layer enforces physical plausibility and enhances robustness against real-world data flaws, such as limited data availability, noisy data, and systematic measurement errors.

4.1 Efficient machine learning based surrogate models for surface kinetics by approximating the rates of the rate-determining steps

The first publication [17] addresses the limited scope of surrogate modelling for surface kinetics which is caused by the fact that the well-known logarithmic transformation cannot be used to linearize source term data if a species changes sign. It makes use of the fact that source terms are linear combinations of elementary rates, which are positive by definition. Therefore, the logarithm of elementary rates can be mapped as a function of the inverse temperature and the logarithm of partial pressures (Eq. 4.1) in analogy to Eq. 3.3. Afterwards, source terms are recovered by multiplication with the stoichiometric matrix (Eq. 2.7).

$$\ln r = f_{\text{surrogate}}\left(\ln \boldsymbol{p}, T^{-1}\right) \tag{4.1}$$

However, even if the elementary rates are accurately modelled, the resulting source terms might be inaccurate. This is due to a mathematical effect called *catastrophic cancellation* and occurs when two very similar numbers are subtracted. For example, the adsorption-desorption reaction typically has a low activation barrier and is therefore prone to this effect. The present publication avoids this by modelling the rates of the rate-determining steps instead. This is an example of physics-informed data curation. Further, a systematic approach is presented to determine the rate-determining steps using reaction path analysis.

Before the publication of this work, surrogate modelling in heterogeneous systems routinely relied on classical interpolation methods like splines. As they are based on pre-computed solutions placed on a regular grid, they suffer from the so-called curse of dimensionality. The number of pre-computed solutions and the memory requirements of the model grow exponentially with the number of input dimensions. This limits the application of surrogate models to medium-sized problems with less than ten gas phase species. The present publication addresses that problem by replacing splines with neural networks. A thorough comparison shows that neural networks outperform splines with regard to accuracy, prediction time, and the required amount of training data and storage space.

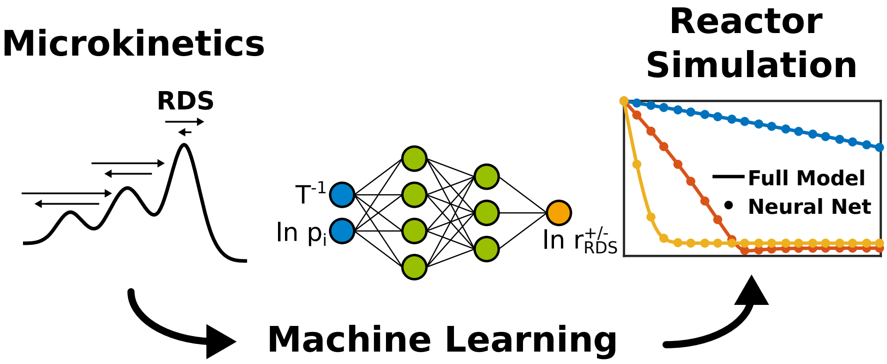


Figure 4.1: Graphical abstract of the first publication contained within this work [17].

Reprinted with permission from F. A. Döppel and M. Votsmeier, *Chemical Engineering Science*, **2022**, *262*, 117964. Copyright 2022 Elsevier.

Chemical Engineering Science 261 (2022) 117964

Contents lists available at ScienceDirect

Chemical Engineering Science

journal homepage: www.elsevier.com/locate/ces

Efficient machine learning based surrogate models for surface kinetics by approximating the rates of the rate-determining steps

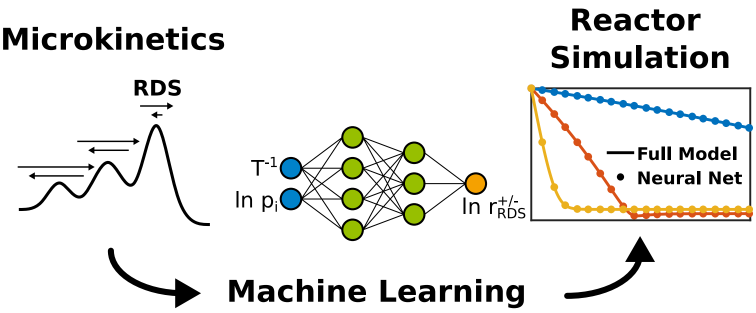
Felix A. Döppel[a], Martin Votsmeier[a,b,*]

[a] Technische Universität Darmstadt, Alarich-Weiss-Str. 8, 64287 Darmstadt, Germany
[b] Umicore AG & Co. KG, Rodenbacher Chaussee 4, 63457 Hanau, Germany

HIGHLIGHTS

- Decomposing source terms into rate-determining reactions yields efficient surrogates.
- Improves accuracy by factor 100 compared to modelling source terms directly.
- Neural networks outperform spline regarding accuracy, storage, speed & required data.
- Even moderate size neural networks yield unprecedented prediction errors far below 1%
- Method can be paired with any regression model (splines, random forests, SVM, ...)

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 4 March 2022
Received in revised form 21 July 2022
Accepted 29 July 2022
Available online 2 August 2022

Keywords:
Machine learning
Heterogeneous catalysis
Microkinetic
Rate-determining step
Surrogate model
Neural network

ABSTRACT

Machine learning based surrogate models that interpolate precomputed solutions of the rate equations can greatly accelerate simulations of catalytic systems. The interpolation quality is significantly increased if reaction rates are mapped using a logarithmic scaling. However, this only works if the source terms $\dot{s}$ are either strictly positive over the entire relevant range of conditions (scaling as ln ($\dot{s}$)) or strictly negative over the entire relevant range of conditions (scaling as ln (-$\dot{s}$)). This excludes most reaction systems of practical relevance, limiting this strategy to special use cases. We propose a new approach based on interpolating the forward- and reverse rates of the rate-determining reactions and thus overcoming this limitation.

The new scheme is demonstrated using a surface reaction mechanism describing the oxidation of CO and H_2 as well as the water gas shift reaction including 5 gas species, 9 surface species and 18 reversible reactions. Multivariate spline interpolation is used for a first evaluation of our approach. With the splines, the new method reproduces the source terms of CO with an error of 0.5 % which is 50 to 100 times more accurate than the established approaches of either mapping the source terms directly (49.9 % error) or mapping adsorption/desorption rates (23.7 % error).

The true potential of the new approach develops in combination with machine learning techniques like neural networks. Even very small neural networks with a single hidden layer of 20 nodes yield an error of 0.35 %. This is about 200 times more accurate than the same neural networks used with the established approaches. Increasing the network size to still moderate 30 nodes in two hidden layers (2342 parameters in total) reduces the error to 0.0058 %.

[*] Corresponding author.
E-mail address: martin.votsmeier@tu-darmstadt.de (M. Votsmeier).

https://doi.org/10.1016/j.ces.2022.117964

Nomenclature

Abbreviations	
A	preexponential factor (s^{-1})
α	factor for combining rates to source terms (-)
β	temperature exponent (-)
c	concentration (mol m^{-3})
c_{Pt}	site concentration (mol m^{-3})
E_{A}	activation energy (J mol^{-1})
e_{rel}	relative error (-)
e_{thres}	thresholded relative error (-)
Γ	site density (mol m^{-2})
K	equilibrium constant (-)
k_{ads}	adsorption rate constant (m^3 mol^{-1} s^{-1})
k_j	rate constant (s^{-1})
$\mathscr{L}_{\text{MSE}}$	mean squared error loss function (-)
$\mathscr{L}_r$	loss function for rates (-)
$\mathscr{L}_r^{\text{rel}}$	loss function for rates minimizing relative error (-)
$\mathscr{L}_{\dot{s}}$	loss functions for source terms (-)
M	molecular mass (kg mol^{-1})
N	number of points in data set (-)
ν	reaction order (-)
p	pressure (atm)
PEI	partial equilibrium index (-)
PEI-share	fraction of relevant conditions leading to partial equilibrium (-)

PROX	preferential oxidation
R	universal gas constant (J mol^{-1} K^{-1})
r	reaction rate (s^{-1})
$\hat{r}$	estimated rate (s^{-1})
r_{forward}	forward rates (s^{-1})
r_{reverse}	reverse rates (s^{-1})
RS	representative set of reactions
$\dot{s}$	source term (mol m^{-3} s^{-1})
$\hat{\dot{s}}$	estimated source term (mol m^{-3} s^{-1})
$\dot{s}_{\text{thres}}$	threshold source term (mol m^{-3} s^{-1})
s_0	sticking coefficient (-)
spline(x)	spline interpolation
T	temperature (K)
T_0	reference temperature (K)
t	time (s)
t_{predict}	time to predict source terms for 100,000 conditions (s)
τ	residence time (s)
τ_{max}	upper limit for the residence time (s)
θ	surface coverage (-)
w	error weight (-)
W	loss function weights (-)
$\underline{X}_{\text{RC}}$	degree of rate control (-)
$\underset{\sim}{X}_{\text{RC}}$	average degree of rate control (-)
$\widetilde{X}_{\text{RC}}$	total degree of rate control (-)

Besides the increased accuracy, the neural networks also outperform spline interpolation by at least an order of magnitude with respect to interpolation time, storage space and required amount of training data.

Due to the extremely low errors achieved with moderate size neural networks and small training data sets, the proposed method based on mapping the rate-determining steps shows promise to scale to much larger and more complex reaction systems in catalysis and other fields like combustion, atmospheric chemistry or systems biology.

1. Introduction

Complex reaction networks play an important role in many real-world systems including catalysis, atmospheric chemistry, combustion, and cell metabolism. The kinetics of the individual reactions are more and more well understood due to experimental progress and improved quantum chemical calculations. However, bridging the scales towards the application of the reaction mechanisms in simulations of macroscopic systems like a catalytic reactor still poses significant challenges. Therefore, new solutions are required. A promising approach to this scale bridging problem is to learn the results of the chemical rate equations by an interpolation- or machine learning technique (Micale et al., 2022). In this way, the simulation of the macroscopic system can be decoupled from the solution of the rate equations by replacing it with a call to the predetermined interpolation function.

Several applications of this rate mapping approach on catalytic systems have been reported in the literature. We used multivariate tensor spline functions, leading to an excellent representation of the surface chemistry (Scheuer et al., 2012; Votsmeier, 2009; Votsmeier et al., 2010), and it has been shown that this approach describes several relevant catalytic systems well (Partopour and Dixon, 2016). However, the storage requirement of the spline representations increases exponentially with the number of input variables (here: the number of gas phase species). The storage requirement could be somewhat reduced by the introduction of Hermite spline interpolation, so that systems with about seven input parameters can be handled on a standard computer (Klingenberger et al., 2017).

Matera *et al.* applied Shepard interpolation to represent surface reaction rates obtained by kinetic Monte Carlo simulations (Lorenzi et al., 2017; Matera et al., 2014; Sutton et al., 2018). There are also several publications that apply the *In Situ* Adaptive Tabulation technique for the interpolation of surface reaction rates which was specifically developed to overcome storage requirement limitations and is highly efficient in doing so (Blasi and Kee, 2016; Bracconi et al., 2017; Daymo et al., 2022; Kumar and Mazumder, 2011; Mazumder, 2005; Varshney and Armaou, 2008). Recently, the focus shifted towards the application of machine learning techniques. In particular, the Random Forest method has been used to represent surface kinetics (Bracconi and Maestri, 2020; Partopour et al., 2018).

Outside the field of catalysis, neural networks have successfully been applied as surrogate models for homogeneous kinetics, for example in combustion (Blasco et al., 1999; Christo et al., 1996; Ji and Deng, 2021; Wan et al., 2020), atmospheric chemistry (Keller and Evans, 2019; Kelp et al., 2022), or systems biology (Wang et al. 2019).

In general, reaction rates for a given species span several orders of magnitude. It has been found previously that the interpolation quality is greatly increased, if reaction rates are mapped using a logarithmic scaling of the rates and concentrations and an inverse scaling of the temperature (Bracconi and Maestri, 2020; Matera et al., 2014; Partopour and Dixon, 2016; Votsmeier, 2009;

Votsmeier et al., 2010). However, such a logarithmic scaling only works if the source terms $\dot{s}$ are strictly positive (scaling as ln ($\dot{s}$)) or strictly negative (scaling as ln (-$\dot{s}$)) over the entire relevant range of conditions. Most reaction systems of practical relevance contain species that are consumed under some conditions and produced under other conditions. This situation cannot be handled by the previous approach of a logarithmic scaling. This limitation is most likely one of the main reasons why so far, the rate mapping approach did not find a wider application. Previous investigations circumvented this problem by either focusing on reactions that only proceed in a forward direction (Votsmeier, 2009), choosing the range of application so that only forward reactions take place (Bracconi and Maestri, 2020), or exploiting the fact that in some special cases the source terms of the critical species can be expressed in terms of other source terms that do not change sign (Votsmeier et al., 2010).

One approach suggested in the literature to avoid negative values during logarithmic scaling of the source terms is to separately learn the adsorption- and desorption rates of the respective species (Partopour et al., 2018; Partopour and Dixon, 2016). To our knowledge the performance of this method has never been systematically investigated. The shortcomings of this approach will be discussed in Section 3.1.4.

In this paper we propose a new approach that is based on separately mapping the rates of the forward- and reverse rates of the rate-determining reactions. We apply this procedure to a surface reaction mechanism describing the oxidation of H_2 and CO by O_2 (including the PROX reaction, the preferential oxidation of CO in an excess of H_2 for removing CO during H_2 production) as well as the (forward and reverse) water–gas shift reaction on platinum. We demonstrate that our new approach increases the accuracy of the interpolation by orders of magnitude, achieving accuracies far better than 0.1 %.

First, we demonstrate the principle of our proposed scheme using multivariate spline interpolation. Spline interpolation allows for a convenient discussion of the approach, since in contrast to most machine learning techniques, setting up the interpolation function is a deterministic process and thus avoids complications due to the choice of hyperparameters and the randomness of the training process.

In a second step, we demonstrate that neural networks allow an even better implementation of the approach, providing better accuracy with less training data and thus offering promise for a scalability towards larger reaction systems.

2. Methodology

2.1. The reaction mechanism

The surface reaction mechanism used in this paper contains five gas species and nine surface species and was developed by by Mhadeshwar and Vlachos to describe CO oxidation, H_2 oxidation, water–gas shift reaction, preferential oxidation of CO and the promoting role of H_2O on CO oxidation on platinum (Mhadeshwar and Vlachos, 2004). The 18 reversible reactions of the mechanism are listed in Table 3. We use the set of kinetic parameters provided by Hauptmann *et al.* as listed in Table S1 in the supporting material.

Reaction rates are calculated using equation (1),

$$r_j = k_j \cdot \prod_i c_i^{\nu_{i,j}} \cdot \prod_l \theta_l^{\nu_{l,j}} \tag{1}$$

with the reaction rate r_j (s^{-1}), the rate constant k_j ($m^3\ mol^{-1}\ s^{-1}$ for adsorption and s^{-1} else), the concentration of gas species c_i (mol m^{-3}), the surface coverage θ_l (unitless) and the reaction order $\nu_{i,j}$ (unitless) (Hauptmann et al., 2011). The rate constants for adsorption reactions are calculated using equation (2) and the rate constants for all other surface reactions are calculated using equation (3),

$$k_{\mathrm{ads}} = \frac{R \cdot T}{\Gamma \cdot \sqrt{2 \cdot \pi \cdot M_i \cdot R \cdot T}} \cdot \left(\frac{T}{T_0}\right)^{\beta} \cdot s_{0,i} \tag{2}$$

$$k_j = A_j \cdot \left(\frac{T}{T_0}\right)^{\beta} \cdot \exp\left(-\frac{E_{\mathrm{A},j}}{R \cdot T}\right) \tag{3}$$

with the universal gas-constant R ($J\ mol^{-1}\ K^{-1}$), the temperature T (K), the site density Γ ($2.49081 \cdot 10^{-5}$ mol m^{-2}), the molecular mass M_i (kg mol^{-1}), the reference temperature T_0 (300 K), the temperature exponent β (unitless), the sticking coefficient $s_{0,i}$ (unitless) as well as the preexponential factor A_j (s^{-1}) and the activation energy $E_{\mathrm{A},j}$ (J mol^{-1}).

For given reaction conditions in form of temperature and the partial pressures of CO, CO_2, H_2, H_2O and O_2, steady state surface coverages are calculated by integrating equation (4) in time, until steady state is reached i.e., $\frac{d\theta_l}{dt} = 0$. Gas composition and temperature are assumed to be invariable during this process. Finally, the obtained steady state surface coverages are used to calculate gas species source terms using equation (5), where c_{Pt} is the site concentration (mol m^{-3}). Numerically, integration is performed using the DASPK solver (van Keken et al., 1995) with an integration time of 10^7 s, a relative tolerance of 10^{-6} and an absolute tolerance of 10^{-50}.

$$\frac{d\theta_l}{dt} = \sum\nolimits_j \nu_{l,j} \cdot r_j \tag{4}$$

$$\frac{dc_i}{dt} = \sum\nolimits_j \nu_{i,j} \cdot r_j \cdot c_{\mathrm{Pt}} \tag{5}$$

2.2. The input range of the surrogate models

The input range of the spline- and neural network interpolation functions was chosen to cover typical operating conditions met in a reactor for the removal of CO from H_2 streams by preferential oxidation of CO with small amounts of added O_2. At the same time, we also cover the operating conditions in a low temperature water gas shift reactor. Table 1 shows the minimal and maximal values used for each variable.

2.3. Error measures

To measure the accuracy of the predicted source terms, a test data set was used with 100,000 data points sampled randomly from the operating range of Table 1 applying logarithmic scaling of partial pressures and inverse scaling of temperature. The source terms of CO and O_2 both span several orders of magnitude, also including very small values that will not lead to significant concentration changes within typical reactor residence times. To avoid overemphasis on these very small values, source terms $\dot{s}_i$ below a certain threshold $\dot{s}_{i,\mathrm{thres}}$ are considered with a reduced weight w_i (Scheuer et al., 2011, 2012; Votsmeier et al., 2010),

$$w_i = \begin{cases} 1 \text{ if } |\dot{s}_i| \geq \dot{s}_{i,\mathrm{thres}} \\ \frac{|\dot{s}_i|}{\dot{s}_{i,\mathrm{thres}}} \text{ else} \end{cases} \tag{6}$$

The thresholded relative error e_{thres} is defined as

$$e_{\mathrm{thres}} = \frac{1}{N}\sum_{i=1}^{N} w_i \left|\frac{\hat{\dot{s}}_i - \dot{s}_i}{\dot{s}_i}\right| \tag{7}$$

which is equivalent to

Table 1
The input range for reaction conditions (temperature and partial pressures) which are solved for steady state.

Quantity	Unit	Minimum	Maximum	Scaling
T	K	280	600	reciprocal
$p(H_2)$	atm	$8 \cdot 10^{-2}$	$8 \cdot 10^{-1}$	logarithmic
$p(O_2)$	atm	$1 \cdot 10^{-7}$	$4 \cdot 10^{-2}$	logarithmic
$p(H_2O)$	atm	$4 \cdot 10^{-2}$	$4 \cdot 10^{-1}$	logarithmic
$p(CO)$	atm	$1 \cdot 10^{-7}$	$4 \cdot 10^{-2}$	logarithmic
$p(CO_2)$	atm	$4 \cdot 10^{-2}$	$4 \cdot 10^{-1}$	logarithmic

$$e_{\mathrm{thres}} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{\dot{s}}_i - \dot{s}_i}{\max\left(|\dot{s}_i|, \dot{s}_{i,\mathrm{thres}}\right)}\right| \qquad (8)$$

Here $\hat{\dot{s}}_i$ is the predicted source term and N the number of points in the test set. The equivalence between equation (7) and (8) is shown in equations (S12-S20) in the supporting material.

To define the threshold rate $\dot{s}_{i,\mathrm{thres}}$, we assume an upper limit for the reactor residence time of $\tau_{\max} = 10$ s. We then define $\dot{s}_{i,\mathrm{thres}}$ as the source term that yields a 100 % change in the respective concentration c_i for a residence time $\tau_{\max}$ (equation (9)).

$$\dot{s}_{i,\mathrm{thres}} = \frac{c_i}{\tau_{\max}} \qquad (9)$$

For fast reactions the thresholded error is identical to the relative error. For slow reactions a thresholded error of 1 % means that the predicted source term is off by $1\% \cdot c/\tau_{\max}$.

In previous publications we used the root mean square of the thresholded relative error instead of the mean (Scheuer et al., 2011, 2012; Votsmeier et al., 2010). However, in the current study we found that the root mean square average leads to unwanted behavior as it puts disproportionate weight on a few points with high errors. For our 100,000-point test set, 5 data points contributed more than 99% of the test set error in some cases, making the result highly dependent on the choice of the test data points. Therefore, in this study we use the mean of the absolute values, which is less sensitive to individual data points. In Section S4 of the supplementary material we compare the different error measures (mean thresholded error, root mean square thresholded error and a simple relative error without threshold) for two neural networks discussed in the paper. While the obtained values differ by up to an order of magnitude, the qualitative findings of the paper do not depend on the choice of the error measure. For all error measures, the best neural network obtained by our new scheme reaches an accuracy well below 0.2 %.

Since the thresholded relative error requires a reference concentration, the error of the rates of individual reactions is reported as the mean relative error,

$$e_{\mathrm{rel}} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{r_i - \hat{r}_i}{r_i}\right| \qquad (10)$$

with the rate r_i and the estimated rate $\hat{r}_i$.

2.4. Spline interpolation

Polynomial spline maps are implemented using the MATLAB® Curve Fitting Toolbox version R2020b with the "*spapi*" function and stored in polynomial form (de Boor, 1978). The Splines use a data set with points located on an orthogonal grid uniformly spaced in inverse temperature and logarithm of partial pressures. Nine breakpoints are used per dimension, yielding 9^6 = 531441 points in total.

2.5. Neural networks

Shallow neural networks are implemented using the MATLAB® Deep Learning Toolbox version R2020b. Training is performed with Bayesian regularization backpropagation using MATLAB®'s "*trainbr*" algorithm. It updates weight and bias values according to Levenberg-Marquardt optimization to minimize a combination of squared errors and weights that generalizes well (Foresee and Hagan, 1997; MacKay, 1992). The standard configuration consists of a single hidden layer using tanh activation. $\dot{s}_{\mathrm{CO}}$, $\dot{s}_{\mathrm{O_2}}$ and ln $\left(-\dot{s}_{\mathrm{O_2}}\right)$ use 40 units in the hidden layer (321 parameters). In case of CO forward and reverse rates two models are built using 20 units each (322 parameters in total). The parameter study shown in Figure 6 uses 5 to 150 nodes per model in one layer and 5 to 30 nodes per model in two layers with a total number of learnable parameters of up to 2500. Per task, three randomly initialized models are trained until no relative decrease of at least 1 % is observed for validation error over 25,000 epochs or the training algorithm fails to reduce the error on the training set. All three models use the same training set. Where not stated otherwise, the training set consists of 25,000 data sampled randomly from the range of conditions and using the scaling as described in Table 1. Full batch learning is performed. The error reported is the median thresholded relative error obtained from the three different models trained, using the 100,000-point test set. This set includes none of the training data points. Figure S7 in the supporting material shows exemplary training curves.

During training, for learning the source terms, backpropagation is used to minimize the loss function:

$$\mathscr{L}_{\dot{s}} = \frac{1}{N}\sum\nolimits_{i=1}^{N}\left(\frac{\dot{s}_i - \hat{\dot{s}}_i}{\max\left(|\dot{s}_i|, \dot{s}_{i,\mathrm{thres}}\right)}\right)^2 \qquad (11)$$

For learning forward- and reverse rates the loss function is:

$$\mathscr{L}_r = \frac{1}{N}\sum\nolimits_{i=1}^{N}\left(\frac{r_i - \hat{r}_i}{\max\left(|\dot{s}_i|, \dot{s}_{i,\mathrm{thres}}\right)}\right)^2 \qquad (12)$$

In both cases, the errors are scaled relative to the thresholded source term. The threshold for the source term $\dot{s}_{i,thres}$ is the same as introduced in section 2.3.

The loss functions in equation (11) and (12) are implemented using the mean squared error loss function $\mathscr{L}_{\mathrm{MSE}}$ (equation (13)) choosing the weights W_i in a way to mimic $\mathscr{L}_{\dot{s}}$ and $\mathscr{L}_r$ as shown in section S3 of the supporting material.

$$\mathscr{L}_{MSE} = \frac{1}{N}\sum\nolimits_{i=1}^{N} W_i \cdot \left(y_i - \hat{y}_i\right)^2 \qquad (13)$$

In section S4 of the supplementary material we show that it is important to adapt the loss function to the error measure used.

2.6. Plug flow reactor model

A simple isothermal and isobaric plug flow reactor is modelled by discretizing its length into cells of equal size. In each cell steady state source terms are calculated according to equation (5). Concentrations in the next cell are updated according to equation (14),

$$c_{i,n+1} = c_{i,n} + \dot{s}_i\left(c_{i,n}, T\right) \cdot \tau_n \qquad (14)$$

with the concentration $c_{i,n}$ (mol m^{-3}) of species i in cell n, the temperature T (K), the source term $\dot{s}_i$ (mol m^{-3} s^{-1}) of species i and τ_n (s) as the residence time in cell n obtained by dividing the total residence time by the number of cells.

A total pressure of 1 atm, a site concentration c_{Pt} of 26.3 mol m^{-3}, a reactor length of 1 m divided into 200 cells and a gas velocity of

1 m/s are used. The resulting residence time τ is 1 s. Molar fractions in the feed are 40 % H_2, 1 % O_2, 10 % H_2O, 1 % CO and 10 % CO_2 with N_2 as balance species.

2.7. Degree of rate control

The degree of rate control $X_{RC,i}$ (unitless) is a measure to identify rate controlling steps in reaction mechanisms. A value of 1 means, the step i is rate controlling, whereas a value of 0 means step i has a negligible influence. $X_{RC,i}$ is defined as.

$$X_{RC,i} = \frac{k_i}{r}\left(\frac{dr}{dk_i}\right)_{k_{j\neq i},K_i} \tag{15}$$

with the rate constant k_i, the net reaction rate r and the equilibrium constant K_i. To keep K_i constant, k_i of the forward rate and k_{-i} of the reverse rate are varied simultaneously by the same factor (Campbell, 2017).

$X_{RC,i}$ can change depending to reaction conditions like temperature, and concentrations. To consolidate the degree of rate control results for a range of conditions into a single value for each reaction, we define the average degree of rate control $\bar{X}_{RC,i}$ as the mean of $X_{RC,i}$ over all N conditions.

$$\bar{X}_{RC,i} = \frac{1}{N}\sum_{m=1}^{N} X_{RC,i}(T_m, p_m(\mathrm{CO}), p_m(\mathrm{O_2}), \cdots) \tag{16}$$

To measure the Degree of Rate Control for a set of multiple reactions, we define a total degree of rate control $\tilde{X}_{RC,S}$ as the sum of $\bar{X}_{RC,i}$ over all reactions i part of representative set RS (equation (17)).

$$\tilde{X}_{RC,RS} = \sum_{i\in RS} \bar{X}_{RC,i} \tag{17}$$

2.8. Hardware

The hardware used for all calculations in this paper was a Ryzen 7 3800X CPU @3.6 GHz with 16 Gb RAM @3600 MHz running Microsoft Windows 10 64-bit as an operating system. MATLAB® Version R2020b was used. Runtime measurements use MATLAB®'s intrinsic tic toc commands. Reported times are an average of five identical calculations.

3. Results and discussion

First, we demonstrate the principle of our proposed scheme using spline regression. Spline regression is well suited for an initial discussion since it avoids difficulties arising from a non-deterministic learning process and there is little bias form the choice of hyperparameters. In a later section we will demonstrate the full potential of our method by combining it with neural networks.

3.1. Demonstration of the interpolation approach using splines

3.1.1. Outline of the problem

The reaction system considered in this work can be described by three global reactions:

$2\,CO + O_2 \rightarrow 2\,CO_2$ CO oxidation
$2\,H_2 + O_2 \rightarrow 2\,H_2O$ H_2 oxidation
$CO + H_2O \rightleftharpoons CO_2 + H_2$ water – gas shift

Since the three reactions are linearly dependent (water–gas shift can be expressed as CO oxidation and reversed H_2 oxidation happening simultaneously), only two source terms are required to fully describe the reaction progress in the system. We focus on the source terms for O_2 and CO, keeping in mind that the source terms for all other species can be computed as linear combinations e. g., $r_{H_2} = 2 \cdot r_{O_2} - r_{CO}$.

The equilibrium of the CO and H_2 oxidation reactions is fully on the right side, so that the source term of O_2 is negative under all conditions considered in our study. Therefore, O_2 is an example where a logarithmic scaling of the source term can be applied (in this case as $\log(-\dot{s}_{O_2})$). A spline map was constructed that interpolates the steady state source term as a function of the gas species partial pressures (H_2, O_2, H_2O, CO, CO_2) and temperature. Table 2 lists the obtained error, when spline maps with and without logarithmic scaling were used to predict the source terms from a 100,000-point test data set. An error of 0.2 % is observed for the logarithmic scaling, while no useful representation of the data is achieved by direct mapping (error > 100 %). This result is in line with our previous results obtained with reaction mechanisms for CH_4 oxidation (Votsmeier, 2009) and NH_3 oxidation (Votsmeier et al., 2010).

The source terms of CO (and all other gas species except O_2) assume positive and negative values, depending on the reaction conditions. In this case a simple logarithmic scaling cannot be applied. We have constructed a spline interpolation function for the steady state CO source terms using a linear scaling. A prediction error of 49.9 % is obtained (see Table 2), rendering it unsuitable for reactor simulations.

As in our example, most reaction mechanisms of practical relevance contain species that appear as both, reactants and products, depending on the reaction conditions. This poses a severe limitation for the representation of source terms by interpolation- and machine learning approaches, since in these cases, no logarithmic scaling can be applied, leading to poor predictions. In the following, we present a new approach that overcomes these problems, allowing for an accurate representation of steady state kinetics regardless of source terms changing sign.

3.1.2. Mapping forward and reverse rates of elementary reactions

In contrast to the species source terms, elementary rates are always positive. They should be easy to model using an Arrhenius-like scaling as $\ln(r) = f\left(T^{-1}, \ln\ p_i\right)$. For demonstration, we built one spline map for each elementary rate i.e., the forward- and reverse rates of all 18 reactions, approximating the steady state rates as a function of temperature and partial pressures. Figure 1 shows the accuracy of all 36 spline maps. Their mean relative error is below 1 % in most cases. Note, that for the interpolation of the rates of individual reactions we report relative errors (equation (10)) instead of the thresholded relative error (equation (8)) used to report the error of source terms.

Table 2
Average error e_{thres} obtained by spline interpolation of the source terms for O_2 (always negative) and CO (positive and negative values, depending on conditions). Quadratic splines are constructed as described in section 2.4, using an equidistant grid of 531,441 input data points spanning the input range provided in Table 1. The thresholded relative error is computed according to equation (8), using a test data set with 100,000 conditions sampled randomly from the input range.

	O_2 error	CO error
Direct mapping $\dot{s} = f\left(T^{-1}, \ln\ p_i\right)$	632 %	49.9%
Arrhenius-like scaling $\ln(\dot{s}) = f\left(T^{-1}, \ln\ p_i\right)$	0.209 %	–

Table 3
Different sets of representative reactions I-IV for the CO source term. Reactions 1–6 and 16–18 are part of none of these sets. Fields with $\alpha_i = 0$ are left empty. For each reaction the mean degree of rate control $\bar{X}_{RC}$ with respect to CO and partial equilibrium index share (PEI-share) are given. Values are rounded to the nearest integer. For each representative set $\tilde{X}_{RC}$, PEI-share and accuracy of spline regression are given.

		α_i for representative set					
No.	Reaction	I	II	III	IV	$\bar{X}_{RC}$/ %	PEI-share / %
H_2 oxidation on Pt							
1	$H_2 + 2\,{*} \rightleftharpoons 2H^*$					1	83
2	$O_2 + 2\,{*} \rightleftharpoons 2O^*$					5	0
3	$OH^* + {*} \rightleftharpoons O^* + H^*$					0	0
4	$H_2O^* + {*} \rightleftharpoons OH^* + H^*$					−1	13
5	$H_2O^* + O^* \rightleftharpoons 2\,OH^*$					0	0
6	$H_2O + {*} \rightleftharpoons H_2O^*$					0	100
CO oxidation on Pt							
7	$CO + {*} \rightleftharpoons CO^*$	−1				8	78
8	$CO_2 + {*} \rightleftharpoons CO_2{}^*$				1	0	100
9	$CO_2{}^* + {*} \rightleftharpoons CO^* + O^*$		1	1		0	0
Coupling reactions							
10	$CO_2{}^* + H^* \rightleftharpoons CO^* + OH^*$		1	1		1	9
11	$COOH^* + {*} \rightleftharpoons CO^* + OH^*$		1			1	9
12	$COOH^* + {*} \rightleftharpoons CO_2{}^* + H^*$			−1		0	100
13	$CO^* + H_2O^* \rightleftharpoons COOH^* + H^*$		−1			**85**	3
14	$CO_2{}^* + OH^* \rightleftharpoons COOH^* + O^*$			1		0	0
15	$CO_2{}^* + H_2O^* \rightleftharpoons COOH^* + OH^*$			1		0	12
16	$CO_2{}^* + H^* \rightleftharpoons HCOO^* + {*}$					0	100
17	$CO_2{}^* + OH^* \rightleftharpoons HCOO^* + O^*$					0	0
18	$CO_2{}^* + H_2O^* \rightleftharpoons HCOO^* + OH^*$					0	12
$\tilde{X}_{RC}$/ %		8	**86**	1	0		
PEI-share / %		78.0	**3.20**	100	100		
e_{thres}/ %		23.7	**0.506**	2.70	3.27		

3.1.3. Computing CO source terms from estimated adsorption/desorption rates

Having shown that the forward- and reverse rates for all reactions are well estimated by spline regression, an obvious approach is to compute the gas species source terms as the difference of adsorption and desorption rate estimates. This procedure has previously been proposed (Partopour and Dixon, 2016; Partopour et al., 2018). We investigate this approach for the source term $\dot{s}_{CO}$ of CO. Although the individual rates of CO adsorption and desorption (reaction 7) are both represented with a relative error below 0.4 %, the net rate of CO adsorption/desorption computed by this technique has a relative error of 110 %. The thresholded error e_{thres} used to characterize the accuracy of source terms amounts to 23.7 % when computed from the forward- and reverse rate estimates. That is not significantly more useful than the 49.9 % error obtained by mapping the source term directly (compare Table 2).

3.1.4. Why using adsorption/desorption rates may result in poor predictions: The reversibility of the reaction

Computing the CO source terms as the difference of adsorption and desorption rate estimates yields poor results. To understand why, we look at the ratio of the forward- and reverse rates. We consider two limiting cases. If a reaction is near partial equilibrium, the forward- and reverse rates both assume high values, while their difference i.e., the net rate, is much smaller. In contrast, reactions far from equilibrium, show a large rate in one direction and a small rate in the other. Their difference is close to the larger rate.

A standard measure for the partial equilibrium status of a reaction is the Partial Equilibrium Index PEI defined by equation (18) (Gupta and Vlachos, 2020; Mhadeshwar and Vlachos, 2005),

$$PEI = \frac{r_{\text{forward}}}{r_{\text{forward}} + r_{\text{reverse}}} \tag{18}$$

with the forward- and reverse rates denoted as r_{forward} and r_{reverse}, respectively. For reactions in partial equilibrium, r_{forward} and r_{reverse} are of equal magnitude and the PEI is close to 0.5. For reactions far from equilibrium, the PEI is near 0 or 1.

To obtain a measure for the 'average' equilibrium status of an elementary reaction over the entire range of relevant operating conditions, we define the PEI-share. It is defined as the fraction of relevant reaction conditions leading to the reaction being close to partial equilibrium i.e., a PEI within 0.5 ± 0.05. A high PEI-share means the reaction is in partial equilibrium for most conditions of interest.

Figure 2 displays a histogram of the PEI values of CO adsorption/desorption (reaction 7). In agreement with (Mhadeshwar and Vlachos, 2005) most conditions show a PEI close to 0.5, indicating that the reaction is close to partial equilibrium. The PEI-share of this reaction is 78 %. This explains the poor accuracy of the source term when computed from the adsorption and desorption rate estimates. Here, the net rate is computed as the difference of two much larger rates. Despite the small relative errors, inaccuracies add up and the total inaccuracy is large compared to the small source term.

Figure 2 also displays a histogram of the PEI values of reaction 13 which Mhadeshwar and Vlachos identified as the rate-determining step (Mhadeshwar and Vlachos, 2005). Most values are near 0 or 1, but few are close to 0.5. The PEI-share is 3 %, so in agreement with (Mhadeshwar and Vlachos, 2005) the reaction is far from equilibrium on average. For most conditions, forward and reverse rates are very different. This suggests that the net rate of reaction 13 can be computed as the difference of forward and reverse rate estimates without losing accuracy. Indeed, this technique of estimating the net rate of reaction 13 has a relative error of 1.18 %, similar to the accuracy obtained for the forward- and reverse rates.

3.1.5. Computing the net rates of all reactions

So far, we learned that an accurate prediction of forward- and reverse rates does not necessarily lead to an accurate prediction

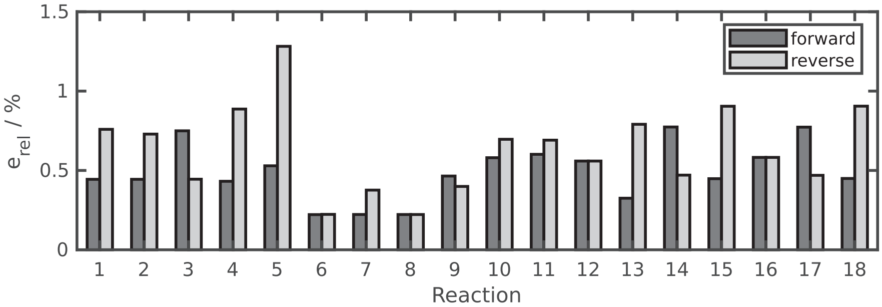


Fig. 1. Mean relative errors for spline maps predicting steady state elementary rates. Forward and reverse rates of all 18 reactions sum up to 36 elementary rate spline maps. Each spline map is based on the steady state solutions of the same 9^6 conditions. The relative error was computed based on equation (10), using the 100,000-point test data set applied throughout this paper.

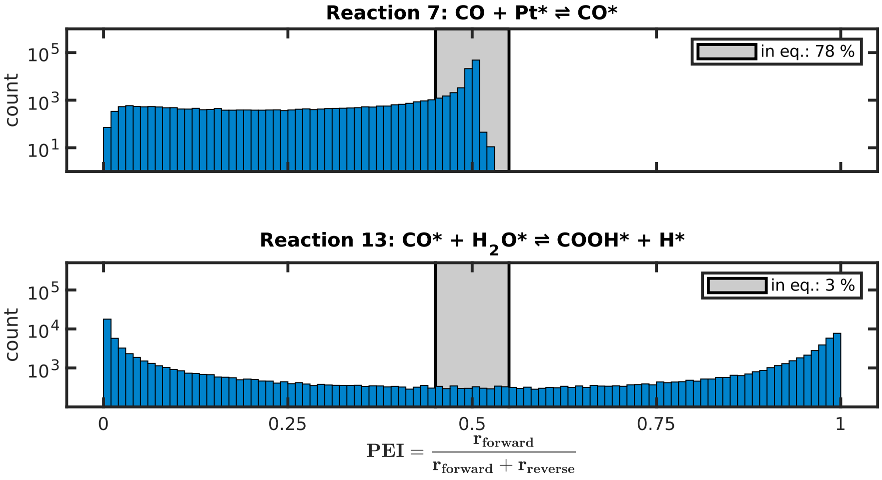


Fig. 2. The distribution of partial equilibrium index (PEI) values for the steady state rates of all 100,000 reaction conditions of the test set. Values close to 0.5 (shaded area) imply partial equilibrium. Here, forward and reverse rates are fast compared to the net rates. Values near 0 and 1 are far from partial equilibrium. Top: CO adsorption/desorption is in partial equilibrium for most reaction conditions (Mhadeshwar and Vlachos, 2005) explaining the poor net rate estimate. Bottom: reaction 13 is far from partial equilibrium (Mhadeshwar and Vlachos, 2005).

of the net rate. Also, we observed that the accuracy of the computed net rates seems to depend on the partial equilibrium status of the reaction. To verify this, we computed the net rate for all the 18 reactions from the corresponding forward- and reverse rate estimates. In each case, a separate spline function was constructed for forward- and reverse reactions using an Arrhenius-like scaling as $\ln(r) = f\left(T^{-1}, \ln\ p_i\right)$. Figure 3 shows the interpolation accuracies obtained for the net rates of all 18 reactions. The obtained errors cover a wide range from relative errors below 1 % to higher than 100 %. Figure 3 also displays the PEI-share of all reactions, showing a strong correlation to interpolation accuracy. A low PEI-share seems to guarantee a high interpolation accuracy ($e_{rel} < 1$ %) for the net rates. For reactions closer to partial equilibrium i.e., with higher PEI-shares, the error increases consistently, although there is no one-to-one correlation.

