Wissen statt Blindflug:
Unsicherheitsbewusste und informierte KI für die industrielle Produktion

Prof. Dr.-Ing. Marco Huber
marco.huber@ipa.fraunhofer.de

Fraunhofer IPA

Direktorat *Digitalisierung und Künstliche Intelligenz*
Forschungsbereich *Künstliche Intelligenz und Maschinelles Sehen*
Fraunhofer IPA, Stuttgart
www.ipa.fraunhofer.de/ki

Universität Stuttgart
Institut für Industrielle Fertigung und Fabrikbetrieb IFF

Kognitive Produktionssysteme
Institut für Industrielle Fertigung und Fabrikbetrieb IFF
Universität Stuttgart
https://www.iff.uni-stuttgart.de

Fraunhofer IPA

Innovationstreiber mit wissenschaftlicher Reputation seit 1959

Auf einen Blick

- 1 000+ Projekte mit Unternehmen pro Jahr
- ~ 1 200 Mitarbeiter an 9 Standorten (Hauptsitz: Stuttgart)

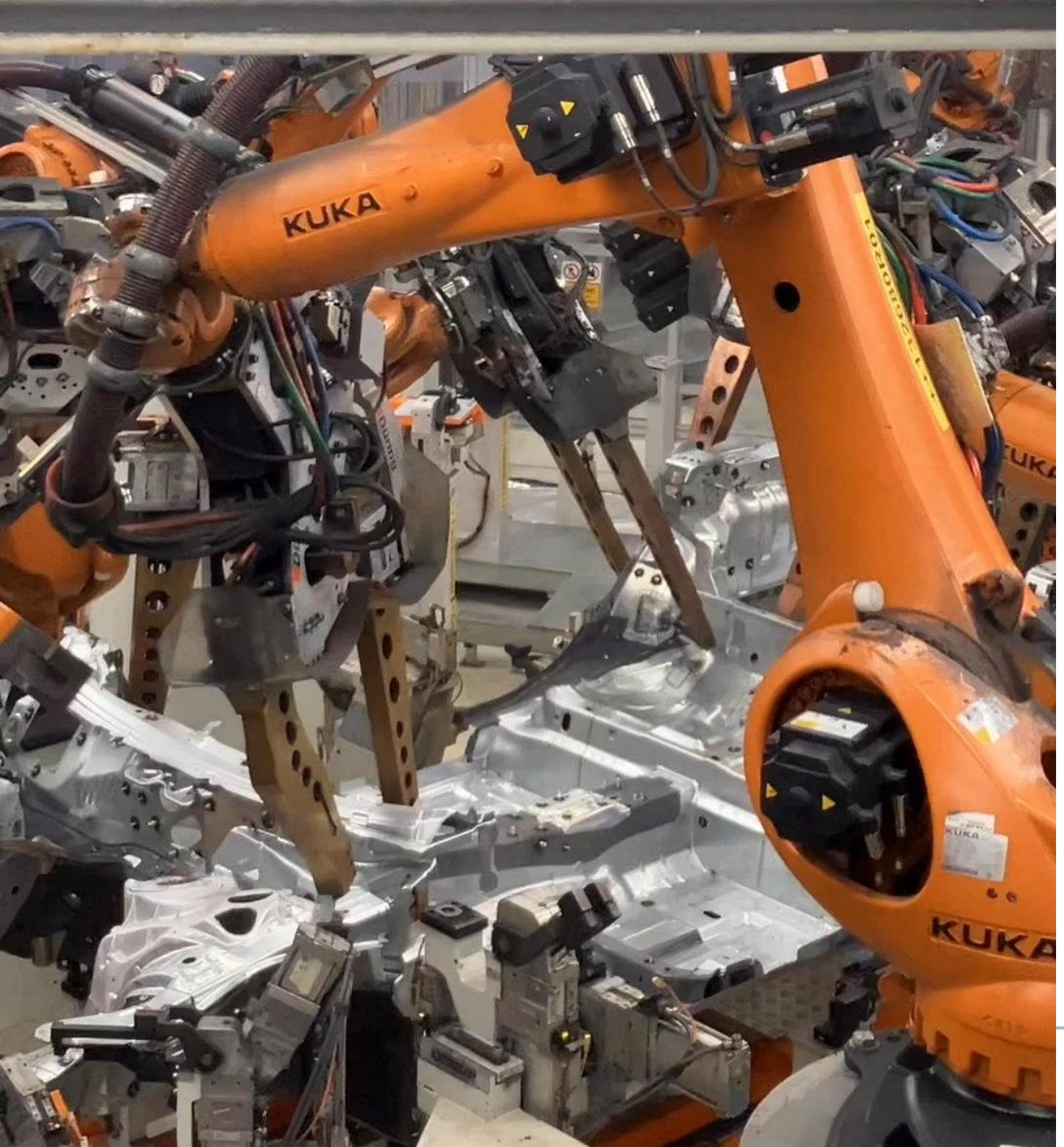
- 28 erteilte Patente im Jahr 2023
  (10 in Deutschland, 18 international)
- 835 Veröffentlichungen im Jahr 2023

- Kennzahlen Gesamtjahr 2023 in Mio. Euro [1)]
  - Haushalt gesamt: 94
  - Betriebshaushalt: 89 [2)]

- **Geschäftsbereiche:** Automobilbau, Maschinen- und Anlagenbau, Prozessindustrie, Energie, Elektronik und Microsystemtechnik, Gesundheitsindustrie

1) Werte inkl. Fraunhofer Austria Research GmbH, Wien, Geschäftsbereich Produktions- u. Logistikmanagement
2) Angepasster Betriebshaushalt: erhöht um kostenentlastende interne Leistungsverrechnungen mit IPA-Wertschöpfung i. H. v. rd. € 3 Mio.

Fraunhofer IPA



Trotz Funkenflug: kein Silvesterfeuerwerk in der Fertigung!

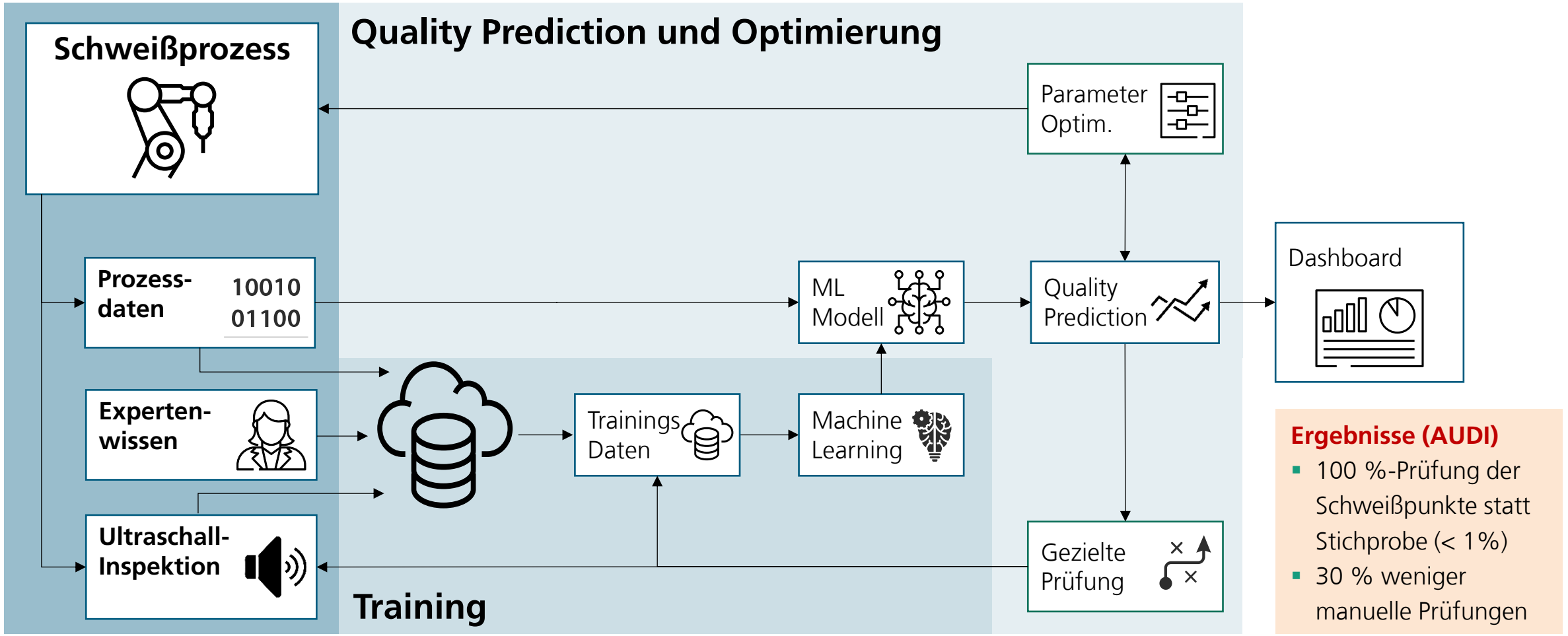
Das sind Schweißfehler

X

Fraunhofer IPA

Referenzbeispiel: Qualitätsprognose

Widerstandspunktschweißen



Ergebnisse (AUDI)

- 100 %-Prüfung der Schweißpunkte statt Stichprobe (< 1%)
- 30 % weniger manuelle Prüfungen

Referenzbeispiel: Maschinenparameteroptimierung

AI-basierte Qualitätskontrolle und Parameteroptimierung beim Laserschneiden

TRUMPF

Problem

- Bewertung der Schnittkantenqualität erfolgt manuell
- Hohe Streuung bei der Bewertung durch menschliche Experten
- Maßgeschneiderte Maschinen, wechselndes Material