3.1.6. Degree of rate control analysis

To identify the rate controlling reactions, we computed the degree of rate control X_{RC} with respect to the CO source term (Campbell, 2017). Generally, the degree of rate control refers to a specific operating condition. Here, we want to gain an overview of elementary reactions that are rate controlling over the whole input range of our interpolation function. Figure S4 in the supplementary material provides a histogram of the degree of rate control values for reactions 7 and 13 obtained at 10,000 operating conditions sampled randomly from the input space. We also determined which reaction is most rate controlling for each operating condition i.e., has the largest degree of rate control. A histogram summarizing the result is presented in Figure S5. In agreement with (Mhadeshwar and Vlachos, 2005) reaction 13 is rate controlling over nearly the entire range of interest (for 92.7 % of all conditions). Reaction 2 is rate controlling for 6.9 % and reaction 7 for 0.4 % of the conditions. All other reactions are never the most rate controlling step.

To consolidate the degree of rate control results into a single value for each reaction, we computed the average degree of rate control $\bar{X}_{RC}$ as the mean of X_{RC} over all conditions. Table 3 shows $\bar{X}_{RC}$ for each reaction.

3.1.7. Systems with a single reaction path: Using the forward and reverse rates of the rate-determining step

Frequently, there will be a single reaction path towards a given product and the rate of the reaction is controlled by a single rate-determining step. In these cases, the species' source term can be

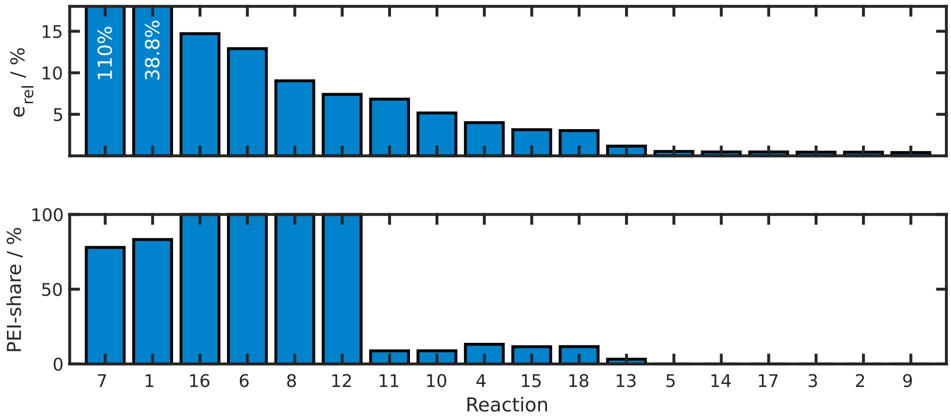


Fig. 3. Top: Mean relative error of net rates obtained as the difference of spline estimations of forward and reverse rates. Bottom: The fraction of conditions leading to partial equilibrium (PEI-share). For PEI-shares near 0 the spline map combination is as precise as the individual maps. If a significant PEI-share is observed, the spline map combination is less precise. In some cases, the error is several orders of magnitude higher.

represented by mapping the forward and reverse rates of this rate controlling step.

If the global reaction proceeds only in one direction, the rate-determining step will be far from equilibrium, allowing an accurate computation of the net rate. As the global reaction approaches equilibrium, also does the rate-determining step. However, it will still be the step furthest from equilibrium and hence yield the best available approximation of the net rate.

3.1.8. Systems with parallel reaction paths: Using the forward and reverse rates of several reactions including the rate-determining step

As shown in Figure 4, our example reaction mechanism is more complex, including several parallel reaction paths between CO and CO_2. For such systems with parallel reaction paths, a combination of several elementary reactions needs to be considered to describe the overall rate. These representative reactions need to be chosen in a way, that the entire flux towards the respective product is captured exactly once. For our example, this means that the source term $\dot{s}_{CO}$ for the formation/consumption of CO is described as:

$$\dot{s}_{CO} = \sum_i \alpha_i \cdot (r_{+,i} - r_{-,i}) \tag{19}$$

where the factor α_i considers the direction and stoichiometry of the reactions i that are part of the set of representative reactions. For reactions that are not part of the respective reaction set $\alpha_i = 0$.

In general, for each mechanism there will be multiple such combinations of representative reactions, defined by the choice of α_i. For our reaction mechanism, several representative reaction sets are listed in Table 3. Here, all α_i are either -1, 0 or 1.

To obtain an estimation of the CO source term, spline interpolations are constructed for the forward- and reverse rates of each reaction in the representative sets I-IV (Table 3). The source terms are then computed from the individual reaction rates as described by equation (20),

$$\dot{s}_{CO} = \sum_i \alpha_i \cdot [\exp(\text{spline}(\ln\ r_{+,i})) - \exp(\text{spline}(\ln\ r_{-,i}))] \tag{20}$$

where i runs over all reactions, $\text{spline}(\ln r_{+,i})$ is the spline interpolation of the logarithm of elementary forward rates, and $\text{spline}(\ln r_{-,i})$ the spline interpolation of the reverse rates, with the splines being a function of the species' partial pressures and the temperature.

Figure 5 shows the interpolation error of the CO source terms of different representative reaction sets and compares it with the PEI-share and degree of rate control. As expected, the best accuracy is obtained for reaction set II, which is furthest from partial equilibrium (lowest PEI-share) and is most rate controlling. It reproduces the source terms of CO with an error of 0.5 % which is 50 to 100 times more accurate than the established approaches of either mapping the source terms directly (49.9 % error) or mapping adsorption/desorption rates (reaction set I with 23.7 % error).

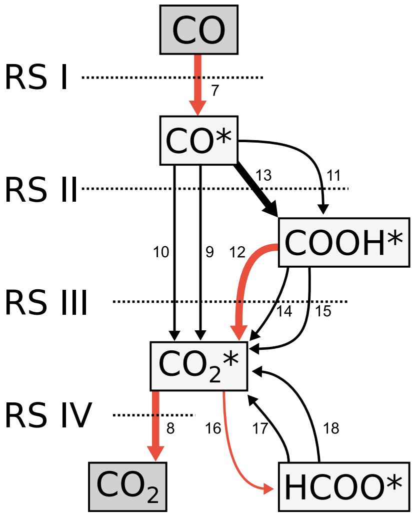


Fig. 4. Reaction path of carbon-containing species, build with ReNView (Gupta and Vlachos, 2020). Arrows represent reactions. Their width represents the net rate. They are colored red if the reaction is in partial equilibrium and black otherwise. Their direction is already adjusted to the sign of α_i as shown in Table 3. The numbers next to the arrows are the numbers of the corresponding reactions. The dashed lines cutting the path between CO and CO_2 are used to build the representative sets RS I-IV.

3.1.9. Reducing the interpolation effort by using combined forward and reverse rates

Our approach summarized by equation (20) requires the construction of a separate interpolation function for each forward- and reverse rate in the set of representative reactions. The number of interpolation functions can be reduced, if a single interpolation function is constructed for the sum of forward- and reverse rates,

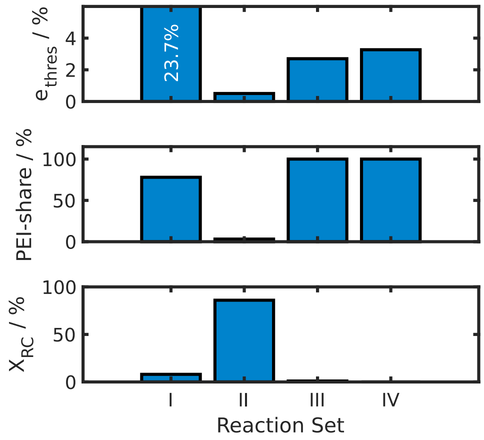


Fig. 5. Interpolation error, partial equilibrium status and degree of rate control for the four representative sets of reactions defined in Table 3. Reaction set I is equivalent to CO adsorption/desorption. Top: thresholded relative error e_{thres} of the interpolation of CO source terms using reaction sets I-IV. Middle: PEI-share of reaction sets I-IV. It is a measure of the average partial equilibrium status of the reaction set defined as the fraction of the 100,000 reaction conditions in the test set close to partial equilibrium i.e., having a PEI of 0.5 ± 0.05. The PEI for a set of reactions is computed by equation (18), replacing the rates r_{forward} and r_{reverse} by the sum of the forward- and reverse rates, respectively. Bottom: Total degree of rate control $\tilde{X}_{\text{RC}}$ of representative reaction sets I-IV.

respectively. As shown in equation (21), logarithmic scaling can be maintained.

$$\dot{s}_{CO} = \exp\left(\text{spline}\left(\ln\sum_i \alpha_i \cdot r_{+,i}\right)\right) - \exp\left(\text{spline}\left(\ln\sum_i \alpha_i \cdot r_{-,i}\right)\right) \tag{21}$$

Table 4 compares the prediction performance of equation (20) against equation (21). The latter achieves comparable performance with only two interpolation functions as opposed to eight.

3.1.10. Identifying the optimal set of representative reactions

Having demonstrated the ability of our scheme to provide an accurate interpolation of the source terms, a remaining open point is a systematic procedure to identify an optimal set of representative reactions. The results presented above provide some guidance in this direction, suggesting the following approach:

1. Apply degree of rate control analysis to identify the most rate controlling reaction.
2. Identify potential additional reaction pathways parallel to the rate controlling reaction.
3. Construct candidates of representative reaction sets containing the rate controlling reaction, in a way that each reaction set captures the entire flow contributing to the given source term exactly once.
4. Rank all sets according to their partial equilibrium index share.
5. Choose the one furthest from partial equilibrium.

Table 4
CO source term prediction error e_{thres} using representative set II according to equations (20) and (21). The former uses one spline per elementary reaction (eight in total). The latter uses two in total.

	CO error
Eight splines (equation (20))	0.506 %
Two splines (equation (21))	0.679 %

Ultimately, one can envisage an automated algorithm for the identification of the optimal representative reaction set. The algorithm would use a graph of the reaction mechanism to find all sets of reactions that capture the flux of a given source term exactly once. Afterwards the Partial Equilibrium Index share would be calculated for each set and the set with the smallest PEI-share would be chosen.

For the moment, the lack of an automated algorithm for the identification of a suitable representative reaction set does not limit the application of our scheme. Most reaction mechanisms contain a few tens of reactions. For mechanisms of this size, the representative sets can be identified manually using the guidelines above.

3.1.11. If the rate-determining step changes as a function of reaction conditions

In many reaction systems the rate-determining step changes as a function of the reaction conditions. So far, we have used a single set of representative reactions for the computation of the reaction rates over the entire input range of the spline map. It can be expected that in cases where the rate-determining step depends on the reaction conditions, the interpolation performance can be improved if different sets of representative reactions are applied in different regions of the input space, so that in each region the set of representative reactions is used that shows the lowest reversibility. Different implementations of such a scheme can be envisaged. Ideally, for each condition the set with the lowest reversibility is chosen. However, this is not guaranteed to lead to smooth outputs. Therefore, we propose to use a weighted sum of representative reaction sets, which will be smooth.

Weights should be high if a set s is far from partial equilibrium and low otherwise. In other words, weights should favor conditions with a high net rate compared to the single direction rates, as this leads to the best error propagation properties. We propose to quantify this using the measure Z_s, which is closely related to the partial equilibrium index, as shown in equation (22). We use absolute values to account for the sign of the net rate.

$$Z_s = \left|\frac{r_{\text{forward},s} - r_{\text{reverse},s}}{r_{\text{forward},s} + r_{\text{reverse},s}}\right| = |2 \cdot PEI - 1| \tag{22}$$

As described in equation (23) we normalize to guarantee that the sum of weights w_s is one.

$$w_s = \frac{Z_s}{\sum_s Z_s} \tag{23}$$

Finally, weights are considered in equation (21), yielding equation (24).

$$\dot{s}_{CO} = \sum_s w_s \cdot \exp\left[\text{spline}\left(\ln\sum_i \alpha_{i,s} \cdot r_{+,i}\right)\right] - w_s \cdot \exp\left[\text{spline}\left(\ln\sum_i \alpha_{i,s} \cdot r_{-,i}\right)\right] \tag{24}$$

This strategy yields an error of 0.303 % which is significantly better than the 0.506 % obtained using just set II as listed in Table 4. Even though we have shown in section 3.1.6 that a single reaction is rate-determining for more than 90 % of the reaction conditions considered, accounting for the other 10 % almost halved the prediction error. In cases where the rate-determining step changes significantly depending on the conditions, this strategy is expected to increase the interpolation accuracy even further.

3.2. Neural networks

So far, we used spline interpolation to demonstrate our approach. Spline interpolation is well suited for an initial discussion since it avoids difficulties arising from a non-deterministic learning process and there is little bias form the choice of hyperparameters. The real potential of our procedure develops in combination with machine learning methods like neural networks, that can overcome the curse of dimensionality and hence have the potential to be used for complex mechanisms e.g., including more gas phase species. Hereafter, we demonstrate the potential of our approach combined with neural networks. For the initial demonstration, we use small neural networks with a single hidden layer containing 20 nodes when learning the rates in a single direction and 40 nodes when learning the source terms directly.

First, we attempt fitting the source terms of CO and O_2 without logarithmic scaling using neural networks. The results are shown in Table 5. As observed with spline interpolation before, this approach does not provide a reasonable approximation of the source terms.

As the source term of O_2 is always negative, a logarithmic scaling of the source term can be applied. The neural network yields an error of 0.01 %, much better than the spline with 0.21 %.

As outlined above, the real challenge is the representation of the CO source terms, since these source terms assume positive and negative values depending on the reaction conditions, so that a logarithmic scaling cannot be applied. Again, we attempt to follow the literature approach and represent the CO source terms as the difference of approximated rates for CO adsorption and desorption. While the adsorption- and desorption rates both are represented well by the neural network, no useful approximation of the source term is obtained. Again, this can be explained by the fact that the CO adsorption/desorption reaction is in partial equilibrium for most reaction conditions. In consequence, the net rate is computed as the difference of two very large values leading to unfavourable error propagation.

Next, we trained neural networks to the forward and reverse rates of the different representative sets of elementary reactions listed in Table 3. Like for the adsorption/desorption reaction (representative set I), the net rates of the reactions in the representative sets II-IV add up to the CO source terms. To simplify the learning process, we compute a total forward rate as the sum of all forward reaction rates and a total reverse rate as the sum of all reverse reaction rates, as described by equation (21). Afterwards, neural networks are trained to predict the total forward- and reverse rates for each representative set separately. Again, total forward and reverse rates are predicted with high accuracy. However, accuracies of source term estimates differ widely. Results are shown in Table 6. As observed for spline interpolation, the best prediction accuracy is obtained using reaction set II which contains the rate controlling reaction and consequently is far from equilibrium. An error e_{thres} of 0.347 % is obtained. As Table 7 shows, this is about 200 times more accurate than the established approaches of direct mapping (61.8 %) or using adsorption/desorption rates (76.2 %). The neural networks also perform better than spline interpolation (0.679 %) using our technique.

So far, very small neural networks were used, having a single hidden layer with 20 nodes (322 parameters in total), and these simple networks achieved a performance similar to the spline maps. Figure 6 shows neural network prediction accuracy for different number of trainable parameters and training data used.

Table 5
Average error e_{thres} obtained by three neural network models each learning the source terms for O_2 (always negative) and CO (positive and negative values, depending on conditions) using 321 learnable parameters. It is computed according to equation (8) using a test set with steady state rates of 100,000 reaction conditions sampled randomly from the input range defined by Table 1. Models learning the CO and O_2 source terms without log scaling get stuck in a local minimum of predicting zero for almost every condition. Predicting zero leads to a relative error of 100 %. The corresponding thresholded relative error is around 50 % for O_2 and 60 % for CO as errors in very small rates are compared to a higher reference value. However, these models are of no use for kinetics estimations.

	O_2 error	CO error
Direct mapping $\dot{s} = f\left(T^{-1}, \ln\ p_i\right)$	51.8 %	61.8 %
Arrhenius-like scaling $\ln(\dot{s}) = f\left(T^{-1}, \ln\ p_i\right)$	0.0118 %	–

Table 6
Results for approximating CO source terms using different representative reaction sets I-IV learned with small neural networks (one with a single hidden layer of 20 nodes for each direction, 322 parameters in total). Set I is equivalent to mapping CO ads-/desorption. Set II contains the rate-determining step.

Representative set	I	II	III	IV
$e_{thres}(\dot{s}_{CO})/\%$	76.2	**0.347**	546	23,600

Table 7
Comparison of mapping approaches for CO source terms showing the prediction error e_{thres}. Direct mapping is the most straight-forward approach using the source terms as targets without scaling them. Mapping adsorption and desorption rates separately makes use of Arrhenius-like scaling as $\ln(r) = f\left(T^{-1}, \ln p_i\right)$. Although this yields precise rate estimates, source terms are computed as their difference which leads to unfavourable error propagation. Our new method maps the rates of the rate-determining steps. Their difference is much less error prone as the reactions are far from equilibrium. This leads to an increase in accuracy of factor 35 to 70 for splines and about 200 for small neural networks without increasing model complexity.

	$e_{thres}(\dot{s}_{CO})$	
Method	Neural network	Spline
Direct mapping	61.8 %	49.9 %
Ads-/desorption mapping	76.2 %	23.7 %
New approach (Mapping rate-determining reactions)	**0.347 %**	**0.679 %**

Table 8
Comparison between neural network (2342 learnable parameters) and spline interpolation models for CO source terms approximated using representative set II according to equation (21). The thresholded relative error was tested using the same steady state source terms of 100,000 conditions. The time needed to predict these 100,000 rates is given as prediction time $t_{predict}$. To estimate a speed up, we consider that the exact solution using integration yields the source term for all species whereas splines and neural networks must be evaluated for at least two species to calculate the source terms of all other species from that. The spline evaluation time does not depend on the interpolated property as all spline maps used are of equal size. The table shows the combined prediction time for forward and reverse rate of CO. The O_2 source term can be obtained by a single spline. In total, speed up can be estimated by comparing 1.5 times $t_{predict}$ to the computation time of the full mechanism using integration which is around 1,200 s for 100,000 conditions. The same factor is used for neural networks as the O_2 model shown in Table 5 is less than half the size of the CO model used in this table. Obtaining the missing source terms after the model evaluations is computationally cheap and can be neglected in this estimation.

	neural network	Spline
$e_{thres}/\%$	**0.00584**	0.679
# Data	**25,000**	531,441
$t_{predict}$/ s	**0.08**	3.95
Storage space / kB	**135**	1,370,000
Speed up	**~10,000**	~200

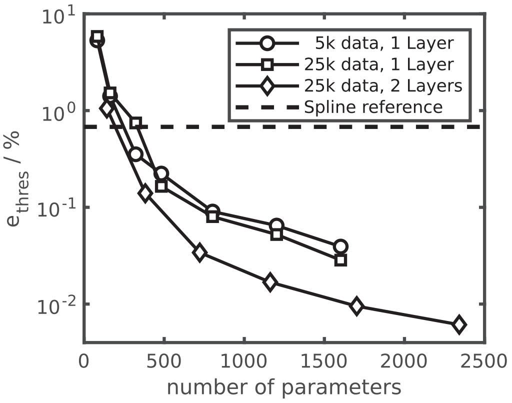


Fig. 6. Thresholded relative error obtained from neural networks predicting the CO source term using representative set II depending on the total number of learnable parameters used. Circles: neural networks with a single hidden layer learned with 5,000 data available during training. Squares: neural networks with a single hidden layer learned with 25,000 data available during training. Diamonds: neural networks with two hidden layers learned with 25,000 data available during training. Dashed line: error obtained by spline regression for reference. The simple neural networks used in Table 5 and Table 6 with 20 hidden nodes for the forward- and reverse rate use 322 parameters in total.

Increasing the number of trainable parameters to 2342 reduces the prediction error to 0.0058%. Although this accuracy is well above that of the kinetic parameters, the obtained accuracy suggests that our method can be applied to much larger and more complex reaction mechanisms.

Also, the new scheme obtains high prediction accuracy with a very small training set containing 5000 data points. This offers the opportunity to learn kinetic models that are numerically expensive to evaluate, such as kinetic Monte Carlo simulations.

Analogous to Figure 6 showing neural network prediction accuracy for different number of trainable parameters using the new approach (mapping rate-determining reactions), Figure S6 in the supporting material shows prediction accuracy for different number of parameters using direct mapping of CO source terms. Within the same parameter range, no model suitable for reactor simulation is found. Most models simply predict a very low value regardless of the condition, reducing the relative error to 100 % and the thresholded error to around 60 %.

Finally, Table 8 compares the performance of neural networks with conventional spline interpolation. The neural networks outperform the splines by several orders of magnitude regarding all performance criteria of interest: They obtain higher accuracy, require less training data, reduce storage requirements and evaluate much faster. The only disadvantage of the neural networks is that they require time for training. This is offset by the reduced number of required training points. Furthermore, the training only happens once for a given mechanism.

Even though the absolute values of the error depend on the applied error measure, qualitatively similar results were obtained if we used other error measures, for example a simple relative error. Throughout this paper we used the mean thresholded relative error defined by equation (8). Table S2 in the supplementary material compares the average errors obtained using different error measures. The best neural network yields average errors for the CO source term well below 0.1 % for all investigated error measures.

3.2.1. Validation in a Plug flow reactor simulation

To demonstrate the usability of the neural network models, we simulated a simple isobaric and isothermal plug flow reactor under PROX conditions. Figure 7 shows the mole fractions of CO and H_2 over the length of the reactor using the exact steady state solutions obtained by numerical solution of the rate equations (full lines) compared to the results when neural networks are used to estimate the source terms instead (dotted line). Visually there is no difference between the solutions obtained. The lower part of Figure 7 reveals the relative error between both solutions to be far below 1 %. Note however, that calling the neural network models is about 10,000 times faster than calculating the exact steady state source terms.

4. Conclusion

Current schemes for approximating complex surface kinetics do not work well for species that appear as both, reactants and products depending on the reaction conditions. We propose a new scheme to overcome this limitation by learning the elementary rates of the rate-determining step. For the PROX mechanism used as our demonstration example, the new scheme improves the accuracy of the interpolated source terms by orders of magnitude.

In a first step we used spline interpolation to show that forward and reverse rates can be represented with high accuracy each.

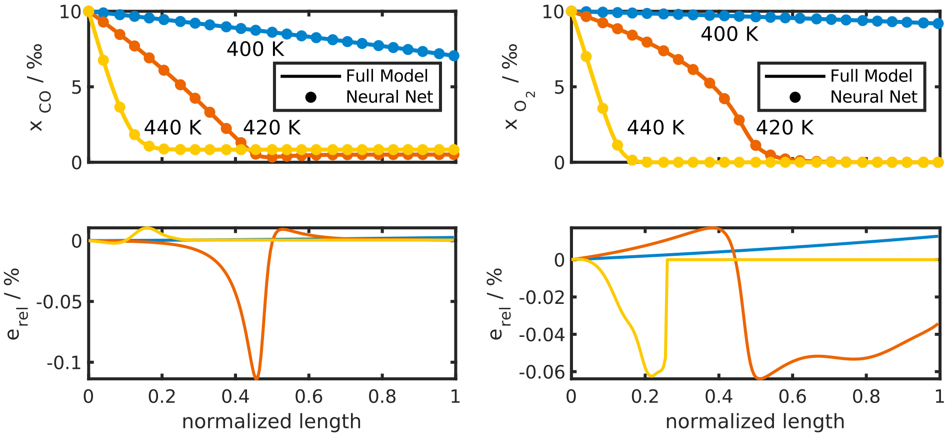


Fig. 7. Mole fractions of CO (left) and O_2 (right) plotted against the length of a simulated isobaric and isothermal Plug Flow Reactor model for three different temperatures. Full lines show the exact solution while circles show the result obtained when source terms are estimated using neural networks (2342 parameters). The relative error of the mole fractions obtained by neural network solution is shown below. Inlet conditions are shown in section 2.6.

However, the accuracy of net rates suffers if a reaction is partially equilibrated, so that the net rate is computed as the difference of two large numbers. The best accuracy is obtained for reactions far from equilibrium e.g., the rate-determining step. We propose a new scheme where source terms are represented by a set of elementary rates containing the rate-determining step. If a mechanism contains a single path from reactant to product, the source terms are computed as the difference of estimations for forward- and reverse rates of a single reaction. In presence of parallel reaction paths, multiple reactions need to be considered to capture the entire flux from reactants to products. This method reproduces the source terms of CO with an error of 0.5 % which is 50 to 100 times more accurate than the established approaches of either mapping the source terms directly (49.9 % error) or mapping adsorption/desorption rates (23.7 % error).

The real potential of the new scheme develops in combination with machine learning techniques. Very small neural networks with a single layer of 20 hidden nodes (322 parameters in total) already provide a prediction error of 0.35 %. This is about 200 times more accurate than the same neural networks used with the established approaches. Increasing the network size to still moderate 30 nodes in two hidden layers (2342 parameters in total) reduces the error to 0.0058 %. To our knowledge, such an accuracy is unprecedented in the literature, especially considering the complexity of our demonstration example mechanism in terms of number of gas species, the complexity of the reaction network and the broad range of conditions chosen. Although this accuracy is well above that of the kinetic parameters, it suggests that our method can be applied to much larger and more complex reaction mechanisms. The proposed neural network models are easily implemented in C and can therefore replace the rate-calculating function in e. g. COMSOL Multiphysics® or OpenFOAM. Besides neural networks and splines, it is expected that our approach should also improve the accuracy of other interpolation and machine learning methods currently used to implement detailed reaction mechanisms in reactor simulations, such as *In Situ* Adaptive Tabulation and random forests.

We further believe that beyond catalysis, the scheme proposed in this paper can be adopted to other systems described by complex reaction networks such as atmospheric chemistry (Keller and Evans, 2019; Kelp et al., 2022), combustion or systems biology.

In catalysis further applications of our scheme arise from the fact that due to the need for high contact area, catalytic systems generally are multi-scale. We have previously shown that a rate learning approach can be extended to also include the transport phenomena at the lower scales in the learned rate function. In this way effective source terms can be predicted that describe the solution of the reaction diffusion equation in the catalyst domain (Nien et al., 2013), or even the solution of the balance equations describing a volume element of the reactor scale simulation (Scheuer et al., 2011).

Overall, the approach proposed in this paper will significantly facilitate the application of detailed mechanistic knowledge in the simulation-based design of realistic catalytic systems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was funded by BMBF in the framework of the project ML-MORE (contract 05M20RDA).

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.ces.2022.117964.

References

Blasco, J.A., Fueyo, N., Larroya, J.C., Dopazo, C., Chen, Y.J., 1999. A single-step time-integrator of a methane-air chemical system using artificial neural networks. Comput. Chem. Eng. 23, 1127–1133. https://doi.org/10.1016/S0098-1354(99)00278-1.

Blasi, J.M., Kee, R.J., 2016. In situ adaptive tabulation (ISAT) to accelerate transient computational fluid dynamics with complex heterogeneous chemical kinetics. Comput. Chem. Eng. 84, 36–42. https://doi.org/10.1016/J.COMPCHEMENG.2015.08.020.

Bracconi, M., Maestri, M., Cuoci, A., 2017. In situ adaptive tabulation for the CFD simulation of heterogeneous reactors based on operator-splitting algorithm. AIChE J. 63, 95–104. https://doi.org/10.1002/aic.15441.

Bracconi, M., Maestri, M., 2020. Training set design for machine learning techniques applied to the approximation of computationally intensive first-principles kinetic models. Chem. Eng. J. 400,. https://doi.org/10.1016/j.cej.2020.125469 125469.

Campbell, C.T., 2017. The degree of rate control: a powerful tool for catalysis research. ACS Catal. 7, 2770–2779. https://doi.org/10.1021/acscatal.7b00115.

Christo, F.C., Masri, A.R., Nebot, E.M., 1996. Artificial neural network implementation of chemistry with pdf simulation of H2/CO2 flames. Combust. Flame 2180, 406–427. https://doi.org/10.1016/0010-2180(95)00250-2.

Daymo, E.A., Hettel, M., Deutschmann, O., Wehinger, G.D., 2022. Accelerating particle-resolved CFD simulations of catalytic fixed-bed reactors with DUO. Chem. Eng. Sci. 250,. https://doi.org/10.1016/j.ces.2021.117408 117408.

de Boor, C., 1978. A Practical Guide to Splines. Springer-Verlag, Springer-Verlag, New York.

Foresee, F.D., Hagan, M.T., 1997. Gauss-Newton approximation to Bayesian learning, in. In: Proceedings of International Conference on Neural Networks (ICNN'97). IEEE, pp. 1930–1935.

Gupta, U., Vlachos, D.G., 2020. Reaction Network Viewer (ReNView): An open-source framework for reaction path visualization of chemical reaction systems. SoftwareX 11,. https://doi.org/10.1016/j.softx.2020.100442 100442.

Hauptmann, W., Votsmeier, M., Vogel, H., Vlachos, D.G., 2011. Modeling the simultaneous oxidation of CO and H2 on Pt - Promoting effect of H2 on the CO-light-off. Appl. Catal. A Gen. 397, 174–182. https://doi.org/10.1016/j.apcata.2011.02.031.

Ji, W., Deng, S., 2021. KiNet: A Deep Neural Network Representation of Chemical Kinetics. arXiv Prepr. arXiv2108.00455 1–23.

Keller, A.C., Evans, J.M., 2019. Application of random forest regression to the calculation of gas-phase chemistry within the GEOS-Chem chemistry model v10. Geosci. Model Dev. 12, 1209–1225. https://doi.org/10.5194/gmd-12-1209-2019.

Kelp, M.M., Jacob, D.J., Lin, H., Sulprizio, M.P., 2022. An online-learned neural network chemical solver for stable long-term global simulations of atmospheric chemistry. J. Adv. Model. Earth Syst. 14, 1–16. https://doi.org/10.1029/2021ms002926.

Klingenberger, M., Hirsch, O., Votsmeier, M., 2017. Efficient interpolation of precomputed kinetic data employing reduced multivariate Hermite Splines. Comput. Chem. Eng. 98, 21–30. https://doi.org/10.1016/j.compchemeng.2016.12.005.

Kumar, A., Mazumder, S., 2011. Adaptation and application of the In Situ Adaptive Tabulation (ISAT) procedure to reacting flow calculations with complex surface chemistry. Comput. Chem. Eng. 35, 1317–1327. https://doi.org/10.1016/j.compchemeng.2010.10.005.

Lorenzi, J.M., Stecher, T., Reuter, K., Matera, S., 2017. Local-metrics error-based Shepard interpolation as surrogate for highly non-linear material models in high dimensions. J. Chem. Phys. 147 (16), 164106.

MacKay, D.J.C., 1992. Bayesian Interpolation. Neural Comput. 4, 415–447. https://doi.org/10.1162/neco.1992.4.3.415.

Matera, S., Maestri, M., Cuoci, A., Reuter, K., 2014. Predictive-quality surface reaction chemistry in real reactor models: INTEGRATING first-principles kinetic Monte Carlo simulations into computational fluid dynamics. ACS Catal. 4, 4081–4092. https://doi.org/10.1021/cs501154e.

Mazumder, S., 2005. Adaptation of the in situ adaptive tabulation (ISAT) procedure for efficient computation of surface reactions. Comput. Chem. Eng. 30, 115–124. https://doi.org/10.1016/j.compchemeng.2005.08.008.

Mhadeshwar, A.B., Vlachos, D.G., 2004. Microkinetic modeling for water-promoted CO oxidation, water-gas shift, and preferential oxidation of CO on Pt. J. Phys. Chem. B 108, 15246–15258. https://doi.org/10.1021/jp048698g.

Mhadeshwar, A.B., Vlachos, D.G., 2005. Is the water-gas shift reaction on Pt simple? Computer-aided microkinetic model reduction, lumped rate expression, and rate-determining step. Catal. Today 105, 162–172. https://doi.org/10.1016/j.cattod.2005.04.003.

Micale, D., Ferroni, C., Uglietti, R., Bracconi, M., Maestri, M., 2022. Computational fluid dynamics of reacting flows at surfaces: Methodologies and applications. Chemie Ingenieur Technik 94 (5), 634–651.

Nien, T., Mmbaga, J.P., Hayes, R.E., Votsmeier, M., 2013. Hierarchical multi-scale model reduction in the simulation of catalytic converters. Chem. Eng. Sci. 93, 362–375. https://doi.org/10.1016/j.ces.2013.01.059.

Partopour, B., Dixon, A.G., 2016. Computationally efficient incorporation of microkinetics into resolved-particle CFD simulations of fixed-bed reactors. Comput. Chem. Eng. 88, 126–134. https://doi.org/10.1016/j.compchemeng.2016.02.015.

Partopour, B., Paffenroth, R.C., Dixon, A.G., 2018. Random Forests for mapping and analysis of microkinetics models. Comput. Chem. Eng. 115, 286–294. https://doi.org/10.1016/j.compchemeng.2018.04.019.

Scheuer, A., Hirsch, O., Hayes, R., Vogel, H., Votsmeier, M., 2011. Efficient simulation of an ammonia oxidation reactor using a solution mapping approach. Catal. Today 175, 141–146. https://doi.org/10.1016/j.cattod.2011.03.036.

Scheuer, A., Hauptmann, W., Drochner, A., Gieshoff, J., Vogel, H., Votsmeier, M., 2012. Dual layer automotive ammonia oxidation catalysts: Experiments and computer simulation. Appl. Catal. B Environ. 111–112, 445–455. https://doi.org/10.1016/j.apcatb.2011.10.032.

Sutton, J.E., Lorenzi, J.M., Krogel, J.T., Xiong, Q., Pannala, S., Matera, S., Savara, A., 2018. Electrons to reactors multiscale modeling: Catalytic CO oxidation over RuO2. ACS Catal. 8, 5002–5016. https://doi.org/10.1021/acscatal.8b00713.

Van Keken, P.E., Yuen, D.A., Petzold, L.R., 1995. DASPK: A new high order and adaptive time-integration technique with applications to mantle convection with strongly temperature-and pressure-dependent rheology. Geophys. Astrophys. Fluid Dyn. 80, 57–74. https://doi.org/10.1080/03091929508229763.

Varshney, A., Armaou, A., 2008. Reduced order modeling and dynamic optimization of multiscale PDE/kMC process systems. Comput. Chem. Eng. 32, 2136–2143. https://doi.org/10.1016/j.compchemeng.2008.03.004.

Votsmeier, M., 2009. Efficient implementation of detailed surface chemistry into reactor models using mapped rate data. Chem. Eng. Sci. 64, 1384–1389. https://doi.org/10.1016/j.ces.2008.12.006.

Votsmeier, M., Scheuer, A., Drochner, A., Vogel, H., Gieshoff, J., 2010. Simulation of automotive NH3 oxidation catalysts based on pre-computed rate data from mechanistic surface kinetics. Catal. Today 151, 271–277. https://doi.org/10.1016/j.cattod.2010.01.018.

Wan, K., Barnaud, C., Vervisch, L., Domingo, P., 2020. Chemistry reduction using machine learning trained from non-premixed micro-mixing modeling: Application to DNS of a syngas turbulent oxy-flame with side-wall effects. Combust. Flame 220, 119–129. https://doi.org/10.1016/j.combustflame.2020.06.008.

Wang, S., Fan, K., Luo, N., Cao, Y., Wu, F., Zhang, C., Heller, K.A., You, L., 2019. Massive computational acceleration by using neural networks to emulate mechanism-based biological models. Nat. Commun. 10, 1–9. https://doi.org/10.1038/s41467-019-12342-y.

Supplementary material

Efficient machine learning based surrogate models for surface kinetics by approximating the rates of the rate-determining steps

Felix A. Döppel[a], Martin Votsmeier[a,b,*]

[a] *Technische Universität Darmstadt, Alarich-Weiss-Str. 8, 64287 Darmstadt, Germany*
[b] *Umicore AG & Co. KG, Rodenbacher Chaussee 4, 63457 Hanau, Germany*

**Corresponding author:* *martin.votsmeier@eu.umicore.com*

S1 The reaction mechanism and kinetic parameters

Table S 1 contains the mechanism used in this work, as provided in table 1 of HAUPTMANN *et al.* (2011). Shown are the sticking coefficient s_0 for adsorption reactions, the preexponential factor A_0 for all other reactions, the temperature exponent β, the activation energy E_A and the surface coverage dependencies of the activation energy $dE_A/d\theta$.

Table S 1: Mechanism used in this work taken from table 1 of H*AUPTMANN et al. (2011). Forward direction parameters are shaded, reverse direction parameters are not.*

No.	Reaction	s_0 or A_0	β	E_A	$\frac{dE_A}{d\theta_H}$	$\frac{dE_A}{d\theta_O}$	$\frac{dE_A}{d\theta_{OH}}$	$\frac{dE_A}{d\theta_{H_2O}}$	$\frac{dE_A}{d\theta_{CO}}$
		unitless or s^{-1}	unitless	kJ/mol	kJ/mol	kJ/mol	kJ/mol	kJ/mol	kJ/mol
H_2 oxidation on Pt									
R1	H_2 +2 * ⇌ 2 H*	1.287×10^{-1}	0.8584	-	-	-	-	-	-
		7.953×10^{12}	1.911	79.09	-25.10	0	0	0	0
R2	O_2 + 2 * ⇌ 2 O*	5.423×10^{-1}	0.7656	-	-	-	-	-	-
		8.406×10^{12}	0.9275	208.9	0	-133.9	0	0	0
R3	OH* + * ⇌ O* + H*	1.950×10^{12}	1.3286	111.2	6.7321	-4.87	0	62.89	0
		6.325×10^{12}	1.0812	32.48	-5.820	22.26	0	-41.71	0
R4	H_2O* + * ⇌ OH* + H*	9.358×10^{12}	-0.3949	74.57	4.795	50.33	52.30	-43.25	0
		9.989×10^{12}	-0.3664	53.43	-7.757	-87.75	-52.30	71.81	0
R5	H_2O* + O* ⇌ 2 OH*	4.316×10^{10}	0.3262	36.74	0	70.10	52.30	-83.68	0
		1.700×10^{10}	0.5285	94.32	0	-139.12	-52.30	136.0	0
R6	H_2O + * ⇌ H_2O*	1.084×10^{-1}	1.1624	-	-	-	-	-	-
		2.033×10^{12}	2.490	39.98	0	0	104.6	-10.46	0
CO oxidation on Pt									
R7	CO + * ⇌ CO*	1.000×10^{0}	0	-	-	-	-	-	-
		5.659×10^{15}	0.3946	165.9	0	0	0	0	-62.70
R8	CO_2 + * ⇌ CO_2*	1.950×10^{-1}	0.2500	-	-	-	-	-	-
		3.626×10^{12}	0.2459	11.60	0	0	0	0	0
R9	CO_2* + * ⇌ CO* + O*	4.178×10^{10}	-0.2778	110.4	0	4.27	0	0	24.21
		2.393×10^{11}	0.4558	85.42	0	-17.69	0	0	-38.53
Coupling reactions between CO- and H_2 oxidation									
R10	CO_2* + H* ⇌ CO* + OH*	8.031×10^{8}	-0.3259	23.22	-6.276	60.25	0	-45.56	18.12
		1.245×10^{9}	0.8237	76.91	6.276	-77.83	0	59.04	-44.63
R11	COOH* + * ⇌ CO* + OH*	8.426×10^{8}	0.02577	22.66	0	60.25	0	-45.56	18.13
		1.187×10^{9}	0.4719	77.46	0	-77.83	0	59.94	-44.63
R12	COOH* + * ⇌ CO_2* + H*	1.058×10^{11}	0.5812	4.975	6.276	0	0	0	0
		9.454×10^{10}	-0.1098	6.079	-6.2844	0	0	0	0
R13	CO* + H_2O* ⇌ COOH* + H*	1.103×10^{11}	0.4911	98.71	4.904	0	52.30	-5.230	-31.38
		9.070×10^{10}	-0.02778	22.79	-7.657	0	-52.30	5.230	31.38
R14	CO_2* + OH* ⇌ COOH* + O*	5.349×10^{10}	0.05272	109.5	0	-41.13	0	52.30	0
		1.870×10^{11}	0.4515	29.70	0	30.00	0	-52.30	0
R15	CO_2* + H_2O* ⇌COOH* + OH*	8.642×10^{10}	-0.0481	71.31	0	52.45	52.30	-44.87	0
		1.157×10^{11}	0.5647	49.07	0	-85.62	-52.30	70.19	0
R16	CO_2* + H* ⇌ HCOO* + *	1.117×10^{11}	-0.07525	73.49	-12.55	0	0	0	0
		8.957×10^{10}	0.4218	0.00	0	0	0	0	0
R17	CO_2* + OH* ⇌ HCOO* + O*	6.168×10^{10}	-0.3443	154.8	0	-67.47	0	100.2	0
		1.621×10^{11}	-0.1617	2.577	0	3.657	0	-4.351	0
R18	CO_2* + H_2O* ⇌ HCOO* + OH*	1.022×10^{11}	-0.3574	111.8	0	118.7	90.99	-98.17	0
		9.785×10^{10}	0.06665	17.15	0	-19.40	-13.61	16.89	0

S2 Implementation of the loss function for neural network training

During training of the neural networks, backpropagation is used to minimize the Mean Squared Error loss $\mathcal{L}_{MSE}$ as given in equation (S1) with the number of test points N, the target value y_i, the estimated value $\widehat{y_i}$ and weighting factor W_i.

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N} W_i \cdot \left(y_i - \widehat{y_i}\right)^2 \tag{S1}$$

When fitting source terms directly, $y_i = \dot{s}_i$. Choosing W_i as given by equation (S2) leads to minimization of the mean squared thresholded relative error $\mathcal{L}_{\dot{s}}$ as shown in equation (S3).

$$W_i = \frac{1}{max\left(|\dot{s}_\mathrm{i}|, \dot{s}_{i,thres}\right)}^{2} \tag{S2}$$

$$\begin{aligned} \mathcal{L}_{MSE} &= \frac{1}{N}\sum_{i=1}^{N} \left(\frac{1}{max\left(|\dot{s}_\mathrm{i}|, \dot{s}_{i,thres}\right)}\right)^2 \cdot \left(\dot{s}_i - \widehat{\dot{s}_i}\right)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N} \left(\frac{\dot{s}_i - \widehat{\dot{s}_i}}{max\left(|\dot{s}_\mathrm{i}|, \dot{s}_{i,thres}\right)}\right)^2 \\ &= \mathcal{L}_{\dot{s}} \end{aligned} \tag{S3}$$

In the case of logarithmically scaled source terms as used for O_2, $y_i = ln\, \dot{s}_i$. Assuming low errors and hence $\Delta\, ln\, \dot{s} \approx \Delta\dot{s}/\dot{s}$, MSE approximates the Mean Squared Relative Error of the rates as shown in equation (S4).

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N} W_i \cdot \left(ln\, \dot{s}_i - \widehat{ln\, \dot{s}_i}\right)^2 \approx \frac{1}{N}\sum_{i=1}^{N} W_i \cdot \left(\frac{\dot{s}_i - \widehat{\dot{s}_i}}{\dot{s}_i}\right)^2 \tag{S4}$$

Choosing W_i as given by equation (S5) leads to minimization of the mean squared thresholded relative error $\mathcal{L}_{\dot{s}}$ as shown in equation (S6).

$$W_i = \frac{\dot{s}_i}{max\left(|\dot{s}_\mathrm{i}|, \dot{s}_{i,thres}\right)}^{2} \tag{S5}$$

$$\begin{aligned} \mathcal{L}_{MSE} &\approx \frac{1}{N}\sum_{i=1}^{N} \left(\frac{\dot{s}_i}{max\left(|\dot{s}_\mathrm{i}|, \dot{s}_{i,thres}\right)}\right)^2 \cdot \left(\frac{\dot{s}_i - \widehat{\dot{s}_i}}{\dot{s}_i}\right)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N} \left(\frac{\dot{s}_{i_i} - \widehat{\dot{s}_i}}{max\left(|\dot{s}_\mathrm{i}|, \dot{s}_{i,thres}\right)}\right)^2 \\ &= \mathcal{L}_{\dot{s}} \end{aligned} \tag{S6}$$

In the case of logarithmically scaled rates as used for representative sets I-IV, $y_i = ln\, r_i$. Assuming low errors and hence $\Delta\, ln\, r \approx \Delta r/\mathrm{r}$, MSE approximates the Mean Squared Relative Error of the rates as shown in equation (S7).

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N} W_i \cdot \left(ln\,\mathrm{r}_i - \widehat{ln\,\mathrm{r}_i}\right)^2 \approx \frac{1}{N}\sum_{i=1}^{N} W_i \cdot \left(\frac{\mathrm{r}_i - \hat{\mathrm{r}}_i}{\mathrm{r}_i}\right)^2 \quad \text{(S7)}$$

Choosing W_i as given by equation (S8) leads to minimization of the mean squared thresholded relative error as shown in equation (S9).

$$W_i = \frac{\mathrm{r}_i}{max\left(|\dot{s}_\mathrm{i}|, \dot{s}_{i,thres}\right)}^2 \quad \text{(S8)}$$

$$\begin{aligned} \mathcal{L}_{MSE} &\approx \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\mathrm{r}_i}{max\left(|\dot{s}_\mathrm{i}|, \dot{s}_{i,thres}\right)}\right)^2 \cdot \left(\frac{\mathrm{r}_i - \hat{\mathrm{r}}_i}{\mathrm{r}_i}\right)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}\left(\frac{r_i - \hat{\mathrm{r}}_\imath}{max\left(|\dot{s}_\mathrm{i}|, \dot{s}_{i,thres}\right)}\right)^2 \\ &= \mathcal{L}_r \end{aligned} \quad \text{(S9)}$$

To minimize the standard (non-thresholded) relative error instead of the thresholded relative error, $\dot{s}_{i,thres}$ can be excluded from the weights W_i as follows.

$$W_i = \left(\frac{\mathrm{r}_i}{\dot{s}_\mathrm{i}}\right)^2 \quad \text{(S10)}$$

In consequence, the loss function approximates $\mathcal{L}_r^{rel}$.

$$\begin{aligned} \mathcal{L}_{MSE} &\approx \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\mathrm{r}_i}{\dot{s}_\mathrm{i}}\right)^2 \cdot \left(\frac{\mathrm{r}_i - \hat{\mathrm{r}}_i}{\mathrm{r}_i}\right)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}\left(\frac{r_i - \hat{\mathrm{r}}_\imath}{\dot{s}_\mathrm{i}}\right)^2 \\ &= \mathcal{L}_r^{rel} \end{aligned} \quad \text{(S11)}$$

S3 Error measure

This paper uses a thresholded relative error e_{thres} as the error measure for source terms. The goal of this section is to show the equivalence of equations (7) and (8). The weights w_i are defined as

$$w_i \ = \begin{cases} 1 \ if \ |\dot{s}_i| \ \geq \ \dot{s}_{i,thres} \\ \dfrac{\dot{s}_i}{\dot{s}_{i,thres}} \ else \end{cases} \tag{S12}$$

$$= \ min\left(1, \frac{|\dot{s}_i|}{\dot{s}_{i,thres}}\right) \tag{S13}$$

$$= \ \frac{1}{max\left(1, \dfrac{\dot{s}_{i,thres}}{|\dot{s}_i|}\right)} \tag{S14}$$

$$= \left|\frac{1}{max\left(1, \dfrac{\dot{s}_{i,thres}}{|\dot{s}_i|}\right)}\right| \tag{S15}$$

The thresholded relative error e_{thres} is defined as

$$e_{thres} = \frac{1}{N}\sum_{i=1}^{N} w_i \left|\frac{\widehat{\dot{s}_\imath} \ - \ \dot{s}_i}{\dot{s}_i}\right| \tag{S16}$$

$$= \frac{1}{N}\sum_{i=1}^{N} w_i \ \frac{\left|\widehat{\dot{s}_\imath} \ - \ \dot{s}_i\right|}{|\dot{s}_i|}. \tag{S17}$$

Using the definition of the weights

$$= \frac{1}{N}\sum_{i=1}^{N} \frac{\left|\widehat{\dot{s}_\imath} \ - \ \dot{s}_i\right|}{\left|max\left(1, \dfrac{\dot{s}_{i,thres}}{|\dot{s}_i|}\right)\right| \cdot |\dot{s}_i|} \tag{S18}$$

$$= \frac{1}{N}\sum_{i=1}^{N} \frac{\left|\widehat{\dot{s}_\imath} \ - \ \dot{s}_i\right|}{\left|max\left(|\dot{s}_i|, \dot{s}_{i,thres}\right) \right|} \tag{S19}$$

$$= \ \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\widehat{\dot{s}_\imath} \ - \ \dot{s}_i}{max(|\dot{s}_i|, \dot{s}_{i,thres})}\right|. \tag{S20}$$

S4 Comparison of different error measures

The absolute values of the average errors depend on the choice of the error measure. Throughout the paper, the error of source terms is measured by a mean thresholded relative error e_{thres} defined by equations 6-9. To simplify the comparison with other papers and with our own earlier work where we used the root mean square thresholded relative error e_{thres}^{RMS} (Scheuer et al., 2011, 2012; Votsmeier et al., 2010), we compare the average errors obtained using different error measures in Table S2 below. Besides the errors e_{thres} and e_{thres}^{RMS} , the table also lists a simple mean relative error (equation 10). The comparison is performed for the spline interpolation, the small neural network with one hidden layer containing 20 nodes and the best Neural Network of Figure 6 which has two hidden layers with 30 nodes each.

Table S 2: Comparison of different error measures for models approximating CO source terms using the representative set of reactions II. The results listed for Neural Networks are obtained as the median error of three differently initialized models. The root mean square thresholded relative error e_{thres}^{RMS} is about one order of magnitude higher than the mean thresholded relative error e_{thres}. Also shown is the relative error e_{rel}. The loss function $\mathcal{L}_r$ aims to minimize e_{thres} while $\mathcal{L}_r^{rel}$ aims to minimize e_{rel}. $\mathcal{L}_{MSE}$ minimizes the relative error of the forward and reverse rates.

Surrogate Model	Nodes	Loss function	e_{thres} / %	e_{thres}^{RMS} / %	e_{rel} / %
Spline	-	-	0.679	9.25	2.39
Neural Network	20	$\mathcal{L}_r$	0.347	2.74	183
Neural Network	20	$\mathcal{L}_r^{rel}$	0.745	10.6	1.92
Neural Network	20	$\mathcal{L}_{MSE}$	1.03	17.2	2.90
Neural Network	2×30	$\mathcal{L}_r$	**0.00584**	**0.0728**	2.72
Neural Network	2×30	$\mathcal{L}_r^{rel}$	0.00640	0.142	**0.0150**

We also show the influence of different loss functions on neural network results. Table S2 shows the importance of adjusting the loss function to the error measure. The loss function $\mathcal{L}_r$ is designed to minimize e_{thres} and assigns a low weight to operating points with source terms below a threshold (see section S2). This leads to a small thresholded error e_{thres} but yields a high relative error e_{rel}. Using the loss function $\mathcal{L}_r^{rel}$ instead, the relative error is minimized for all operating points leading to a relative error of 0.015 %. However, this comes at the cost of an increased thresholded error e_{thres}. Using the common mean squared error loss without any error weights basically minimizes the relative error of the forward- and reverse rates separately also yielding a reasonable error e_{thres} as shown for the small Neural Network.

S5 Partial Equilibrium Index data for all reactions

Figure S 1 - Figure S 3 show the Partial Equilibrium Index distributions for all 18 reactions. The logarithmic y-axis shows that for some reactions there is not a single condition leading to partial equilibrium. A reaction is considered in partial equilibrium if the PEI is between 0.45 and 0.55 (Gupta and Vlachos, 2020). This Interval is shaded gray in the figure. The portion of conditions leading to partial equilibrium (PEI-share) is given in each subfigure. If the PEI-share is low, the net rate can be estimated by building two separate models for its respective forward and reverse rate.

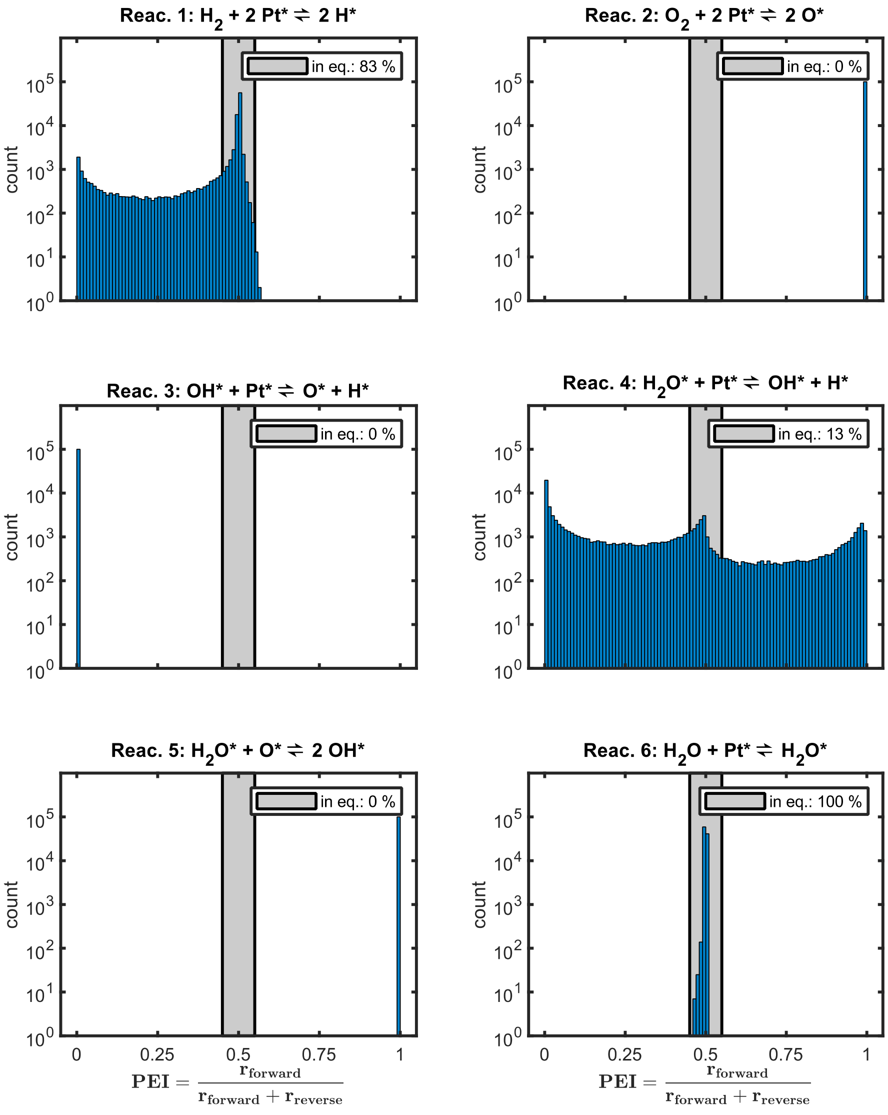


Figure S 1: Histogram of the Partial Equilibrium Index PEI of Reactions 1-6. Occurrences in the shaded area are in partial equilibrium.

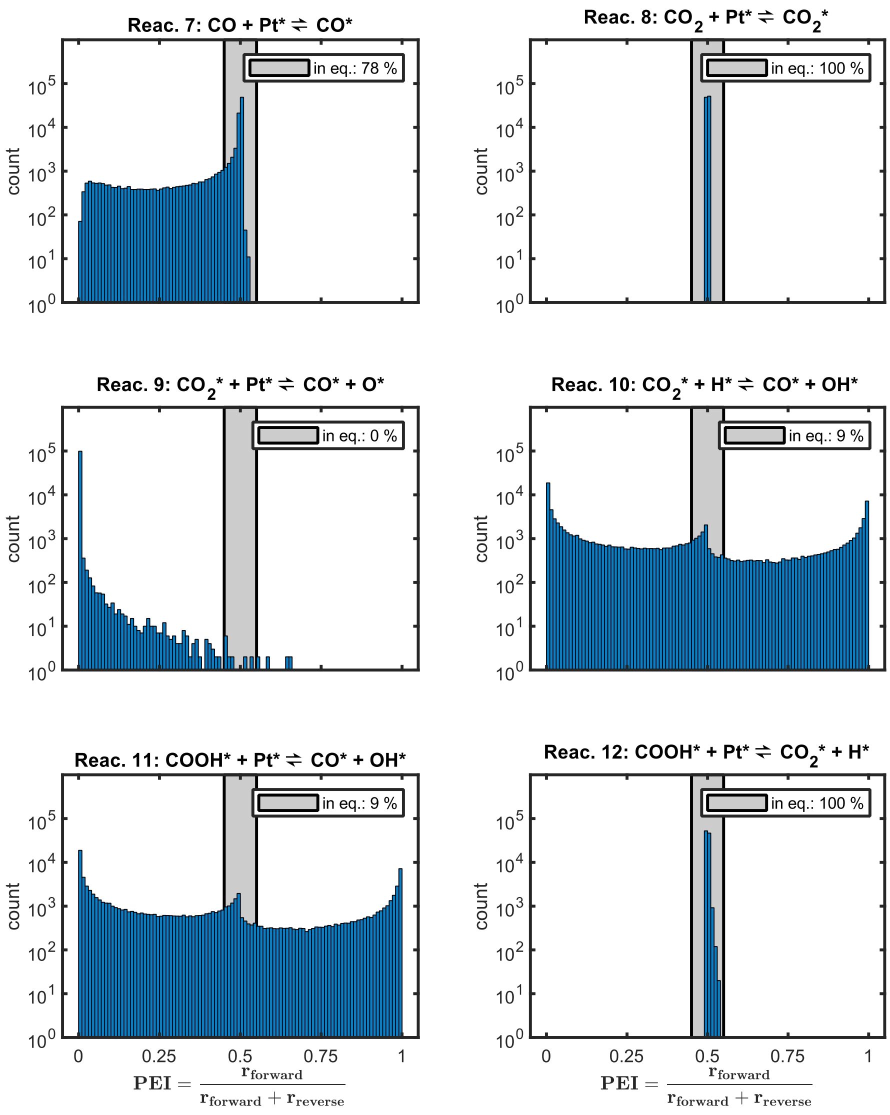


Figure S 2: Histogram of the Partial Equilibrium Index PEI of Reactions 7-12. Occurrences in the shaded area are in partial equilibrium.

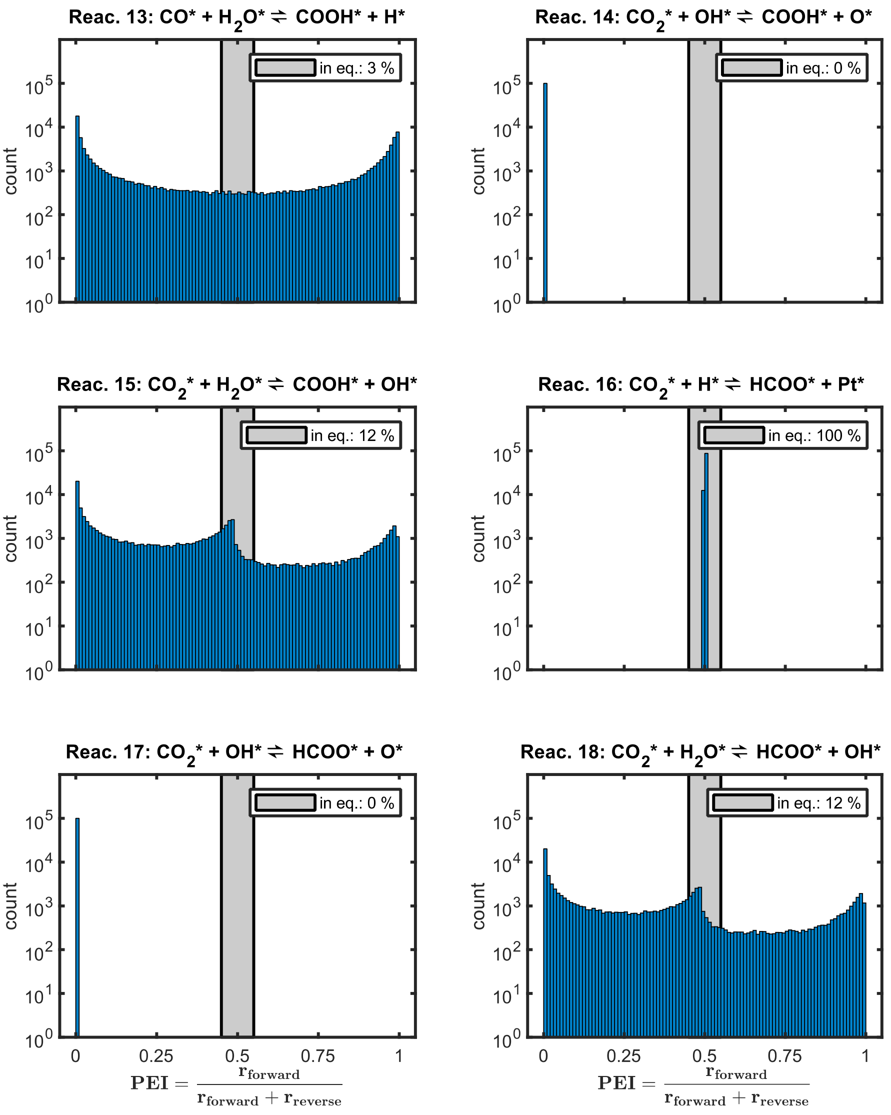


Figure S 3: Histogram of the Partial Equilibrium Index PEI of Reactions 13-18. Occurrences in the shaded area are in partial equilibrium.

S6 Degree of Rate Control data

Figure S4 shows a histogram of the Degree of Rate Control X_{RC} of reaction 7 and 13 for 10,000 operating conditions sampled randomly from the input space. Reaction 13 shows a X_{RC} near 1 for many conditions indicating, that it is the rate controlling step for CO formation.

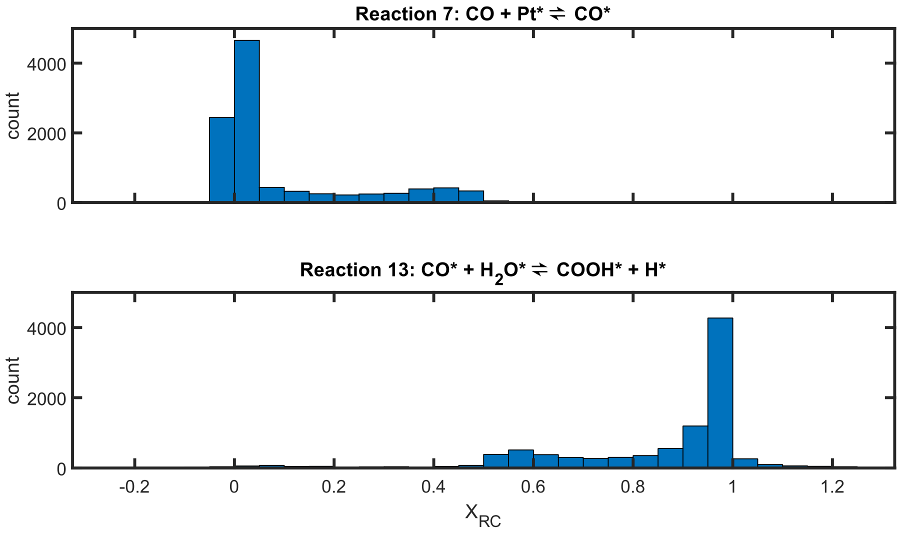


Figure S 4: Histogram of the Degree of Rate Control X_{RC} with respect to CO for reactions 7 and 13 for 10,000 operating conditions sampled randomly from the input space (see table 1).

Figure S5 shows which reaction has the highest X_{RC} with respect to CO for all conditions. Reaction 13 has the highest X_{RC} for over 90% of the conditions. Again, this indicates that reaction 13 is the rate controlling step for CO formation.

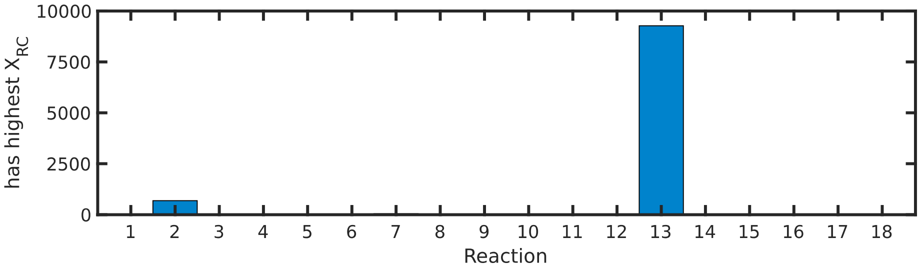


Figure S 5: Histogram of the reactions with the highest Degree of Rate Control X_{RC} with respect to CO for 10,000 reaction conditions. Reactions 2, 7 and 13 are the most rate controlling reaction for 6.9%, 0.4% and 92.7% of the conditions respectively. All other reactions are never the most rate controlling reaction.

S7 Mapping CO source terms directly using larger neural networks

Figure S6 shows the error of learning CO source terms directly using neural networks with up to 2342 parameters with one or two hidden layers. Three differently initialized models are trained per network configuration. As in some cases the results differ widely within the same configuration, all three results are shown individually instead of reporting an average error. Within this parameter range, no model suitable for reactor simulation is found.

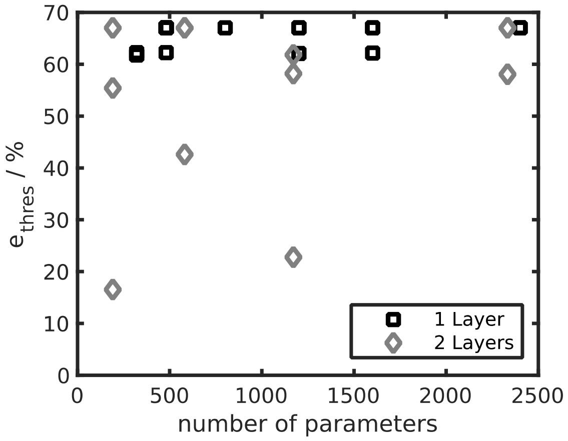
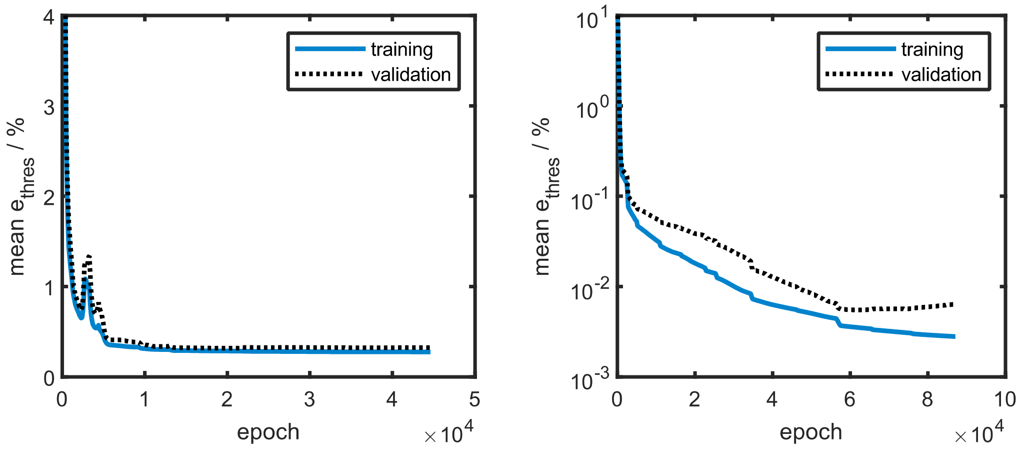


Figure S 6: Thresholded relative error obtained from neural networks predicting the CO source term directly depending on the total number of learnable parameters used. Squares: neural networks with a single hidden layer learned with 25,000 data available during training. Diamonds: neural networks with two hidden layers learned with 25,000 data available during training.

S8 Neural network learning curves

Figure S7 shows two exemplary neural network learning curves showing the progress of the thresholded error of CO source terms using separate models for forward and reverse rates of set II. The shown error is evaluated using both sub models predictions at the shown epoch.



Figures S7: Neural network learning curve for CO source terms mapped via two separate models of forward and reverse rates of set II. Left: Both models combined use 322 parameters in total. Right: Both models combined use 2342 parameters in total.

References

Gupta, U., Vlachos, D.G., 2020. Reaction Network Viewer (ReNView): An open-source framework for reaction path visualization of chemical reaction systems. SoftwareX 11, 100442. https://doi.org/10.1016/j.softx.2020.100442

Hauptmann, W., Votsmeier, M., Vogel, H., Vlachos, D.G., 2011. Modeling the simultaneous oxidation of CO and H2 on Pt - Promoting effect of H2 on the CO-light-off. Appl. Catal. A Gen. 397, 174–182. https://doi.org/10.1016/j.apcata.2011.02.031

Scheuer, A., Hauptmann, W., Drochner, A., Gieshoff, J., Vogel, H., Votsmeier, M., 2012. Dual layer automotive ammonia oxidation catalysts: Experiments and computer simulation. Appl. Catal. B Environ. 111–112, 445–455. https://doi.org/10.1016/j.apcatb.2011.10.032

Scheuer, A., Hirsch, O., Hayes, R., Vogel, H., Votsmeier, M., 2011. Efficient simulation of an ammonia oxidation reactor using a solution mapping approach. Catal. Today 175, 141–146. https://doi.org/10.1016/j.cattod.2011.03.036

Votsmeier, M., Scheuer, A., Drochner, A., Vogel, H., Gieshoff, J., 2010. Simulation of automotive NH3 oxidation catalysts based on pre-computed rate data from mechanistic surface kinetics. Catal. Today 151, 271–277. https://doi.org/10.1016/j.cattod.2010.01.018

4.2 Efficient neural network models of chemical kinetics using a latent asinh rate transformation

The method presented in the first publication [17] relies on detailed mechanistic insights to identify the rate-determining steps and to compute their rates. Often, however, mechanistic insights are not readily available. The second publication [18] aims to further broaden the scope of kinetic surrogate modelling to work directly with differential reactor data. This requires a way to normalize the data distribution of the source terms, which is characterized by spanning multiple orders of magnitude in both, positive and negative domain. This is done with a family of logarithm-like functions, e.g. the inverse hyperbolic sine $\sinh^{-1}(x)$. The source term values are normalized using this function in a pre-processing step and then modelled using a neural network (Eq. 4.2). This is an example of physics-informed data curation.

$$\sinh^{-1}(\dot{s}) = \ln\left(\dot{s} + \sqrt{\dot{s}^2 + 1}\right) \approx f_{\text{surrogate}}\left(\ln \boldsymbol{p}, T^{-1}\right) \tag{4.2}$$

As expected, this procedure improves the accuracy of the surrogates by orders of magnitude compared with the raw (non-normalized) data. However, especially shallow neural networks with one hidden layer perform significantly worse than expected. This is because data pre-processing changes the training objective. The relative error is a common choice to assess the quality of surrogate models, regardless of the abundance of a species or the magnitude of their source term [86–89, 96, 98, 128, 129]. Therefore, it is an example of a physics-informed loss function. The new approach (Eq. 4.2), however, optimizes for the mean error of the pre-processed values. The present publication introduces latent data transformation as a way to decouple data handling and training objective. Instead of pre-processing the data, the transformation is performed within the model, posing an example of a physics-informed architecture. With this approach, the model output directly corresponds to source term values (Eq. 4.3), allowing to freely choose the training objective. As a result, model accuracy is further improved by up to two orders of magnitude.

$$\dot{s} \approx \sinh\left(f_{\text{surrogate}}\left(\ln \boldsymbol{p}, T^{-1}\right)\right) \tag{4.3}$$

Reprinted from F. A. Döppel and M. Votsmeier, *Reaction Chemistry and Engineering*, **2023**, *8*, 2620-2631. The article is licensed under a Creative Commons Attribution 3.0 Unported License.

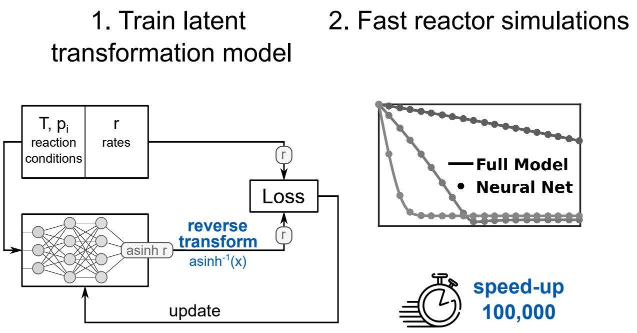


Figure 4.2: Graphical abstract of the second publication contained within this work [18].

Reaction Chemistry & Engineering

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal

Check for updates

Cite this: DOI: 10.1039/d3re00212h

Efficient neural network models of chemical kinetics using a latent asinh rate transformation†

Felix A. Döppel[a] and Martin Votsmeier*[ab]

Received 8th April 2023,
Accepted 7th July 2023

DOI: 10.1039/d3re00212h

rsc.li/reaction-engineering

We propose a new modeling strategy to build efficient neural network representations of chemical kinetics. Instead of fitting the logarithm of rates, we embed the hyperbolic sine function into neural networks and fit the actual rates. We demonstrate this approach on two detailed surface mechanisms: the preferential oxidation of CO in the presence of H_2 and the ammonia oxidation under industrially relevant conditions of the Ostwald process. Implementing the surrogate models into reactor simulations shows accurate results with a speed-up of 100 000. Overall, the approach proposed in this work will significantly facilitate the application of detailed mechanistic knowledge to the simulation-based design of realistic catalytic systems. We foresee that combining this approach with neural ordinary differential equations will allow building machine learning representations of chemical kinetics directly from experimental data.

1. Introduction

Detailed multi-scale modeling provides valuable insights into the complex phenomena of catalytic systems that typically occur in a wide range of time and length scales.[1,2] While such highly complex models would allow for rational catalyst and reactor design[3,4] they will be computationally infeasible for the foreseeable future.[2,5] The computationally most demanding part of those simulations is the solution of the chemical kinetics that often takes 70% to 90% of the computation time for both gas-phase[5,6] and surface-reactive[1,7] systems. Therefore, there is huge interest in accelerating the kinetic calculations.[1,2,8,9] This can be done by tabulating the kinetics or even a time integration step.[9,10] Latter is often done for gas-phase reactive systems[11–13] because the integration of the stiff ODE system resulting from the gas-phase kinetics is very time-consuming. For heterogeneous catalysis, timescales of surface reactions and the gas phase are usually separable *via* the steady state approximation.[2,14–17] For each simulation time step, the surface kinetics can be solved for steady state separately to avoid unnecessarily small time steps in the computational fluid dynamics simulation. Even with that simplification, evaluating the surface kinetics still poses a severe computational bottleneck.[1,7,18–20]

1.1. State of the art

Several works are mapping steady-state solutions of surface kinetics in a tabulation approach. Those maps can be built before a simulation using pre-computed solutions or during a simulation with so-called on-the-fly techniques. Some of the most used on-the-fly techniques exploit prior solutions to estimate new queries like the *in situ adaptive tabulation* (ISAT)[7,8,18,20–23] and *piecewise reusable implementation of solution mapping* (PRISM)[24] technique. In contrast, agglomeration algorithms exploit similarities of open queries to reduce the number of calls to the kinetic solver.[8,20,25] Surrogate models like splines have been extensively used to map pre-computed steady-state solutions of chemical kinetics for accelerating reactor simulations[14,15,19,26–29] or even subsystems of the reactor.[30,31] The (error-based modified) Shepard interpolation approach has been used to replace very demanding but detailed kinetic Monte-Carlo calculations in reactor simulations.[17,32,33] Recently, machine learning techniques gained growing attention for modeling kinetic data because they can overcome the so-called *curse of dimensionality*.[34] State of the art methods are random forests[35,36] and neural networks,[29] both of which have been used for accurate predictions of steady-state surface kinetics.

1.2. Data transformation

Not only the model type but also the way data are presented to the model strongly determine its accuracy. Besides scaling also transforming data is known to make models of wide-range data such as chemical kinetics more efficient. Logarithmic transformations have been used for gas-phase mass fractions[6,11,37] while preprocessing data as $\log(r)$, $\log(p_i)$ and $1/T$ is well known to facilitate modeling of surface

[a] *Technical University of Darmstadt, Peter-Grünberg-Straße 8, 64287 Darmstadt, Germany. E-mail: martin.votsmeier@tu-darmstadt.de*

[b] *Umicore AG & Co. KG, Rodenbacher Chaussee 4, 63457 Hanau, Germany*

† Electronic supplementary information (ESI) available. See DOI: **https://doi.org/10.1039/d3re00212h**

kinetics.[14,15,17,19,27,29] This can be accounted to the fact that it makes the target function more linear over a wide range of reaction conditions[17,29] as demonstrated in Eqn. S1–S3 in the ESI.† However, because these transformations rely on the logarithm a problem arises for source terms that are not strictly positive or strictly negative over the entire range of interest. This presents a substantial limitation as most systems of practical interest contain species that are both, consumed and produced depending on the reaction conditions. In our previous work we showed that this limitation can be overcome by modeling the rates of the rate-determining steps.[29] Since elementary rates are always positive, they can be modeled accurately using the logarithm. Source terms can then be constructed as a linear combination of the modeled elementary rates. Choosing the rates of the rate-determining steps instead of *e.g.*, the adsorption/desorption reaction avoids subtracting two very similar numbers, which would lead to unfavorable error propagation. However, this approach requires insights into the mechanism that are not always available for example when modeling experimental data. This leaves us with the challenge to accurately model source terms changing sign without prior insights into the reaction mechanism.

In contrast to the logarithm, logarithm-like functions like the inverse hyperbolic sine asinh(x) are not limited to positive inputs but can process negative and zero values in a meaningful way. As shown in Fig. 1, asinh(x) behaves logarithmically for values $|x| \gg 1$ while it is linear in the interval $-1 < x < 1$. This function is commonly used to analyze financial data when zero or negative values occur.[38–40] Like economic data, the net rates of chemical kinetics usually span many orders of magnitude and assume both, positive and negative values or zero.

1.3. Scope of this work

In this work, we propose a neural network architecture specialized to efficiently model steady-state solutions of detailed surface kinetics thus removing the computational bottleneck from reactor simulations. It consists of two major points:

1. We transform the rates using the logarithm-like function asinh(x) that can be applied to negative values and zero, which is crucial for modeling systems of practical interest *e.g.*, when they include intermediate species.

2. We work with latent (hidden) representations of the transformed data. This means we embed data transformation directly into the model instead of the conventional preprocessing of data, see Fig. 2. This allows minimizing meaningful error metrics like the relative error of reaction rates while preserving the advantage of data transformation. In other words, we avoid spending model capacities to regions that are not important for its application in reactor simulations.

With this setup, neural networks can accurately model wide-range data changing sign such as chemical kinetics. No prior knowledge about the reaction mechanism is required, paving the way for learning kinetics directly from experimental data or highly detailed first principles simulations. The approach is validated by reactor simulations. The preferential oxidation of CO in the presence of H_2 is simulated in a plug-flow reactor showing a speed-up of 100 000 when using neural networks instead of solving the full mechanism. Further, we model the ammonia oxidation under conditions of the Ostwald process.

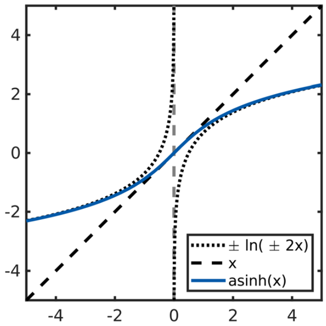


Fig. 1 Plot of the asinh(x) function which approximates the logarithm of 2x for large positive and negative inputs while being linear near the origin.

2. Methodology

2.1. Preferential oxidation of CO

2.1.1. The reaction mechanism. We consider the same reaction mechanism as used in our previous work for surrogate modeling of detailed surface kinetics.[29] The mechanism was developed by Mhadeshwar and Vlachos to describe CO oxidation, H_2 oxidation, water-gas shift reaction as well as the preferential oxidation of CO and the promoting role of H_2O on CO oxidation on platinum.[41] We use the kinetic parameters provided by Hauptmann *et al.* that are listed in Table S1 in the ESI† for all 36 elementary reactions.[42]

Reaction rates r_j (s^{-1}) are calculated as

$$r_j = k_j \cdot \prod_i c_i^{\nu_{i,j}} \cdot \prod_l \theta_l^{\nu_{l,j}} \tag{1}$$

with the rate constant k_j of reaction j (m^3 mol^{-1} s^{-1} for adsorption and s^{-1} else), the concentration c_i of gas species i (mol m^3), the surface coverage θ_l of species l (unitless) and the reaction order $\nu_{i,j}$ (unitless).[42] The rate constants for adsorption reactions k_j^{ads} and the rate constants for all other surface reactions k_j^{surf} are calculated using eqn (2) and (3) respectively.

$$k_j^{\text{ads}} = \frac{R \cdot T}{\Gamma \cdot \sqrt{2 \cdot \pi \cdot M_i \cdot R \cdot T}} \cdot \left(\frac{T}{T_0}\right)^{\beta} \cdot s_{0,i} \tag{2}$$

$$k_j^{\text{surf}} = A_j \cdot \left(\frac{T}{T_0}\right)^{\beta} \cdot \exp\left(-\frac{E_{A,j}}{R \cdot T}\right) \tag{3}$$

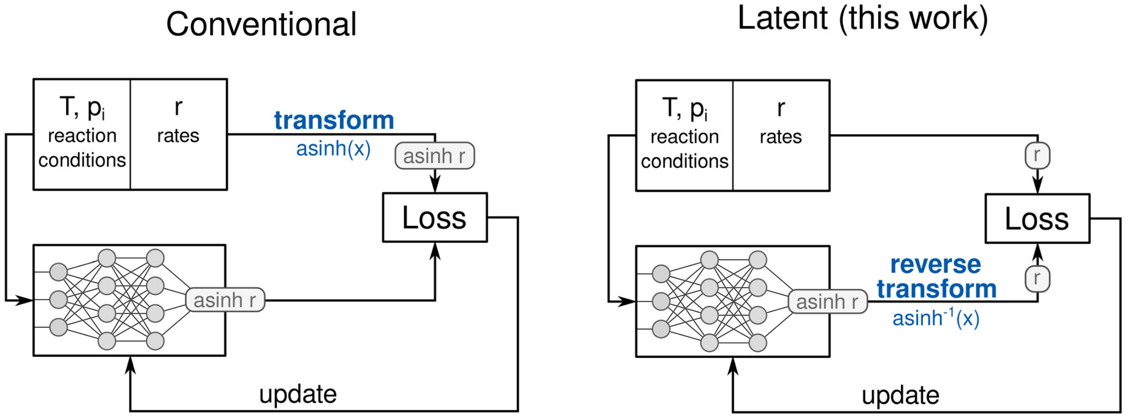


Fig. 2 Comparing conventional and latent training strategy. They differ in how data transformation is applied and which error metric is optimized during training. In both cases, a machine learning model predicts chemical rates using reaction conditions as inputs. The loss is computed to evaluate the prediction accuracy and the model parameters are updated accordingly. Conventionally, the transformation is applied to the data in a preprocessing step. The transformed values asinh(*r*) are then learned by a standard neural network. The disadvantage of this approach is that during training a loss function with respect to the transformed values has to be used instead of the actual error measure of interest. We propose to work with latent (hidden) representations of the transformed data. This means that a model with standard fully connected layers learns a latent representation of the transformed rates. Afterwards, the inverse of the transformation function is applied as a custom output activation in the final layer so that outputs represent the original rates. Hence, the error metric of interest can be optimized during training. If all parameters of the transformation function are fixed before the optimization, the inverse transformation can alternatively be implemented in a customized loss function.

with the universal gas constant R (J mol^{-1} K^{-1}), the temperature T (K), the site density Γ (2.49081 × 10^{-5} mol m^{-2}), the molecular mass M_i (kg mol^{-1}), the reference temperature T_0 (300 K), the temperature exponent β (unitless), the sticking coefficient $s_{0,i}$ (unitless), the preexponential factor A_j (s^{-1}) and the activation energy $E_{A,j}$ (J mol^{-1}).[42]

For each reaction condition given by a temperature and the partial pressures of CO, CO_2, H_2, H_2O and O_2, steady state surface coverages are calculated. This is done by integrating eqn (4) in time until $d\theta_l/dt = 0$. Gas composition and temperature are assumed to be constant during this process. The obtained surface coverages are used in eqn (5) to calculate steady state source terms $\dot{s}_i$.[29,42] Numerically, integration is performed using the DASPK solver[43] with an integration time of 10^7 s, a relative tolerance of 10^{-6} and an absolute tolerance of 10^{-50}.

$$\frac{d\theta_l}{dt} = \sum_j \nu_{l,j} \cdot r_j \tag{4}$$

$$\frac{dc_i}{dt} = \dot{s}_i = \sum_j \nu_{i,j} \cdot r_j \cdot c_{Pt} \tag{5}$$

2.1.2. The input range of the surrogate model. The input range was chosen to cover typical operating conditions met in a reactor for the removal of CO from H_2 streams by preferential oxidation of CO with small amounts of added O_2. Also, operating conditions in a low temperature water-gas shift reactor are covered.[29] The input ranges are shown in Table 1.