Lösung

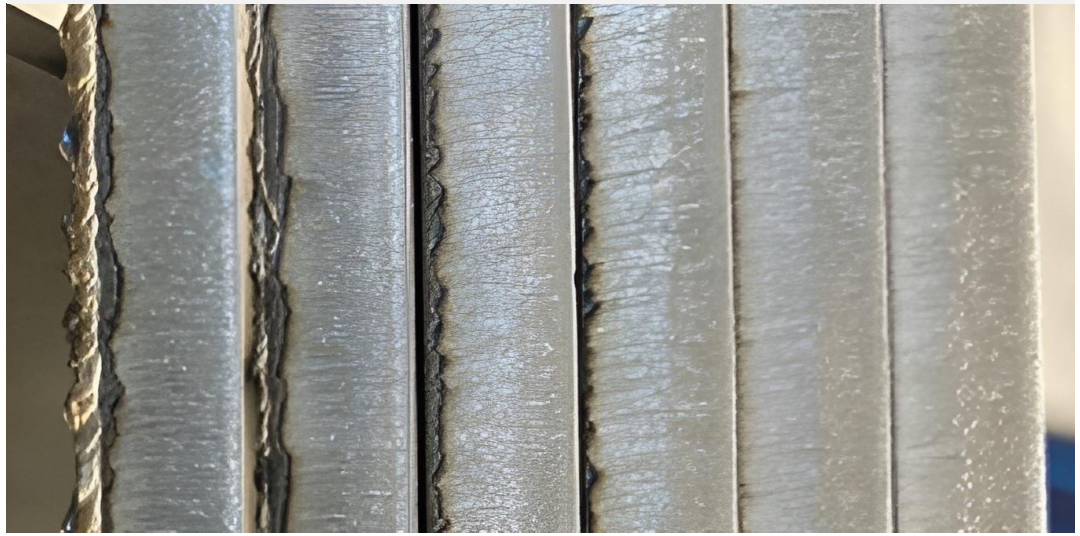
- Neuronale Netze zur *Vorhersage verschiedener Qualitätsgrößen*, z. B. Rauheit, Rillennachlauf, Neigungstoleranz, Grathöhe
- KI-basierte Berechnung *optimaler Prozessparameter*

Produkt: Cutting Assistant

- *Automatische und objektive Bewertung* der Schnittqualität
- *Verbesserte Schnittqualität* und zugleich *höherer Vorschub*
  → Stets so gut oder besser als menschliche Prozessexperten
- *Automatische Adaptierung* an wechselndes Material

	Prozessexpert	KI
Anzahl Parameter	5 – 7	Mehr al 100
Testschnitte	15 – 20	3 – 5
Prozessgeschwindigk.	Schnellerer Prozess als beim Mensch	
Grathöhe	Bis zu 50% reduziert	

Verbesserung des Grats von hoch (links) zu kaum sichtbar (rechts) →



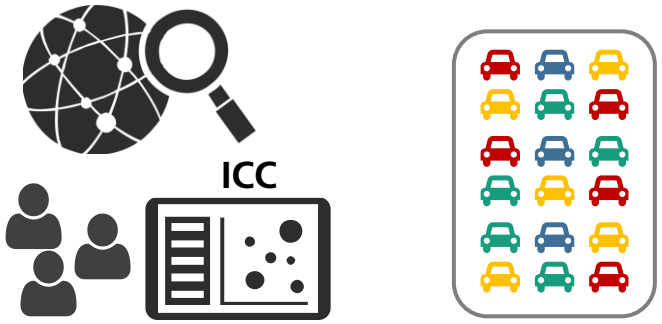
Philipp Wagner, Tobias Nagel, Philipp Leube, Marco F. Huber: *Sample-Efficient Bayesian Transfer Learning for Online Machine Parameter Optimization*. IEEE Conference on Artificial Intelligence (CAI), 2025.

Fraunhofer IPA

Referenzbeispiel: Auftragsplanung

Porsche Intelligent Planning and Ordering

Konfigurationsgenerator

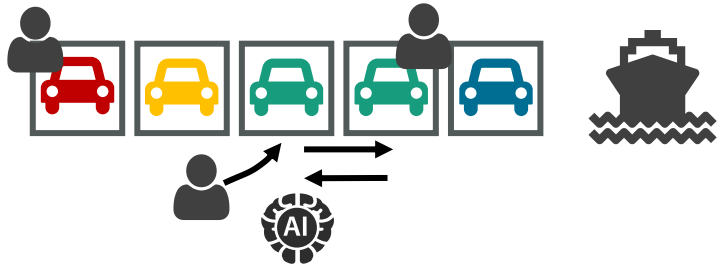


- Antizipation des Kundenwunsches mit Methoden der KI
- Erstellung marktspezifischer und baubarer Planaufträge

Ersteinplanung Auftragsbuch

- Einplanung der Planaufträge und Bildung des Produktionsprogramms
- Automatisierte und datenbasierte Ableitung aller Options- und Materialbedarfe

Matching & Rekalibrierung



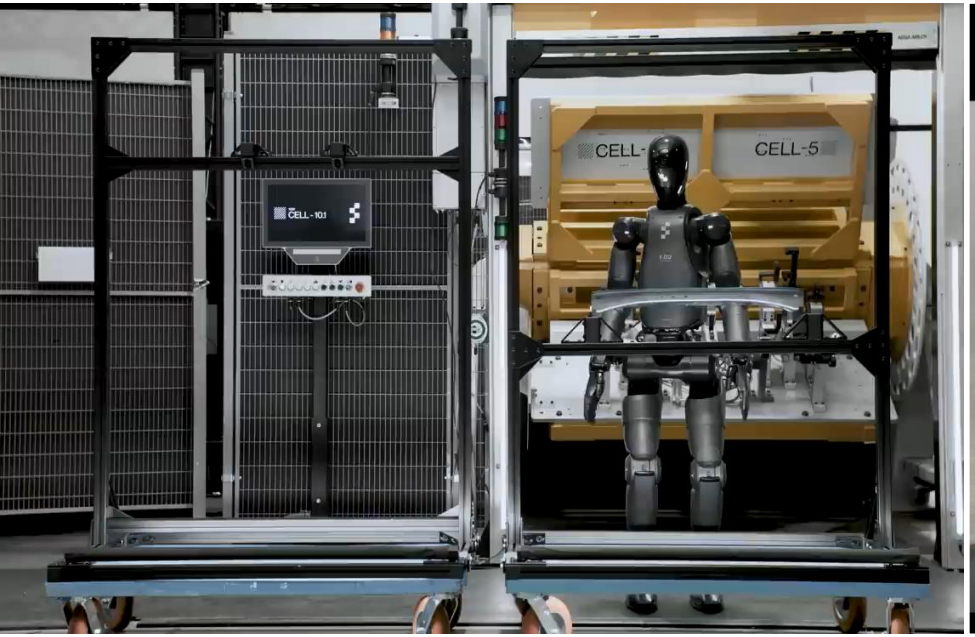
- Zuordnung realer Kundenaufträge zu optimalen Planaufträgen
- Automatisierte Steuerung von Optionstauschen

ICC: Internet Car Configurator

PORSCHE Fraunhofer IPA

Motivation

Steuerungsaufgaben in der Robotik und Fertigung




Source: Project RoboGrind

Source: Fraunhofer IPA, https://www.youtube.com/watch?v=wtR413ipMtQ

Maschinenbeladung | **Bearbeiten** | **De-/Montage**

Herausforderungen: teilweise bekannte Umgebungen, verrauschte Beobachtungen, lange Trainingszeiten, Nutzung von Expertenwissen, …

Fraunhofer IPA

Übersicht

Bayesian Machine Learning

- Online-Lernen Bayes‘scher neuronaler Netze
- Modellprädiktive Regelung mit Bayesschen neuronalen Netzen

Physics-Informed Machine Learning

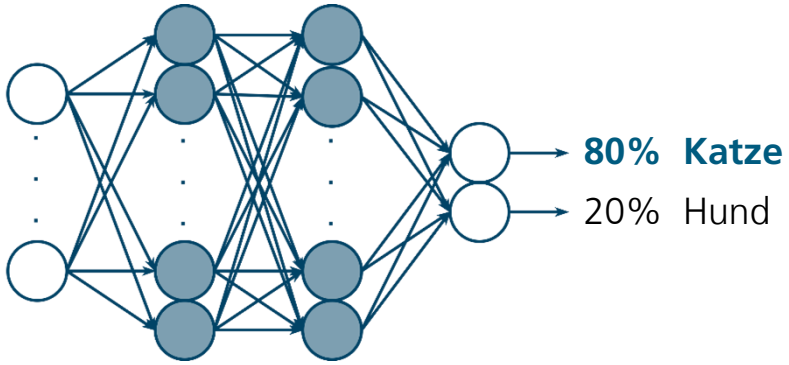
- Systemidentifikation mit physikalisch informierten neuronalen Netzen
- Modellprädiktive Regelung mit gelernten Differentialgleichungen

Bayes‘sche Zustandsschätzung | **Informiertes maschinelles Lernen** | **Planen und Regeln**

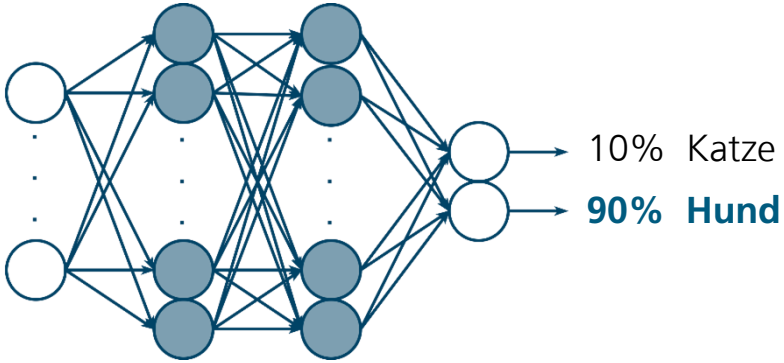
Das Problem

Neuronale Netze sind übermäßig selbstsicher

In-distribution

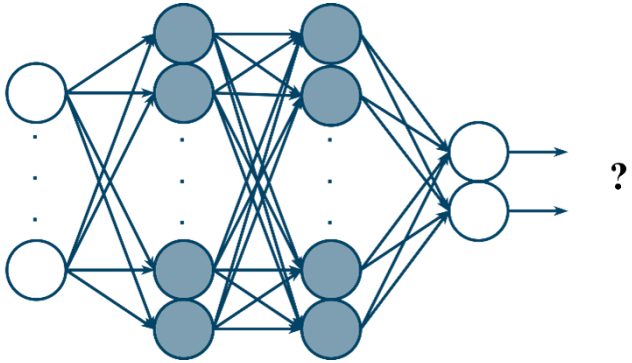


80% Katze
20% Hund



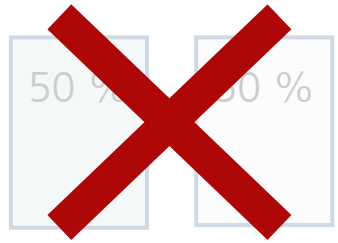
10% Katze
90% Hund

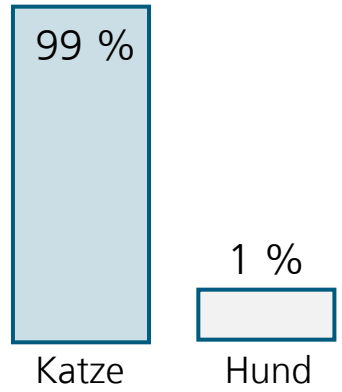
Out-of-Distribution



?

Oft zu
selbstsicher[1]

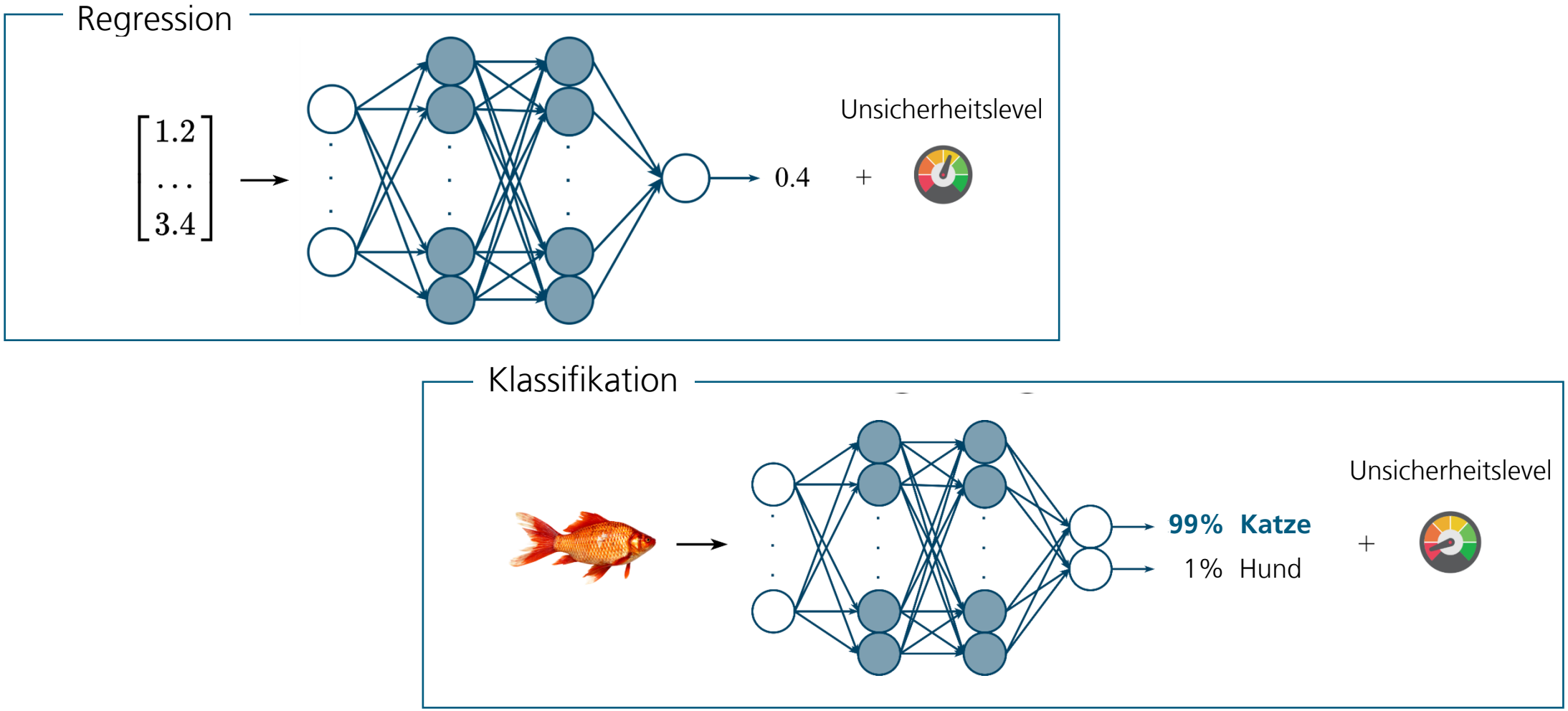




[1]Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf: *Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem*. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019

Gewünscht

Quantifizierung der Unsicherheit



Fraunhofer IPA

Die Lösung

Bayes'sche Inferenz

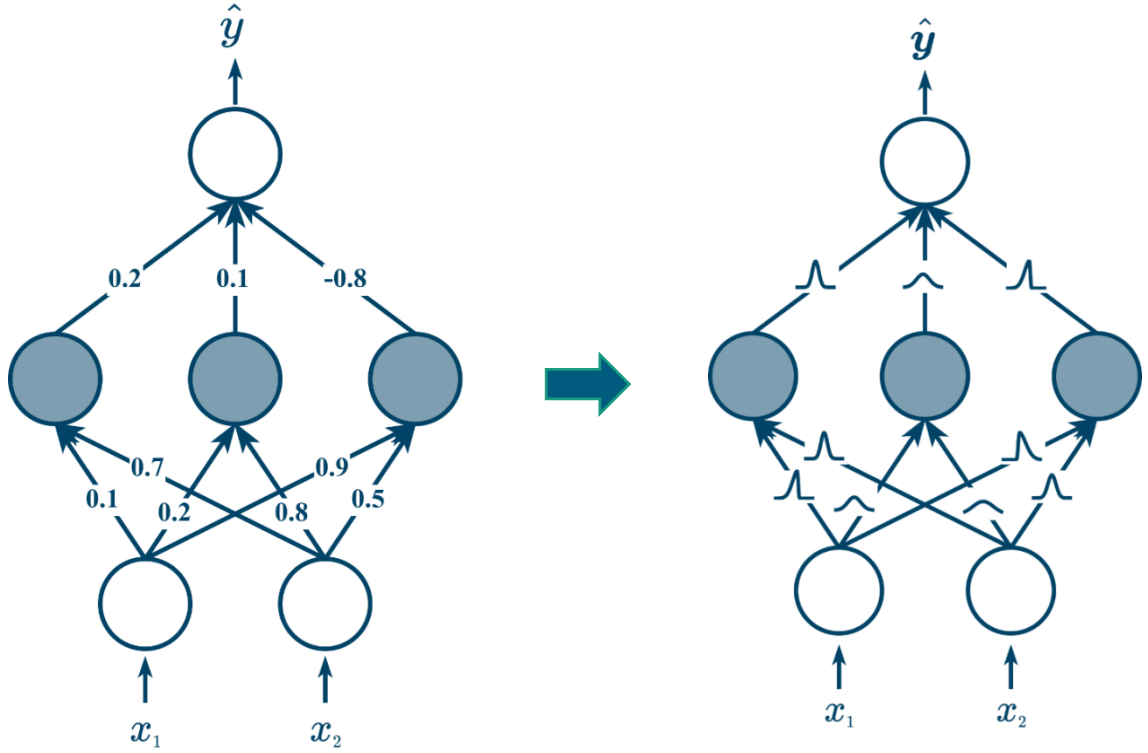
☛ Bayes'sche Regel

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

Die Lösung

Bayes'sche Neuronale Netze (BNN)