2.1.3. Plug-flow reactor model. We use a simple isothermal and isobaric plug-flow reactor model as described in our previous work[29] to showcase the suitability of the surrogate models for reactor simulations. The model is discretized in 200 cells of equal size in axial direction. For each cell, steady state kinetics are determined, and the gas phase concentrations are updated according to eqn (6),

$$c_{i,n+1} = c_{i,n} + \dot{s}_i\left(c_{i,n}, T\right) \cdot \tau_n \tag{6}$$

with the concentration $c_{i,n}$ (mol m^{-3}) of species i in cell number n, the temperature T (K), the source term $\dot{s}_i$ (mol m^{-3} s^{-1}) of species i and the residence time τ_n (s) in cell n obtained by dividing the total residence time by the number of cells.[29]

A total pressure of 1 atm, a site concentration c_{Pt} of 26.3 mol m^{-3}, a reactor length of 1 m divided into 200 cells and a

Table 1 Input range for reaction conditions (temperature and partial pressures) which are solved for steady state. The ranges are identical to the ones used in our previous work[29]

Quantity	Unit	Minimum	Maximum	Scaling
T	K	280	600	Reciprocal
$p(H_2)$	atm	8 × 10^{-2}	8 × 10^{-1}	Logarithmic
$p(O_2)$	atm	1 × 10^{-7}	4 × 10^{-2}	Logarithmic
$p(H_2O)$	atm	4 × 10^{-2}	4 × 10^{-1}	Logarithmic
$p(CO)$	atm	1 × 10^{-7}	4 × 10^{-2}	Logarithmic
$p(CO_2)$	atm	4 × 10^{-2}	4 × 10^{-1}	Logarithmic

gas velocity of 1 ms^{-1} are used. The resulting residence time τ is 1 s. The feed consists of 40% H_2, 1% O_2, 10% H_2O, 1% CO and 10% CO_2 with N_2 as the balance species.

If conditions outside the input range defined in Table 1 occur, they are set to the corresponding minimum or maximum values to avoid extrapolation of the neural network models.

2.2. Ammonia oxidation in the Ostwald process

2.2.1. The reaction mechanism. Ammonia oxidation on platinum is the key step in nitric acid production *via* the Ostwald process and plays an important role in automotive catalysis where it is used to remove excess ammonia from the exhaust of diesel vehicles. We consider the mechanism Ma and Schneider developed based on density functional theory (DFT) calculations.[44] This mechanism aims to describe the reaction kinetics of both applications despite the widely differing operating conditions. It consists of 15 reversible reactions featuring six gas phase species and ten surface species as shown in Table S2 in the ESI.†

Reaction rates r_j (s^{-1}) are calculated as

$$r_j = k_j \cdot \prod_i c_i^{\nu_{i,j}} \cdot \prod_l \theta_l^{\nu_{l,j}} \tag{7}$$

with the rate constant k_j of reaction j ($m^3\ mol^{-1}\ s^{-1}$ for adsorption and s^{-1} else), the concentration c_i of gas species i ($mol\ m^{-3}$), the surface coverage θ_l of species l (unitless) and the reaction order $\nu_{i,j}$ (unitless).

For each reaction condition given by a temperature and the partial pressures of NH_3, O_2, H_2O, NO, N_2O and N_2, steady state surface coverages are calculated. This is done by integrating eqn (8) in time until $d\theta_l/dt = 0$. Gas composition and temperature are assumed to be constant during this process. The obtained surface coverages are used in eqn (9) to calculate steady state source terms $\dot{s}_i$ ($mol\ m^{-2}\ s^{-1}$) using the site density Γ which is assumed to be 2.3×10^{-5} $mol\ m^{-2}$.

$$\frac{d\theta_l}{dt} = \sum_j \nu_{l,j} \cdot r_j \tag{8}$$

$$\dot{s}_i = \Gamma \cdot \sum_j \nu_{i,j} \cdot r_j \tag{9}$$

Numerically, integration is performed using MATLAB's *ode15s* solver[45] with an integration time of 10^{15} s, a relative tolerance of 10^{-8} and an absolute tolerance of 10^{-50}.

The rate constants for surface reactions k_j^{surf} are calculated as

$$k_j^{surf} = A_j \cdot \exp\left(-\frac{E_{A,j}}{R \cdot T}\right) \tag{10}$$

$$A_j = \frac{k_B T}{h} \cdot \frac{q_{TS}}{q_{IS}} \tag{11}$$

with the universal gas constant R ($J\ mol^{-1}\ K^{-1}$), the temperature T (K), the activation energy $E_{A,j}$ ($J\ mol^{-1}$) and the preexponential factor A_j (s^{-1}). Latter is calculated using transition state theory with the partition functions q_{TS} for transition states and q_{IS} for initial states as shown in eqn (11) with the Boltzmann constant k_B and the Planck constant h. The partition functions (unitless) are calculated using the harmonic oscillator model.

$$q = \prod_{n=1}^{N_{vib}} \frac{1}{1 - \exp\left(-\frac{h\nu_n}{k_B T}\right)} \tag{12}$$

with the vibrational frequencies ν obtained by DFT calculations (s^{-1}, excluding the imaginary ones, values are given in Table S3 in the ESI†) of the N_{vib} vibrational degrees of freedom. The rate constants for adsorption reactions k_j^{ads} are calculated as

$$k_j^{ads} = \frac{R \cdot T}{\Gamma \cdot \sqrt{2 \cdot \pi \cdot M_i \cdot R \cdot T}} \cdot s_{0,i} \tag{13}$$

with the molecular mass M_i ($kg\ mol^{-1}$) and the sticking coefficient $s_{0,i}$ (unitless). Desorption rate constants k_j^{des} are calculated using the equilibrium constant K_p as follows.

$$k_j^{des} = \frac{k_j^{ads}}{K_p} \tag{14}$$

$$K_p = \exp\left(-\frac{\Delta E - T\Delta S}{RT}\right) \cdot \frac{RT}{p} \tag{15}$$

with the energy differences ΔE ($J\ mol^{-1}$) obtained by DFT calculations and the reaction entropies ΔS ($J\ mol^{-1}\ K^{-1}$). Gas phase entropies are obtained from the NIST database[46] using data from ref. 47 while surface species entropies are calculated using the harmonic oscillator model as shown in eqn (16).

$$S_{surf} = R \cdot \sum_{n=1}^{N_{vib}} \frac{\frac{h\nu_n}{k_B T}}{\exp\left(\frac{h\nu_n}{k_B T}\right) - 1} - \ln\left(1 - \exp\left(-\frac{h\nu_n}{k_B T}\right)\right) \tag{16}$$

We chose this mechanism because in contrast to simpler ammonia oxidation mechanisms considered in our earlier works[15,26,28,30] it is more detailed and does not neglect the consumption of several gas species. In consequence, all species except NH_3, H_2O and N_2 show source terms changing sign in the range of reaction conditions considered. Therefore, it is not possible to rely on modeling only strictly positive source terms and compute all other species source terms from the atom balance. Rather, at least one species with sign changing source terms has to be modeled for use in a reactor simulation. We focus on predicting the source terms of NH_3, N_2 and N_2O.

2.2.2. The input range of the surrogate model. The input range was chosen to cover typical operating conditions met in a reactor for the Ostwald process *i.e.*, maximal 12% ammonia in air at up to 5 bar. The input ranges are shown in

Table 2 Input range for reaction conditions (temperature and partial pressures) which are solved for steady state within the ammonia oxidation mechanism by Ma and Schneider[44]

Quantity	Unit	Minimum	Maximum	Scaling
T	K	10^3	1.3×10^3	Reciprocal
$p(NH_3)$	Pa	0.5	6×10^4	Logarithmic
$p(O_2)$	Pa	1.25×10^4	1×10^5	Logarithmic
$p(H_2O)$	Pa	0.5	9×10^4	Logarithmic
$p(NO)$	Pa	0.5	6×10^4	Logarithmic
$p(N_2O)$	Pa	0.5	3×10^3	Logarithmic

Table 2 and are sampled uniformly in the inverse temperature and the logarithmic partial pressures.

2.3. Neural networks

Neural networks are implemented using PyTorch.[48] All neural networks are fully connected, use tanh activation and have an equal number of nodes in all hidden layers. The number of nodes per hidden layer is chosen to meet a total number of adjustable parameters up to 5000. Hidden layer weights are initialized using PyTorch's *kaiming uniform* function.

2.3.1. Architecture. The proposed neural network architecture is shown in Fig. 3. It takes the thermo-chemical state of the reactor simulation consisting of temperature (K) and partial pressures p_i (bar) as input. Those values are transformed as shown in eqn (17) and (18) and further a linear transformation is applied which maps the training data to the interval (−1, 1) (eqn (19)). Since these operations do not change during training, they can alternatively be done in a data preprocessing step.

$$x_T = (T/1\ \mathrm{K})^{-1} \tag{17}$$

$$x_{p,\mathrm{i}} = \log(p_\mathrm{i}/1\ \mathrm{atm}) \tag{18}$$

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \cdot 2 - 1 \tag{19}$$

The preprocessed thermo-chemical state is fed to the hidden layer(s) which are fully connected and use tanh activation. Per key species (CO and O_2 for test case 1 and NH_3, N_2O and N_2 for test case 2) we train a separate neural network with a single output node which contains a latent representation of the transformed source terms y = asinh ($\dot{s}/z$), see eqn (20). This node uses the inverse of this function z sinh(y) as activation to restore outputs in the form of the original source term target values $\dot{s}$. The only parameters to be learned are the weights in and out of the hidden layer(s) and optionally the parameter z of the sinh activation. To mimic the behavior of the well-known logarithmic transformation, we will choose the parameter z in a way that all modeled rates lie within the logarithmic part of the function, *i.e.* by assigning it the smallest absolute source term occurring in the training data.

$$\operatorname{asinh}(x/z) = \ln\left(\frac{x}{z} + \sqrt{\left(\frac{x}{z}\right)^2 + 1}\right) \tag{20}$$

When alternatives to the hyperbolic sine function are discussed, the output activation is replaced by either $z \cdot \mathrm{nl}^{-1}(y)$, $\mathrm{gpow}^{-1}(y, n)$ or $\exp(y)$.

In contrast to the latent transformation approach, the conventional approach computes transformed target values in a preprocessing step (asinh ($\dot{s}/z$) in our case) and uses a standard fully connected neural network to learn the transformed target values. Consequently, during training the differences between exact and estimated transformed values, *e.g.* measured by the root-mean-square error of transformed

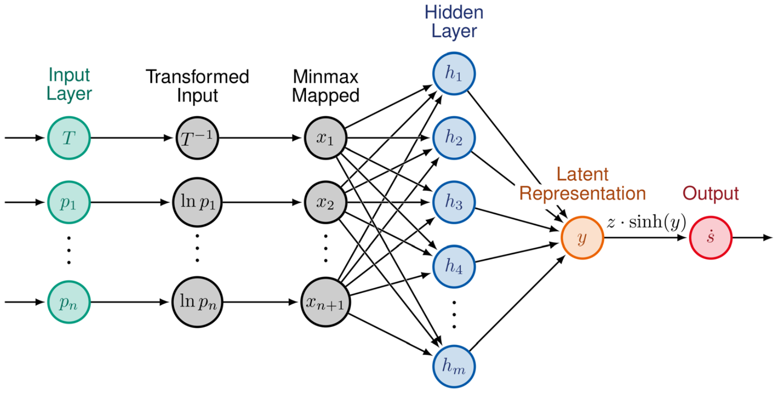


Fig. 3 Scheme of the recommended architecture. The neural network takes reaction conditions in form of temperature and partial pressures as input. Those values are transformed and linearly scaled before being fed into conventional hidden layers with tanh activation. The last layer holds a single node per gas phase species and contains y, a latent representation of the transformed target values. A hyperbolic sine activation is applied to obtain outputs in the form of source terms.

values, are minimized instead of a typical error metric of interest like the relative error of the source terms.

Direct modeling means dropping both, the input transformation in eqn (17) and (18) as well as the output activation. However, the original steady state source terms are used as targets.

2.3.2. Mass balance. Due to the mass balance, the source terms of gas phase species are linearly dependent. Therefore, it is sufficient to model only the source terms of selected key species. The net production rates of all other species are derived *via* the atom balance. Therefore, the mass balance is always exactly closed.

For test case 1 we model CO and O_2 source terms. The other species are calculated as follows:

$$\dot{s}_{\mathrm{CO_2}} = -\dot{s}_{\mathrm{CO}} \tag{21}$$

$$\dot{s}_{\mathrm{H_2}} = 2\dot{s}_{\mathrm{O_2}} - \dot{s}_{\mathrm{CO}} \tag{22}$$

$$\dot{s}_{\mathrm{H_2O}} = -2\dot{s}_{\mathrm{O_2}} + \dot{s}_{\mathrm{CO}} \tag{23}$$

For test case 2 we model NH_3, N_2 and N_2O source terms. The other species are calculated as follows:

$$\dot{s}_{\mathrm{H_2O}} = -1.5\dot{s}_{\mathrm{NH_3}} \tag{24}$$

$$\dot{s}_{\mathrm{NO}} = -\dot{s}_{\mathrm{NH_3}} - 2\dot{s}_{\mathrm{N_2O}} - 2\dot{s}_{\mathrm{N_2}} \tag{25}$$

$$\dot{s}_{\mathrm{O_2}} = 1.25\dot{s}_{\mathrm{NH_3}} + 0.5\dot{s}_{\mathrm{N_2O}} + \dot{s}_{\mathrm{N_2}} \tag{26}$$

2.3.3. Data sets. This work uses 35 000 input–output pairs of reaction conditions and resulting steady state source terms for both test cases. The training set contains 25 000, the validation set contains 5000 and the test set contains another 5000 input–output pairs. Every input–output pair is contained in only one of the three data sets. The data for the preferential oxidation test case are identical to the ones used in our previous work.[29]

2.3.4. Training. Neural network training is performed using full batch. The *LBFGS* algorithm with *strong wolfe* line search and an initial learning rate of 1 is used to update weights during training until the chosen loss evaluated on the validation set did not reduce over the last 1000 epochs.

We do not perform excessive hyper-parameter tuning as the focus of this work lies on the general modeling strategy for steady state source terms.

2.3.5. Error measure. In physics (and chemistry) small quantities are typically as important as others.[49] Therefore, also slow reactions have to be modeled with high precision for successful reactor simulations.[50] Consequently, we use the mean absolute relative error (MARE, eqn (27)) of the test set to measure the performance of the regression models built.

$$\mathrm{MARE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - h_i}{y_i}\right| \tag{27}$$

with the number of points N, the target y and the prediction h.

2.3.6. Loss functions. Different loss functions are used depending on the modeling strategy. The root mean squared relative loss $\mathscr{L}_{\mathrm{rel}}$ (eqn (28)) is minimized when source terms are used as target data.

$$\mathscr{L}_{\mathrm{rel}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{y_i - h_i}{y_i}\right)^2}. \tag{28}$$

The root mean squared absolute loss $\mathscr{L}_{\mathrm{abs}}$ (eqn (29)) of transformed values is minimized when using transformed source terms as targets.

$$\mathscr{L}_{\mathrm{abs}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y - h)^2}. \tag{29}$$

2.4. Hard- and software

Datasets for this work were produced using MATLAB Version R2021a and a faster fortran-based in-house code. Neural network training and inference were performed using PyTorch Version 1.10. Prediction times were measured using a Ryzen 7 5800X CPU @3800 MHz and a NVIDIA GFORCE RTX 3070 GPU running Linux Mint 20.3 as an operating system and averaged over 1000 identical calculations.

3. Results and discussion

We discuss the proposed latent hyperbolic sine transformation in detail using the preferential oxidation of CO as a showcase mechanism. The obtained models are validated in a plug-flow reactor simulation and compared to our previous work based on approximating the rates of the rate-determining steps.[29] A DFT-based mechanism for the ammonia oxidation under conditions of the industrial Ostwald process is used as a second test case. Finally, we discuss alternatives to the hyperbolic sine function.

3.1. Test case 1: preferential oxidation of CO

The latent hyperbolic sine transformation will be presented in detail for the preferential oxidation of CO in the presence of H_2 with a platinum catalyst. This system is important in H_2 production for fuel cell applications[42] and has been the first detailed surface mechanism modeled with neural networks in literature.[29] It can be described by three global reactions:

$$2CO + O_2 \rightarrow 2CO_2 \qquad \text{(CO oxidation)}$$

$$2H_2 + O_2 \rightarrow 2H_2O \qquad \text{(}H_2\text{ oxidation)}$$

$$CO + H_2O \rightleftarrows CO_2 + H_2 \qquad \text{(water-gas shift)}$$

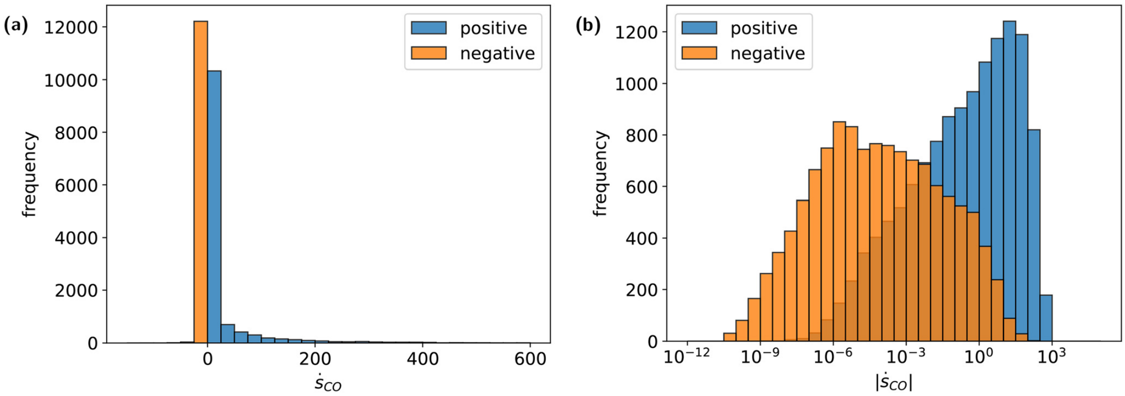


Fig. 4 Histogram of the CO source term distribution: (a) in linear scale, and (b) in logarithmic scale.

As the mechanism contains five gas phase species and three elements, at least 5 − 3 = 2 species source terms must be modeled to fully describe the reaction progress in the system. Analogous to the procedure in our previous work[29] we train a separate neural network to model the source terms of both key species O_2 and CO each. The net production rates of all other species are derived *via* the atom balance. Therefore, the mass balance is always exactly closed.

The equilibrium of the CO and H_2 oxidation reactions is fully on the right side, so that the source term of O_2 is negative under all relevant reaction conditions. Consequently, a logarithmic transformation can be applied to model the O_2 source terms.[14,15,17,19,27,29] As typical for systems of practical relevance, other species in the mechanism (including CO) change the sign of their source term depending on the reaction conditions. For those, logarithmic transformation cannot be applied in a meaningful way. We propose the latent hyperbolic sine transformation, overcoming this limitation. Fig. 4a shows a histogram of the distribution of CO source term values while Fig. 4b shows the same data on a logarithmic scale.

3.1.1. Modeling CO source terms. As the well-known logarithmic transformation cannot be applied to CO source terms due to the occurrence of negative values, the fallback approach is to directly model source terms without any transformation. However, in alignment with the results of our previous work[29] standard neural network models of reasonable size are not suited for capturing the strong non-linear character of the data. Fig. 5 shows that relative prediction errors are around 100% or higher. Using the inverse hyperbolic sine transformation conventionally, *i.e.* in a preprocessing step, reduces the prediction errors considerably. Even better results can be obtained when the transformation is implemented in a latent fashion, *i.e.* directly embedded into the neural network. This can be attributed to the fact that the conventional approach has to minimize an error measure defined in terms of the transformed values (see section 5 in the ESI†) while the latent approach operates on the original target values and therefore directly minimizes the relative prediction error. In other words, the latent transformation approach avoids spending model capacities on regions that are not important for the application in reactor simulations.

Fig. 6 shows how the prediction error of the latent transformation models scales with the number of parameters. For example, application ready models with relative prediction errors of 1% can be obtained with 15 nodes each in five hidden layers (≈1000 parameters) and less than 15 minutes of training time (see Fig. S1 in the ESI†). As neural networks are usually

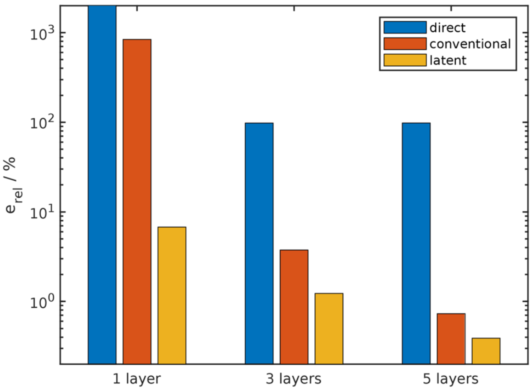


Fig. 5 Comparing the CO source term prediction accuracy of different neural network training strategies on 5000 unseen test data randomly sampled from the input range of Table 1. Using a standard feed-forward neural network without data transformation ("direct" modeling) does not yield accurate results. Using the asinh transformation conventionally, *i.e.* in a preprocessing step, reduces the prediction errors considerably. When the asinh transformation is implemented in a latent fashion, the models are even more accurate and yield application ready predictions with relative errors below 1%. All models contain about 5000 parameters distributed over one (625 nodes), three (48 nodes each) or five hidden layers (34 nodes each) and were trained with 25 000 data points.

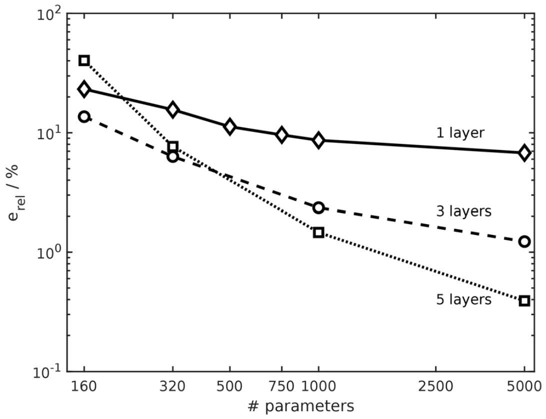


Fig. 6 Relative prediction error of CO source terms dependent of the total number of learnable parameters in a neural network using the latent hyperbolic sine transformation.

deployed with orders of magnitude more parameters and layers, all models used in this work can be considered small. Wan *et al.* for example used about 180 000 parameters for modeling chemical kinetics.[51]

In our previous work we proposed another method dealing with chemical source terms changing sign by approximating the rates of the rate-determining step.[29] As shown in Fig. S4 in the ESI,† it yields more accurate models than the latent hyperbolic sine transformation proposed in this work. That is achieved by exploiting detailed insights from a reaction path analysis. An analysis, however, is not feasible when dealing with experimental data or highly complex computational models. In contrast, the latent hyperbolic sine transformation is designed to work without any previous knowledge about the mechanism and therefore poses the first method to obtain accurate and lightweight surrogate models for detailed surface kinetics when dealing with experimental data or highly complex computational models.

In summary, the latent hyperbolic sine transformation works well because of two major points: 1. The inverse hyperbolic sine transformation brings target data into a similar order of magnitude and leads to a more linear input–output relation. 2. Using the transformation in a latent fashion gives full control over the training objective while maintaining the advantages of data transformation.

3.1.2. Validation in a plug-flow reactor simulation. We validate the neural network models by simulating an isobaric and isothermal plug-flow reactor under conditions of the preferential oxidation of CO in H_2 rich environments. Although the operating conditions are within the range of the training data, all simulations shown work with purely unseen data. We used the neural network representations of CO (34 nodes each in 5 hidden layers (≈5000 parameters) with a relative prediction error on the validation set of about 0.5%) and O_2 (1 hidden layer with 94 nodes (≈750 parameters) and a relative prediction error on the validation set of about 0.05%) kinetics to replace the steady state source term calculations in the reactor simulation as show in eqn (6). The source terms of all other species are calculated using the atom balance.

Fig. 7 shows that the concentration profiles obtained from the neural network models (dotted line) cannot be visually separated from the exact solution (full lines). The lower part of Fig. 7 shows, that the relative difference between both solutions is about 1% or lower. Note however, that calculating the neural network estimation of the source terms is approximately 50 000 times faster than evaluating the exact steady state kinetics, see Table S5 in the ESI.† Using a consumer grade graphics card for inference increases the speed-up to 100 000. As the obtained accuracy is well above that of the kinetic parameters, these results suggest that our method can be applied to much larger and more complex reaction mechanisms.

In summary, the neural network models obtained with latent hyperbolic sine transformation are well suited for replacing the computationally expensive steady state source term calculations associated with heterogeneous catalysis. They yield accurate solutions and speed-up the calculations by factor 100 000.

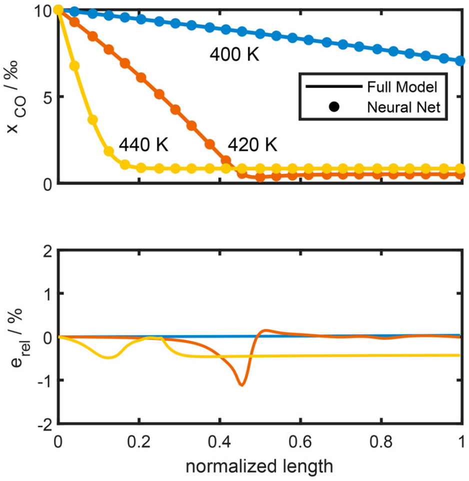
3.2. Test case 2: ammonia oxidation in the Ostwald process

To test the generality of the latent hyperbolic sine transformation approach we apply it to a second detailed surface mechanism. We consider the DFT-based ammonia oxidation mechanism from Ma and Schneider[44] for the Ostwald process under industrially relevant conditions.

Neural networks are used to predict the steady state source terms as a function of temperature and gas composition. The training data set covers all industrially relevant reaction conditions of a medium pressure ammonia oxidation reactor and contains 25 000 samples. As the mechanism contains six gas phase species and three elements, at least 6 − 3 = 3 species source terms (*e.g.* NH_3, N_2 and N_2O) must be modeled to fully describe the reaction progress in the system.

Since ammonia is burned at high temperatures, NH3 source terms are negative for all training conditions. N_2 shows only positive source terms because it is the thermodynamically favored product. Consequently, NH_3 and N_2 source terms can be modeled using the well-known logarithmic transformation. For both species a separate lightweight neural network with 63 nodes in a single hidden layer (≈500 parameters) is trained, resulting in relative prediction errors around 0.1%.

N_2O source terms, however, do change sign and the logarithmic transformation cannot be applied. Again, using a standard feed-forward neural network without data transformation ("direct" modeling) does not yield usable models as it leads to relative prediction errors near 100%. Using the asinh transformation in the conventional way increases accuracy to about 15%. The latent variant of the asinh



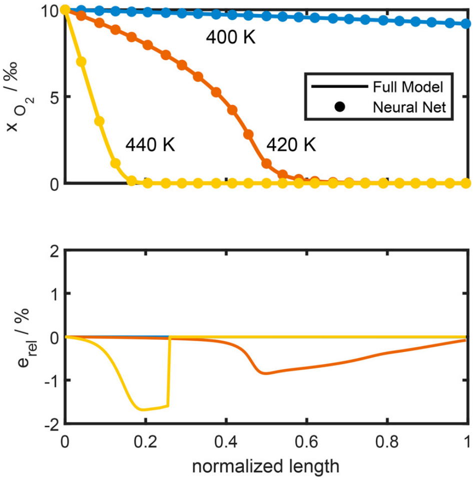


Fig. 7 Plug-flow reactor model of the preferential oxidation of CO in H_2 rich environments at three different temperatures. The upper part shows CO and O_2 molar fractions along the reactor length. The neural network solution (dotted lines) cannot be visually separated from the exact solution (full lines). The lower part shows the relative difference between both solutions. None of the conditions shown were part of the 25 000 training data which were randomly sampled from the input range of Table 1. Feed composition and other details are described in section 2.1.3.

transformation performs even better leading to errors near 1%, see Fig. 8a. See section 2.3.1 for a detailed comparison between the three modeling approaches.

The neural network models are validated by computing the product selectivities at Ostwald process conditions and zero ammonia conversion. None of these conditions are part of the training data set. Analogous to the original publication of the mechanism[44] mass transfer was not considered. The model based on the newly proposed latent transformation is in excellent alignment with the results from the full kinetic model, see Fig. 8b. In contrast, the model based on the conventional data transformation shows significant deviations. Most notably it overestimates the N_2O selectivity by an order of magnitude at lower temperatures.

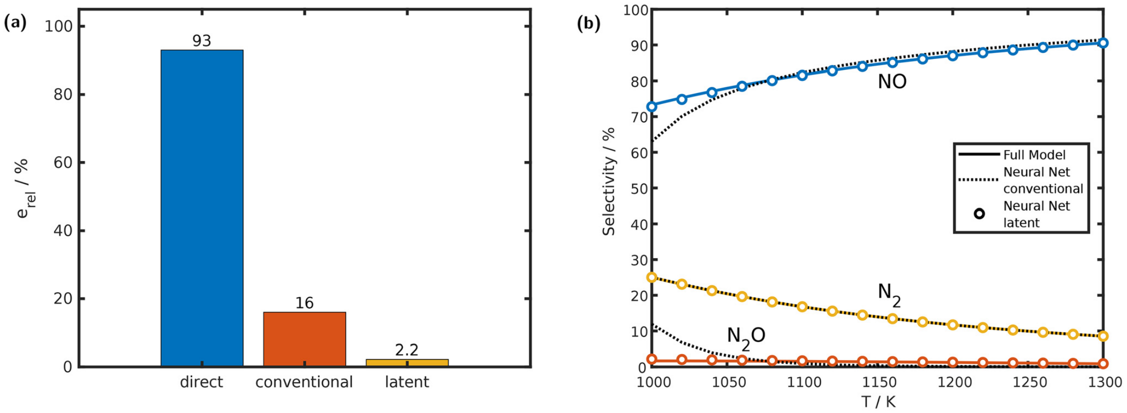


Fig. 8 Comparing different neural network training strategies. All models contain 63 nodes in a single hidden layer (≈500 parameters) and were trained with 25 000 data points. (a) Comparing the N_2O source term prediction accuracy of different neural network training strategies on 5000 unseen test data randomly sampled from the input range of Table 2. Using a standard feed-forward neural network without data transformation ("direct" modeling) does not yield accurate results. Using the asinh transformation conventionally, *i.e.* in a preprocessing step, reduces the prediction errors to approximately 15%. When the asinh transformation is implemented in a latent fashion, the models yield application ready predictions with relative errors near 1%. (b) Models are validated by comparing product selectivities at unseen Ostwald process conditions and zero ammonia conversion (10% NH_3 in air at 5 bar). The model using asinh data transformation in the conventional way covers the general trend of selectivities but fails at lower temperatures. In contrast, the model using the latent asinh transformation is in excellent agreement with the full model over the whole temperature range. Besides N_2O, neural network models of NH_3 and N_2 are used to fully describe the reaction progress in the system and use the well-known logarithmic transformation.

Overall, the latent asinh transformation allows lightweight and therefore computationally cheap neural network models ready for use in reactor simulations.

3.3. Alternatives to the hyperbolic sine

While in this work, we focused on the inverse hyperbolic sine, there are other functions able to flatten unfavorable data distributions such as wide-range data changing sign.

The *Bi-Symmetric log transformation* nl(x) was introduced by Webber to depict data that cover a wide range of scales and have both positive and negative components.[52–54] It is defined as

$$\mathrm{nl}(x,z) = \mathrm{sgn}(x)\cdot\ln\left(1+\left|\frac{x}{z}\right|\right) \tag{30}$$

with the scale parameter z and the standard mathematical sign function sgn. Like the asinh, this function approximates logarithmic behavior for $|x| \gg z$. In parallel to this work, the *Bi-Symmetric log transformation* has been used by Klumpers *et al.* for the representation of catalytic reaction rates by neural networks.[55]

Power transformations pose another way to normalize the skewed distribution of wide-range data that assume both, positive and negative values. To this end, a generalized n-th root of x can be defined as

$$\mathrm{gpow}(x,n) = \mathrm{sgn}(x)\cdot|x|^{1/n} \tag{31}$$

The three functions asinh(x), nl(x) and gpow(x,n) perform similarly for CO source term predictions when applied with the latent approach, see Fig. S5a in the ESI.† However, for O_2 source term predictions gpow(x,n) performs significantly worse than asinh(x), nl(x) and log(x), see Fig. S5b in the ESI.† This might be attributed to the fact, that the logarithmic rate transformation that is commonly used can be motivated by the Arrhenius equation and the power law expressions the rate calculations are based on and might therefore be ideal for transforming source term data without sign changes. Consequently, deviating from logarithm-like behavior can be expected to have a negative effect on accuracy. Since the adjustable parameter variant did not lead to higher accuracy, we conclude that the lowest target value occurring in the training data is a good initial guess for z. However, there seems to be no obvious initial guess for the parameter n of the generalized power function. The data shown use n = 12 for CO and n = 18 for O_2 source terms as these values provided the most accurate results in an initial testing phase.

In summary, all three functions studied in this work are suitable for latent transformation of steady state source terms changing sign and perform about similar. We suggest using the inverse hyperbolic sine function to get started as nearly all numerical libraries provide an efficient implementation.

4. Conclusions and future work

This work proposes the latent hyperbolic sine transformation for efficient neural network models of detailed surface kinetics. As the standard logarithmic transformation is not applicable to source terms changing sign we introduced the asinh function that behaves similar to the logarithm but can deal with negative numbers and zero. Further, we work with latent (hidden) representations of the transformed data. This means we embed data transformation directly into the model instead of the conventional preprocessing of data. This allows to decouple the error metric optimized during training from the data transformation and therefore increases the model accuracy significantly.

The development of the new approach is demonstrated using two test cases. The first test case is a detailed surface reaction mechanism describing the oxidation of CO in presence of H_2 as well as the water-gas shift reaction. It includes 5 gas species, 9 surface species, and 36 reactions. Models are validated by implementing them in plug-flow reactor simulations. While the neural network-assisted solution is visually not separable from the exact solution, it is computed 100 000 times faster. Neural network training used 25 000 data points and takes less than an hour on a consumer grade PC.

The second test case is a detailed surface mechanism based on density functional theory calculations of the ammonia oxidation on platinum. The latent hyperbolic sine transformation increases model accuracy significantly and allows using very small and thus computationally efficient neural networks in detailed reactor simulations.

In our previous work, we reached similarly good results by performing a reaction path analysis to exploit the detailed insights into the reaction mechanism available.[29] The present work, however, can produce accurate models of detailed surface kinetics without any previous knowledge about the underlying mechanism.

Currently, there is huge interest in determining kinetic models directly from experimental data. Especially neural ODEs[56] are promising for generating a representation of the reaction kinetic ODEs from experimental data directly.[57–59]

In accordance with our findings, it is reported that the parameter z of the inverse hyperbolic sine transformation (eqn (20)) can substantially affect regression results.[39] While several works developed strategies for finding the best value others even argue not to use this transformation at all, emphasizing that the optimal parameter value is not given by theory.[40] In this work we embedded the transformation function into a neural network. This allows optimizing all transformation parameters automatically during training and could potentially be used to identify the optimal parameter value for related problems like economic analyses.

The concept of latent data transformation is not limited to neural networks and can be used in all machine learning methods that allow customizing the loss function. For this purpose we define the custom loss function $\mathscr{L}^*$ that applies the inverse of the desired data transformation $f^{-1}(x)$ to the model outputs h before comparing them to the target values y in the conventional loss function $\mathscr{L}$, see eqn (32). This approach yields the same results as embedding $f^{-1}(x)$ in the output layer of a neural network but does not allow optimizing transformation parameters like z from eqn (20) during training.

$$\mathscr{L}^{*}(y,h) = \mathscr{L}\big(y, f^{-1}(h)\big) \tag{32}$$

Overall, the approach proposed in this work will not only significantly facilitate the application of detailed mechanistic knowledge in the simulation-based design of realistic catalytic systems, but it also presents a first step towards learning detailed surface kinetics directly from experimental data.

Author contributions

Felix A. Döppel: conceptualization, methodology, software, validation, formal analysis, investigation, data curation, writing – original draft, writing – review and editing and visualization. Martin Votsmeier: supervision, writing – review and funding acquisition.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This work was funded by BMBF in the framework of the project ML-MORE (contract 05M20RDA). The authors gratefully acknowledge the computing time provided to them on the high-performance computer Lichtenberg at the NHR Centers NHR4CES at TU Darmstadt. This is funded by the Federal Ministry of Education and Research, and the state governments participating on the basis of the resolutions of the GWK for national high performance computing at universities.

References

1 D. Micale, C. Ferroni, R. Uglietti, M. Bracconi and M. Maestri, *Chem. Ing. Tech.*, 2022, **94**, 634–651.

2 G. D. Wehinger, M. Ambrosetti, R. Cheula, Z.-B. Ding, M. Isoz, B. Kreitz, K. Kuhlmann, M. Kutscherauer, K. Niyogi, J. Poissonnier, R. Réocreux, D. Rudolf, J. Wagner, R. Zimmermann, M. Bracconi, H. Freund, U. Krewer and M. Maestri, *Chem. Eng. Res. Des.*, 2022, **184**, 39–58.

3 M. Bracconi, *Chem. Eng. Process.*, 2022, **181**, 109148.

4 G. D. Wehinger, *Chem. Ing. Tech.*, 2022, **94**, 1215–1216.

5 T. S. Brown, H. Antil, R. Löhner, F. Togashi and D. Verma, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer International Publishing, 2021, vol. 12761, LNCS, pp. 23–39.

6 V. F. Nikitin, I. M. Karandashev, M. Y. Malsagov and E. V. Mikhalchenko, *Acta Astronaut.*, 2022, **194**, 376–382.

7 M. Bracconi, M. Maestri and A. Cuoci, *AIChE J.*, 2017, **63**, 95–104.

8 E. A. Daymo, M. Hettel, O. Deutschmann and G. D. Wehinger, *Chem. Eng. Sci.*, 2022, **250**, 117408.

9 S. Barwey and V. Raman, *Energies*, 2021, **14**, 2710.

10 X. Han, M. Jia, Y. Chang and Y. Li, *Combust. Flame*, 2022, **238**, 111934.

11 A. J. Sharma, R. F. Johnson, D. A. Kessler and A. Moses, *AIAA Scitech 2020 Forum*, Reston, Virginia, 2020.

12 M. Haghshenas, P. Mitra, N. Dal Santo and D. P. Schmidt, *Energies*, 2021, **14**, 1–15.

13 J. Blasco, N. Fueyo, C. Dopazo and J. Ballester, *Combust. Flame*, 1998, **113**, 38–52.

14 M. Votsmeier, *Chem. Eng. Sci.*, 2009, **64**, 1384–1389.

15 M. Votsmeier, A. Scheuer, A. Drochner, H. Vogel and J. Gieshoff, *Catal. Today*, 2010, **151**, 271–277.

16 O. Deutschmann, *Handbook of Heterogeneous Catalysis*, Wiley-VCH, 2nd edn, 2008, ch. Comutatio, pp. 1811–1821.

17 S. Matera, M. Maestri, A. Cuoci and K. Reuter, *ACS Catal.*, 2014, **4**, 4081–4092.

18 J. M. Blasi and R. J. Kee, *Comput. Chem. Eng.*, 2016, **84**, 36–42.

19 B. Partopour and A. G. Dixon, *Comput. Chem. Eng.*, 2016, **88**, 126–134.

20 R. Uglietti, M. Bracconi and M. Maestri, *React. Chem. Eng.*, 2020, **5**, 278–288.

21 S. Mazumder, *Comput. Chem. Eng.*, 2005, **30**, 115–124.

22 S. Pope, *Combust. Theory Modell.*, 1997, **1**, 41–63.

23 R. Uglietti, M. Bracconi and M. Maestri, *React. Chem. Eng.*, 2018, **3**, 527–539.

24 S. R. Tonse, N. W. Moriarty, N. J. Brown and M. Frenklach, *Isr. J. Chem.*, 1999, **39**, 97–106.

25 S. Rebughini, A. Cuoci, A. G. Dixon and M. Maestri, *Comput. Chem. Eng.*, 2017, **97**, 175–182.

26 A. Scheuer, W. Hauptmann, A. Drochner, J. Gieshoff, H. Vogel and M. Votsmeier, *Appl. Catal., A*, 2012, **111–112**, 445–455.