Standard NN



Bayesian NN

☛ Bayes'sche Regel

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

💡 Idee für Neuronales Netz

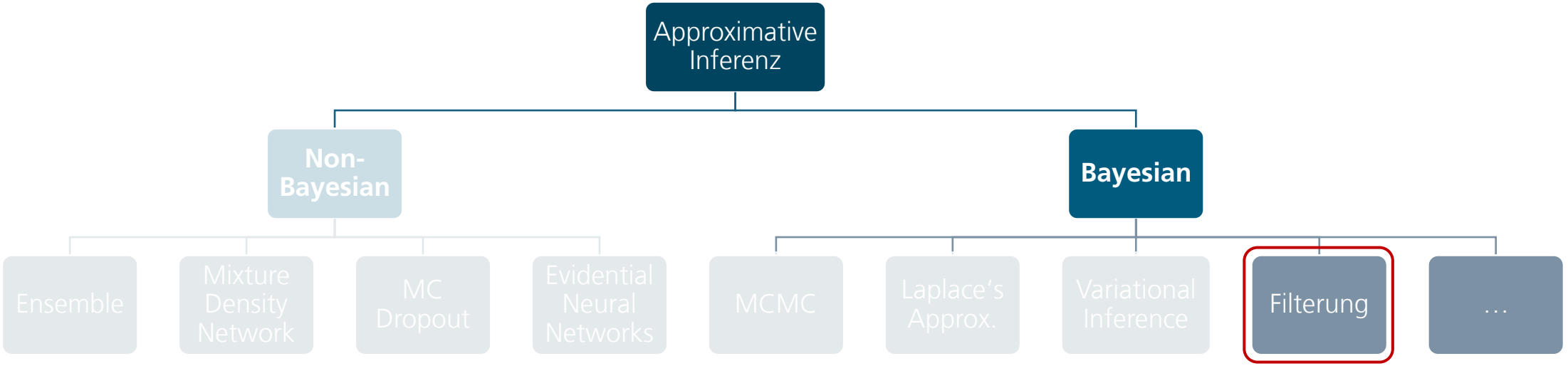
Betrachtung aller **Gewichte** als **Zufallsvariablen**

! Vorteile

Bessere Prädiktionen und **kalibrierte Unsicherheiten**

Schätzung der Netzwerkgewichte

Stand der Technik



Die Aktualisierung der Gewichte ist sehr anspruchsvoll

- Berechnung der Posterior-Wahrscheinlichkeit ist unlösbar → Approximationen sind unvermeidlich
- Gradientenabstieg erfordert Iterationen über Datensatz → Schlechte Skalierung
- Stapelverarbeitung von Datensätzen → neue Daten erfordern erneutes Training

Kalman Bayesian Neural Network (KBNN)

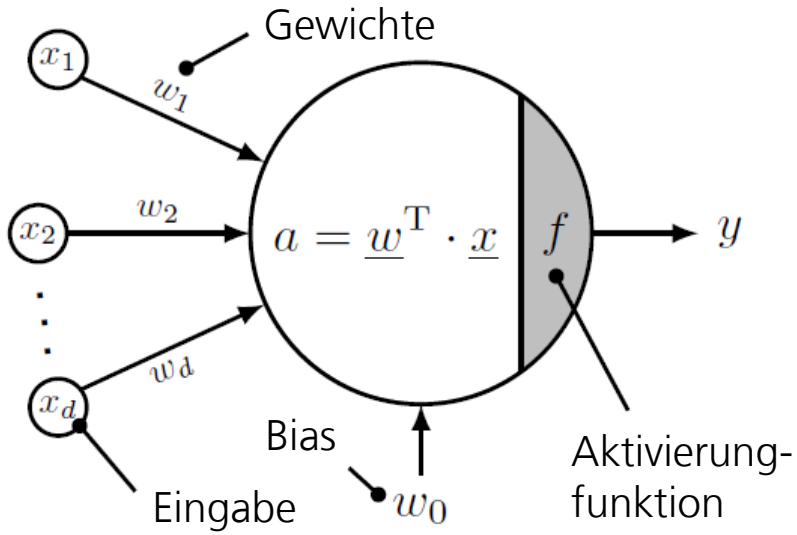
Einzelnes Neuron: Bayes'sches Perzeptron

Einzelnes Neuron (= Perzepton)

$$a = \boldsymbol{w}^T \cdot \boldsymbol{x} \qquad \text{(Lineare Funktion)}$$
$$= \sum_i w_i \cdot x_i + w_0$$

$$y = f(a) \qquad \text{(Nichtlineare Aktivierung)}$$

Annahme: Gewichte $\boldsymbol{w}$ sind normalverteilt



💡 Nutzung der Theorie der (exakten) Gauß'schen Zustandsschätzung

Unter der zusätzlichen Annahme, dass a and y gemeinsam normalverteilt sind, ist es hinreichend die folgenden drei Momente zu berechnen, um y vorherzusagen und $a, \boldsymbol{w}$ zu aktualisieren:

$$\mu_y = \mathrm{E}[f(a)] \qquad \sigma_y^2 = \mathrm{E}[f(a)^2] - \mu_y^2 \qquad \sigma_{ay}^2 = \mathrm{Cov}[a, f(a)]$$

→ Exakte Berechnung für stückweise lineare Aktivierungen (z. B. ReLU) und neuartige, enge Approximation für Sigmoid/Tanh.

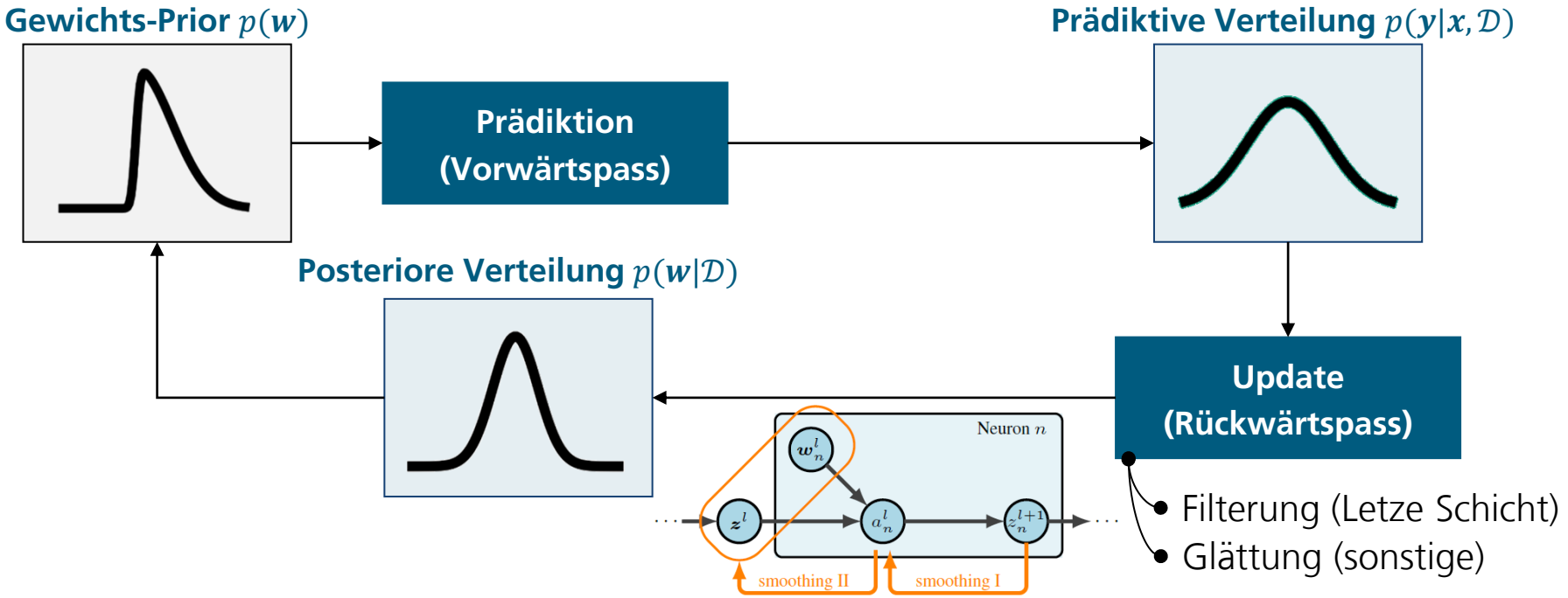
Marco F. **Huber**: *Bayesian Perceptron: Towards fully Bayesian Neural Networks*. Proceedings of the 59th IEEE Conference on Decision and Control (CDC), 2020.

Kalman Bayesian Neural Network (KBNN)

Volles Netz

💡 Idee

Rekursive Schätzung der posterioren Gewichtsverteilung unter Verwendung von Bayes'schen Filter- und Glättungsgleichungen, ähnlich des **Kalman-Filters**.



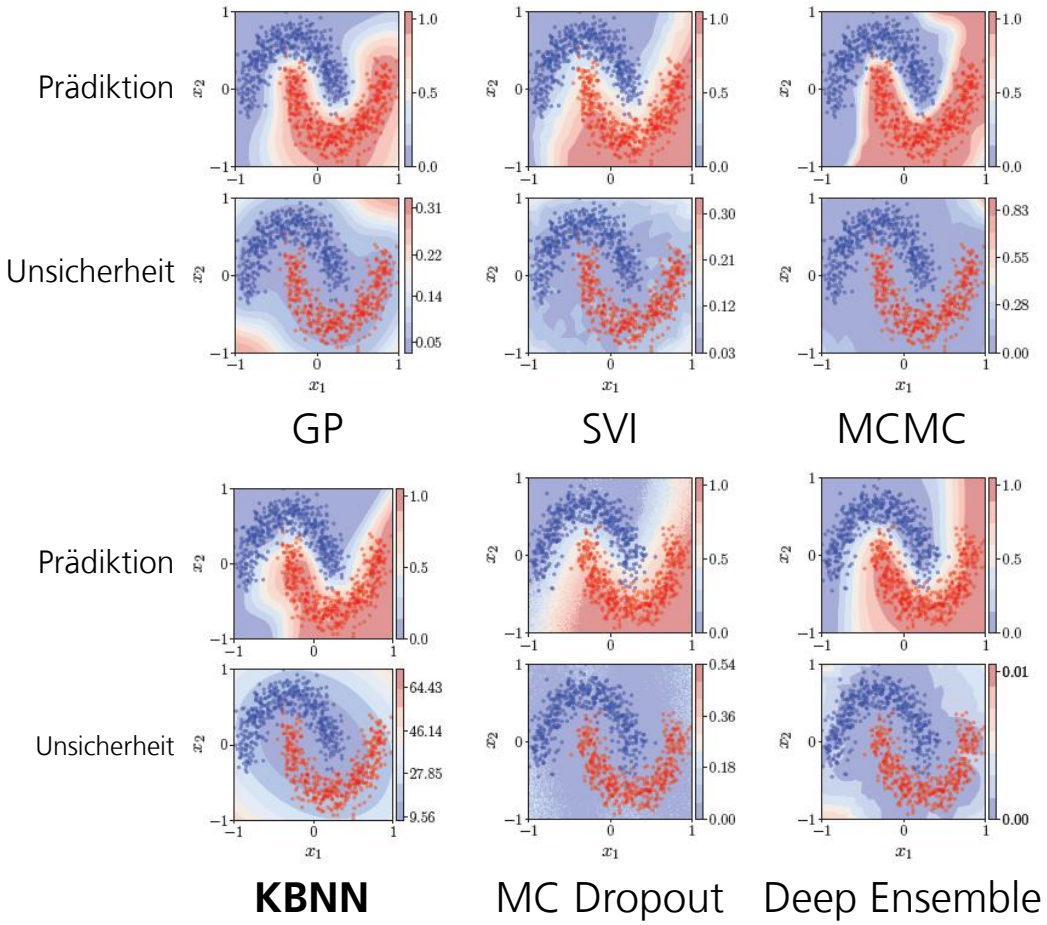
! Wichtig

- Ein Bayes'sches neuronales Netz wird als **Markov-Netz** betrachtet.
  → Schichtenweise Verarbeitung
- **Geschlossene Berechnung** von Mittelwert und Kovarianz für gängige Operatoren wie ReLU, Faltungen, Max-Pooling
  → Moment Matching
- **Nur eine Epoche** und **gradientenfrei!**

Philipp Wagner, Xinyang Wu, and Marco **Huber**: *Kalman Bayesian Neural Networks for Closed-Form Online Learning*. 37th AAAI Conference on Artificial Intelligence (AAAI), 2023.

Ergebnisse

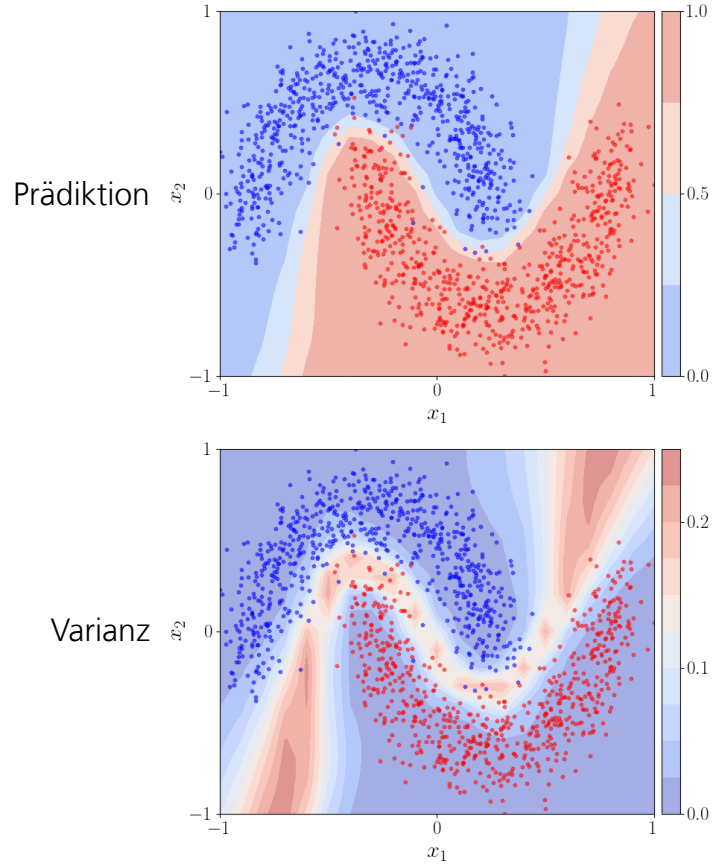
Toy Problem: Mond Datensatz



Diskussion

- Zunehmende Unsicherheit bei Out-of-Distribution Daten
- Online-Lernfähigkeit

Rotierender Mond (nur KBNN)



Fraunhofer IPA

Benchmark

UCI Datensatz

Root mean square error (RMSE) ↓

Dataset	N	d	SVI	MCMC	PBP	KBNN 1	KBNN 10
Boston	506	13	3.434 ± 0.131	$\mathbf{2.553 \pm 0.027}$	2.740 ± 0.095	3.893 ± 0.200	2.695 ± 0.155
Concrete	1,030	8	7.597 ± 0.283	6.227 ± 0.108	5.874 ± 0.054	8.396 ± 0.497	$\mathbf{5.703 \pm 0.183}$
Energy	768	8	4.025 ± 0.074	$\mathbf{0.906 \pm 0.049}$	3.274 ± 0.049	4.155 ± 0.087	2.404 ± 0.259
Wine	4,898	11	0.726 ± 0.007	$\mathbf{0.656 \pm 0.004}$	0.667 ± 0.002	0.719 ± 0.011	0.666 ± 0.006
Naval	11,934	16	0.025 ± 0.012	0.008 ± 0.001	$0.006 \pm 6.12 * 10^{-5}$	0.034 ± 0.005	$\mathbf{0.004 \pm 0.001}$
Yacht	308	6	1.157 ± 0.222	0.879 ± 0.294	$\mathbf{0.867 \pm 0.047}$	3.752 ± 0.240	1.584 ± 0.178

Trainingszeit / s ↓

Dataset	SVI	MCMC	PBP	KBNN 1	KBNN 10
Boston	21.4	446.0	8.2	**0.8**	8.7
Concrete	22.5	481.7	12.7	**1.7**	17.5
Energy	21.6	405.9	10.2	**1.2**	13.2
Wine	23.8	520.3	49.1	**8.3**	86.7
Naval	42.8	367.0	116.1	**20.5**	205.3
Yacht	21.5	357.4	5.9	**0.5**	5.0

Diskussion

- Leistung von KBNN 1 ähnlich zu SVI
- KBNN 10 ähnlich zu PBP
- KBNN 1 ist schnellste Methode

Philipp Wagner, Xinyang Wu, and Marco **Huber**: *Kalman Bayesian Neural Networks for Closed-Form Online Learning*. 37th AAAI Conference on Artificial Intelligence (AAAI), 2023.

Fraunhofer IPA

In Arbeit

Offene Probleme und zukünftige Arbeiten

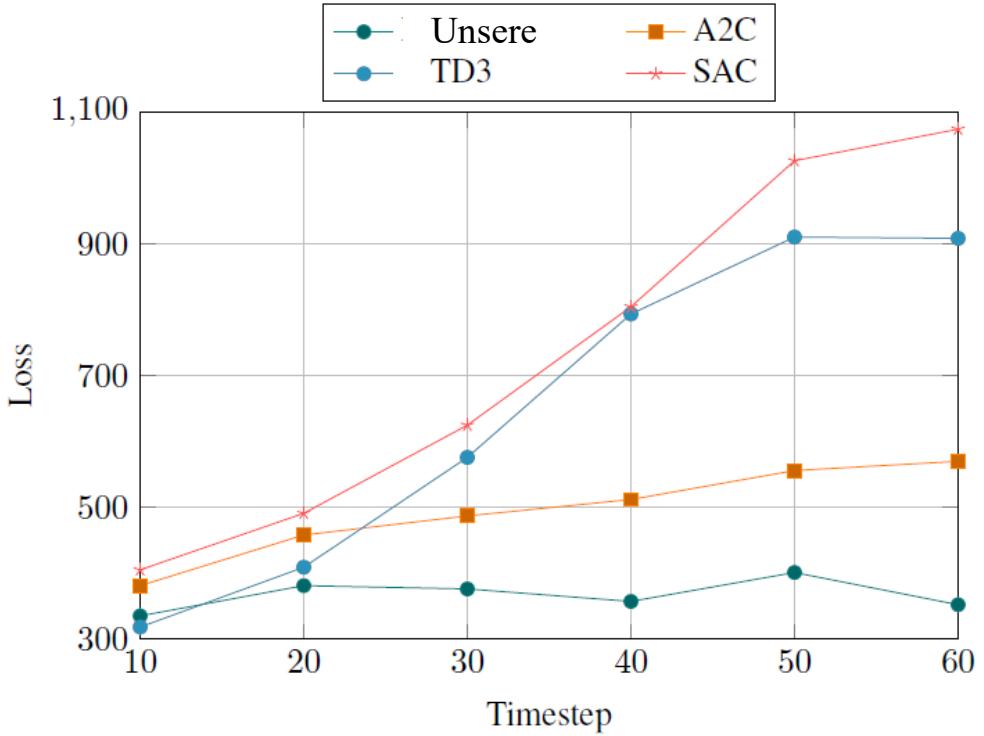
- Die **priore Verteilung** hat einen starken Einfluss auf die Leistung, und die Auswahl des Priors ist schwierig.

→ Idee: Einführung eines niedrigdimensionalen Hyper-Priors

- **Doppelte Gauß'sche Verteilungsannahme** ist nicht (vollständig) gerechtfertigt, und es bestehen weiterhin Probleme mit Stichproben außerhalb der Verteilung.