27 B. Partopour and A. G. Dixon, *AIChE J.*, 2017, **63**, 87–94.

28 M. Klingenberger, O. Hirsch and M. Votsmeier, *Comput. Chem. Eng.*, 2017, **98**, 21–30.

29 F. A. Döppel and M. Votsmeier, *Chem. Eng. Sci.*, 2022, **262**, 117964.

30 A. Scheuer, O. Hirsch, R. Hayes, H. Vogel and M. Votsmeier, *Catal. Today*, 2011, **175**, 141–146.

31 T. Nien, J. Mmbaga, R. Hayes and M. Votsmeier, *Chem. Eng. Sci.*, 2013, **93**, 362–375.

32 J. M. Lorenzi, T. Stecher, K. Reuter and S. Matera, *J. Chem. Phys.*, 2017, **147**, 164106.

33 J. E. Sutton, J. M. Lorenzi, J. T. Krogel, Q. Xiong, S. Pannala, S. Matera and A. Savara, *ACS Catal.*, 2018, **8**, 5002–5016.

34 S. Matera, W. F. Schneider, A. Heyden and A. Savara, *ACS Catal.*, 2019, **9**, 6624–6647.

35 B. Partopour, R. C. Paffenroth and A. G. Dixon, *Comput. Chem. Eng.*, 2018, **115**, 286–294.

36 M. Bracconi and M. Maestri, *Chem. Eng. J.*, 2020, **400**, 125469.

37 W. Ji and S. Deng, *arXiv*, 2021, preprint, pp. 1–23, DOI: 10.48550/arXiv.2108.00455.

38 M. F. Bellemare and C. J. Wichman, *Oxf. Bull. Econ. Stat.*, 2020, **82**, 50–61.

39 G. B. D. Aihounton and A. Henningsen, *J. Econom.*, 2021, **24**, 334–351.

40 J. Mullahy and E. Norton, *Why Transform Y? A Critical Assessment of Dependent-Variable Transformations in Regression Models for Skewed and Sometimes-Zero Outcomes*, National bureau of economic research technical report, 2022.

41 A. B. Mhadeshwar and D. G. Vlachos, *J. Phys. Chem. B*, 2004, **108**, 15246–15258.

42 W. Hauptmann, M. Votsmeier, H. Vogel and D. G. Vlachos, *Appl. Catal., A*, 2011, **397**, 174–182.

43 P. E. Van Keken, D. A. Yuen and L. R. Petzold, *Geophys. Astrophys. Fluid Dyn.*, 1995, **80**, 57–74.

44 H. Ma and W. F. Schneider, *ACS Catal.*, 2019, **9**, 2407–2414.

45 L. F. Shampine and M. W. Reichelt, *SIAM J. Sci. Comput.*, 1997, **18**, 1–22.

46 *NIST Chemistry WebBook*, https://webbook.nist.gov/cgi/cbook.cgi?Source=1998CHA1-1951, Last accessed at 20.04.2020.

47 M. Chase, C. Davies, J. Downey and D. Frurip, *J. Phys. Chem. Ref. Data*, 1998, **14**, 927–1856.

48 A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai and S. Chintala, *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035.

49 T. Zhang, Y. Yi, Y. Xu, Z. X. Chen, Y. Zhang, W. E and Z.-Q. J. Xu, *Combust. Flame*, 2022, **245**, 112319.

50 X. Huang, X. Li, W. Xiao and Z. Wei, *AIChE J.*, 2023, **69**, e17945.

51 K. Wan, C. Barnaud, L. Vervisch and P. Domingo, *Combust. Flame*, 2020, **220**, 119–129.

52 J. Whittaker, C. Whitehead and M. Somers, *J. R. Stat. Soc., C: Appl. Stat.*, 2005, **54**, 863–878.

53 J. B. W. Webber, *Meas. Sci. Technol.*, 2013, **24**, 027001.

54 J. A. John and N. R. Draper, *Appl. Stat.*, 1980, **29**, 190.

55 B. Klumpers, T. Luijten, S. Gerritse, E. Hensen and I. Filot, *SSRN Journal*, 2023, 1–20.

56 R. T. Chen, Y. Rubanova, J. Bettencourt and D. K. Duvenaud, *Adv. Neural. Inf. Process. Syst.*, 2018, **31**, 1–13.

57 J. Yin, J. Li, I. A. Karimi and X. Wang, *Chem. Eng. J.*, 2023, **452**, 139487.

58 W. Ji and S. Deng, *J. Phys. Chem. A*, 2021, **125**, 1082–1092.

59 G. S. Gusmão, A. P. Retnanto, S. C. da Cunha and A. J. Medford, *Catal. Today*, 2022, **417**, 113701.

Electronic Supplementary Information

Efficient Neural Network Models of Chemical Kinetics Using a Latent asinh Rate Transformation

Felix A. Döppel and Martin Votsmeier*
*martin.votsmeier@tu-darmstadt.de

June 12, 2023

1 Linearizing Rate Expressions

$$r_{\mathrm{j}} = k_{\mathrm{j},0} \cdot \exp\left(-\frac{E_A}{RT}\right) \cdot \prod_{\mathrm{i}} c_{\mathrm{i}}^{\nu_{\mathrm{i,j}}} \cdot \prod_{\mathrm{k}} \theta_{\mathrm{k}}^{\nu_{\mathrm{k,j}}} \tag{S1}$$

$$\ln r_{\mathrm{j}} = \ln k_{\mathrm{j},0} - \frac{E_A}{R} \cdot \frac{1}{T} + \sum_{\mathrm{i}} \nu_{\mathrm{i,j}} \cdot \ln c_{\mathrm{i}} + \sum_{\mathrm{k}} \nu_{\mathrm{k,j}} \cdot \ln \theta_{\mathrm{k}} \tag{S2}$$

$$y = b + m \cdot x \tag{S3}$$

2 Preferential Oxidation Mechanism

Table S1 Reactions and parameters for the PROX mechanism taken from[1] originating from[2]

No.	Reaction	s_0 or A_0		E_A	$\frac{dE_A}{d\theta_H}$	$\frac{dE_A}{d\theta_O}$	$\frac{dE_A}{d\theta_{OH}}$	$\frac{dE_A}{d\theta_{H_2O}}$	$\frac{dE_A}{d\theta_{CO}}$
		unitless or s^{-1}	unitless	kJ/mol	kJ/mol	kJ/mol	kJ/mol	kJ/mol	kJ/mol
H_2 oxidation on Pt									
R1	$H_2 + 2^* \rightleftarrows 2H^*$	1.287×10^{-1}	0.8584	-	-	-	-	-	-
		7.953×10^{12}	1.911	79.09	−25.10	0	0	0	0
R2	$O_2 + 2^* \rightleftarrows 2O^*$	5.423×10^{-1}	0.7656	-	-	-	-	-	-
		8.406×10^{12}	0.9275	208.9	0	−133.9	0	0	0
R3	$OH^* + ^* \rightleftarrows O^* + H^*$	1.950×10^{12}	13.286	111.2	67.321	−4.87	0	62.89	0
		6.325×10^{12}	10.812	32.48	−5.820	22.26	0	−41.71	0
R4	$H_2O^* + ^* \rightleftarrows OH^* + H^*$	9.358×10^{12}	−0.3949	74.57	4.795	50.33	52.30	−43.25	0
		9.989×10^{12}	−0.3664	53.43	−7.757	−87.75	−52.30	71.81	0
R5	$H_2O^* + O^* \rightleftarrows 2OH^*$	4.316×10^{10}	0.3262	36.74	0	70.10	52.30	−83.68	0
		1.700×10^{10}	0.5285	94.32	0	−139.12	−52.30	136.0	0
R6	$H_2O + ^* \rightleftarrows H_2O^*$	1.084×10^{-1}	11.624	-	-	-	-	-	-
		2.033×10^{12}	2.490	39.98	0	0	104.6	−10.46	0
CO oxidation on Pt									
R7	$CO + ^* \rightleftarrows CO^*$	1.000	0	-	-	-	-	-	-
		5.659×10^{15}	0.3946	165.9	0	0	0	0	−62.70
R8	$CO_2 + ^* \rightleftarrows CO_2{}^*$	1.950×10^{-1}	0.2500	-	-	-	-	-	-
		3.626×10^{12}	0.2459	11.60	0	0	0	0	0
R9	$CO_2{}^* + ^* \rightleftarrows CO^* + O^*$	4.178×10^{10}	−0.2778	110.4	0	4.27	0	0	24.21
		2.393×10^{11}	0.4558	85.42	0	−17.69	0	0	−38.53
Coupling reactions between CO- and H_2 oxidation									
R10	$CO_2{}^* + H^* \rightleftarrows CO^* + OH^*$	8.031×10^{8}	−0.3259	23.22	−6.276	60.25	0	−45.56	18.12
		1.245×10^{9}	0.8237	76.91	6.276	−77.83	0	59.04	−44.63
R11	$COOH^* + ^* \rightleftarrows CO^* + OH^*$	8.426×10^{8}	0.025 77	22.66	0	60.25	0	−45.56	18.13
		1.187×10^{9}	0.4719	77.46	0	−77.83	0	59.94	−44.63
R12	$COOH^* + ^* \rightleftarrows CO_2{}^* + H^*$	1.058×10^{11}	0.5812	4.975	6.276	0	0	0	0
		9.454×10^{10}	−0.1098	6.079	−62.844	0	0	0	0
R13	$CO^* + H_2O^* \rightleftarrows COOH^* + H^*$	1.103×10^{11}	0.4911	98.71	4.904	0	52.30	−5.230	−31.38
		9.070×10^{10}	−0.027 78	22.79	−7.657	0	−52.30	5.230	31.38
R14	$CO_2{}^* + OH^* \rightleftarrows COOH^* + O^*$	5.349×10^{10}	0.052 72	109.5	0	−41.13	0	52.30	0
		1.870×10^{11}	0.4515	29.70	0	30.00	0	−52.30	0
R15	$CO_2{}^* + H_2O^* \rightleftarrows COOH^* + OH^*$	8.642×10^{10}	−0.0481	71.31	0	52.45	52.30	−44.87	0
		1.157×10^{11}	0.5647	49.07	0	−85.62	−52.30	70.19	0
R16	$CO_2{}^* + H^* \rightleftarrows HCOO^* + ^*$	1.117×10^{11}	−0.075 25	73.49	−12.55	0	0	0	0
		8.957×10^{10}	0.4218	0.00	0	0	0	0	0
R17	$CO_2{}^* + OH^* \rightleftarrows HCOO^* + O^*$	6.168×10^{10}	−0.3443	154.8	0	−67.47	0	100.2	0
		1.621×10^{11}	−0.1617	2.577	0	3.657	0	−4.351	0
R18	$CO_2{}^* + H_2O^* \rightleftarrows HCOO^* + OH^*$	1.022×10^{11}	−0.3574	111.8	0	118.7	90.99	−98.17	0
		9.785×10^{10}	0.066 65	17.15	0	−19.40	−13.61	16.89	0

3 Ammonia Oxidation Mechanism

Table S2 shows all 15 reversible reactions of the ammonia oxidation mechanism considered in this work, the activation energy E_A associated with the forward direction and the energy difference ΔE of the reaction. Table S3 lists all non-imaginary frequencies ν of the initial- and transition states. Values have to be multiplied by the elementary charge e and divided by Planck's constant h to convert to the unit Hz. Sticking coefficients of all adsorption reactions are assumed to be one.

Table S2 All 15 reversible ammonia oxidation reactions, the activation energy E_A associated with the forward direction and the energy difference ΔE of the reaction

Reaction	E_A / J mol^{-1}	ΔE / J mol^{-1}
$O_2 + 2^* \rightleftarrows 2O^*$	0	−238 319
$NH_3 + ^* \rightleftarrows NH_3{}^*$	0	−66 575
$NH_3{}^* + O^* \rightleftarrows NH_2{}^* + OH^*$	67 540	39 559
$NH_2{}^* + O^* \rightleftarrows NH^* + OH^*$	78 153	−14 473
$NH^* + O^* \rightleftarrows N^* + OH^*$	154 377	31 840
$NH_3{}^* + OH^* \rightleftarrows NH_2{}^* + H_2O^*$	33 770	−13 508
$NH_2{}^* + OH^* \rightleftarrows NH^* + H_2O^*$	965	−67 540
$NH^* + OH^* \rightleftarrows N^* + H_2O^*$	39 559	−22 192
$2OH^* \rightleftarrows O^* + H_2O^*$	0	−53 067
$H_2O^* \rightleftarrows H_2O + ^*$	18 332	18 332
$2N^* \rightleftarrows N_2 + 2^*$	244 108	−52 102
$N^* + O^* \rightleftarrows NO^* + ^*$	213 233	3859
$NO^* \rightleftarrows NO + ^*$	184 287	184 287
$N^* + NO^* \rightleftarrows N_2O^* + ^*$	164 990	89 731
$N_2O^* \rightleftarrows N_2O + ^*$	0	−2895

Table S3 A list of all non-imaginary frequencies ν of the initial states IS and transition states TS

Species	Type	ν / V
$NH_3{}^*$	IS	0.4320 0.4319 0.4152 0.1936 0.1934 0.1342 0.0820 0.0819 0.0458 0.0150 0.0149
$NH_2{}^*$	IS	0.4327 0.4189 0.1800 0.0969 0.0954 0.0825 0.0595 0.0436 0.0269
NH^*	IS	0.4242 0.0994 0.0992 0.0665 0.0588 0.0587
N^*	IS	0.0612 0.0612 0.0612
N_2O^*	IS	0.2980 0.1614 0.0541 0.0540 0.0225 0.0150 0.0149 0.0033
NO^*	IS	0.1924 0.0512 0.0511 0.0388 0.0187 0.0184
H_2O^*	IS	0.4578 0.4448 0.1919 0.0635 0.0573 0.0167 0.0122 0.0114
OH^*	IS	0.4391 0.0930 0.0896 0.0466 0.0244 0.0225
O^*	IS	0.0558 0.0454 0.0453
*	IS	–
$NH_3{-}O$	TS	0.4379 0.4262 0.3916 0.1802 0.1211 0.1125 0.0810 0.0685 0.0641 0.0588 0.0418 0.0316 0.0201 0.0137
$NH_2{-}O$	TS	0.4129 0.1870 0.1610 0.1155 0.1045 0.0786 0.0669 0.0576 0.0458 0.0332 0.0272
$NH{-}O$	TS	0.2167 0.1402 0.0776 0.0656 0.0569 0.0523 0.0355 0.0298
$NH_3{-}OH$	TS	0.4564 0.4366 0.4309 0.4057 0.1952 0.1826 0.1033 0.0825 0.0671 0.0638 0.0426 0.0333 0.0270 0.0201 0.0143 0.0088 0.0065
$NH_2{-}OH$	TS	0.4623 0.4009 0.3930 0.1915 0.1032 0.0995 0.0750 0.0607 0.0569 0.0502 0.0460 0.0197 0.0178 0.0113
$NH{-}OH$	TS	0.4531 0.2032 0.1503 0.1021 0.0783 0.0637 0.0626 0.0563 0.0448 0.0306 0.0198
$OH{-}OH$	TS	0.4549 0.1881 0.1673 0.1026 0.0799 0.0709 0.0632 0.0534 0.0368 0.0231 0.0180
$N{-}N$	TS	0.0716 0.0702 0.0537 0.0474 0.0228
$N{-}O$	TS	0.0722 0.0613 0.0525 0.0299 0.0287
$N{-}NO$	TS	0.2166 0.0731 0.0574 0.0515 0.0367 0.0339 0.0159 0.0060

4 Training Times

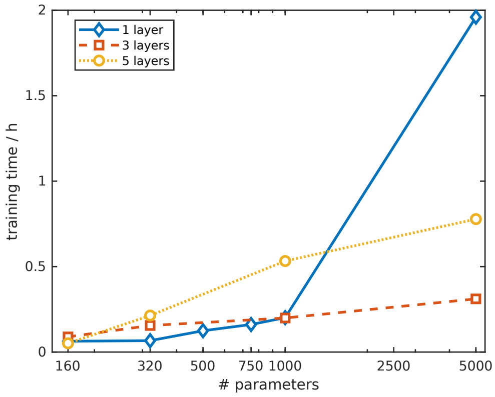


Figure S1 Median training times of the neural networks predicting CO source terms with the latent hyperbolic sine transformation strategy. Measured using three threads of a Intel® Xeon® Platinum 9242 Processor each.

5 Comparing Relative Error and MATE

So far, we applied the inverse hyperbolic sine transformation in a latent way. To investigate the effects of the latent approach, we also applied the inverse hyperbolic sine transformation in the conventional way. This means data are transformed in a preprocessing step and the transformed values are used as targets to be learned by a conventional neural network. As shown in table S4, the conventional approach leads to relative prediction errors above 1000 %, while the latent approach achieves 15 %. This can be attributed to the fact that instead of the relative error, the conventional approach minimizes an error measure defined in terms of the transformed values $\mathrm{asinh}(\dot{s})$ which we call MATE. This error measure, however, is not relevant for reactor simulations.

Table S4 Prediction errors of lightweight neural networks with 40 nodes in a single hidden layer (≈320 parameters) modeling the CO source terms $\dot{s}$ with two different approaches: Latent transformation minimizes the relative error during training resulting in an average accuracy of 15 %. Conventional transformation minimizes the error of transformed values MATE instead.[1] Therefore, its predictions are two orders of magnitude less accurate, as measured by the relative error. Because MATE is not a relevant measure for the application in reactor simulations, the slightly better MATE score of the conventional approach poses no considerable advantage over the latent approach. The equations show how the errors are computed using the neural network predictions h

error measure	equation	latent (this work)	conventional
relative error	$\left\lvert\frac{\dot{s}-h(\dot{s})}{\dot{s}}\right\rvert$	15 %	>1000 %
MATE	$\lvert\mathrm{asinh}(\dot{s}) - h\,(\mathrm{asinh}(\dot{s}))\rvert$	120 %	110 %

Figure S2 shows the relative error and the mean absolute transformed error MATE as a function of the predicted value. The exact value is arbitrarily chosen to be one. Both error measures share the same minimum and show similar asymptotic behavior for underestimations. Because of the logarithmic x-axis, the relative error grows exponentially for overestimations while MATE grows linearly. This leads to much more significant overestimations occuring in models trained with the conventional transformation approach as opposed to the latent approach.

[1] Actually, both approaches minimize the root mean square of the term shown in the respective equation, not the mean absolute which is reported in table S4. As the root mean square is more sensitive to outliers it is well suited for model training but not an intuitive indicator of the model quality.

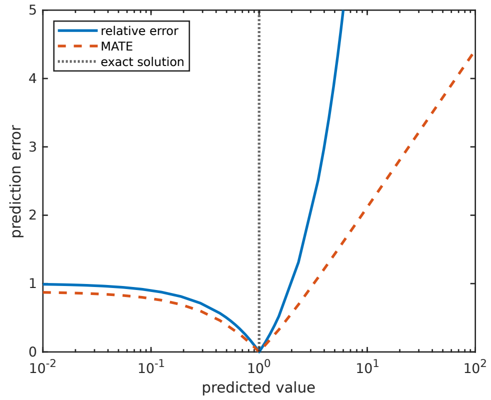


Figure S2 Comparison between the relative error and the mean absolute transformed error MATE (y-axis) for different predicted values (x-axis) assuming an exact solution of one.

6 Comparing Conventional and Latent Transformation Results

Figure S3 shows the prediction accuracy gain when using latent inverse hyperbolic sine transformation instead of the conventional approach modeling steady state CO source terms from test case one. The numbers shown are computed by dividing the relative prediction error obtained with the conventional approach by the error of the latent approach.

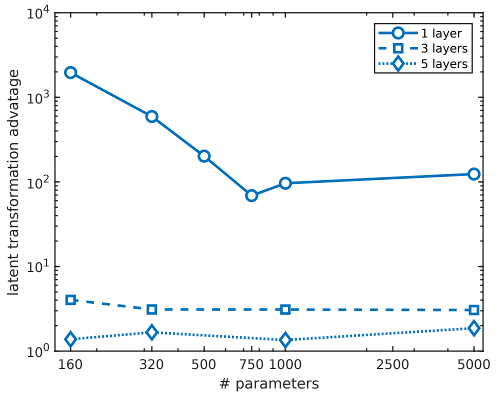


Figure S3 The accuracy gain of latent over conventional data transformation is shown as the quotient between relative prediction errors of models with identical complexity modeling steady state CO source terms. Results are shown for different numbers of hidden layers as a function of the number of model parameters.

7 Inference Time

Source term prediction time t_{predict} for 100 000 reaction conditions is averaged over 1000 runs with the models used for the preferential oxidation plug-flow reactor simulations. Results are shown in table S5 using an AMD Ryzen 7 5800X CPU and a GEFORCE RTX 3070 GPU. Computing the exact solution on the same machine takes 800 s resulting in a speed-up of 45 700 on the CPU and 95 200 on the GPU.

Table S5 Neural network prediction times t_{predict} for 100 000 reaction conditions averaged over 1000 measurements with the same models as used for the preferential oxidation plug-flow reactor simulations

Hardware	$t_{\text{predict}}(\text{O}_2)$ / ms	$t_{\text{predict}}(\text{CO})$ / ms
CPU: AMD Ryzen 7 5800X	5.5	12
GPU: GEFORCE RTX 3070	3.0	5.4

8 Comparing to our Previous Work Using Mechanistic Insights

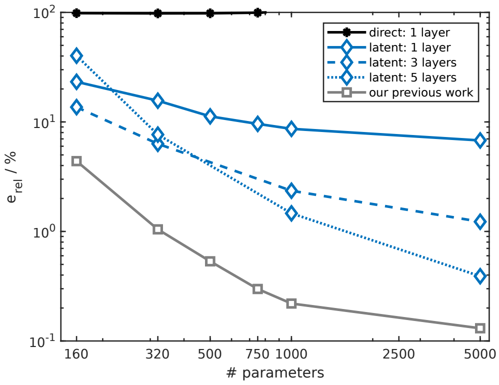
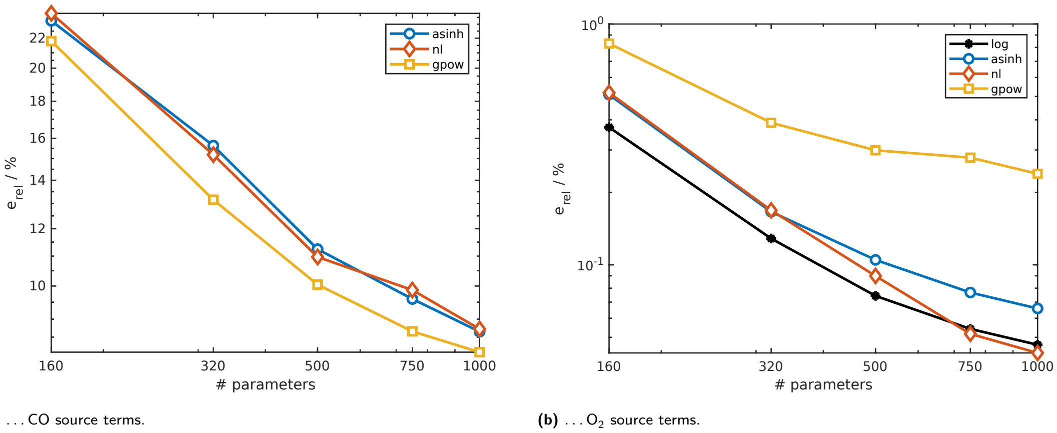


Figure S4 Relative prediction error of CO source terms dependent of the total number of learnable parameters in a neural network compared between different modeling strategies. Our previous work[1] performs best but requires a reaction path analysis and therefore cannot be used with experimental data or highly complex computational models.

9 Alternatives to the Hyperbolic Sine



(a) . . . CO source terms.

(b) . . . O_2 source terms.

Figure S5 Comparing the prediction accuracy of different transformation functions as a function of trainable model parameters in a single hidden layer for . . .

References

[1] F. A. Döppel and M. Votsmeier, *Chemical Engineering Science*, 2022, **262**, 117964.

[2] W. Hauptmann, M. Votsmeier, H. Vogel and D. G. Vlachos, *Applied Catalysis A: General*, 2011, **397**, 174–182.

4.3 Goal-Oriented Two-Layered Kernel Models as Automated Surrogates for Surface Kinetics

Having demonstrated that physics-informed machine learning models can provide efficient surrogates for surface kinetics, the third publication [19] addresses data efficiency and model reliability. The rates of the rate-determining step are mapped as suggested in [17], this time using novel two-layered kernel models. A goal-oriented selection criterion is introduced to decouple data transformation and training objective to increase prediction accuracy in analogy to the latent data transformation approach for neural networks. This is an example of a physics-informed loss function.

The limited reliability and extrapolation capabilities of conventional machine learning models call for the use of uncertainty quantification. Kernel-based models allow computing an exact upper bound to their prediction errors. Here, this measure is utilized to detect possibly inaccurate surrogate predictions during reactor simulations. If necessary, the model is updated on-the-fly, ensuring reliable results, even outside of the original training range. Interestingly, the model's accuracy can already be determined during training. This fact builds the basis of the proposed grid-free training set design scheme. Here, only those training data are evaluated that provide a significant contribution to the accuracy of the model.

Although the resulting kernel-based surrogates are overall less accurate than corresponding neural networks [17, 18], this work highlights the importance of uncertainty quantification. Reliable measures of the prediction accuracy are beneficial throughout the entire process, from data generation to model parametrization and its use in simulations.

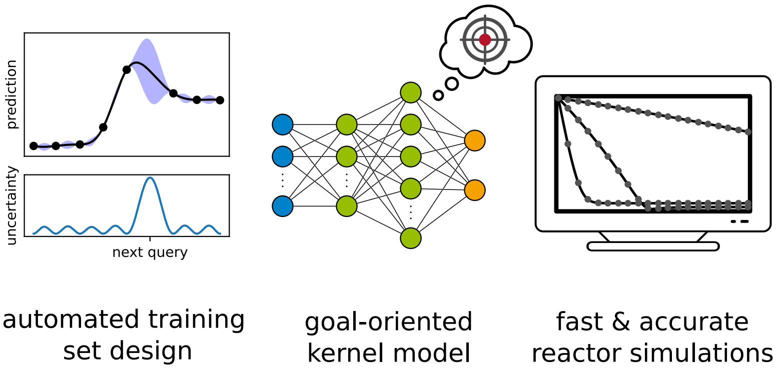

prediction
uncertainty
next query
automated training set design
goal-oriented kernel model
fast & accurate reactor simulations

Figure 4.3: Graphical abstract of the third publication contained within this work [19].

Reprinted from F. A. Döppel, T. Wenzel, R. Herkert, B. Haasdonk and M. Votsmeier, *Chemie Ingenieur Technik*, **2024**, *96*, No.5, 1-11. The article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 4.0 International License.

Goal-Oriented Two-Layered Kernel Models as Automated Surrogates for Surface Kinetics in Reactor Simulations

Felix Döppel*, Tizian Wenzel, Robin Herkert, Bernard Haasdonk*, and Martin Votsmeier*

DOI: 10.1002/cite.202300178

Supporting Information available online

Multi-scale modeling allows the description of real reactive systems under industrially relevant conditions. However, its application to rational catalyst and reactor design is hindered by the prohibitively high computational cost associated with the chemical kinetics on the catalyst scale. Here, the computational cost is drastically reduced by introducing goal-oriented kernel models that serve as surrogates for the chemical kinetics. This special model type allows for automated training set design and reliable results, even outside the training region. Therefore, it can be envisioned as a plug-and-play solution for accelerating reactive flow simulations with guaranteed accuracy.

Keywords: Adaptive modeling, Extrapolation, Machine learning, Microkinetic, Surrogate

Received: September 30, 2023; *revised:* November 17, 2023; *accepted:* December 18, 2023

1 Introduction

Machine learning methods such as neural networks find increasing application in chemical engineering [1], e.g., to model complex reaction kinetics [2,3], predict reactor behavior [4], and assist catalyst design and discovery [5]. A promising alternative to neural networks are kernel methods, especially the *Vectorial Kernel Orthogonal Greedy Algorithm (VKOGA) [6]*.

The objective of this work is to showcase the use of the VKOGA for chemical engineering problems like modeling detailed surface kinetics. We show that, in contrast to standard machine learning methods, the VKOGA reduces the demand for costly training data and provides accurate predictions, even outside the training range. To do so, we apply the recently developed two-layer kernels [7,8] and extend the VKOGA procedure by goal-oriented selection criteria.

1.1 Modeling Chemical Kinetics

Multi-scale models provide valuable insights into real reactive systems by describing the complex interplay between the physical phenomena occurring over a wide range of time and length scales [9,10]. Their use for knowledge-based engineering of surface-reactive systems [11,12], however, is currently hindered by the high computational cost [9,10,13,14] associated with the detailed chemical kinetics on the catalyst scale. Even if the timescales of surface reactions and the gas phase are separated via the steady-state approximation [10,15–18], the surface kinetics pose a severe computational bottleneck [9,10,13,19–21]. Therefore, there is huge interest in accelerating the kinetic calculations [9,10,22,23]. To this end, steady-state solutions of surface kinetics are usually mapped in a tabulation approach. The *in-situ adaptive tabulation* [13,19,21,22,24–26] builds such a map during the simulation by exploiting prior solutions to estimate new queries. In contrast, surrogate models like splines or the (error-based modified) Shepard interpolation approach map precomputed solutions to accelerate reactor simulations [2,15,16,18,20,27–31] or even spatial subsystems of the reactor [32,33]

[1]Felix Döppel https://orcid.org/0000-0003-4733-9872 (felix.doeppel@tu-darmstadt.de), [2]Tizian Wenzel, [2]Robin Herkert, [2]Prof. Bernard Haasdonk (bernard.haasdonk@ians.uni-stuttgart.de), [1,3]Prof. Martin Votsmeier https://orcid.org/0000-0002-3842-7136 (martin.votsmeier@tu-darmstadt.de)

[1]Technical University of Darmstadt, Peter-Grünberg-Straße 8, 64287 Darmstadt, Germany.

[2]University of Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany.

[3]Umicore AG & Co. KG, Rodenbacher Chaussee 4, 63457 Hanau, Germany.

and breakthrough curves [34]. Lately, primarily machine learning techniques like random forests [35, 36] or neural networks [2, 3, 37] have been used for accurate predictions of steady-state surface kinetics because they can overcome the so-called *curse of dimensionality* [38], i.e., the exponentially increasing difficulty to learn high-dimensional data. A promising alternative are kernel methods because their training is deterministic and data efficient even for high-dimensional problems. Especially interesting are greedy kernel methods like the VKOGA [6, 39], which has just recently been applied to the efficient prediction of reactive flows [40].

1.2 Kernel Methods for Regression

Kernel methods [41] comprise several tools and algorithms revolving around the notion of a kernel, which in our case is a symmetric function, $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. For regression tasks, we consider provided input data, $X_N := \{x_1, \ldots, x_N\} \subset \mathbb{R}^d$, and corresponding (vector-valued) output values, $Y_N := \{y_1, \ldots, y_N\} \subset \mathbb{R}^b$. In this work, we will model the steady-state source terms of the preferential oxidation of CO, and the input x consists of the inverse temperature as well as the logarithm of the partial pressures of CO, O_2, H_2, H_2O, and CO_2, i.e., we have $d = 6$. The output y will be either the scalar logarithm of the oxygen source terms or a two-dimensional vectorial quantity containing the logarithm of the forward and reverse rates of the CO oxidation according to [2], i.e., $b = 1$ or $b = 2$.

Then, a standard kernel representer theorem states that the kernel model can be expressed as

$$\hat{f}_{X_N}(x) = \sum_{j=1}^{N} \kappa\big(x, x_j\big)\alpha_j \tag{1}$$

where the coefficients $\alpha_j \in \mathbb{R}^b$ can usually be calculated directly based on the input and output data. Popular choices of kernels are so-called radial basis function (RBF) kernels, which can be written with the help of a scalar valued function, $\Phi : \mathbb{R}_{\geq 0} \to \mathbb{R}$, as

$$\kappa(x, z) = \Phi\big(\varepsilon \|x - z\|_2\big) \tag{2}$$

where $\varepsilon > 0$ is a shape parameter. The class of Matérn kernels, which will also be used in this work, is a prominent example of RBF kernels, and the basic Matérn kernel is given by $\Phi(r) = \exp(-r)$. Kernel methods can be understood as a linear model on nonlinearly transformed features, where the feature map is implicitly given by the chosen nonlinear kernels.

In order to adapt the kernel to the considered dataset (X_N, Y_N), recently two-layered kernels were introduced [7, 8], which can be understood as a generalization of standard shape parameter-tuned kernels. Indeed, such a two-layered kernel is given by

$$\kappa_\theta(x, z) = \Phi\big(\|\mathbf{A}_\theta(x - z)\|_2\big) \tag{3}$$

for a matrix $\mathbf{A}_\theta \in \mathbb{R}^{d \times d}$. The choice $\mathbf{A}_\theta := \varepsilon \cdot I_d$, i.e., a multiple of the identity matrix, allows the recovery of standard shape parameter-tuned kernels. The kernel κ_θ (i.e., the first layer mapping provided by the matrix $\mathbf{A}_\theta$) can be optimized efficiently using a mini-batch training (see Sect. 2.4 for details). Due to this optimization, the subsequent two-layered kernel model usually outperforms standard "shallow" kernel models such as Eq. (1).

In order to reduce the computational complexity of the kernel model from Eq. (1) for large data sets $N >> 1$, a suitable subset $X_n \subset X_N$ using $n << N$ centers can be extracted using greedy algorithms. An overview of available greedy methods and an analysis of their convergence properties are provided in [42]. The combination of the two-layered kernel optimization in conjunction with a greedy selection of suitable center points was implemented in the *2L-VKOGA* algorithm [7, 8].[1)]

1.3 Training Set Design and Reliable Extrapolation

Training data for sampling-based surrogate models are usually generated on a simple grid without considering further information about the system's behavior [2, 3, 15, 16, 20, 29, 35]. However, the input-output behavior typically varies strongly over the input domain, e.g., due to sharp transitions between regions of high and low catalytic activity [18]. Therefore, the "information content" of samples on a uniform grid is very inhomogeneous, rendering this method very inefficient. Bracconi and Maestri [36] used the training set design to focus data sampling onto regions that are difficult to model, thereby requiring 60–80 % fewer data to achieve the same model accuracy as with evenly distributed data. We automate the training set design for kernel models by utilizing the VKOGA prediction error estimator to identify regions that are difficult to model. We use this information to evaluate only those target data that significantly increase the prediction accuracy.

Once trained and validated, surrogate models still tend to produce large errors when they are applied to regions they were not trained on. Frequently, however, this remains undetected. Therefore, unreliable and potentially unnoticed extrapolation stands as a major point of concern about machine learning in chemical engineering [1]. During application, we use the VKOGA error estimator to identify regions that are not safe to extrapolate to. We use this information to build online-adaptive models, which are refined on-the-fly where necessary without diminishing the prediction quality in already known regions.

While other methods estimate their prediction error by the distance to known points [25] or the slope of the func-

[1)] https://gitlab.mathematik.uni-stuttgart.de/pub/ians-anm/2L-VKOGA tion [36], for VKOGA an exact upper error bound is known at any point, further increasing data efficiency.

2 Methods

2.1 Reaction Mechanism

The reaction mechanism considered in this work was developed by Mhadeshwar and Vlachos [43] to describe CO oxidation, H_2 oxidation, the water-gas shift reaction as well as the preferential oxidation of CO, and the promoting role of H_2O on CO oxidation on platinum. It features 36 elementary reactions, 5 gas species and 9 surface species. The kinetic parameters are taken from [44] as listed in Supporting Information Tab. S1.

Reaction rates r_j (s^{-1}) are calculated using Eq. (4) using the rate constant k_j of reaction j ($m^3\,mol^{-1}s^{-1}$ for adsorption and s^{-1} else), the concentration c_i of gas species i ($mol\,m^{-3}$), the surface coverage θ_ℓ of species ℓ (unitless), and the reaction order $\nu_{i,j}$ (unitless).

$$r_j = k_j \prod_i c_i^{\nu_{i,j}} \prod_\ell \theta_\ell^{\nu_{\ell,j}} \tag{4}$$

The rate constants for adsorption reactions $k_j^{ads,i}$ of species i and the rate constants for all other surface reactions k_j^{surf} are calculated using Eqs. (5) and (6), respectively, with the site density Γ ($2.49081\times10^{-5}\,mol\,m^{-2}$), the universal gas constant R ($J\,mol^{-1}K^{-1}$), the temperature T (K), the molecular mass M_i ($kg\,mol^{-1}$), the reference temperature T_0 (300 K), the temperature exponent β (unitless), the sticking coefficient $s_{0,i}$ (unitless), the preexponential factor A_j (s^{-1}), and the activation energy $E_{A,j}$ ($J\,mol^{-1}$) [2, 3, 44].

$$k_j^{ads,i} = \frac{1}{\Gamma}\sqrt{\frac{R\cdot T}{2\cdot\pi\cdot M_i}}\cdot\left(\frac{T}{T_0}\right)^{\beta} s_{0,i} \tag{5}$$

$$k_j^{surf} = A_j\left(\frac{T}{T_0}\right)^{\beta}\exp\left(-\frac{E_{A,j}}{R\cdot T}\right) \tag{6}$$

2.2 Generating Kinetic Data

Steady-state surface coverages are calculated for each reaction condition, given by a temperature and the partial pressures of CO, CO_2, H_2, H_2O, and O_2. This is done by integrating the rates of change of the surface coverages

$$\frac{d\theta_\ell}{dt} = \sum_j \nu_{\ell,j} r_j \tag{7}$$

in time until they are sufficiently small:

$$\frac{d\theta_\ell}{dt} \approx 0 \tag{8}$$

The DASPK solver [45] is used to perform the integration with an integration time of 10^7 s, a relative tolerance of 10^{-6}, and an absolute tolerance of 10^{-50} (which emulates an absolute tolerance of 0, but a value of 0 is not accepted by the solver implementation used). During integration, the gas composition and temperature are assumed to be constant. The obtained surface coverages are used in

$$\dot{s}_i = \sum_j \nu_{i,j} r_j c_{Pt} \tag{9}$$

to calculate the steady-state source terms $\dot{s}_i$ [2, 44].

Operating conditions (Tab. 1) are chosen to cover the removal of CO from H_2 streams by preferential oxidation of CO with small amounts of added O_2 and low-temperature water-gas shift [2]:

$$CO + H_2O \rightleftharpoons CO_2 + H_2 \tag{10}$$

Table 1. Input range for reaction conditions (temperature and partial pressures) that are solved for steady state.[a]

Quantity	Unit	Minimum	Maximum
T	[K]	280	600
$p(H_2)$	[atm]	8×10^{-2}	8×10^{-1}
$p(O_2)$	[atm]	10^{-7}	4×10^{-2}
$p(H_2O)$	[atm]	4×10^{-2}	4×10^{-1}
$p(CO)$	[atm]	10^{-7}	4×10^{-2}
$p(CO_2)$	[atm]	4×10^{-2}	4×10^{-1}

a) The ranges are identical to the ones used in previous works [2, 3].

As described in our previous work [2], CO source terms are modeled as the difference of the forward and reverse rates of the rate-determining steps, which are obtained as

$$r_{fwd} = r_{17} + r_{19} + r_{21} + r_{26} \tag{11}$$

$$r_{rev} = r_{18} + r_{20} + r_{22} + r_{25} \tag{12}$$

The 25 000 training, 5000 validation, and 5000 test data points from our previous works [2, 3] are used. Reaction conditions are randomly sampled with uniform distribution in $1/T$ and $\ln(p_i)$.

2.3 Plug-Flow Reactor Model

An isothermal and isobaric plug-flow reactor model

$$\frac{dc_i}{dt} = \dot{s}_i \tag{13}$$

is discretized in 200 cells of equal size in axial direction

$$c_{i,n+1} = c_{i,n} + \dot{s}_i\left(c_{i,n}, T\right)\tau_n \tag{14}$$

with the concentration $c_{i,n}$ (mol m^{-3}) of species i in cell number n, and the residence time τ_n (s) in cell n obtained by dividing the total residence time by the number of cells [2, 3]. A site concentration c_{Pt} of 26.3 mol m^{-3} and a gas velocity of 1 m s^{-1} are used.