→ Idee 1: Alternative Formulierungen des Rückwärtspasses zur Vermeidung der gemeinsamen Gauß'schen Annahme.

→ Idee 2: Verwendung von Heavy-Tail-Verteilungen

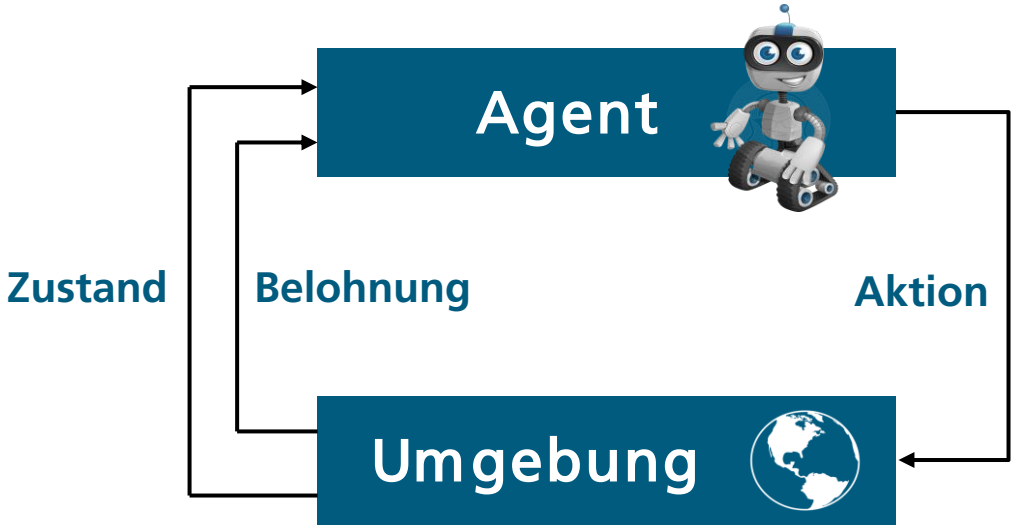


Vorläufig: Leistung beim Siemens Industrial Benchmark[1]

[1]Daniel Hein, Stefan Depeweg, Michel Tokic, Steffen Udluft, Alexander Hentschel, Thomas A Runkler, Volkmar Sterzing. A benchmark environment motivated by industrial control problems. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), 2017.

Reinforcement Learning

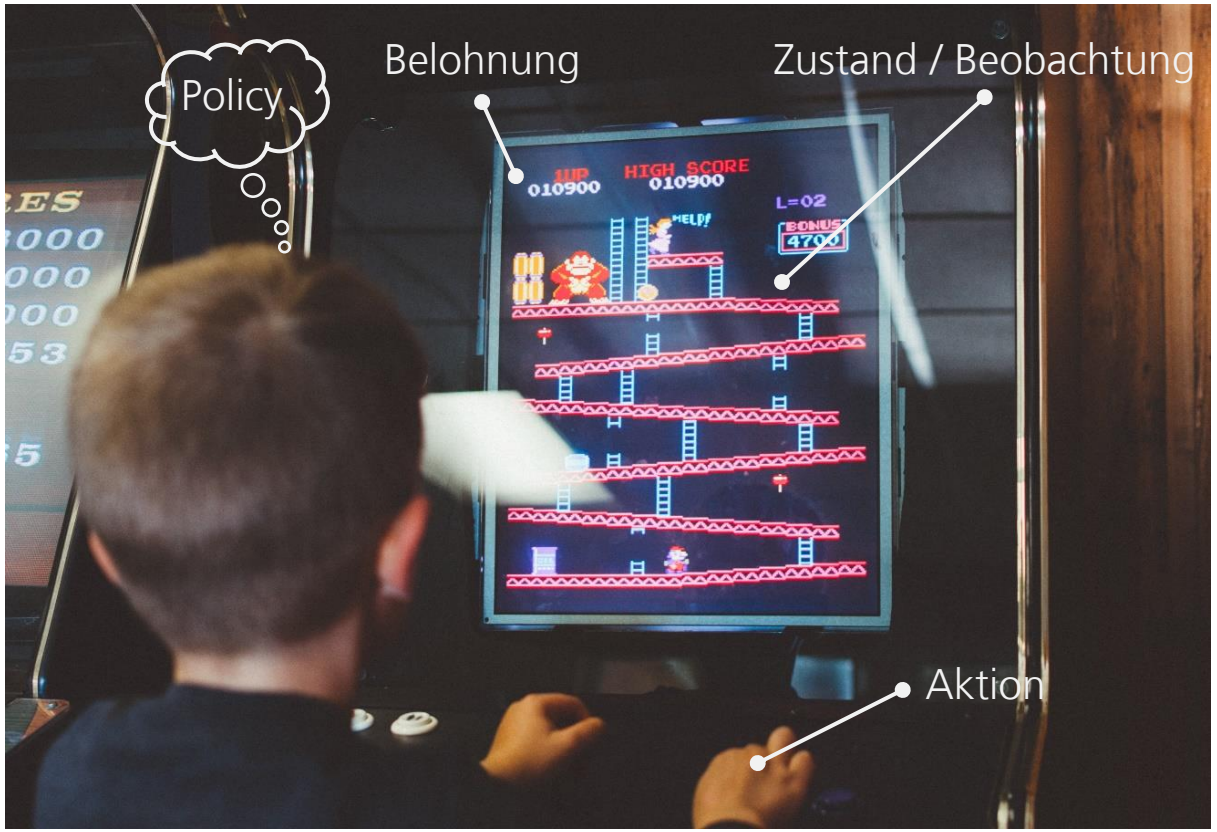
Eine kurze Einführung



- **Zustand:** Pixel des Spiels / Position des Roboters
- **Aktion:** Joystick-Bewegungen / Gelenkmomente
- **Belohnung:** Punktzahl / Abschluss der Aufgabe
- **Ziel:** Maximieren der numerischen Belohnung

→ Die meisten RL-Methoden sind modellfrei

→ **Sehr dateneffizient!**

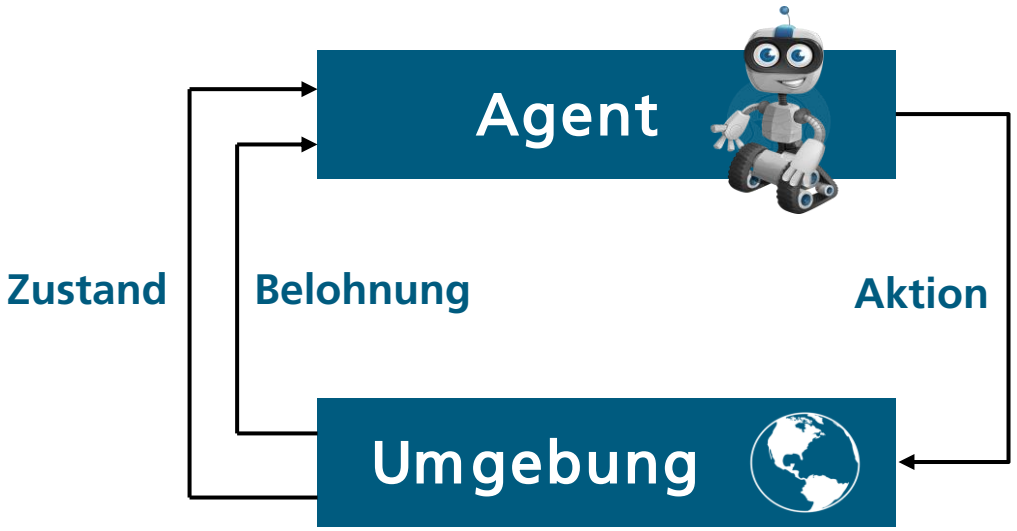


Bildquelle: https://medium.com/swlh/how-i-spent-my-summer-of-1982-59638293f358

Fraunhofer IPA

Reinforcement Learning

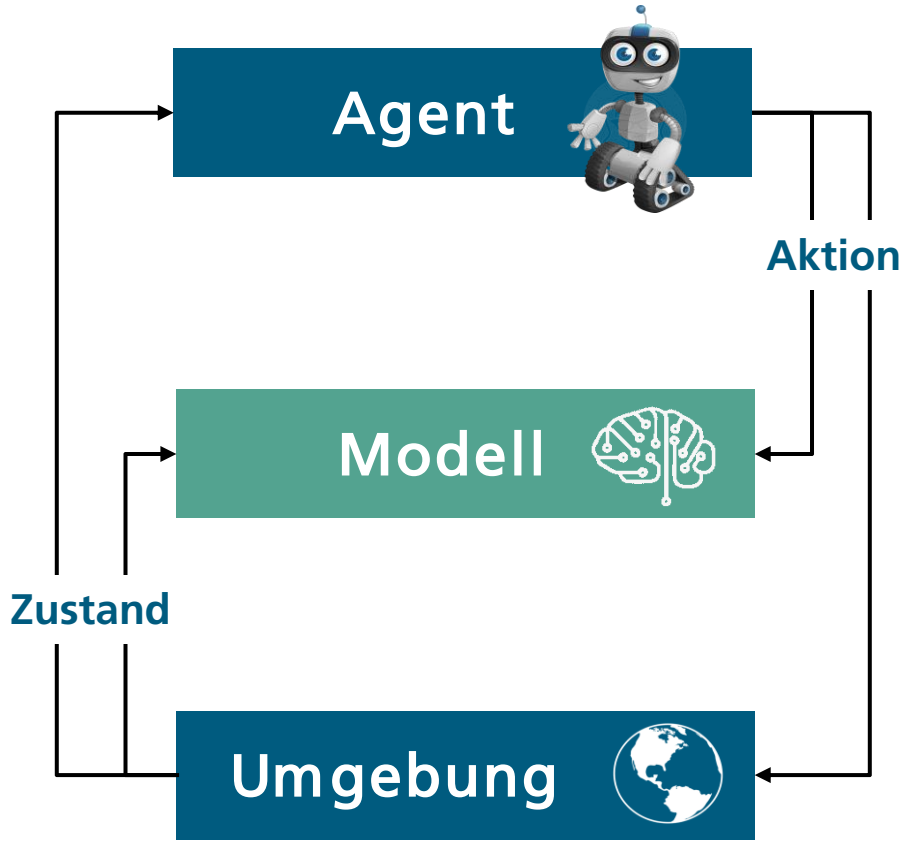
Eine kurze Einführung



- **Zustand:** Pixel des Spiels / Position des Roboters
- **Aktion:** Joystick-Bewegungen / Gelenkmomente
- **Belohnung:** Punktzahl / Abschluss der Aufgabe
- **Ziel:** Maximieren der numerischen Belohnung

→ Die meisten RL-Methoden sind modellfrei

→ **Sehr dateneffizient!**



Lernt Modell durch Interaktion mit der Umgebung

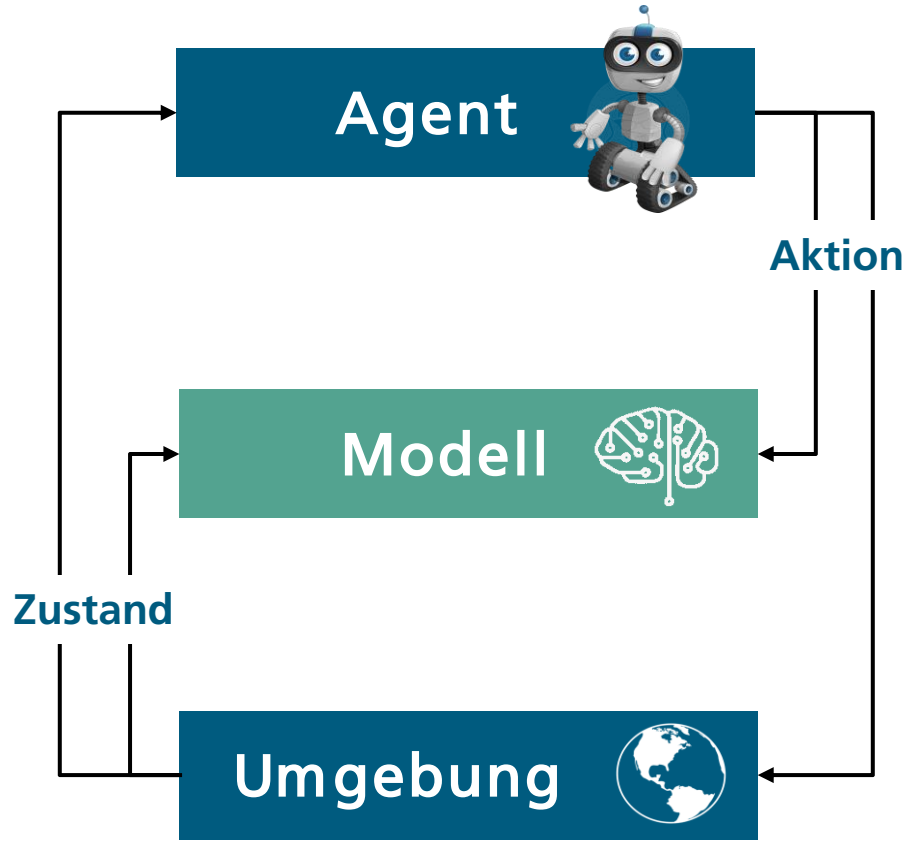
Fraunhofer IPA

Modellbasiertes Reinforcement Learning

Abwechselndes Lernen und Planen

Modell lernen

- Überwachtes ML-Problem, bei dem die Trainingsdaten online erzeugt werden
- BNN als Modell → Unsicherheit während des Lernens nutzen
- Verwendung KBNN aufgrund dessen Online-Lernfähigkeit



Lernt Modell durch Interaktion mit der Umgebung

Modellbasiertes Reinforcement Learning

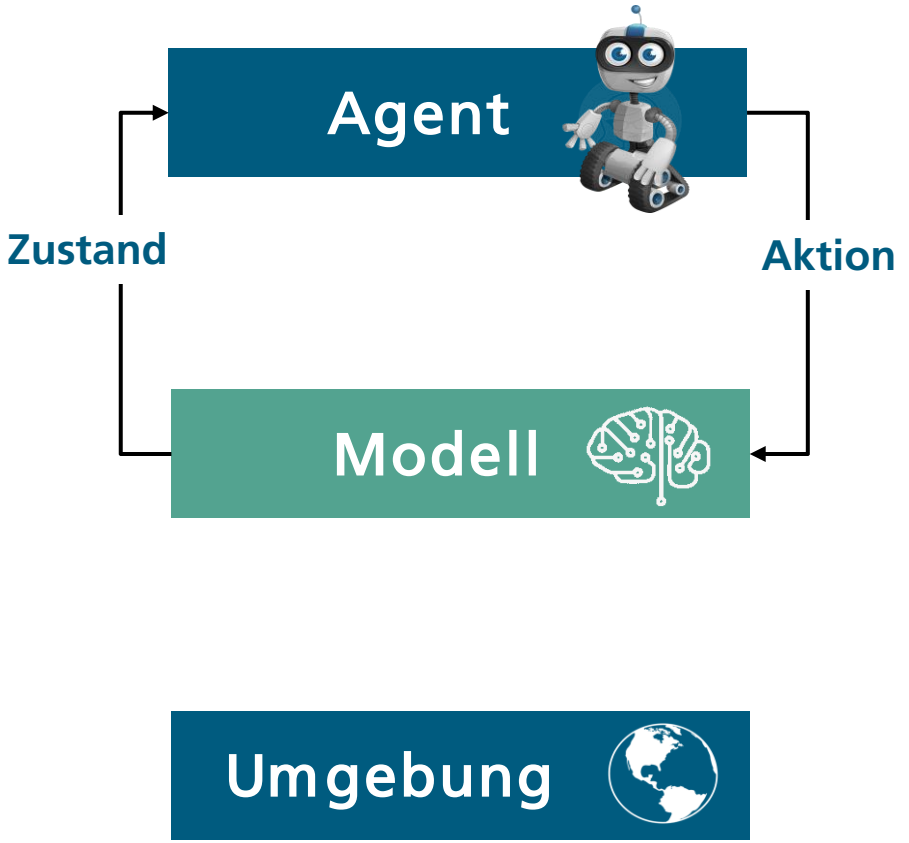
Abwechselndes Lernen und Planen

Modell lernen

- Überwachtes ML-Problem, bei dem die Trainingsdaten online erzeugt werden
- BNN als Modell → Unsicherheit während des Lernens nutzen
- Verwendung KBNN aufgrund dessen Online-Lernfähigkeit

Aktionen planen

- Optimierungsproblem
- Verwendung von modellprädiktiver Regelung (MPC)
  → Ermöglicht Online-Planung
  → Aktionsplanung über langen, rollierenden Zeithorizont
- Vorhersage der Unsicherheit über Zeithorizont mittels Vorwärtspass des KBNN (oder anderer Vorhersagetechnik)



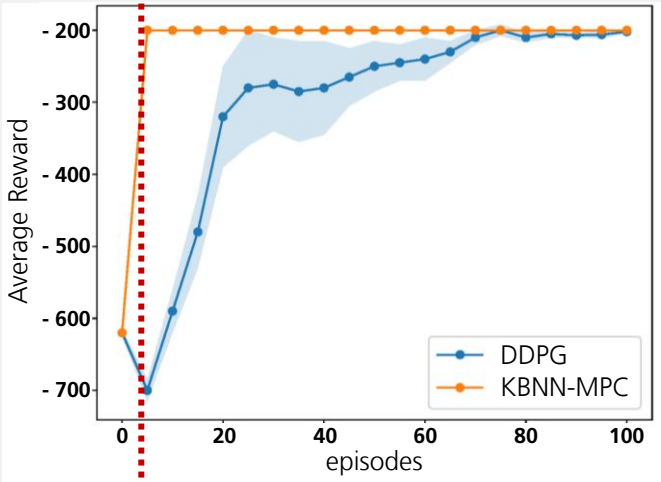
Plant Aktionen per Simulation

Ergebnisse

Beispiel: Inverses Pendel

Diskussion

- Vergleich von DDPG und KBNN-MPC



- **DDPG:** erfordert 20+ Episoden mit je 200 Zeitschritten → 4,000 Zeitschritte
- **KBNN-MPC:** weniger als 1 Episode

Video ist beschleunigt (4x)

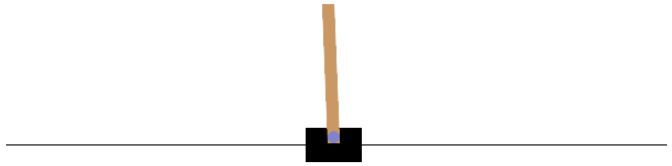
Xinyang Wu, Elisabeth Wedernikow, Marco F. **Huber**: *Data-Efficient Uncertainty-Guided Model-Based Reinforcement Learning with Unscented Kalman Bayesian Neural Networks*. American Control Conference (ACC), 2024.