To close the mass balance with kernel models, only CO and O_2 source terms will be predicted and the source terms of all other species are obtained from the mass balance [3] as follows:

$$\dot{s}_{\mathrm{CO}_2} = -\dot{s}_{\mathrm{CO}} \tag{15}$$

$$\dot{s}_{\mathrm{H}_2} = 2\dot{s}_{\mathrm{O}_2} - \dot{s}_{\mathrm{CO}} \tag{16}$$

$$\dot{s}_{\mathrm{H}_2\mathrm{O}} = -2\dot{s}_{\mathrm{O}_2} + \dot{s}_{\mathrm{CO}} \tag{17}$$

For the preferential oxidation reactor, a total pressure of 1 atm, a length of 1 m, and a feed composition of 40 % H_2, 1 % O_2, 10 % H_2O, 1 % CO, and 10 % CO_2 are assumed. For the water-gas shift reactor, a total pressure of 10 atm, a length of 0.01 m, and a feed composition of 5 % H_2, 0 % O_2, 20 % H_2O, 15 % CO, and 5 % CO_2 are assumed. In both cases, N_2 acts as the balance species.

To avoid extrapolation during model validation in the preferential oxidation reactor, conditions outside the training range (Tab. 1) are set to the corresponding minimum or maximum values. In simulations of the water-gas shift reactor, all conditions are used as-is.

2.4 Sparsified Two-Layered Kernel Optimization

In order to adapt the kernel to the data set before running the subsequent (greedy) model computation, we make use of two-layered kernels from Eq. (3). For the kernel optimization, i.e., the optimization of the first-layer matrix $\mathbf{A}_\theta$, we make use of the mini-batch training described in [7, 8]: It uses a mini-batch gradient descent on an efficiently computed cross-validation loss, whereby the mini-batches of size 64 are randomly drawn from the whole training data set of size 25 000.

In order to further improve the training procedure of the two-layered kernels, we introduce a sparsification of the original training procedure: Instead of using the full data set for the optimization, we note that it is sufficient to only use a smaller subset with just 200 data points, while increasing the number of epochs by a factor of 25 000/200. In this way, the overall number of training iterations stays the same, and we obtain a similarly well-performing two-layered kernel. However, we require only the knowledge of 200 target values, which pays off whenever the computation of the target values is expensive. This is, for example, of importance when using the unsupervised goal-oriented *P*-greedy algorithm (see Sect. 3.3).

2.5 Standard Greedy Selection Criteria

Standard, widely used selection criteria like *f*-greedy or *P*-greedy focus on selecting kernel centers x_1, x_2, …, such that the error in the prediction of the state y is minimized. For *f*-greedy, the centers are iteratively selected following the rule

$$x_{n+1} = \mathrm{argmax}_{x\in X_N\setminus X_n} \left\| f(x) - \hat{f}_{X_n}(x) \right\|_2 \tag{18}$$

and for *P*-greedy according to the rule

$$x_{n+1} = \mathrm{argmax}_{x\in X_N\setminus X_n} \mathcal{P}_n(x) \tag{19}$$

where $\mathcal{P}_n$ denotes the so-called power function. The power function can be computed from the kernel k and the currently selected centers and can be used as an error indicator because the pointwise error between f and the kernel model $\hat{f}_{X_n}$ can be bounded as

$$\left\| \left(f - \hat{f}_{X_n}\right)(x) \right\|_2 \leq C\mathcal{P}_n(x) \tag{20}$$

for some constant C.

The advantage of the *P*-greedy selection in comparison to the *f*-greedy one is that the function evaluations $f(x_i)$, $i = 1$, …, N do not have to be known beforehand and only the function evaluations $f(x_i)$, $i = 1$, …, n for $n << N$ have to be computed during the selection process. For large datasets and if the function evaluation of f is computationally intensive, this drastically decreases the computational effort. However, in terms of error decay rates, the *f*-greedy usually behaves better because more knowledge of f is used during the selection process.

2.6 Goal-Oriented Greedy Selection Criteria

In cases like the estimation of the CO source term, we are not primarily interested in an approximation of the states y (logarithm of forward and reverse rates) itself, but in approximating a derived "goal" property, $g(y) \in \mathbb{R}$ (steady-state source terms $\dot{s}_{\mathrm{CO}}$). We now propose two new selection criteria that select centers in such a goal-oriented manner:

1) the *goal-oriented f-greedy* and
2) the *goal-oriented P-greedy.*

In contrast to the *f*-greedy criterion that minimizes the absolute error from the kernel model to *f*, the goal-oriented f-greedy selects centers according to the relative error in *g*:

$$x_{n+1} = \mathrm{argmax}_{x\in X_N\setminus X_n} \frac{\left| g(f(x)) - g\left(\hat{f}_{X_n}(x)\right) \right|}{|g(f(x))|} \tag{21}$$

The idea of the goal-oriented P-greedy is to select centers according to a bound for the relative error. For that purpose, we bound the numerator in Eq. (21) from above with

$$\left|g(f(x)) - g\left(\hat{f}_{X_n}(x)\right)\right| \leq \Delta_{\text{abs,goal}}(x) \tag{22}$$

and the denominator from below with

$$|g(f(x))| \geq g_{LB}(x) \tag{23}$$

In contrast to the non-goal-oriented case, these bounds generally cannot be computed exactly but only estimated (see Sect. 2 in the Supporting Information for details on the estimation procedure).

3 Results and Discussion

First, we showcase the performance of the newly proposed two-layered goal-oriented kernel models for approximating steady-state source terms of a detailed surface mechanism. Such surrogate models are necessary to remove the computational bottleneck from reactive flow simulations, facilitating their use for knowledge-based engineering of surface-reactive systems. In particular, we will show (1) that two-layered models consistently outperform the common one-layered models, (2) that the goal-oriented models select more relevant data and are therefore more data efficient, and (3) that the VKOGA allows deriving an upper bound to the prediction error, which we will use to automate the training set design and thereby reduce the demand for costly training data by more than one order of magnitude. We validate the kernel models in simulations of the partial oxidation reactor at different temperatures. Due to their computational efficiency, the kernel models accelerate the simulation by factor 180. Finally, we tackle one of the major concerns about machine learning in chemical engineering by showing that so-called online-adaptive kernel models allow for reliable results, even outside the training range.

3.1 Modeling Detailed Surface Kinetics

We chose a mechanism proposed by Mhadeshwar and Vlachos [43], describing the selective removal of CO from hydrogen on platinum to prevent catalyst poisoning. It features 36 elementary reactions, 9 surface species, and 5 gas phase species. The target of the kernel models is to predict the steady-state source terms as a function of the reaction conditions (temperature T and partial pressures p_i). To this end, a training data set consisting of 25 000 reaction conditions and corresponding steady-state source terms was computed. As typical for chemical kinetics [2, 3, 15, 18], the source term values cover many orders of magnitude (Fig. 1).

It is well known that modeling the source terms as a function of the reaction conditions (T and p_i) will fail [2, 3] because standard machine learning methods are not designed to capture the multiplicative and exponential relations [46] that are inherent to chemical kinetics. The strongly nonlinear input-output relation (Supporting Information Fig. S1) can be simplified by fitting the logarithm of the source terms as a function of $1/T$ and $\ln(p_i)$ [2, 15, 16, 18, 20]. We followed this approach to approximate the steady-state O_2 source terms with kernel models using the VKOGA (Fig. 2). Kernel models gradually grow during training by adding kernels at the data point with the largest (estimated) prediction error. From now on, the number of kernels added will be referred to as the size of the model.

3.1.1 Two-Layered Kernel Models

The recently developed two-layered kernel is a data-efficient modeling tool [7, 8] that is well suited for high-dimensional problems. Here, the common kernel layer is extended by an additional linear layer, which adjusts the kernel to the magnitude and importance of the features (here: reaction conditions) and thereby significantly increases the model accuracy. The parameters of this linear transformation are obtained by gradient descent optimization of small kernel models on

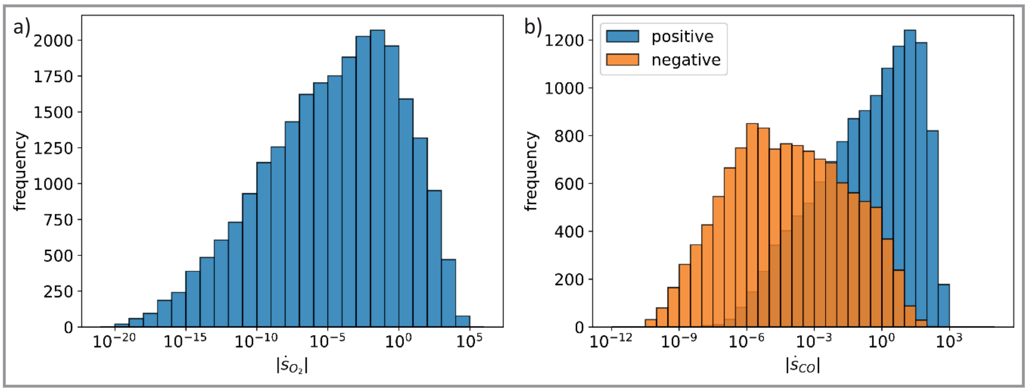


Figure 1. Histogram of the source term distribution with logarithmic scale. (a) O_2 source term values span more than 20 orders of magnitude. (b) Both positive (blue, darker) and negative (orange, lighter) CO source term values span more than 10 orders of magnitude.

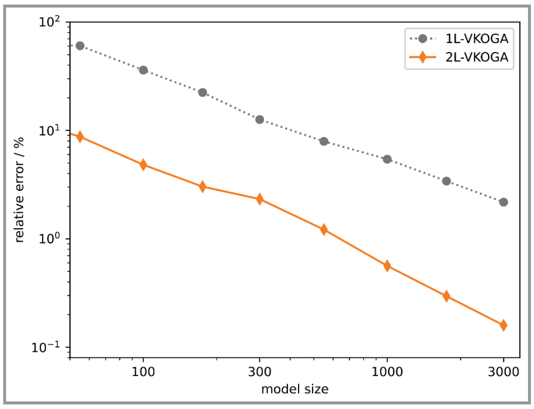


Figure 2. Relative test set prediction error over the number of *f*-greedily selected data points (model size) out of the 25 000 training data points for both one-layered (1L) and two-layered (2L) kernels approximating O_2 source terms.

mini batches (64 data points) of a small, randomly chosen subset (200 data points) of the training data (Sect. 2.4). These parameters are kept constant during the subsequent optimization of the nonlinear kernel layer, which utilizes the full training data set. As a result, accurate two-layered kernel models with a relative prediction error below 1 % are obtained using only 500 kernel centers while the standard one-layered models are ten times less accurate (Fig. 2).

3.1.2 Goal-Oriented Kernel Models

In most reactive systems of industrial interest, the source term values of multiple species change sign depending on the reaction conditions. In the preferential oxidation reactor considered in this work, this is the case for all species except O_2 (namely CO, CO_2, H_2, and H_2O). This makes modeling the chemical kinetics particularly difficult, because the standard logarithmic transformation cannot be used to simplify the input-output relation of the data. In an earlier work [2], we showed that an efficient way to model such data is to approximate the logarithm of forward, r_{fwd}, and reverse rates, r_{rev}, of the rate-determining steps and obtain the source terms as their difference. We will follow this approach to approximate the steady-state CO source terms with kernel models. However, the accuracy of predicting these auxiliary quantities (r_{fwd}, r_{rev}) is not a reliable indicator of the resulting source term prediction accuracy because near the partial equilibrium two very similar numbers are subtracted, leading to an amplification of modeling errors [2]. Therefore, we introduce goal-oriented selection criteria that directly consider the accuracy of the actual target property (here: steady-state source terms) and thereby automatically focus on regions that are difficult to model. The effect can be seen by comparing the kernel centers chosen by different modeling strategies (Fig. 3). In contrast to standard (non-goal-oriented) models, goal-oriented models primarily select kernel centers near the partial equilibrium (partial equilibrium index (PEI) ≈ 0.5).

Goal-oriented models select more relevant data. Therefore, they require 42 % fewer parameters and 60 % less training time than standard (non-goal-oriented) two-layered models to approximate steady-state CO source terms with a relative prediction error of 1 % (2L goal-oriented *f*-greedy models need 1000 kernel centers, while standard 2L *f*-greedy models need 1750 kernel centers, see Fig. 4). One-layered kernels do not reach this accuracy with a reasonable amount of data available (Fig. 4).

3.2 Validation in Reactor Simulations

We validate the surrogate models from Sect. 3.1 in simulations of the preferential oxidation reactor. The computationally intensive rate calculations of the full microkinetic model are replaced with calls to the faster goal-oriented kernel models. To this end, the two-layered goal-oriented *f*-greedy O_2 kernel model of size 500 and the goal-oriented *f*-greedy CO kernel model of size 1000 are used because they each show a prediction accuracy of 1 % on the test data set. The source terms $\dot{s}$ of the other species are obtained from the mass balance (Sect. 2.3). Therefore, the model predictions will always have an exactly closed mass balance [2, 3].

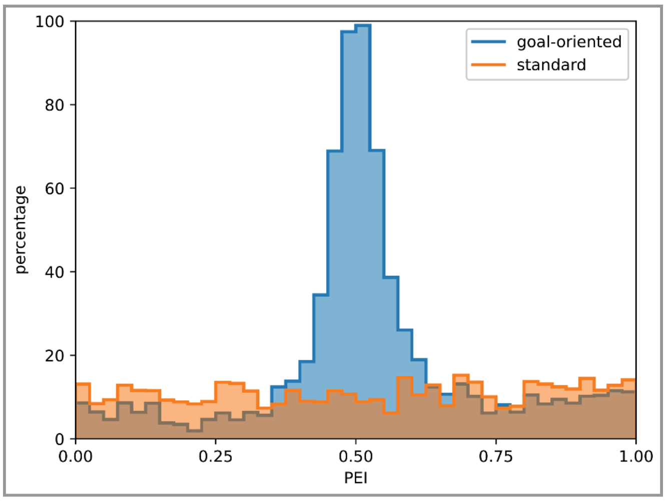


Figure 3. Visualization of the percentage of training data points selected as kernel centers (from the respective histogram bin) depending on the partial equilibrium index, $\mathrm{PEI} = \frac{r_{\mathrm{fwd}}}{r_{\mathrm{fwd}} + r_{\mathrm{rev}}}$, for standard and goal-oriented kernel models. Both models chose 3000 kernels in total.

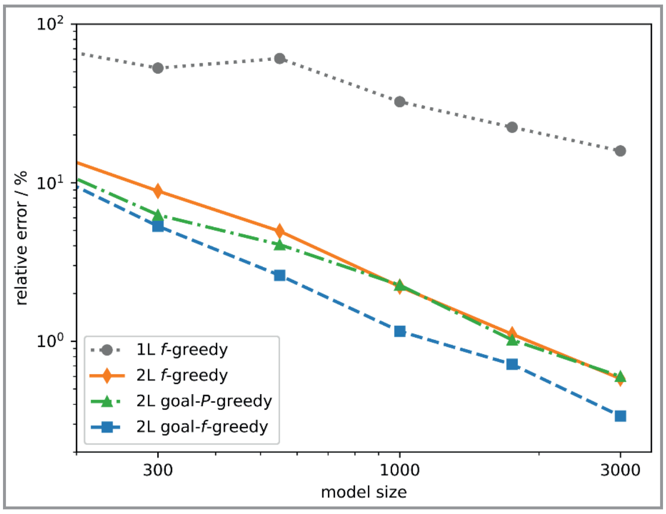


Figure 4. Relative test set prediction error over the number of greedily selected data points (model size) out of the 25 000 training data points for both one-layered (1L) and two-layered (2L) kernels approximating CO source terms. The two-layered kernels consistently outperform the standard one-layered kernels.

Fig. 5 shows the obtained concentration profiles of CO and O_2 as molar fractions, which are predicted with a difference below 0.1 ‰ at any point (Supporting Information Fig. S2). While the kernel model solution is so accurate that it cannot be visually separated from the exact solution, source terms are calculated 180 times faster (Supporting Information Tab. S2).

3.3 Automated Training Set Design

Usually, training data for surrogate models are generated on a simple grid without considering further information about the system behavior [2, 3, 15, 16, 20, 29, 35]. In this way, many expensive data are collected even if they do not significantly contribute to the accuracy of the model. We harness the unique properties of the VKOGA to derive an upper bound for the prediction error of any kernel model. This bound can be understood as an error estimator which automatically identifies regions that are difficult to model. It thereby indicates which data will contribute to the model's accuracy and are therefore worth collecting.

We use the estimated prediction error as a new selection criterion (goal-oriented *P*-greedy) and compute only those target data that are explicitly requested by the model (Fig. 6). As a result, 13 times fewer data are needed to reach 1 % prediction accuracy. (Goal-oriented *P*-greedy models reach 1 % prediction accuracy with 1750 selected kernel centers plus 200 data for training the linear layer (see Fig. 4), while the full training data set used for all other models holds 25 000 points.) When dealing with expensive data as obtained from kinetic Monte-Carlo simulations or experiments, this directly translates to a 13-fold reduction of data generation costs.

3.4 Reliable Extrapolation

While surrogate models are usually not suited for use outside their training region, they might still produce seemingly plausible results. Such compromised model reliability can pose an important safety issue, especially when unexpected or extreme conditions occur. Therefore, unreliable and potentially unnoticed extrapolation is considered one of the major concerns about machine learning in chemical engineering [1]. We show that the prediction accuracy of goal-oriented kernel models can be easily tracked and used to achieve reliable extrapolation. For demonstration, we force extrapolation by predicting source terms in a water-gas shift reactor with a model that was trained on preferential oxidation conditions. In particular, we employ the same 2L-VKOGA models that were used in Sect. 3.2. We consider two operation modes:

1) "Static": After the training phase, the model is not altered anymore. It is solely used to predict source terms. This is the standard procedure and identical to the approach used in Sect. 3.2.

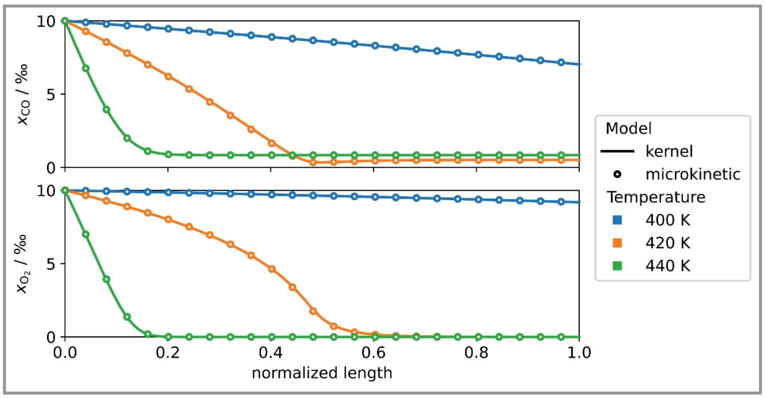


Figure 5. Predicted concentration profiles over the length of a plug-flow reactor (lines) match the exact solution with the microkinetic model (symbols) as shown for three different temperatures (see Sect. 2.3 for details).

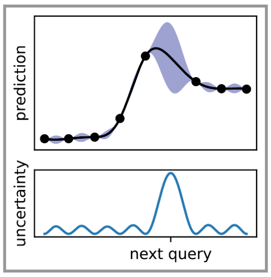


Figure 6. Schematic visualization of the automated training set design. The prediction of the kernel model (black line) is based on all previously known data points (symbols). Iteratively, the next data point is requested where the model's uncertainty (shaded area and bottom plot) is highest.

2) "Online-adaptive": During deployment, in addition to predicting source terms, the prediction error is estimated. If it passes a threshold of 5 %, the full model solution is computed, the new data point is added to the model as a kernel center, and the local error decreases to zero (Fig. 7).

Even though operated outside its training region, the static model produces seemingly plausible results: It shows low activity below 600 K, maximal conversion around 750 K, and slightly decreasing conversion due to equilibrium limitation at higher temperatures (Fig. 8). However, these results vary from the correct solution by almost up to 20 % and do not even reach the correct equilibrium state. In contrast, the online-adaptive model is fully accurate and exactly predicts the conversion at any temperature (Fig. 8). Still, it requests 70 times fewer function calls to the high-cost microkinetic model than the full solution, leading to a corresponding reduction of data generation cost (Supporting Information Fig. S3). Future use of the kernel model in this temperature range will not require additional calls to the full model anymore. Therefore, the computational advantage grows with repetitive use.

It is important to note the local character of the kernel models. Common machine learning methods like neural networks tend to forget old information as new information is acquired [47, 48]. When training such a model sequentially, it is likely that it loses accuracy on the task it was originally trained on. Kernel models, however, are made up of local functions, such as Gaussians. Adding a new kernel center at water-gas shift reaction conditions hardly influences the model's behavior in the original training range. Therefore, the validation for preferential oxidation conditions in Sect. 3.2 still holds and does not have to be repeated.

4 Conclusion

In this work, we showcased the use of novel two-layered kernel models for the accurate and efficient prediction of chemical kinetics. The additional layer adjusts the kernel to the magnitude and importance of the input values (reaction conditions) and thereby increased accuracy by at least one order of magnitude. We extended the kernel methods by introducing goal-oriented selection criteria. Goal-oriented models automatically focus on regions that are difficult to model and therefore required 42 % fewer parameters and 60 % less training time than standard non-goal-oriented models. Especially when dealing with high-cost data, e.g., from kinetic Monte-Carlo simulations or laboratory measurements, data generation expenses are a significant factor in model development. We derived an exact upper bound to the prediction error of the kernel models and used it as an error estimator to compute only such target data that significantly contribute to the model accuracy. Thereby, we reduced the demand for costly training data by factor 13. The resulting models show negligible error (1 % relative deviation), provide steady-state surface kinetics 180 times faster than the full kinetic mechanism, and are easily implemented in common simulation software like gPROMS or OpenFOAM. Finally, we tackled one of the major concerns about machine learning in chemical engineering by showing that so-called online-adaptive kernel models allow for reliable extrapolation results.

In conclusion, two-layered goal-oriented kernel models are a promising tool for chemical engineering problems like providing accurate and efficient approximations to detailed surface kinetics. In contrast to standard machine learning methods, they reduce the demand for costly training data and accurately extrapolate outside the training range. We envision them as a plug-and-play solution for reducing the computational demand of reactive flow simulations that maintains any user-specified accuracy.

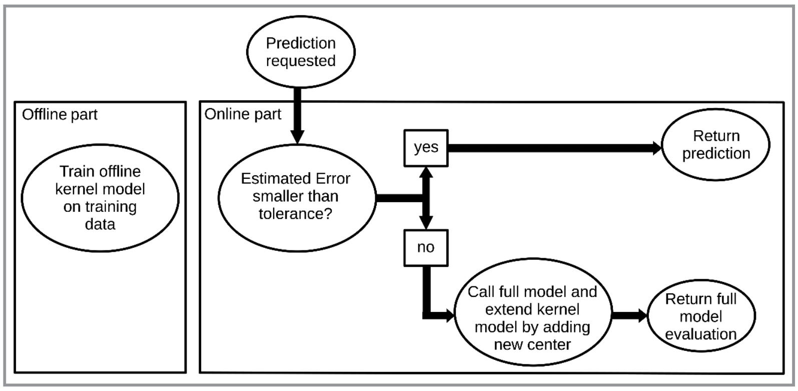

Prediction requested
Offline part
Online part
Train offline kernel model on training data
Estimated Error smaller than tolerance?
yes
no
Return prediction
Call full model and extend kernel model by adding new center
Return full model evaluation

Figure 7. Schematic workflow for an online-adaptive kernel model.

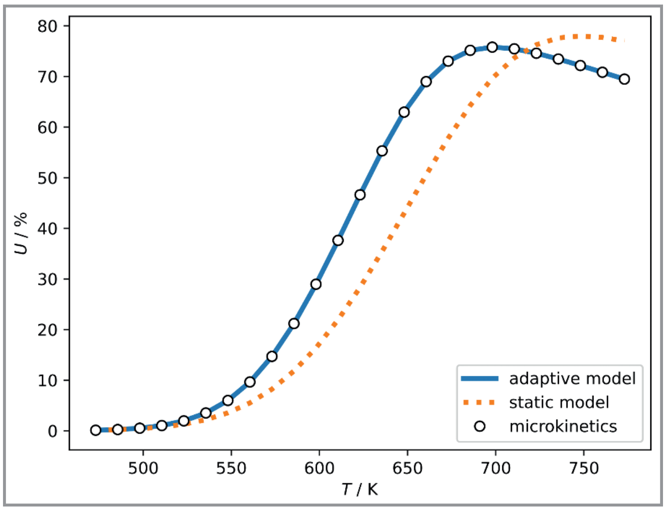


Figure 8. CO conversion U in a water-gas shift reactor as predicted by the online-adaptive kernel model (solid line), the static kernel model (dotted line), and the full kinetic model (symbols) for different temperatures T. The kernel models have not been trained for these conditions (see Sect. 2.3 for details).

Supporting Information

Supporting Information for this article can be found under DOI: https://doi.org/10.1002/cite.202300178

Acknowledgments

Funded by BMBF under contracts 05M20RDA and 05M20VSA. The authors acknowledge the funding of the project by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2075 – 390740016. Open access funding enabled and organized by Projekt DEAL.

Symbols used

A	$[s^{-1}]$	preexponential factor
$\mathbf{A}_\theta$	[–]	kernel optimization matrix
b	[–]	number of output values
c	$[mol\,m^{-3}]$	concentration
d	[–]	number of output values
E_A	$[J\,mol^{-1}]$	activation energy
f	[–]	generic function
g	[–]	goal property
g_{LB}	[–]	lower bound on g
I_d	[–]	identity matrix
k_{ads}	$[m^3\,mol^{-1}s^{-1}]$	adsorption rate constant
k_j	$[s^{-1}]$	rate constant
M	$[kg\,mol^{-1}]$	molar mass
N	[–]	number of data
p_i	[Pa]	partial pressure
$\mathcal{P}_n$	[–]	power function
R	$[J\,mol^{-1}K^{-1}]$	universal gas constant
r	$[s^{-1}]$	reaction rate
$\dot{s}$	$[mol\,m^{-3}s^{-1}]$	source term
s_0	[–]	sticking coefficient
T	[K]	temperature
t	[s]	time
U	[–]	conversion
x	[–]	input values
y	[–]	output values

Greek symbols

α	[–]	kernel parameter
β	[–]	temperature exponent
$\Delta_{abs,goal}$	[–]	upper bound to the absolute error in g
Φ	[–]	generic, scalar-valued function
κ	[–]	kernel
ν	[–]	stochiometric coefficient
θ	[–]	surface coverages
τ	[s]	residence time

Sub-/superscripts

fwd	forward direction
i	species i
j	reaction j
rev	reverse direction

References

[1] M. R. Dobbelaere, P. P. Plehiers, R. Van de Vijver, C. V. Stevens, K. M. Van Geem, *Engineering* **2021**, *7 (9)*, 1201–1211. DOI: https://doi.org/10.1016/j.eng.2021.03.019

[2] F. A. Döppel, M. Votsmeier, *Chem. Eng. Sci.* **2022**, *262*, 117964. DOI: https://doi.org/10.1016/j.ces.2022.117964

[3] F. A. Döppel, M. Votsmeier, *React. Chem. Eng.* **2023**, *8*, 2620–2631. DOI: https://doi.org/10.1039/D3RE00212H

[4] J. Yin, J. Li, I. A. Karimi, X. Wang, *Chem. Eng. J.* **2023**, *452 (P3)*, 139487. DOI: https://doi.org/10.1016/j.cej.2022.139487

[5] B. R. Goldsmith, J. Esterhuizen, J.-X. Liu, C. J. Bartel, C. Sutton, *AIChE J.* **2018**, *64*, 2311–2323.

[6] G. Santin, B. Haasdonk, in *Model Order Reduction: System- and Data-Driven Methods and Algorithms* (Eds: P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. Schilders, L. M. Silveira), vol. 2, De Gruyter, Berlin **2021**.

[7] T. Wenzel, F. Marchetti, E. Perracchione, *arXiv preprint arXiv:2301.08047*, **2023**.

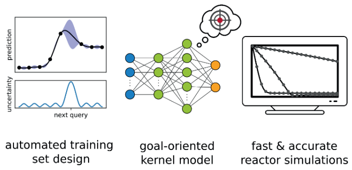
[8] T. Wenzel, B. Haasdonk, H. Kleikamp, M. Ohlberger, F. Schindler, in *Large-Scale Scientific Computing* (Eds: I. Lirkov, S. Margenov), Springer International Publishing, New York **2023**.

[9] D. Micale, C. Ferroni, R. Uglietti, M. Bracconi, M. Maestri, *Chem. Ing. Tech.* **2022**, *94 (5)*, 634–651. DOI: https://doi.org/10.1002/cite.202100196
[10] G. D. Wehinger, M. Ambrosetti, R. Cheula, Z.-B. Ding, M. Isoz, B. Kreitz, K. Kuhlmann, M. Kutscherauer, K. Niyogi, J. Poissonnier, et al., *Chem. Eng. Res. Des.* **2022**, *184*, 39–58. DOI: https://doi.org/10.1016/j.cherd.2022.05.030
[11] M. Bracconi, *Chem. Eng. Process.* **2022**, *181*, 109148. DOI: https://doi.org/10.1016/j.cep.2022.109148
[12] G. D. Wehinger, *Chem. Ing. Tech.* **2022**, *94 (9)*, 1215–1216. DOI: https://doi.org/10.1002/cite.202255272
[13] M. Bracconi, M. Maestri, A. Cuoci, *AIChE J.* **2017**, *63 (1)*, 95–104. DOI: https://doi.org/10.1002/aic.15441
[14] T. S. Brown, H. Antil, R. Löhner, F. Togashi, D. Verma, in *Lecture Notes in Computer Science (LCNS)*, vol. 12761, Springer International Publishing, New York **2021**.
[15] M. Votsmeier, *Chem. Eng. Sci.* **2009**, *64 (7)*, 1384–1389. DOI: https://doi.org/10.1016/j.ces.2008.12.006
[16] M. Votsmeier, A. Scheuer, A. Drochner, H. Vogel, J. Gieshoff, *Catal. Today* **2010**, *151 (3/4)*, 271–277. DOI: https://doi.org/10.1016/j.cattod.2010.01.018
[17] O. Deutschmann, in *Handbook of Heterogeneous Catalysis* (Eds: G. Ertl, H. Knözinger, F. Schüth, J. Weitkamp), 2nd ed., Wiley-VCH, Weinheim **2008**.
[18] S. Matera, M. Maestri, A. Cuoci, K. Reuter, *ACS Catal.* **2014**, *4 (11)*, 4081–4092. DOI: https://doi.org/10.1021/cs501154e
[19] J. M. Blasi, R. J. Kee, *Comput. Chem. Eng.* **2016**, *84*, 36–42. DOI: https://doi.org/10.1016/j.compchemeng.2015.08.020
[20] B. Partopour, A. G. Dixon, *Comput. Chem. Eng.* **2016**, *88*, 126–134. DOI: https://doi.org/10.1016/j.compchemeng.2016.02.015
[21] R. Uglietti, M. Bracconi, M. Maestri, *React. Chem. Eng.* **2020**, *5 (2)*, 278–288. DOI: https://doi.org/10.1039/C9RE00440H
[22] E. A. Daymo, M. Hettel, O. Deutschmann, G. D. Wehinger, *Chem. Eng. Sci.* **2022**, *250*, 117408. DOI: https://doi.org/10.1016/j.ces.2021.117408
[23] S. Barwey, V. Raman, *Energies* **2021**, *14 (9)*, 2710. DOI: https://doi.org/10.3390/en14092710
[24] S. Mazumder, *Comput. Chem. Eng.* **2005**, *30 (1)*, 115–124. DOI: https://doi.org/10.1016/j.compchemeng.2005.08.008
[25] S. B. Pope, *Combust. Theor. Model.* **1997**, *1 (1)*, 41–63. DOI: https://doi.org/10.1088/1364-7830/1/1/006
[26] R. Uglietti, M. Bracconi, M. Maestri, *React. Chem. Eng.* **2018**, *3 (4)*, 527–539. DOI: https://doi.org/10.1039/C8RE00050F
[27] A. Scheuer, W. Hauptmann, A. Drochner, J. Gieshoff, H. Vogel, M. Votsmeier, *Appl. Catal., B* **2012**, *111/112*, 445–455. DOI: https://doi.org/10.1016/j.apcatb.2011.10.032
[28] B. Partopour, A. G. Dixon, *AIChE J.* **2017**, *63 (1)*, 87–94. DOI: https://doi.org/10.1002/aic.15422
[29] M. Klingenberger, O. Hirsch, M. Votsmeier, *Comput. Chem. Eng.* **2017**, *98*, 21–30. DOI: https://doi.org/10.1016/j.compchemeng.2016.12.005
[30] J. M. Lorenzi, T. Stecher, K. Reuter, S. Matera, *J. Chem. Phys.* **2017**, *147 (16)*, 164106. DOI: https://doi.org/10.1063/1.4997286
[31] J. E. Sutton, J. M. Lorenzi, J. T. Krogel, Q. Xiong, S. Pannala, S. Matera, A. Savara, *ACS Catal.* **2018**, *8 (6)*, 5002–5016. DOI: https://doi.org/10.1021/acscatal.8b00713
[32] A. Scheuer, O. Hirsch, R. Hayes, H. Vogel, M. Votsmeier, *Catal. Today* **2011**, *175 (1)*, 141–146. DOI: https://doi.org/10.1016/j.cattod.2011.03.036
[33] T. Nien, J. P. Mmbaga, R. E. Hayes, M. Votsmeier, *Chem. Eng. Sci.* **2013**, *93*, 362–375. DOI: https://doi.org/10.1016/j.ces.2013.01.059
[34] D. Fokina, O. Iliev, P. Toktaliev, I. Oseledets, F. Schindler, *arXiv preprint arXiv:2204.11719*, **2022**.
[35] B. Partopour, R. C. Paffenroth, A. G. Dixon, *Comput. Chem. Eng.* **2018**, *115*, 286–294. DOI: https://doi.org/10.1016/j.compchemeng.2018.04.019
[36] M. Bracconi, M. Maestri, *Chem. Eng. J.* **2020**, *400*, 125469. DOI: https://doi.org/10.1016/j.cej.2020.125469
[37] T. Kircher, F. A. Döppel, M. Votsmeier, *ChemRxiv preprint*, **2023**. DOI: https://doi.org/10.26434/chemrxiv-2023-rpr35
[38] S. Matera, W. F. Schneider, A. Heyden, A. Savara, *ACS Catal.* **2019**, *9 (8)*, 6624–6647. DOI: https://doi.org/10.1021/acscatal.9b01234
[39] D. Wirtz, B. Haasdonk, *Dolomites Res. Notes Approx.* **2013**, *6*, 83–100.
[40] B. Haasdonk, H. Kleikamp, M. Ohlberger, F. Schindler, T. Wenzel, *SIAM J. Sci. Comput.* **2023**, *45 (3)*, A1039–A1065.
[41] H. Wendland, *Scattered Data Approximation*, vol. 17, Cambridge University Press, Cambridge **2005**.
[42] T. Wenzel, G. Santin, B. Haasdonk, *Constr. Approx.* **2023**, *57 (1)*, 45–74.
[43] A. B. Mhadeshwar, D. G. Vlachos, *J. Phys. Chem. B.* **2004**, *108 (39)*, 15246–15258. DOI: https://doi.org/10.1021/jp048698g
[44] W. Hauptmann, M. Votsmeier, H. Vogel, D. G. Vlachos, *Appl. Catal., A* **2011**, *397 (1/2)*, 174–182. DOI: https://doi.org/10.1016/j.apcata.2011.02.031
[45] P. E. Van Keken, D. A. Yuen, L. R. Petzold, *Geophys. Astrophys. Fluid Dyn.* **1995**, *80 (1/2)*, 57–74. DOI: https://doi.org/10.1080/03091929508229763
[46] J. W. Hines, in *Proceedings of International Conference on Neural Networks (ICNN'96)*, vol. 2, IEEE, Piscataway **1996**.
[47] R. M. French, *Trends Cognit. Sci.* **1999**, *3 (4)*, 128–135.
[48] J. Kirkpatrick et al., *Proc. Natl. Acad. Sci. U.S.A.* **2017**, *114 (13)*, 3521–3526. DOI: https://doi.org/10.1073/pnas.1611835114

DOI: 10.1002/cite.202300178

Goal-Oriented Two-Layered Kernel Models as Automated Surrogates for Surface Kinetics in Reactor Simulations

Felix Döppel, Tizian Wenzel, Robin Herkert, Bernard Haasdonk*, Martin Votsmeier**


prediction
uncertainty
next query
automated training set design
goal-oriented kernel model
fast & accurate reactor simulations

Research Article: Specialized kernel models are introduced as a plug-and-play solution to significantly accelerate simulations of reactive systems without requiring machine learning expertise from the users. The workflow reduces the need for expensive training data and ensures accuracy even for extrapolation to unseen reaction conditions. .... ■

Supporting Information
available online

Chemie Ingenieur Technik

Supporting Information

Goal-Oriented Two-Layered Kernel Models as Automated Surrogates for Surface Kinetics in Reactor Simulations

Felix Döppel[1,*], Tizian Wenzel[2], Robin Herkert[2], Bernard Haasdonk[2], Martin Votsmeier[1,3,*]

DOI: 10.1002/cite.202300178

[1]Technical University of Darmstadt, Peter-Grünberg-Straße 8, 64287 Darmstadt, Germany
[2]University of Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany
[3]Umicore AG & Co. KG, Rodenbacher Chaussee 4, 63457 Hanau, Germany
[*]felix.doeppel@tu-darmstadt.de ; martin.votsmeier@tu-darmstadt.de

1 Preferential Oxidation Mechanism

Table S1: Reactions and parameters for the PROX mechanism taken from [1] originating from [2]

No.	Reaction	s_0 or A_0	β	E_A	$\frac{dE_A}{d\theta_H}$	$\frac{dE_A}{d\theta_O}$	$\frac{dE_A}{d\theta_{OH}}$	$\frac{dE_A}{d\theta_{H_2O}}$	$\frac{dE_A}{d\theta_{CO}}$
		unitless or s^{-1}	unitless	kJ/mol	kJ/mol	kJ/mol	kJ/mol	kJ/mol	kJ/mol
H_2 oxidation on Pt									
R1	$H_2 + 2 * \rightleftharpoons 2 H*$	1.287×10^{-1}	0.8584	-	-	-	-	-	-
		7.953×10^{12}	1.911	79.09	-25.10	0	0	0	0
R2	$O_2 + 2 * \rightleftharpoons 2 O*$	5.423×10^{-1}	0.7656	-	-	-	-	-	-
		8.406×10^{12}	0.9275	208.9	0	-133.9	0	0	0
R3	$OH* + * \rightleftharpoons O* + H*$	1.950×10^{12}	1.3286	111.2	6.7321	-4.87	0	62.89	0
		6.325×10^{12}	1.0812	32.48	-5.820	22.26	0	-41.71	0
R4	$H_2O* + * \rightleftharpoons OH* + H*$	9.358×10^{12}	-0.3949	74.57	4.795	50.33	52.30	-43.25	0
		9.989×10^{12}	-0.3664	53.43	-7.757	-87.75	-52.30	71.81	0
R5	$H_2O* + O* \rightleftharpoons 2 OH*$	4.316×10^{10}	0.3262	36.74	0	70.10	52.30	-83.68	0
		1.700×10^{10}	0.5285	94.32	0	-139.12	-52.30	136.0	0
R6	$H_2O + * \rightleftharpoons H_2O*$	1.084×10^{-1}	1.1624	-	-	-	-	-	-
		2.033×10^{12}	2.490	39.98	0	0	104.6	-10.46	0
CO oxidation on Pt									
R7	$CO + * \rightleftharpoons CO*$	1.000×10^{0}	0	-	-	-	-	-	-
		5.659×10^{15}	0.3946	165.9	0	0	0	0	-62.70
R8	$CO_2 + * \rightleftharpoons CO_2*$	1.950×10^{-1}	0.2500	-	-	-	-	-	-
		3.626×10^{12}	0.2459	11.60	0	0	0	0	0
R9	$CO_2* + * \rightleftharpoons CO* + O*$	4.178×10^{10}	-0.2778	110.4	0	4.27	0	0	24.21
		2.393×10^{11}	0.4558	85.42	0	-17.69	0	0	-38.53
Coupling reactions between CO- and H_2 oxidation									
R10	$CO_2* + H* \rightleftharpoons CO* + OH*$	8.031×10^{8}	-0.3259	23.22	-6.276	60.25	0	-45.56	18.12
		1.245×10^{9}	0.8237	76.91	6.276	-77.83	0	59.04	-44.63
R11	$COOH* + * \rightleftharpoons CO* + OH*$	8.426×10^{8}	0.02577	22.66	0	60.25	0	-45.56	18.13
		1.187×10^{9}	0.4719	77.46	0	-77.83	0	59.94	-44.63
R12	$COOH* + * \rightleftharpoons CO_2* + H*$	1.058×10^{11}	0.5812	4.975	6.276	0	0	0	0
		9.454×10^{10}	-0.1098	6.079	-6.2844	0	0	0	0
R13	$CO* + H_2O* \rightleftharpoons COOH* + H*$	1.103×10^{11}	0.4911	98.71	4.904	0	52.30	-5.230	-31.38
		9.070×10^{10}	-0.02778	22.79	-7.657	0	-52.30	5.230	31.38
R14	$CO_2* + OH* \rightleftharpoons COOH* + O*$	5.349×10^{10}	0.05272	109.5	0	-41.13	0	52.30	0
		1.870×10^{11}	0.4515	29.70	0	30.00	0	-52.30	0
R15	$CO_2* + H_2O* \rightleftharpoons COOH* + OH*$	8.642×10^{10}	-0.0481	71.31	0	52.45	52.30	-44.87	0
		1.157×10^{11}	0.5647	49.07	0	-85.62	-52.30	70.19	0
R16	$CO_2* + H* \rightleftharpoons HCOO* + *$	1.117×10^{11}	-0.07525	73.49	-12.55	0	0	0	0
		8.957×10^{10}	0.4218	0.00	0	0	0	0	0
R17	$CO_2* + OH* \rightleftharpoons HCOO* + O*$	6.168×10^{10}	-0.3443	154.8	0	-67.47	0	100.2	0
		1.621×10^{11}	-0.1617	2.577	0	3.657	0	-4.351	0
R18	$CO_2* + H_2O* \rightleftharpoons HCOO* + OH*$	1.022×10^{11}	-0.3574	111.8	0	118.7	90.99	-98.17	0
		9.785×10^{10}	0.06665	17.15	0	-19.40	-13.61	16.89	0

2 Goal-oriented-P-greedy

Using the bound on the absolute prediction error also a bound for the relative error in the target value g can be derived and from that we develop a selection criterion. Using the mean value theorem the absolute error $\left|g(f(x)) - g\left(\hat{f}_{X_n}(x)\right)\right|$ can be bounded by

$$\left|g(f(x)) - g\left(\hat{f}_{X_n}(x)\right)\right| \le \max_{\xi\in\mathrm{conv}\left(f(x),\hat{f}_{X_n}(x)\right)} \left|\left|\nabla g(\xi)\right|\right|_2 \cdot \left|\left|f(x) - \hat{f}_{X_n}(x)\right|\right|_2, \tag{$S1$}$$

where conv(⋅,⋅) indicates the convex hull, which here is the line segment between two points. Next, we assume that we can bound the term $\max_{\xi\in\mathrm{conv}\left(f(x),\hat{f}_{X_n}(x)\right)} \left|\left|\nabla g(\xi)\right|\right|_2$ with

$$\max_{\xi\in\mathrm{conv}\left(f(x),\hat{f}_{X_n}(x)\right)} \left|\left|\nabla g(\xi)\right|\right|_2 \le J_{\mathrm{UB}}\left(C, \mathcal{P}_n(x), \hat{f}_{X_n}(x)\right) \tag{$S2$}$$

where $J_{\mathrm{UB}}\left(C, \mathcal{P}_n(x), \hat{f}_{X_n}(x)\right)$ denotes a fast computable upper bound which may be depending on the power function evaluation $\mathcal{P}_n(x)$, the prediction $\hat{f}_{X_n}(x)$ and the constant C. Then, by inserting $\left|\left|f(x) - \hat{f}_{X_n}(x)\right|\right|_2 \le C\mathcal{P}_n(x)$ in Eq. (S1), we get the bound

$$\left|g(f(x)) - g\left(\hat{f}_{X_n}(x)\right)\right| \le CJ_{\mathrm{UB}}\left(C, \mathcal{P}_n(x), \hat{f}_{X_n}(x)\right)\mathcal{P}_n(x) =: \Delta_{\mathrm{abs,goal}}(x). \tag{$S3$}$$

As the constant C is usually not known in advance, we assume that we have estimates $C_n \approx C$. These estimates can be computed from the current kernel model $\hat{f}_{X_n}$. Replacing the constant C in the error bound by the estimates C_n yields the error estimator (and no longer rigorous upper bound)

$$\hat{\Delta}^n_{\mathrm{abs,goal}}(x) \coloneqq C_n J_{\mathrm{UB}}\left(C_n, \mathcal{P}_n(x), \hat{f}_{X_n}(x)\right)\mathcal{P}_n(x). \tag{$S4$}$$

In order to derive a relative error estimator, we have to find a lower bound for $g(f(x))$. We start with the Taylor expansion

$$g(f(x)) = g\left(\hat{f}_{X_n}(x)\right) + \nabla g(\xi)^T\left(f(x) - \hat{f}_{X_n}(x)\right). \tag{$S5$}$$

For $\left|\left|\nabla g(\xi)\right|\right|_2$ and $\left|\left|f(x) - \hat{f}_{X_n}(x)\right|\right|_2$ from our previous calculations we have the bounds Eq. (S2) and Eq. (20) from the main manuscript. Thus, if

$$CJ_{\mathrm{UB}}\left(C, \mathcal{P}_n(x), \hat{f}_{X_n}(x)\right)\mathcal{P}_n(x) < \left|g\left(\hat{f}_{X_n}(x)\right)\right|, \tag{$S6$}$$

we have the lower bound

$$\begin{aligned} g(f(x)) &\ge g_{LB}\left(x, C, \mathcal{P}_n(x), \hat{f}_{X_n}(x)\right) \\ &\coloneqq \left|g(\hat{f}_{X_n}(x)\right| - CJ_{\mathrm{UB}}\left(C, \mathcal{P}_n(x), \hat{f}_{X_n}(x)\right)\mathcal{P}_n(x). \end{aligned} \tag{$S7$}$$

Replacing the constant C with the estimate C_n yields the approximation

$$\hat{g}^n_{LB}(x) \coloneqq g_{LB}\left(x, C_n, \mathcal{P}_n(x), \hat{f}_{X_n}(x)\right) \approx g_{LB}\left(x, C, \mathcal{P}_n(x), \hat{f}_{X_n}(x)\right). \tag{$S8$}$$

We now define the relative goal error estimate as

$$\hat{\Delta}^n_{\mathrm{rel,goal}}(x) \coloneqq \frac{\hat{\Delta}^n_{\mathrm{abs,goal}}(x)}{\hat{g}^n_{LB}(x)}. \tag{$S9$}$$

If

$$\hat{g}^n_{LB}(x) \le 0 \tag{$S10$}$$

this kind of lower bound cannot be used because the relative error estimate would be negative. This especially may happen if $g(f(x))$ is close to zero. In this case, a relative error estimation is not possible and we use a weighted absolute error estimation $w \cdot \hat{\Delta}^n_{\text{abs,goal}}(x)$ with weight $w \in \mathbb{R}$. As *goal-oriented-P-greedy* selection we define the center selection following the rule

$$x_{n+1} = \operatorname{argmax}_{x \in X_N \setminus X_n} \hat{\Delta}^n_{\text{rel,w-abs,goal}}(x). \qquad (S11)$$

with

$$\hat{\Delta}^n_{\text{rel,w-abs,goal}}(x) := \begin{cases} \hat{\Delta}^n_{\text{rel,goal}}(x) & \text{if } \hat{g}^n_{LB}(x) > \epsilon \\ w \cdot \hat{\Delta}^n_{\text{abs,goal}}(x) & \text{else} \end{cases} \qquad (S12)$$

for some $\epsilon > 0$.

2.1 Application to surface kinetics

In order to make the *goal-oriented P-greedy* applicable to the chemical problem we need to compute the constant $J_{\text{UB}}\left(C, \mathcal{P}_n(x), \hat{f}_{X_n}(x)\right)$. The function $g: \mathbb{R}^2 \to \mathbb{R}$ is given by $g(y) = e^{y_1} - e^{y_2}$ as it computes the net reaction rate from the logarithmic forward and backward rate. The derivative of g is $\nabla g(y) = (e^{y_1}, -e^{y_2})^T$. Thus, $||\nabla g(y)||_2 = \sqrt{e^{2y_1} + e^{2y_2}}$. Because

$$\left|\hat{f}_{X_N,i}(x) - f_i(x)\right| \leq \left|\left|\hat{f}_{X_n}(x) - f(x)\right|\right| \leq C\mathcal{P}_n(x) \qquad (S13)$$

we have $\xi_i \leq \hat{f}_{X_N,i}(x) + C\mathcal{P}_n(x)$ for all $\xi \in \operatorname{conv}\left(f(x), \hat{f}_{X_n}(x)\right)$. Finally, because $||\nabla g||_2$ is monotically increasing in y_1 and y_2 we get

$$\begin{aligned} \max_{\xi \in \operatorname{conv}\left(f(x), \hat{f}_{X_n}(x)\right)} ||\nabla g(\xi)|| &= \max_{\xi \in \operatorname{conv}\left(f(x), \hat{f}_{X_n}(x)\right)} \sqrt{e^{2\xi_1} + e^{2\xi_2}} \\ &\leq \sqrt{e^{2\left(\hat{f}_{X_N,1}(x) + C\mathcal{P}_n(x)\right)} + e^{2\left(\hat{f}_{X_N,2}(x) + C\mathcal{P}_n(x)\right)}} \\ &=: J_{\text{UB}}\left(C, \mathcal{P}_n(x), \hat{f}_{X_n}(x)\right). \end{aligned} \qquad (S14)$$

Lastly, we have to choose the weight w for the *goal-oriented-P-greedy*. In [3] a threshold of $\frac{c_i}{\tau}$ for $|g(f(x))|$ is assumed, where c_i denotes the current concentration. Thus, we choose $w = \left(\frac{c_i}{\tau}\right)^{-1}$.

3 Input-Output Relation

Kernel models are used to predict the steady state source terms of O_2 and CO as a function of the reaction conditions (temperature *T* and partial pressures p_i). The corresponding input-output relation is strongly non-linear (Fig. S1).

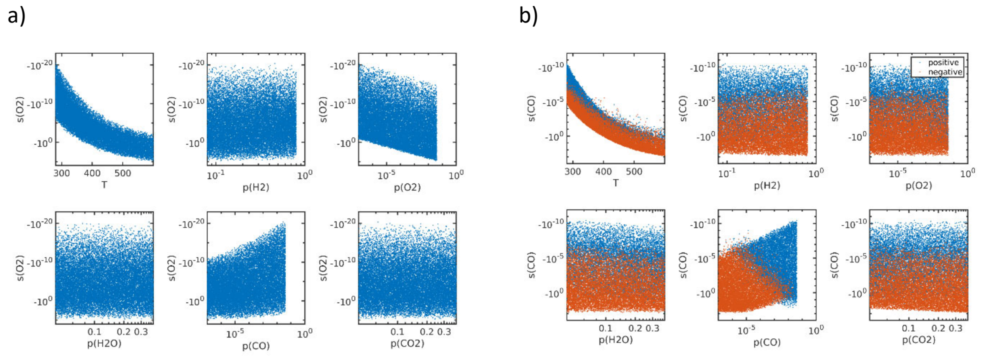


Figure S1: Scatter plot of **a)** O_2 and **b)** CO source terms as a function of reaction conditions.

4 Reactor Simulation Accuracy

Fig. 5 in the main manuscript shows a comparison of the exact and predicted concentration profiles in a preferential oxidation reactor. Fig. S2 shows that the deviation between both solutions is below 0.1 **‰** at any point.

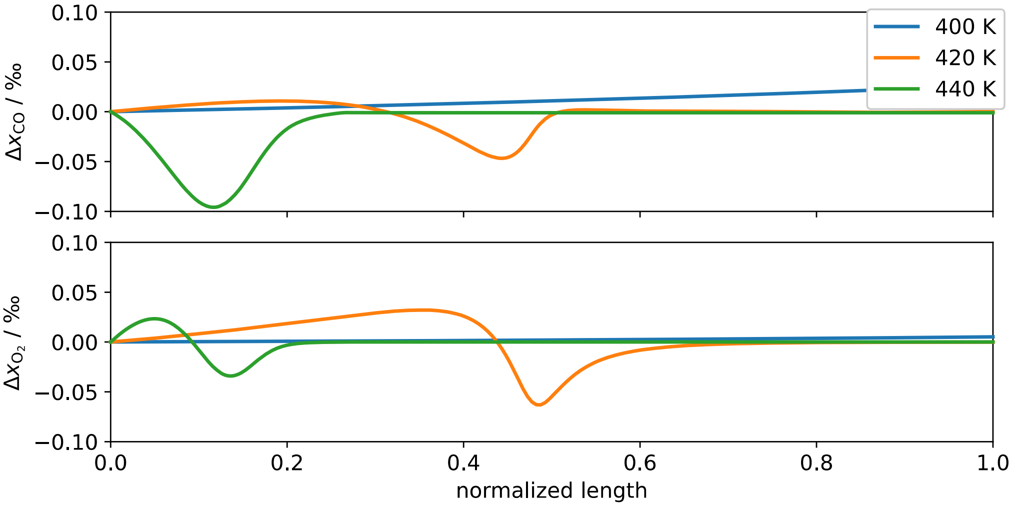


Figure S2: Deviation between exact and predicted concentrations in the preferential oxidation reactor.

5 Inference Time

Source term prediction time t_{predict} for 10000 reaction conditions is averaged over 10 runs with the models used for the preferential oxidation plug-flow reactor simulations on an AMD Ryzen 7 3700X CPU (Tab. S2). Computing the exact solution on the same machine takes 183 times longer.

Table S2: Kernel model prediction times t_{predict} for 10000 reaction conditions averaged over 10 measurements with the same models as used for the preferential oxidation plug-flow reactor simulations.

t_{predict}(VKOGA) / s	t_{predict}(Fortran) / s	Speed-up
1.268	231.9	183

6 Online adaptive model evaluations

Online adaptive kernel models are updated in simulations on-the-fly whenever necessary. Fig. S1 shows how the number of costly microkinetic model evaluations grows during the simulation of the 25 plug-flow reactor model simulations contained in Fig. 5 in the main manuscript up to 72. The standard simulation without kernel model surrogate to the surface kinetics evaluates the microkinetic model in each of the 200 cells in all 25 simulations, resulting in 5000 costly function calls in total. Consequently, the online-adaptive kernel models requires ≈70 times fewer calls to the full microkinetic model.

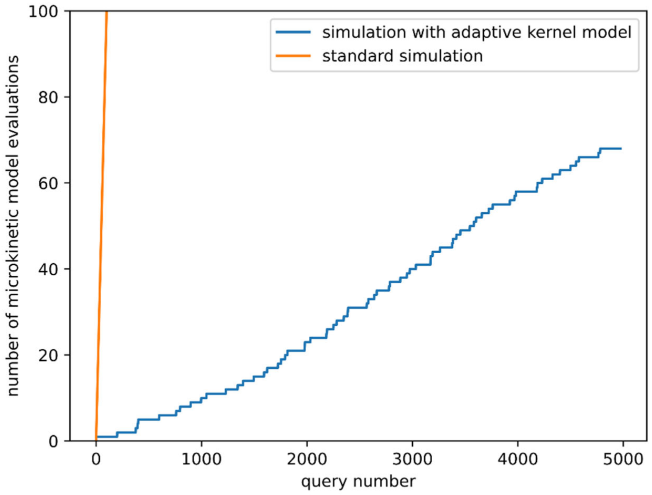


Figure S3: Number of microkinetic model evaluations of a standard plug-flow reactor simulation compared to a simulation using the online-adaptive kernel model as surrogates to the surface kinetics. The standard simulation evaluates the microkinetic model for all of the 5000 queries. The adaptive kernel model requests evaluations only, if the estimated prediction error is above 5%.

References

[1] W. Hauptmann, M. Votsmeier, H. Vogel, D. G. Vlachos, *Appl. Catal., A*. **2011**, *397* (1-2), 174–182. DOI: 10.1016/j.apcata.2011.02.031.

[2] A. B. Mhadeshwar, D. G. Vlachos, *J. Phys. Chem. B*. **2004**, *108* (39), 15246–15258. DOI: 10.1021/jp048698g.

[3] F. A. Döppel, M. Votsmeier, *Chem. Eng. Sci*. **2022**, *262*, 117964. DOI: 10.1016/j.ces.2022.117964.

4.4 Robust Mechanism Discovery with Atom Conserving Chemical Reaction Neural Networks

Atom conservation is one of the most fundamental laws of chemical systems. Traditionally, it is enforced in kinetic mechanisms by multiplying reaction rates with a consistent stoichiometric matrix. The chemical reaction neural network aims to discover unknown reaction pathways from experimental concentration data by learning kinetic parameters within a Neural ODE setting. The fourth publication [21] presents a novel atom conservation layer which constrains the learned stoichiometric coefficients to physically plausible solutions. The weights of this layer are computed before the training of a model using readily available implementations of linear algebra algorithms. Importantly, this step requires only the knowledge about the elemental composition of the relevant chemical species. As the resulting mechanisms are physically consistent, they are more robust towards limited data availability, noisy data and systematic measurement errors.

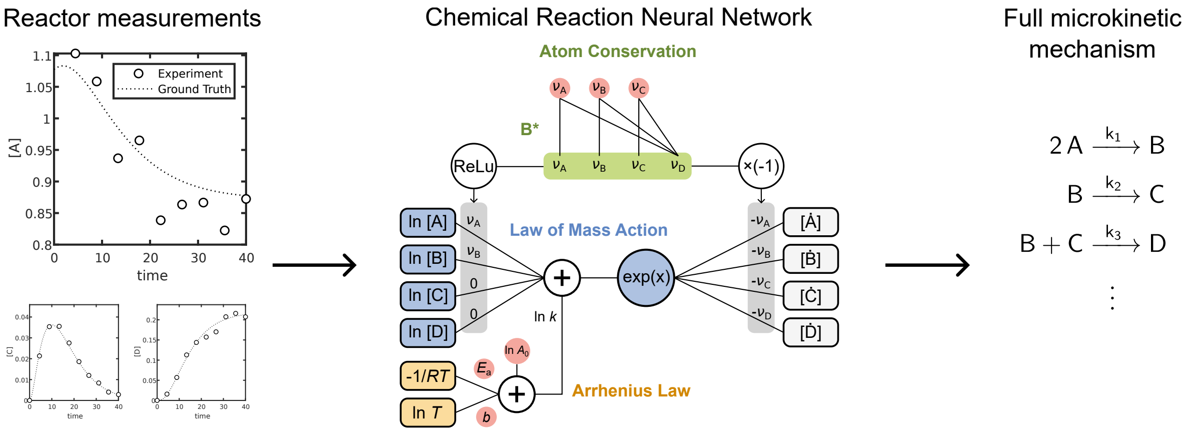


Figure 4.4: Graphical abstract of the fourth publication contained within this work [21].

Reprinted from F. A. Döppel and M. Votsmeier, *ChemRxiv*, **2023**, Version 1, DOI: 10.26434/chemrxiv-2023-1r389. The article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 4.0 International License.

Robust Mechanism Discovery with Atom Conserving Chemical Reaction Neural Networks

Felix A. Döppel[1], Martin Votsmeier[1,2,*]

[1] *Technical University of Darmstadt, Peter-Grünberg-Straße 8, 64287 Darmstadt, Germany*
[2] *Umicore AG & Co. KG, Rodenbacher Chaussee 4, 63457 Hanau, Germany*
[*] *martin.votsmeier@tu-darmstadt.de*

Abstract

Chemical reaction neural networks (CRNNs) established as the state-of-the-art tool for autonomous mechanism discovery. While they encode some fundamental physical laws, mass- and atom conservation are still violated. We enforce atom conservation by adding a dedicated neural network layer which can be interpreted as constraining the model to physically realizable stoichiometries. Using the standard test cases of the original CRNN paper, we show that the resulting atom conserving chemical reaction neural networks improve training stability and speed, offer robustness against noisy and missing data, and require less data overall. As a result, we anticipate increased model reliability and greater utilization of the potential of real-world data sets. We also discuss the potential of the new atom balance layer for other applications in combustion modeling and beyond, such as mechanism reduction and kinetic surrogate models for reactive flow simulations.

Keywords: Atom conservation, Chemical reaction neural networks, Kinetic model, Mechanism discovery, Physics enhanced machine learning

1. Introduction

Machine learning has emerged as an important tool in combustion chemistry discovery, reduction and acceleration [1]. The performance of neural networks for these applications is significantly improved by implementing *a priori* physical knowledge in the model's structure. Frequently studied examples in the combustion context are implementing the overall mass or species balances [2, 3]. In this work we focus on implementing the atom balance in neural networks. Previously, this has either been done explicitly through a soft constraint in the loss function [4, 5], or a post-processing step [6], or indirectly by embedding the stoichiometric matrix into the model's structure [7–10]. However, in case of mechanism discovery and reduction, the stoichiometric matrix is generally unknown and subject to optimization. We propose a dedicated element balance layer for neural networks models of chemical kinetics that enforces atom conservation as a hard constraint without requiring the stoichiometric matrix. We implement this layer into the chemical reaction neural network (CRNN) recently developed by JI and DENG [11], and demonstrate that enforcing the atom conservation greatly increases the model's ability to identify reaction mechanisms from low quality data.

The CRNN is a digital twin of the classic chemical reaction network that encodes the Arrhenius equation (Eq. 1) and the mass-action law (Eq. 2) in a neural network

$$k_j = A_0 \cdot T^{\beta} \cdot \exp\left(\frac{-E_A}{R \cdot T}\right) \tag{1}$$

$$r_j = k_j \cdot \prod_i a_i^{\nu_{i,j}} \tag{2}$$

with the rate constant k_j of reaction j, the pre-exponential A_0, the temperature T, the temperature exponent β, the activation energy E_A, the universal gas constant R, the reaction rate r_j, the activity a_i of species i and the reaction orders $\nu_{i,j}$.

BARWEY and RAMAN used such a digital twin to accelerate chemical source term evaluations of large combustion mechanisms to facilitate high-fidelity simulations of turbulent flames [12]. The main advantage of the CRNN, however, is the ability to autonomously discover and reduce mechanisms using readily available integral reactor measurements with the neural ordinary differential equation (neural ODE) technique [13]. JI and DENG used this approach to obtain reaction mechanisms from several chemical and biochemical engineering systems [11]. It has further been applied to biomass pyrolysis [3], decomposition of energetic materials [14–16], hydrogen as well as methane combustion [17], and *HyChem* models [18].

The current CRNN implementation infringes the fundamental law of atom conservation. We enforce this law through our atom balance layer, building on the original CRNN implementation. The resulting atom conserving chemical reaction neural networks (AC-CRNN) increase training stability and speed, provide robustness against noisy and missing data, and reduce the overall amount of data required. This is an important step to learn from imperfect data as they are typically obtained from experiments.

For simplicity, we will consider the standard example systems from JI and DENG [11]. We showcase the superior AC-CRNN performance under the influence of three realistic imperfect data scenarios: 1. limited data availability, 2. noisy data and 3. systematic measurement errors.

2. Atom Conserving Chemical Reaction Neural Networks

Conventional CRNNs find the stoichiometric coefficients of a reaction mechanism by freely optimizing the weights of their output layer. However, not all combinations of stoichiometric coefficients satisfy the fundamental law of atom conservation. A physically realizable set of stoichiometric coefficients ν_i fulfills the equation

$$0 = \sum_i N_{k,i} \cdot \nu_i \tag{3}$$

with the molecular matrix $\mathbf{N}$ that contains the number of atoms of type k per species i [19]. In other words: All physically realizable vectors ν are in the null space of $\mathbf{N}$. A basis $\mathbf{B}$ of this null space is conveniently computed by the MATLAB function `null` or the Julia function `LinearAlgebra.nullspace` which are based on singular value decomposition. Now all valid sets of stoichiometric coefficients can be expressed as a weighted sum of the basis vectors

$$\nu = \mathbf{B} \cdot w. \tag{4}$$

A neural network that learns the weights w and multiplies them with the basis $\mathbf{B}$ to obtain stoichiometric coefficients will always fulfill atom conservation. We implement Eq. 4 by adding a dedicated element balance layer to the CRNN (Fig 1) and call the resulting architecture *atom conserving chemical reaction neural networks* (AC-CRNN). The matrix $\mathbf{B}$ has $N_{\text{key}} = N_{\text{species}} - \text{rank}(\mathbf{N})$ columns, which is fewer than the total number of species present in the reaction system N_{species} [20]. Therefore, the element balance layer does not only enforce atom conservation but also reduces the number of trainable parameters. In our AC-CRNN, the stoichiometric coefficients are further used to derive the reaction orders of the reactants using the ReLu function

$$\text{ReLu}(x) = \max(x, 0) \tag{5}$$

To further increase the interpretability of the AC-CRNN, we propose to apply the concept of key species by converting the null space basis matrix $\mathbf{B}$ into reduced column echelon form. This means that the first N_{key} rows contain a identity matrix. The matrix conversion can be conveniently achieved using

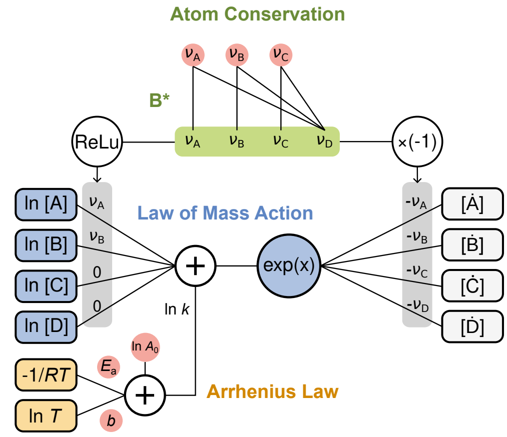


Fig. 1: Schematic representation of an AC-CRNN reaction node. It encodes the law of mass action, the Arrhenius law and in contrast to the original CRNN also the atom conservation. Latter is achieved by multiplication of the key species coefficients with the conservation matrix $\mathbf{B}^*$. Multiple such reaction nodes are combined in a single hidden layer to build up a chemical reaction neural network. Trainable parameters are highlighted in red.

the `rref` function in MATLAB or the Julia library *RowEchelon.jl*. We call this basis the atom conservation matrix $\mathbf{B}^*$. For test case 1,

$$\mathbf{B}^* = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1/3 & -2/3 & -1/3 & -1/3 \end{bmatrix} \tag{6}$$

Here, the first N_{key} weights w are directly mapped to the first N_{key} stoichiometric coefficients. We call those species whose coefficients are identical to the learned weights w the key species. Which species are treated as key species can be chosen by the user through the order of the species in the vector ν. Always the first N_{key} species are treated as key species. The coefficients of the other species are a weighted sum of the key species coefficients.

Altogether, implementing the element balance layer adds a minimum amount of additional code to the original CRNN. The matrix B* is computed in a fully automated preprocessing step. The code for this step is supplied in the appendix. Due to the reduced number of trainable parameters and the regularization provided by the additional physical constraints, AC-CRNNs generally train faster than the corresponding CRNN. All results presented in this work are obtained using the basis $\mathbf{B}^*$. It has been found, that the AC-CRNN that uses $\mathbf{B}^*$ shows better performance than the one using the unconverted basis $\mathbf{B}$. Further, the performance depends on the choice of the key species.

3. Methods

In practice, species concentrations at certain positions of the reactor are measured instead of source terms. Therefore, the CRNN is trained in the context of a neural ODE [13], i.e. wrapped with an ODE solver. The resulting CRNN concentration profiles $c^{\text{CRNN}}(t)$ are compared to the provided concentration data $c^{\text{data}}(t)$. We use the Julia language implementation of CRNN available at `https://github.com/DENG-MIT/CRNN`. It uses the differential programming package *DifferentialEquations.jl* [21] to enable backpropagation of gradients through the ODE solver. The mean absolute error MAE loss (Eq. 7) of the normalized concentrations (Eq. 8) is used and minimized using the ADAM optimizer [22] to adjust the CRNN parameters.

$$\text{loss} = \mathrm{MAE}\left(c_{\text{norm}}^{\text{CRNN}}(t), c_{\text{norm}}^{\text{data}}(t)\right) \tag{7}$$

$$c_{\text{norm}} = \frac{c(t)}{\text{range}(c^{\text{data}}(t))} \tag{8}$$

In test case 1 the initial CRNN parameters are randomly drawn from a standard normal distribution and divided by 1000. The ADAM algorithm is used for 15 000 epochs with a learning rate of 0.001, an exponential decay for the first (0.9) and second (0.999) momentum estimate and a weight decay of 10^{-8}. In test case 2 the initial CRNN parameters are randomly drawn from a normal distribution and divided by 10, for $\ln A_0$ and activation energies 0.8 is added and absolute values are used for activation energies. The ADAM algorithm is used for 10 000 epochs with a learning rate of 0.005, an exponential decay for the first (0.9) and second (0.999) momentum estimate and a weight decay of 10^{-6}.

4. Results and Discussion

We demonstrate, that embedding the atom balance into neural networks facilitates mechanism discovery. Test case 1 is a demonstration system considering mass conservation to reduce the required amount of training data. Test case 2 is a realistic example of biodiesel production kinetics where embedding the atom balance increases the model's robustness against noise and offsets in the training data. Finally, we discuss further applications in surrogate modeling and mechanism reduction.

4.1. Test Case 1 - Trimerization

Test case 1 is a representative example for mechanism discovery introduced by SEARSON *et al.* [23] and describes the trimerization of a generic molecule A. This could for example be the formation of benzene from ethyne or the homotrimerization of proteins, such as porins [24] or hemagglutinin [25]. The reaction system consists of five species called A, A_2, A^*, A^{**}, and A_3 that are involved in four reactions:

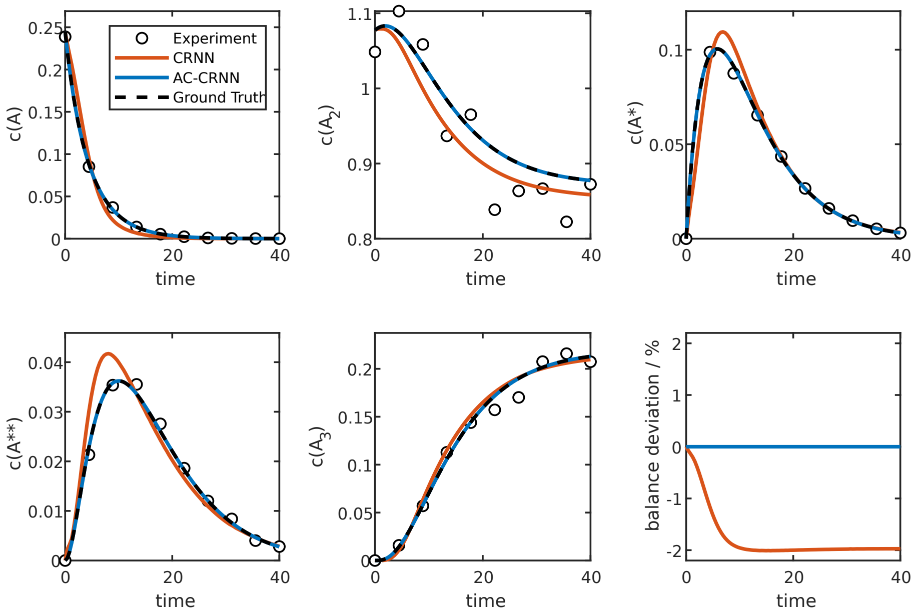


Fig. 2: Predicted concentration profiles obtained by fitting standard CRNN and atom conserving CRNN (AC-CRNN) to 20 synthetic integral reactor experiments with 10 noisy concentration measurements each are compared with the reference model (groud truth). The CRNN overestimates the consumption of species B and looses about 2% of the total mass. The AC-CRNN is biased towards a physically plausible solution by the embedded element balance and therefore hardly distinguishable from the exact solution.

$$
\begin{aligned}
2\,\mathrm{A} &\xrightarrow{\mathrm{k_1}} \mathrm{A_2}\\
\mathrm{A} &\xrightarrow{\mathrm{k_2}} \mathrm{A^*}\\
\mathrm{A^*} &\xrightarrow{\mathrm{k_3}} \mathrm{A^{**}}\\
\mathrm{A_2} + \mathrm{A^{**}} &\xrightarrow{\mathrm{k_4}} \mathrm{A_3}
\end{aligned}
$$

The rate constants $k_1 - k_4$ are not temperature dependent and are assumed to have the values 0.3, 0.1, 0.2, and 0.13 respectively. Initial concentrations of species A and $\mathrm{A_2}$ are randomly chosen with uniform distribution between 0.2 and 1.2, the other species are not present in the initial mixture. Experimental concentration measurements are emulated by integrating this initial value problem with the Tsitouras 5/4 Runge-Kutta method [21] up to a reaction time of 40 s, sampling data at equidistant time intervals and adding 5% gaussian noise.

Ji and Deng showed that CRNNs are able to recover the mechanism from 20 of those simulated isothermal experiments with 100 data points each [11]. We test the CRNN performance for even fewer data (10 points per experiment), increasing the problem difficulty significantly. To tackle this problem, we introduce atom conserving CRNN (AC-CRNN) that embed the atom conservation matrix $\mathbf{B}^*$ into the CRNN (Fig. 1). The conservation matrix is obtained as the reduced column echelon form of the null space basis of the molecular matrix (Tab. 1) using the MATLAB functions `null` and `rref`.

Table 1: The molecular matrix shows the composition of the species of the trimerization case.

	A
A	1
$\mathrm{A_2}$	2
$\mathrm{A^*}$	1
$\mathrm{A^{**}}$	1
$\mathrm{A_3}$	3

In this example, the rank of the molecular matrix is one, so there will be one dependent species and four key species. Without loss of generality, we choose species $\mathrm{A_3}$ as the dependent species. The resulting atom conservation matrix (Eq. 6) is used to calculate the stoichiometric coefficient of $\mathrm{A_3}$ as a weighted sum of the coefficients of the other four species. This guarantees atom conservation and reduces the number of trainable parameters.

Figure 2 shows that the original CRNN models tends to overestimate the formation of the intermediate species. Resulting inconsistencies in the stoichiometric matrix of the model lead to a violation of atom conservation. The AC-CRNN, however, is

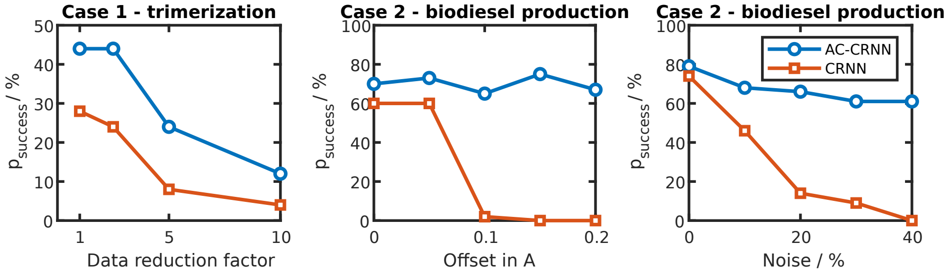


Fig. 3: Fraction of successful mechanism discovery attempts p_{success} of atom conserving AC-CRNN compared to standard CRNN in various scenarios of realistic data flaws. Due to the additional physical bias (here, atom conservation), AC-CRNN are much more robust against left: reduced data availability middle: systematic measurement errors and right: lower data quality.

forced to obey atom conservation and therefore much more likely to identify the correct mechanism. For example, AC-CRNN solutions that overestimate the formation of intermediate species automatically underestimate other species and are therefore penalized with a higher loss.

To further characterize the performance of the networks, we discuss the success probability p_{success} of the training, see Saerson et al. [23]. It is defined as the fraction of successful mechanism discovery attempts from different initial model states. Here, we consider a mechanism discovery attempt successful, if every estimated stoichiometric coefficient differs by less than 0.1 from the ground truth. The success probability of the original CRNN drops from 25% using 100 data points to 4% using only 10 data points per experiment (Fig. 3). Enforcing atom conservation increases the success probability to 44% and 12% respectively.

4.2. Test Case 2 - Biodiesel Production

Test case 2 considers biodiesel production, as studied by BURNHAM *et al.* [26]. DARNOKO and CHERYAN [27] described the transesterification of palm oil derived palmitin glycerides (TG, DG, and MG) with methanol MeOH to smaller methyl esters $\mathrm{RCO_2Me}$ by three consecutive reactions:

$$\mathrm{TG} + \mathrm{MeOH} \xrightarrow{\mathrm{k_1}} \mathrm{DG} + \mathrm{RCO_2Me}$$
$$\mathrm{DG} + \mathrm{MeOH} \xrightarrow{\mathrm{k_2}} \mathrm{MG} + \mathrm{RCO_2Me}$$
$$\mathrm{MG} + \mathrm{MeOH} \xrightarrow{\mathrm{k_3}} \mathrm{GL} + \mathrm{RCO_2Me}$$

The temperature dependence of the rate constants is described by the arrhenius equation with the preexponentials A_0 (18.60, 19.13, and 7.93), the activation energies E_A (14.54, 14.42, and 6.47) kcal/mol and a temperature exponent of 0 for all three reactions. The Tsitouras 5/4 Runge-Kutta method with automatic switching to an order 2/3 L-Stable Rosenbrock method [21] is used to integrate the potentially stiff initial value problem.

A CRNN is used to identify reaction orders, stoichiometric coefficients, activation energies and the preexponential factors. For this, 20 experiments with random initial concentrations between 0.2 and 2.2 arbitrary units at temperatures randomly chosen between 323 K to 343 K are provided. Each experiment consists of 50 noisy (5% gaussian noise) concentration measurements taken after a time step of 1 seconds each. Collecting such an amount of precise measurements is experimentally very challenging, so after we showed with test case 1 that AC-CRNN provide accurate results even with small amounts of data, we now test the robustness against systematic measurement errors in the provided training data in form of a sensor offset that overestimates the concentrations of species TG by 0.2.

Table 2: The molecular matrix shows the elemental composition of the species of the biodiesel production case.

	C	H	O
TG	51	98	6
MeOH	1	4	1
DG	35	68	5
MG	19	38	4
GL	3	8	3
$\mathrm{RCO_2Me}$	17	34	2

The system contains six species and can be balanced in terms of the three elements carbon, hydrogen and oxygen (Tab. 2). Without loss of generality we choose methanol, the di- and the triglyceride as key species. The stoichiometric coefficients of the other three species are inferred from the key species coefficients using the atom conservation matrix

$$B^* = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -3 & 1 & -2 \\ 2 & -1 & 1 \\ 0 & -1 & 0 \end{bmatrix} \tag{9}$$

where the rows correspond to all species (TG, MeOH, DG, MG, GL, and $\mathrm{RCO_2Me}$) and the columns correspond to the key species (TG, MeOH, and DG). Bal-

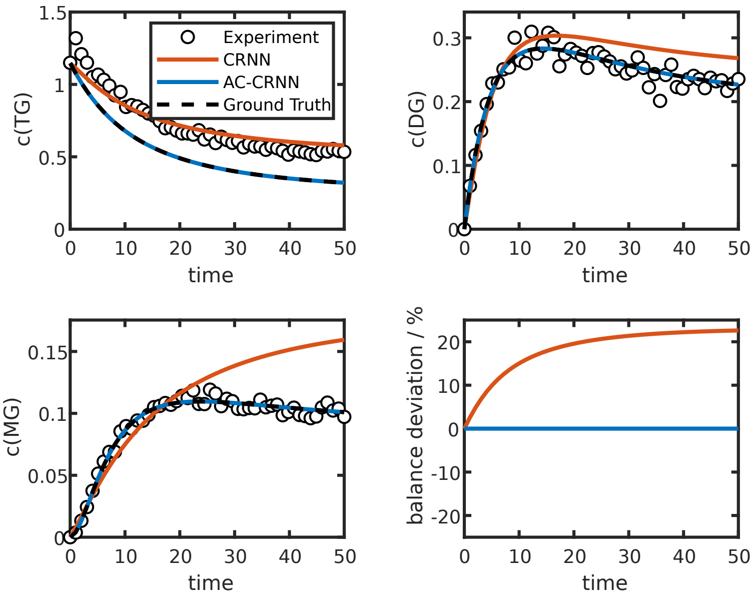


Fig. 4: Biodiesel production concentration profiles obtained from mechanism discovery by standard CRNN and atom conserving CRNN (AC-CRNN) using synthetic measurements from 20 integral reactor experiments. To mimic realistic concentration measurements the data given for model training (”Experiment”) are perturbed by 5% gaussian noise. The triglyceride data are additionally shifted by an offset of 0.2. As the CRNN adapts to the artificially high concentrations in the reactant (TG), too much of the glyceride products (DG and MG) is formed, resulting in a significant deviation from the element balance. The AC-CRNN is constrained by the embedded element balance and therefore not affected by the sensor offset.

ancing in terms of molecular groups such as glycerol, acid rest, and methyl rest would lead to the same conservation matrix.

Figure 4 shows a typical CRNN prediction in case of a sensor offset. As the model adapts to the erroneously high concentrations of the reactant TG, an excessive amount of the other glycerides DG and MG is formed. This leads to an error in the atom balance by more than 20%. The AC-CRNN has 37.5% fewer parameters to optimize and trains correspondingly faster. Further, it is stable towards the sensor offset (Fig. 4) and shows a perfectly closed atom balance. It successfully recovers the correct mechanism in 70% of the runs, whereas the original CRNN is not successful in any out of 60 runs. Similarly, increasing the amount of gaussian noise applied to the concentration measurements from 5% to 40%, the success probability of the standard CRNN drops to 0%, while the AC-CRNN is remains successful in 60% of the runs (Fig. 3). Here, mechanism discovery is considered successful, if every estimated stoichiometric coefficient differs by less than 0.2 from the ground truth.

4.3. Applications of the atom balance layer beyond CRNNs

Because the proposed atom balance layer can be combined with any feed forward neural network that predicts kinetics, it should find widespread use in combustion and beyond, for example in surrogate modeling and mechanism reduction.

Aside from CRNN, our element balance layer should be useful in conjunction with other network structures that discover reaction mechanisms and thus lack an a priori stoichiometric matrix. One example is the PolyODEnet by Wu *et al.* [28].

The proposed key species approach can be applied intuitively to small reaction systems without explicitly using the matrix $\mathbf{B}^*$. This is a common way to achieve atom conservation in surrogate models of catalytic systems [29–33]. Our element balance layer formalizes this approach, allowing its application to more complex systems.

Finally, our approach is easily extended to also implement the charge balance relevant for redox- and electrochemical reactions.

5. Conclusion

Chemical reaction neural networks (CRNN) have established as the most advanced tool for autonomous mechanism discovery and are used in many fields, such as (bio-) chemical engineering, pyrolysis, and combustion. While they encode the law of mass action as well as the Arrhenius law, mass- and atom conservation are still violated.

We enforce the fundamental law of atom conservation by adding a dedicated neural network layer which can be interpreted as constraining stoichiometric coefficients to physically realizable combinations. The resulting atom conserving chemical reaction neural networks (AC-CRNN) improve training stability and speed, offer robustness against noisy and missing data, and require less data overall. As a result, we anticipate increased model reliability and greater utilization of the potential of real-world data sets. Our proposed element conservation layer is compatible with any feed forward neural network that predicts kinetics and should therefore be useful also for surrogate modeling and mechanism reduction.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was funded by BMBF in the framework of the project ML-MORE (contract 05M20RDA). The authors gratefully acknowledge the computing time provided on the high-performance computer Lichtenberg at the NHR Centers NHR4CES at TU Darmstadt.

Appendix

The generation of the atom conservation matrix $\mathbf{B}^*$ from the molecular matrix $\mathbf{N}$ is demonstrated using the following MATLAB code using the example of test case 1:

```
N = [1,2,1,1,3]';
B_star = rref(null(N')')'
```

References

[1] T. Echekki, A. Farooq, M. Ihme, S. M. Sarathy, Machine Learning for Combustion Chemistry, Springer International Publishing, Cham, 2023, pp. 117–147. doi:10.1007/978-3-031-16248-0_5.

[2] A. J. Sharma, R. F. Johnson, D. A. Kessler, A. Moses, Deep Learning for Scalable Chemical Kinetics, in: AIAA Scitech 2020 Forum, Vol. 1 PartF, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2020. doi:10.2514/6.2020-0181.

[3] W. Ji, F. Richter, M. J. Gollner, S. Deng, Autonomous kinetic modeling of biomass pyrolysis using chemical reaction neural networks, Combustion and Flame 240 (2022) 111992. arXiv:2105.11397, doi:10.1016/j.combustflame.2022.111992.

[4] C. Chi, X. Xu, D. Thévenin, Efficient premixed turbulent combustion simulations using flamelet manifold neural networks: A priori and a posteriori assessment, Combustion and Flame 245 (August) (2022) 112325. doi:10.1016/j.combustflame.2022.112325.

[5] A. Almeldein, N. Van Dam, Accelerating Chemical Kinetics Calculations With Physics Informed Neural Networks, Journal of Engineering for Gas Turbines and Power 145 (9) (2023) 1–14. doi:10.1115/1.4062654.

[6] K. Wan, C. Barnaud, L. Vervisch, P. Domingo, Machine learning for detailed chemistry reduction in DNS of a syngas turbulent oxy-flame with side-wall effects, Proceedings of the Combustion Institute 38 (2) (2021) 2825–2833. doi:10.1016/j.proci.2020.06.047.

[7] P. O. Sturm, A. S. Wexler, Conservation laws in a neural network architecture: enforcing the atom balance of a Julia-based photochemical model (v0.2.0), Geoscientific Model Development 15 (8) (2022) 3417–3431. doi:10.5194/gmd-15-3417-2022.

[8] T. Kircher, F. A. Döppel, M. Votsmeier, A neural network with embedded stoichiometry and thermodynamics for learning kinetics from reactor data (2023) 1–35doi:10.26434/chemrxiv-2023-rpr35.

[9] A. Fedorov, A. Perechodjuk, D. Linke, Kinetics-Constrained Neural Ordinary Differential Equations: Artificial Neural Network Models tailored for Small Data to boost Kinetic Model Development (2023) 1–25doi:10.26434/chemrxiv-2023-x39xt.

[10] F. Sorourifar, Y. Peng, I. Castillo, L. Bui, J. Venegas, J. A. Paulson, Physics-Enhanced Neural Ordinary Differential Equations: Application to Industrial Chemical Reaction Systems, Industrial & Engineering Chemistry Research 62 (38) (2023) 15563–15577. doi:10.1021/acs.iecr.3c01471.

[11] W. Ji, S. Deng, Autonomous Discovery of Unknown Reaction Pathways from Data by Chemical Reaction Neural Network, Journal of Physical Chemistry A 125 (4) (2021) 1082–1092. arXiv:2002.09062, doi:10.1021/acs.jpca.0c09316.

[12] S. Barwey, V. Raman, A Neural Network-Inspired Matrix Formulation of Chemical Kinetics for Acceleration on GPUs, Energies 14 (9) (2021) 2710. doi:10.3390/en14092710.

[13] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud, Neural Ordinary Differential Equations (jun 2018). arXiv:1806.07366.

[14] H. Wang, Y. Xu, M. Wen, W. Wang, Q. Chu, S. Yan, S. Xu, D. Chen, Kinetic modeling of CL-20 decomposition by a chemical reaction neural network, Journal of Analytical and Applied Pyrolysis 169 (November 2022) (2023) 105860. doi:10.1016/j.jaap.2023.105860.

[15] G. Tang, H. Wang, C. Chen, Y. Xu, D. Chen, D. Wang, Y. Luo, X. Li, Thermal decomposition of nano Al-based energetic composites with fluorinated energetic polyurethane binders: experimental and theoretical understandings for enhanced combustion and energetic performance, RSC Advances 12 (37) (2022) 24163–24171. doi:10.1039/d2ra03781e.

[16] Y. Xu, Q. Chu, X. Chang, H. Wang, S. Wang, S. Xu, D. Chen, Thermal decomposition mechanism of 1,3,5-trinitroperhydro-1,3,5-triazine: Experiments and reaction kinetic modeling, Chemical Engineering Science 282 (June) (2023) 119234. doi:10.1016/j.ces.2023.119234.

[17] J. Huang, Y. Zhou, W. A. Yong, Data-driven discovery of multiscale chemical reactions governed by the law of mass action, Journal of Computational Physics 448 (2022) 110743. arXiv:2101.06589, doi:10.1016/j.jcp.2021.110743.

[18] W. Ji, J. Zanders, J.-W. Park, S. Deng, Machine Learning Approaches to Learn HyChem Models (2021). arXiv:2104.07875.

[19] D. R. Schneider, G. Reklaitis, On material balances for chemically reacting systems, Chemical Engineer-

ing Science 30 (2) (1975) 243–247. doi:10.1016/0009-2509(75)80012-1.

[20] M. Baerns, A. Behr, H. Hofmann, J. Gmehling, U. Onken, A. Renken, K.-O. Hinrichsen, R. Palkovits, Technische chemie, John Wiley & Sons, 2013.

[21] C. Rackauckas, Q. Nie, DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia, Journal of Open Research Software 5 (1) (2017) 15. doi:10.5334/jors.151.

[22] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[23] D. P. Searson, M. J. Willis, A. Wright, Reverse Engineering Chemical Reaction Networks from Time Series Data, Statistical Modelling of Molecular Descriptors in QSAR/QSPR 2 (2012) 327–348. doi:10.1002/9783527645121.ch12.

[24] S. Galdiero, A. Falanga, M. Cantisani, R. Tarallo, M. Elena Della Pepa, V. D'Oriano, M. Galdiero, Microbe-Host Interactions: Structure and Role of Gram-Negative Bacterial Porins, Current Protein and Peptide Science 13 (8) (2012) 843–854. doi:10.2174/138920312804871120.

[25] C. S. Copeland, R. W. Doms, E. M. Bolzau, R. G. Webster, A. Helenius, Assembly of influenza hemagglutinin trimers and its role in intracellular transport., The Journal of cell biology 103 (4) (1986) 1179–1191. doi:10.1083/jcb.103.4.1179.

[26] S. C. Burnham, D. P. Searson, M. J. Willis, A. R. Wright, Inference of chemical reaction networks, Chemical Engineering Science 63 (4) (2008) 862–873. doi:10.1016/j.ces.2007.10.010.

[27] D. Darnoko, M. Cheryan, Kinetics of palm oil transesterification in a batch reactor, JAOCS, Journal of the American Oil Chemists' Society 77 (12) (2000) 1263–1267. doi:10.1007/s11746-000-0198-y.

[28] Q. Wu, T. Avanesian, X. Qu, H. Van Dam, PolyODENet: Deriving mass-action rate equations from incomplete transient kinetics data, Journal of Chemical Physics 157 (16) (2022). doi:10.1063/5.0110313.

[29] M. Votsmeier, Efficient implementation of detailed surface chemistry into reactor models using mapped rate data, Chemical Engineering Science 64 (7) (2009) 1384–1389. doi:10.1016/j.ces.2008.12.006.

[30] F. A. Döppel, M. Votsmeier, Efficient machine learning based surrogate models for surface kinetics by approximating the rates of the rate-determining steps, Chemical Engineering Science 262 (2022) 117964. doi:10.1016/j.ces.2022.117964.

[31] F. A. Döppel, M. Votsmeier, Efficient neural network models of chemical kinetics using a latent asinh rate transformation, Reaction Chemistry & Engineering 8 (10) (2023) 2620–2631. doi:10.1039/D3RE00212H.

[32] F. A. Döppel, T. Wenzel, R. Herkert, B. Haasdonk, M. Votsmeier, Goal-Oriented Two-Layered Kernel Models as Automated Surrogates for Surface Kinetics in Reactor Simulations, Chemie Ingenieur Technik (2023). doi:10.1002/cite.202300178.

[33] B. Klumpers, T. Luijten, S. Gerritse, E. Hensen, I. Filot, Direct coupling of microkinetic and reactor models using neural networks, Chemical Engineering Journal (2023) 145538doi:10.1016/j.cej.2023.145538.

5 Outlook

High-fidelity reactive flow simulations This work developed reliable and physically consistent surrogate models for the accurate prediction of surface kinetics. They have been implemented and validated in one-dimensional reactor simulations, where they accelerated the numerical solution by up to five orders of magnitude. The next step is to use these tools where they are most urgently needed: high-fidelity simulations of chemical reactors at an industrially relevant scale. First results indicate that the numerical acceleration allows for systematic optimization within days rather than months, representing a significant advancement towards the knowledge-based design of catalytic devices. Furthermore, the accurate Jacobians obtainable through automatic differentiation of the neural network surrogates appear to enhance simulation stability significantly [130].

Global reaction neural networks The follow-up [124] of the first two publications presented in this work [17, 21] makes use of readily available thermodynamic information about gas phase species rather than the much more specific knowledge required for the rate-determining step approach. Specifically, it applies the latent transformation technique [21] to incorporate the De Donder relation [24] into the neural network surrogate structure. This "global reaction neural network" (GRNN) assumes a simplified network of effective global reactions to relate reaction rates and species source terms. This approach has been shown to work well for surrogate modelling of the preferential oxidation of CO as well as steam reforming for fuel cell applications. Further, it can be trained as a Neural ODE to learn kinetics from readily available integral reactor data [124, 125]. In parallel to this work, similar approaches have been reported for ammonia synthesis on Ru and the non-oxidative coupling of methane over a single-atom Fe/SiO_2 catalyst [131], the catalytic hydrogenation of CO_2 to methane and the hydrogenation of CO_2 to higher hydrocarbons [126], as well as methanol synthesis on Cu/Zn-based catalysts [132]. Currently, the optimal choice of the assumed global reactions remains an open question. It is anticipated that establishing best practices and gaining a better understanding of the recently developed Neural ODE technique will further drive the success of the GRNNs.

Benchmarks This work provided several methods of discovering chemical kinetics from data and efficiently modelling existing mechanisms. Now, openly available benchmark problems have to be established to drive further development and maintain comparability between existing and emerging methods. Importantly, the ideal choice of model architecture will depend on the specific problem setting, for example modelling thermodynamically consistent versus inconsistent mechanisms, considering reaction conditions close to or far away from chemical equilibrium, or deriving kinetics from differential or integral reactor data.

Surrogate modelling beyond surface reactive systems The present work builds surrogate models of steady state surface kinetics. Such models are useful because the time scale of surface reactions is generally much shorter than the dynamics in the gas phase. This separation of time scales can also be observed in homogeneous reaction systems. Combustion mechanisms, for instance, typically include multiple “short-lived” radical species that can be safely assumed to instantaneously adjust to the concentrations of the “stable” gas phase species [133]. This effect can also be relevant to atmospheric chemistry, biological systems, and more. There are also similarities with modelling financial data, as the data considered span several orders of magnitude in the positive and negative domains. Therefore, the surrogate modelling approaches proposed in this work should also be useful in these settings.

Bayesian neural networks Bayesian neural networks are a promising approach to quantify the uncertainty of neural network model parameters as well as predictions [108, 109, 134]. In analogy to [19] they could be employed to further improve data efficiency and model reliability without the need to sacrifice the structural flexibility of neural networks.

Chemical reaction neural networks For the chemical reaction neural network to be a truly autonomous approach to mechanism discovery it requires a reliable method to identify the physically correct number of reactions and species involved in a mechanism. To expand the scope towards state-of-the-art combustion mechanisms, more flexibility regarding the reaction type has to be implemented, e.g. to cover third body and falloff reactions. Even more important could be the application to surface reactive systems to elucidate the mechanisms behind catalyst deactivation phenomena, which currently pose a critical technological problem for CO_2 conversion processes [135].

Enforcing conservation laws Currently, the optimal choice of the key species in the atom conservation layer remains an open question. First results imply that it is beneficial to choose those species that undergo the smallest concentration changes. Overall, the conservation layer can be applied to a significantly broader scope than shown within this work. Firstly, analogous to stoichiometric coefficients, it enforces atom conservation of source terms and can therefore be directly implemented for surrogate modelling as well. Finally, the concept can be generalized to enforce the conservation of other properties such as electric charge, momentum, and energy.

Bibliography

[1] D. Agar, M. Bertau, M. Börnhorst, M. Busch, M. Casapu, P. Claus, D. Demtröder, R. Dittmeyer, C. Dreiser, D. Eckes, K. Ehrhardt, B. Etzold, G. Fieg, H. Freund, J. Friedland, J.-D. Grundwaldt, R. Güttel, E. von Harbou, J. Heck, K. Herrera Delgado, T. Herrmann, K.-O. Hinrichsen, R. Horn, J. Khinast, E. Klemm, N. Kockmann, G. Kolios, U. Krewer, R. Kuwertz, M. Löffelholz, G. Luinstra, G. Nilles, H. Marschall, L. Möltner, M. Muhler, U. Nieken, M. Nilles, J. Osiewacz, S. Palkovits, M. Paul, K. Pflug, S. Pinnow, S. Radl, J. Sauer, V. Schallhart, S. Schunk, A. Seidel-Morgenstern, M. Schlüter, M. Schubert, K. Sundmacher, T. Turek, I. Vittorias, H. Vogel, O. Wachsen, G. D. Wehinger, H.-W. Zanthoff, D. Ziegenbalg, *Roadmap Chemical Reaction Engineering : an initiative of the DECHEMA/VDI Subject Division Chemical Reaction Engineering*, 3rd, (Ed.: K. Rübberdt), Gesellschaft für Chemische Technik und Biotechnologie (DECHEMA), **2023**.

[2] M. J. Palys, P. Daoutidis, *Computers & Chemical Engineering* **2022**, *165*, 107948.

[3] G. D. Wehinger, M. Ambrosetti, R. Cheula, Z.-B. Ding, M. Isoz, B. Kreitz, K. Kuhlmann, M. Kutscherauer, K. Niyogi, J. Poissonnier, R. Réocreux, D. Rudolf, J. Wagner, R. Zimmermann, M. Bracconi, H. Freund, U. Krewer, M. Maestri, *Chemical Engineering Research and Design* **2022**, *184*, 39–58.

[4] D. Micale, C. Ferroni, R. Uglietti, M. Bracconi, M. Maestri, *Chemie Ingenieur Technik* **2022**, *94*, 634–651.

[5] M. Bracconi, *Chemical Engineering and Processing - Process Intensification* **2022**, *181*, 109148.

[6] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, D. Hassabis, *Nature* **2021**, *596*, 583–589.

[7] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds, Z. Eaton-Rosen, W. Hu, A. Merose, S. Hoyer, G. Holland, O. Vinyals, J. Stott, A. Pritzel, S. Mohamed, P. Battaglia, *Science* **2023**, *382*, 1416–1421.

[8] A. Krizhevsky, I. Sutskever, G. E. Hinton, *Advances in Neural Information Processing Systems* **2012**.

[9] K. He, X. Zhang, S. Ren, J. Sun, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2015**, 770–778.

[10] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* **2016**, 779–788.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, *Advances in Neural Information Processing Systems* **2017**, *2017-Decem*, 5999–6009.

[12] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, *Advances in Neural Information Processing Systems* **2020**.

[13] A. M. Schweidtmann, E. Esche, A. Fischer, M. Kloft, J. U. Repke, S. Sager, A. Mitsos, *Chemie-Ingenieur-Technik* **2021**, *93*, 2029–2039.

[14] P. Bollini, M. Diwan, P. Gautam, R. L. Hartman, D. A. Hickman, M. Johnson, M. Kawase, M. Neurock, G. S. Patience, A. Stottlemyer, D. G. Vlachos, B. Wilhite, *ACS Engineering Au* **2023**, *3*, 364–390.

[15] M. Raissi, P. Perdikaris, G. Karniadakis, *Journal of Computational Physics* **2019**, *378*, 686–707.

[16] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, *Nature Reviews Physics* **2021**, *3*, 422–440.

[17] F. A. Döppel, M. Votsmeier, *Chemical Engineering Science* **2022**, *262*, 117964.

[18] F. A. Döppel, M. Votsmeier, *Reaction Chemistry & Engineering* **2023**, *8*, 2620–2631.

[19] F. Döppel, T. Wenzel, R. Herkert, B. Haasdonk, M. Votsmeier, *Chemie Ingenieur Technik* **2024**, DOI `10.1002/cite.202300178`.

[20] W. Ji, S. Deng, *Journal of Physical Chemistry A* **2021**, *125*, 1082–1092.

[21] F. Döppel, M. Votsmeier, *ChemRxiv preprint* **2023**, DOI `10.26434/chemrxiv-2023-1r389`.

[22] G. Wedler, H.-J. Freund, *Lehr-und arbeitsbuch physikalische chemie*, Wiley-VCH, Weinheim, **2018**.

[23] A. Jess, P. Wasserscheid, *Chemical technology: An integral textbook*, Wiley-VCH, Weinheim, **2013**.

[24] J. Dumesic, *Journal of Catalysis* **1999**, *185*, 496–505.

[25] W. Reschetilowski, *Einführung in die Heterogene Katalyse*, Springer-Verlag, Berlin, Heidelberg, **2015**.

[26] M. Pineda, M. Stamatakis, *Journal of Chemical Physics* **2022**, *156*, DOI `10.1063/5.0083251`.

[27] S. Matera, W. F. Schneider, A. Heyden, A. Savara, *ACS Catalysis* **2019**, *9*, 6624–6647.

[28] O. Deutschmann in *Handbook of Heterogeneous Catalysis*, (Eds.: G. Ertl, H. Knözinger, F. Schüth, J. Weitkamp), Wiley-VCH, **2008**, Chapter Computational Fluid Dynamics Simulation of Catalytic Reactors, pp. 1811–1821.

[29] E. Hairer, G. Wanner, *Solving Ordinary Differential Equations II*, Springer, Berlin Heidelberg New York, **1996**.

[30] L. Petzold, *SIAM Journal on Scientific and Statistical Computing* **1982**, *3*, 367–384.

[31] M. Baerns, A. Behr, A. Brehm, J. Gmehling, K.-O. Hinrichsen, H. Hofmann, M. Kleiber, N. Kockmann, U. Onken, R. Palkovits, A. Renken, D. Vogt, *Technische Chemie*, 3rd ed., WILEY-VCH Verlag & Co. KGaA, Weinheim, **2023**.

[32] A. L. Samuel, *IBM Journal of research and development* **1959**, *3*, 210–229.

[33] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*, (Eds.: R. Roumeliotis, N. Tache), O'Reilly Media, **2019**.

[34] Google, `scholar.google.com`, Online; accessed 01.03.2024.

[35] T. S. Guzella, W. M. Caminhas, *Expert Systems with Applications* **2009**, *36*, 10206–10222.

[36] M. L. Giger, *Journal of the American College of Radiology* **2018**, *15*, 512–520.

[37] F. Amato, A. López, E. M. Peña-Méndez, P. Vaňhara, A. Hampl, J. Havel, *Journal of Applied Biomedicine* **2013**, *11*, 47–58.

[38] S. Dara, S. Dhamercherla, S. S. Jadav, C. M. Babu, M. J. Ahsan, *Artificial Intelligence Review* **2022**, *55*, 1947–1999.

[39] Mapwith.ai, `www.mapwith.ai`, Online; accessed 30.09.2020.

[40] PyTorch, `pytorch.org`, Online; accessed 30.09.2020.

[41] TensorFlow, `www.tensorflow.org`, Online; accessed 30.09.2020.

[42] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, Q. Zhang, *JAX: composable transformations of Python+NumPy programs*, version 0.3.13, **2018**.

[43] MathWorks, `www.mathworks.com/solutions/machine-learning.html`, Online; accessed 30.09.2020.

[44] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, **2021**.

[45] J. Bezanson, A. Edelman, S. Karpinski, V. B. Shah, *SIAM review* **2017**, *59*, 65–98.

[46] Reddit, `www.reddit.com/r/learnmachinelearning/`, Online; accessed 30.09.2020.

[47] TwoMinutePapers, `www.youtube.com/c/KárolyZsolnai`, Online; accessed 01.03.2024.

[48] T. Hastie, R. Tibshirani, J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, Springer Science & Business Media, **2009**.

[49] C. C. Aggarwal, *Neural networks and deep learning*, Springer, **2018**.

[50] M. Kubat, *An introduction to machine learning*, Springer, **2017**.

[51] R. A. Berk, *Statistical learning from a regression perspective*, *Vol. 14*, Springer, **2008**.

[52] W. S. McCulloch, W. Pitts, *The bulletin of mathematical biophysics* **1943**, *5*, 115–133.

[53] I. J. Goodfellow, J. Pouget-Abadie, B. Mirza, Mehdi and Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, *Advances in neural information processing systems* **2014**, *27*, DOI `10.48550/arXiv.1406.2661`.

[54] N. Shenvi, J. Geremia, H. Rabitz, *The Journal of chemical physics* **2004**, *120*, 9942–9951.

[55] G. Cybenko, *Mathematics of control signals and systems* **1989**, *2*, 303–314.

[56] Z. Lu, H. Pu, F. Wang, Z. Hu, L. Wang, *Advances in neural information processing systems* **2017**, *30*.

[57] D. E. Rumelhart, G. E. Hinton, R. J. Williams, *nature* **1986**, *323*, 533–536.

[58] D. P. Kingma, J. Ba, *arXiv preprint arXiv:1412.6980* **2014**.

[59] D. C. Liu, J. Nocedal, *Mathematical Programming* **1989**, *45*, 503–528.

[60] B. R. Goldsmith, J. Esterhuizen, J.-X. Liu, C. J. Bartel, C. Sutton, *AIChE Journal* **2018**, *64*, 2311–2323.

[61] T. Mou, H. S. Pillai, S. Wang, M. Wan, X. Han, N. M. Schweitzer, F. Che, H. Xin, *Nature Catalysis* **2023**, *6*, 122–136.

[62] A. Corso, D. Karamadian, R. Valentin, M. Cooper, M. J. Kochenderfer, *arXiv preprint arXiv:2307.10586* **2023**.

[63] C. Rudin, *Nature Machine Intelligence* **2019**, *1*, 206–215.

[64] M. R. Dobbelaere, P. P. Plehiers, R. Van de Vijver, C. V. Stevens, K. M. Van Geem, *Engineering* **2021**, *7*, 1201–1211.

[65] S. L. Brunton, J. N. Kutz, *Data-Driven Science and Engineering*, Cambridge University Press, **2022**.

[66] O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, K. R. Müller, *Chemical Reviews* **2021**, *121*, 10142–10186.

[67] J. Brandstetter, M. Welling, D. E. Worrall, *Proceedings of Machine Learning Research* **2022**, *162*, 2241–2256.

[68] S. Mallat, *Philosophical Transactions of the Royal Society A: Mathematical Physical and Engineering Sciences* **2016**, *374*, 20150203.

[69] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, S. Ho, *arXiv preprint arXiv:2003.04630* **2020**.

[70] M. Lutter, J. Peters, *International Journal of Robotics Research* **2023**, *42*, 83–107.

[71] J. Ling, A. Kurzawski, J. Templeton, *Journal of Fluid Mechanics* **2016**, *807*, 155–166.

[72] L. Linden, D. K. Klein, K. A. Kalina, J. Brummund, O. Weeger, M. Kästner, *Journal of the Mechanics and Physics of Solids* **2023**, *179*, 105363.

[73] K. A. Kalina, L. Linden, J. Brummund, M. Kästner, *Computational Mechanics* **2023**, *71*, 827–851.

[74] F. Sorourifar, Y. Peng, I. Castillo, L. Bui, J. Venegas, J. A. Paulson, *Industrial & Engineering Chemistry Research* **2023**, *62*, 15563–15577.

[75] K. Duraisamy, G. Iaccarino, H. Xiao, *Annual Review of Fluid Mechanics* **2019**, *51*, 357–377.

[76] K. Kashinath, M. Mustafa, A. Albert, J. L. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, A. Manepalli, D. Chirila, R. Yu, R. Walters, B. White, H. Xiao, H. A. Tchelepi, P. Marcus, A. Anandkumar, P. Hassanzadeh, Prabhat, *Philosophical Transactions of the Royal Society A: Mathematical Physical and Engineering Sciences* **2021**, *379*, 20200093.

[77] Z. Hao, S. Liu, Y. Zhang, C. Ying, Y. Feng, H. Su, J. Zhu, *arXiv preprint arXiv:2211.08064* **2022**.

[78] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, D. K. Duvenaud, *Advances in neural information processing systems* **2018**, *31*.

[79] W. Ji, S. Deng, *arXiv preprint arXiv:2108.00455* **2021**.

[80] V. F. Nikitin, I. M. Karandashev, M. Y. Malsagov, E. V. Mikhalchenko, *Acta Astronautica* **2022**, *194*, 376–382.

[81] T. S. Brown, H. Antil, R. Löhner, F. Togashi, D. Verma in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *Vol. 12761 LNCS*, **2021**, pp. 23–39.

[82] M. Bracconi, M. Maestri, A. Cuoci, *AIChE Journal* **2017**, *63*, 95–104.

[83] K. McBride, K. Sundmacher, *Chemie-Ingenieur-Technik* **2019**, *91*, 228–239.

[84] C. de Boor, *A Practical Guide to Splines*, *Vol. 27*, Springer-Verlag, New York, **1978**.

[85] M. Votsmeier, *Chemical Engineering Science* **2009**, *64*, 1384–1389.

[86] M. Votsmeier, A. Scheuer, A. Drochner, H. Vogel, J. Gieshoff, *Catalysis Today* **2010**, *151*, 271–277.

[87] A. Scheuer, W. Hauptmann, A. Drochner, J. Gieshoff, H. Vogel, M. Votsmeier, *Applied Catalysis B: Environmental* **2012**, *111-112*, 445–455.

[88] B. Partopour, A. G. Dixon, *Computers & Chemical Engineering* **2016**, *88*, 126–134.

[89] B. Partopour, A. G. Dixon, *AIChE Journal* **2017**, *63*, 87–94.

[90] M. Klingenberger, O. Hirsch, M. Votsmeier, *Computers & Chemical Engineering* **2017**, *98*, 21–30.

[91] S. Matera, M. Maestri, A. Cuoci, K. Reuter, *ACS Catalysis* **2014**, *4*, 4081–4092.

[92] J. M. Lorenzi, T. Stecher, K. Reuter, S. Matera, *Journal of Chemical Physics* **2017**, *147*, 164106.

[93] J. E. Sutton, J. M. Lorenzi, J. T. Krogel, Q. Xiong, S. Pannala, S. Matera, A. Savara, *ACS Catalysis* **2018**, *8*, 5002–5016.

[94] G. Micula, S. Micula, *Handbook of splines*, Springer Science & Business Media, Dodrecht, **1999**.

[95] L. Breiman, *Machine Learning* **2001**, *45*, 5–32.

[96] B. Partopour, R. C. Paffenroth, A. G. Dixon, *Computers & Chemical Engineering* **2018**, *115*, 286–294.

[97] C. T. Campbell, *ACS Catalysis* **2017**, *7*, 2770–2779.

[98] M. Bracconi, M. Maestri, *Chemical Engineering Journal* **2020**, *400*, 125469.

[99] R. Balestriero, J. Pesenti, Y. LeCun, *arXiv preprint arXiv:2110.09485* **2021**.

[100] C. W. Coley, N. S. Eyke, K. F. Jensen, *Angewandte Chemie - International Edition* **2020**, *59*, 22858–22893.

[101] C. W. Coley, N. S. Eyke, K. F. Jensen, *Angewandte Chemie - International Edition* **2020**, *59*, 23414–23436.

[102] J. T. Margraf, K. Reuter, *ACS Omega* **2019**, *4*, 3370–3379.

[103] C. W. Gao, J. W. Allen, W. H. Green, R. H. West, *Computer Physics Communications* **2016**, *203*, 212–225.

[104] M. Liu, A. Grinberg Dana, M. S. Johnson, M. J. Goldman, A. Jocher, A. M. Payne, C. A. Grambow, K. Han, N. W. Yee, E. J. Mazeau, K. Blondal, R. H. West, C. F. Goldsmith, W. H. Green, *Journal of Chemical Information and Modeling* **2021**, *61*, 2686–2696.

[105] F. H. Vermeire, S. U. Aravindakshan, A. Jocher, M. Liu, T.-C. Chu, R. E. Hawtof, R. Van de Vijver, M. B. Prendergast, K. M. Van Geem, W. H. Green, *Energy & Fuels* **2022**, *36*, 1304–1315.

[106] B. Kreitz, K. Blöndal, K. Badger, R. H. West, C. F. Goldsmith, *Digital Discovery* **2024**, *3*, 173–185.

[107] J. Hines in Proceedings of International Conference on Neural Networks (ICNN'96), *Vol. 2*, IEEE, **1996**, pp. 1245–1250.

[108] E. Nieves, R. Dandekar, C. Rackauckas, *Frontiers in Systems Biology* **2024**, *4*, DOI `10.3389/fsysb.2024.1338518`.

[109] Q. Li, H. Chen, B. C. Koenig, S. Deng, *Physical Chemistry Chemical Physics* **2023**, *25*, 3707–3717.

[110] W. Ji, F. Richter, M. J. Gollner, S. Deng, *Combustion and Flame* **2022**, *240*, 111992.

[111] B. C. Koenig, P. Zhao, S. Deng, *Journal of Power Sources* **2023**, *581*, 233443.

[112] H. Wang, Y. Xu, M. Wen, W. Wang, Q. Chu, S. Yan, S. Xu, D. Chen, *Journal of Analytical and Applied Pyrolysis* **2023**, *169*, 105860.

[113] G. Tang, H. Wang, C. Chen, Y. Xu, D. Chen, D. Wang, Y. Luo, X. Li, *RSC Advances* **2022**, *12*, 24163–24171.

[114] Y. Xu, Q. Chu, X. Chang, H. Wang, S. Wang, S. Xu, D. Chen, *Chemical Engineering Science* **2023**, *282*, 119234.

[115] W. Ji, J. Zanders, J.-W. Park, S. Deng, *arXiv preprint arXiv:2104.07875* **2021**.

[116] X. Su, W. Ji, J. An, Z. Ren, S. Deng, C. K. Law, *Combustion and Flame* **2023**, *251*, 112732.

[117] J. Huang, Y. Zhou, W. A. Yong, *Journal of Computational Physics* **2022**, *448*, 110743.

[118] S. Barwey, V. Raman, *Energies* **2021**, *14*, 2710.

[119] T. Kumar, A. Kumar, P. Pal, *arXiv preprint arXiv:2312.00038* **2023**.

[120] C. Chi, X. Xu, D. Thévenin, *Combustion and Flame* **2022**, *245*, 112325.

[121] A. Almeldein, N. Van Dam, *Journal of Engineering for Gas Turbines and Power* **2023**, *145*, 1–14.

[122] K. Wan, C. Barnaud, L. Vervisch, P. Domingo, *Proceedings of the Combustion Institute* **2021**, *38*, 2825–2833.

[123] P. O. Sturm, A. S. Wexler, *Geoscientific Model Development* **2022**, *15*, 3417–3431.

[124] T. Kircher, F. A. Döppel, M. Votsmeier, *Chemical Engineering Journal* **2024**, *485*, 149863.

[125] T. Kircher, F. A. Döppel, M. Votsmeier, *ChemRxiv preprint* **2024**, DOI `10.26434/chemrxiv-2024-10xzj`.

[126] A. Fedorov, A. Perechodjuk, D. Linke, *Chemical Engineering Journal* **2023**, *477*, 146869.

[127] W. Hauptmann, M. Votsmeier, H. Vogel, D. G. Vlachos, *Applied Catalysis A: General* **2011**, *397*, 174–182.

[128] T. Nien, J. Mmbaga, R. Hayes, M. Votsmeier, *Chemical Engineering Science* **2013**, *93*, 362–375.

[129] A. Scheuer, O. Hirsch, R. Hayes, H. Vogel, M. Votsmeier, *Catalysis Today* **2011**, *175*, 141–146.

[130] F. Biermann, R. Uglietti, F. A. Döppel, T. Kircher, M. Bracconi, M. Maestri, M. Votsmeier, “AI accelerated micro-kinetic modelling in heterogeneous catalysis: an application of physically enhanced ANNs to CFD simulations of industrial packed-bed reactors”, Manuscript in preparation, presentation at the Annual Meeting on Reaction Engineering and Electrochemical Processes, **2024**.

[131] S. Kasiraju, D. G. Vlachos, *Reaction Chemistry & Engineering* **2024**, *9*, 119–131.

[132] B. Lacerda de Oliveira Campos, A. Oliveira Souza da Costa, K. Herrera Delgado, S. Pitter, J. Sauer, E. Ferreira da Costa Junior, *Reaction Chemistry & Engineering* **2024**, DOI `10.1039/D3RE00409K`.

[133] S. H. Lam, *Combustion Science and Technology* **1993**, *89*, 375–404.

[134] R. Dandekar, K. Chung, V. Dixit, M. Tarek, A. Garcia-Valadez, K. V. Vemula, C. Rackauckas, *arXiv preprint arXiv:2012.07244* **2020**.

[135] M. I. Alam, R. Cheula, G. Moroni, L. Nardi, M. Maestri, *Catalysis Science and Technology* **2021**, *11*, 6601–6629.

Danksagung

Mein Dank gilt all denen, die mich über die letzten Jahre hinweg fachlich wie persönlich unterstützt und damit wesentlich zum Zustandekommen dieser Arbeit beigetragen haben.

An erster Stelle möchte ich mich bei Prof. Dr. Martin Votsmeier für die Möglichkeit bedanken, in seiner Gruppe an diesem spannenden Thema zu arbeiten. Vielen Dank für die weitreichenden Freiräume in der Bearbeitung und die Möglichkeit, die Ergebnisse dieser Arbeit auf internationalen Konferenzen vorzustellen.

Prof. Dr. Bernard Haasdonk, Dr. Tizian Wenzel und Robin Herkert danke ich für die angenehme Zusammenarbeit und die interessanten Einblicke in die angewandte Mathematik.

Vielen Dank an all meine Kollegen, ausdrücklich auch aus den Arbeitskreisen von Prof. Dr. Bastian Etzold und Prof. Dr. Marc Ledendecker. Aus der gemeinsamen Zeit sind wichtige Freundschaften hervorgegangen. An vorderster Stelle sind hier Felix Biermann, Mareike Schmidt, Lars-Christian Stoltenberg und Marko Malinovic zu nennen. Zudem danke ich insbesondere Riccardo Bono für die humorvolle Zerstreuung in Pausen sowie 陶柳 (Liu Tao) für seine Gastfreundschaft und spannende kulinarische Einblicke.

Des Weiteren möchte ich mich bei allen Masteranden und Forschungsstudenten bedanken, mit denen ich zusammen arbeiten durfte. Die Betreuung hat mir stets Freude bereitet und die gemeinsam erarbeiteten Ergebnisse haben nicht zuletzt mit zur Entstehung dieser Arbeit beigetragen.

Danke auch an alle, die mich schon während des Studiums begleitet haben. Ein besonderer Dank gebührt dabei David Büchner und Dominique Lumpp. Ohne euch wäre diese Zeit ein vollkommen anderes Erlebnis gewesen.

Abschließend möchte ich mich herzlich bedanken bei meiner Mutter und meiner Schwester sowie meiner Partnerin Eva und ihrer Familie für Unterstützung und Verständnis in jeglicher Hinsicht.

Erklärungen laut Promotionsordnung

§ 8 Abs. 1 lit. c PromO

Ich versichere hiermit, dass die elektronische Version meiner Dissertation mit der schriftlichen Version übereinstimmt.

§ 8 Abs. 1 lit. d PromO

Ich versichere hiermit, dass zu einem vorherigen Zeitpunkt noch keine Promotion versucht wurde. In diesem Fall sind nähere Angaben über Zeitpunkt, Hochschule, Dissertationsthema und Ergebnis dieses Versuchs mitzuteilen.

§ 9 Abs. 1 PromO

Ich versichere hiermit, dass die vorliegende Dissertation selbstständig und nur unter Verwendung der angegebenen Quellen verfasst wurde.

§ 9 Abs. 2 PromO

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 26. März 2024

Felix Antonidas Döppel

Erklärung zum Eigenanteil an den Veröffentlichungen der kumulativen Dissertation

Im Folgenden ist aufgelistet, mit welchem Anteil ich an den Veröffentlichungen beteiligt war.

Mein Anteil an der folgenden Veröffentlichung beträgt **80%**.
F. A. Döppel and M. Votsmeier, Efficient machine learning based surrogate models for surface kinetics by approximating the rates of the rate-determining steps, *Chemical Engineering Science*, **2022**, *262*, 117964. DOI: 10.1016/j.ces.2022.117964

Mein Anteil an der folgenden Veröffentlichung beträgt **85%**.
F. A. Döppel and M. Votsmeier, Efficient Neural Network Models of Chemical Kinetics Using a Latent asinh Rate Transformation, *Reaction Chemistry and Engineering*, **2023**, *8*, 2620-2631. DOI: 10.1039/D3RE00212H

Mein Anteil an der folgenden Veröffentlichung beträgt **60%**.
F. A. Döppel, T. Wenzel, R. Herkert, B. Haasdonk and M. Votsmeier, Goal-Oriented Two-Layered Kernel Models as Automated Surrogates for Surface Kinetics in Reactor Simulations, *Chemie Ingenieur Technik*, **2024**, *96*, No.5, 1-11. DOI: 10.1002/cite.202300178

Mein Anteil an der folgenden Veröffentlichung beträgt **85%**.
F. A. Döppel and M. Votsmeier, Robust Mechanism Discovery with Atom Conserving Chemical Reaction Neural Networks, *ChemRxiv*, **2023**, Version 1. DOI: 10.26434/chemrxiv-2023-1r389.

Darmstadt, 26. März 2024

Felix Antonidas Döppel

Erklärung zur Begutachtung der Veröffentlichungen

Weder Referent (Prof. Dr. Martin Votsmeier) noch Korreferent (Prof. Dr. Oliver Weeger) der vorliegenden kumulativen Doktorarbeit waren an der Begutachtung nachstehender Veröffentlichungen beteiligt:

1. F. A. Döppel and M. Votsmeier, Efficient machine learning based surrogate models for surface kinetics by approximating the rates of the rate-determining steps, *Chemical Engineering Science*, **2022**, *262*, 117964. DOI: 10.1016/j.ces.2022.117964
2. F. A. Döppel and M. Votsmeier, Efficient Neural Network Models of Chemical Kinetics Using a Latent asinh Rate Transformation., *Reaction Chemistry & Engineering*, **2023**, *8*, 2620-2631. DOI: 10.1039/D3RE00212H
3. F. A. Döppel, T. Wenzel, R. Herkert, B. Haasdonk and M. Votsmeier, Goal-Oriented Two-Layered Kernel Models as Automated Surrogates for Surface Kinetics in Reactor Simulations, *Chemie Ingenieur Technik*, **2024**, *96*, No. 5, 1-11. DOI: 10.1002/cite.202300178
4. F. A. Döppel and M. Votsmeier, Robust Mechanism Discovery with Atom Conserving Chemical Reaction Neural Networks, *ChemRxiv*, **2023**, Version 1. DOI: 10.26434/chemrxiv-2023-1r389

Datum

Referent
Prof. Dr. Martin Votsmeier

Korreferent
Prof. Dr. Oliver Weeger