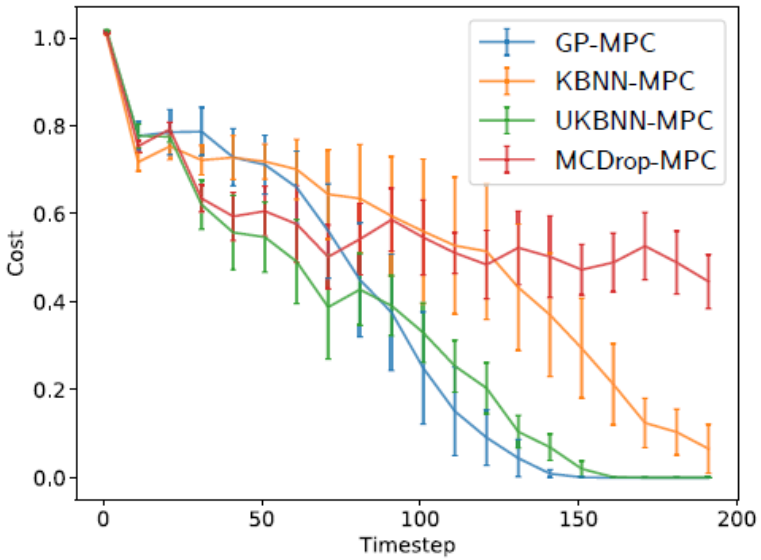
Ergebnisse

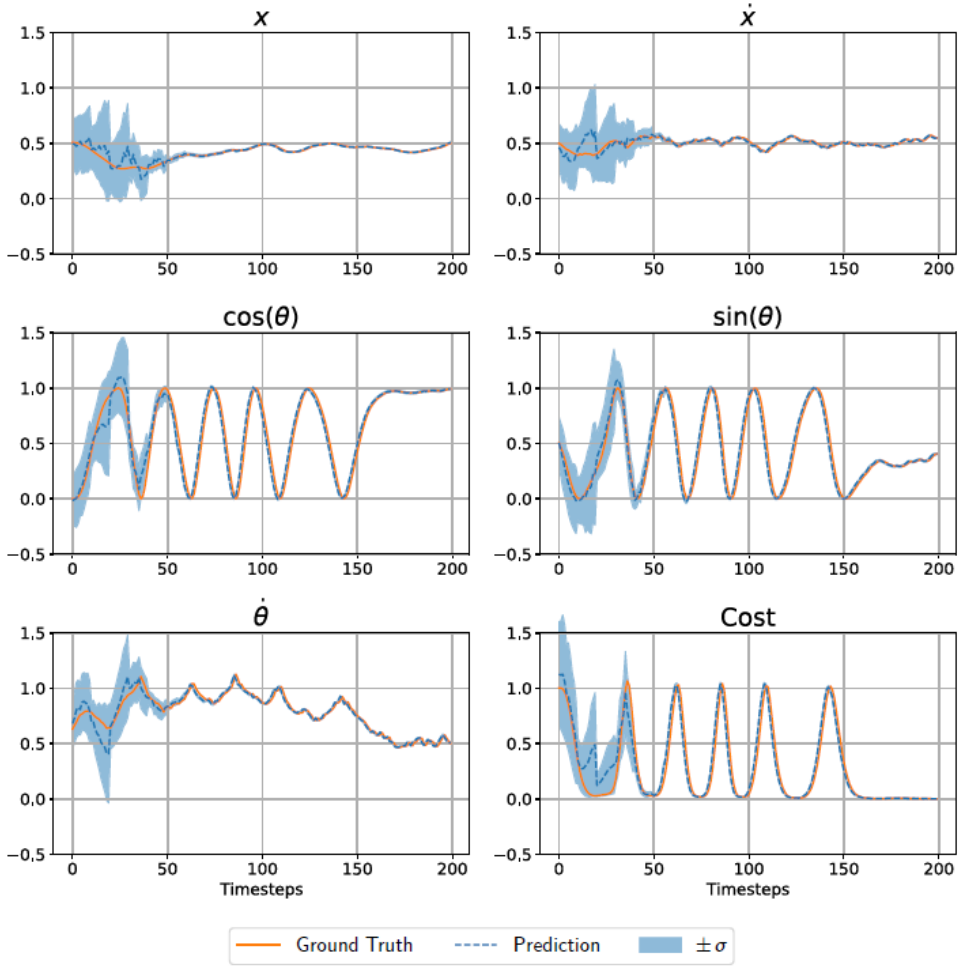
Beispiel: Cart Pole

Diskussion

- Aufschwingen und Stabilisieren eines inversen Pendels mittels eines Schlittens
- System ist nichtlinear und unteraktuiert
- Klassischer Benchmark der Regelungstechnik
- KBNN lernt die Systemdynamik zur Laufzeit in einem Durchgang







Xinyang Wu, Elisabeth Wedernikow, Marco F. **Huber**: *Data-Efficient Uncertainty-Guided Model-Based Reinforcement Learning with Unscented Kalman Bayesian Neural Networks*. American Control Conference (ACC), 2023.

Fraunhofer IPA

Übersicht

Bayesian Machine Learning

- Online-Lernen Bayes‘scher neuronaler Netze
- Modellprädiktive Regelung mit Bayesschen neuronalen Netzen

Physics-Informed Machine Learning

$$\dot{x} = f(x,u)$$
$$y = g(x,v)$$

- Systemidentifikation mit physikalisch informierten neuronalen Netzen
- Modellprädiktive Regelung mit gelernten Differentialgleichungen

Bayes‘sche Zustandsschätzung | **Informiertes maschinelles Lernen** | **Planen und Regeln**

Motivation

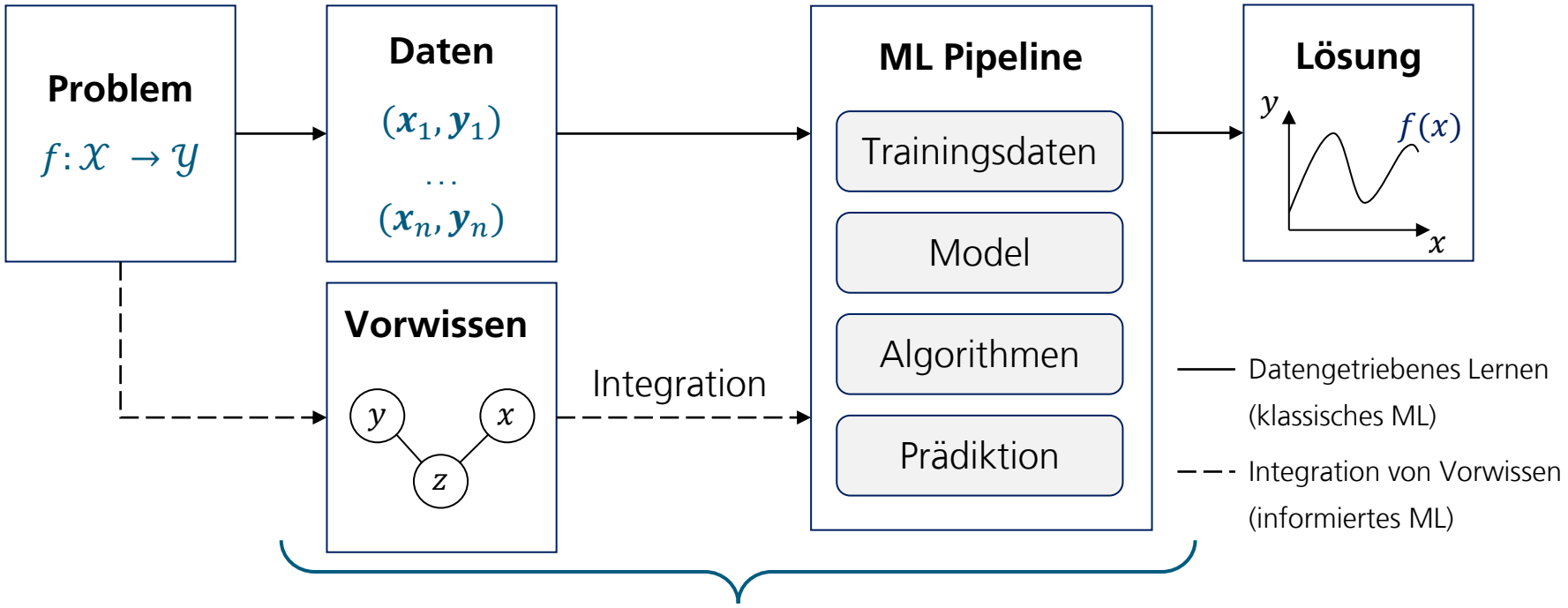
Informiertes Maschinelle Lernen

Ausgangssituation

- **Bisher:** Datengetriebene Modellierung des dynamischen Systems
- **Allerings:** Domänenwissen oft in Form mathematischer Modelle

Informiertes ML

Informationsfluss



Informierte ML Pipeline erfordert hybride Informationsquellen: Daten + Vorwissen

Vorwissen:

- Stammt von einer unabhängigen Quelle
- Ist mittes formaler Repräsentation gegeben
- Wird explizit in ML Pipeline integriert

Vorteile*


Leistungsfähigkeit


Dateneffizienz


Extrapolation


Erklärbarkeit

* Die Liste ist weder erschöpfend noch gilt jeder Vorteil für jeden informierten ML-Ansatz.

Vorwissen

Repräsentationsformen

Algebraische Gleichungen

$$E = m \cdot c^2$$

$$v \leq c$$

Differential-gleichungen

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

$$\mathrm{F(x)} = \mathrm{m} \frac{\partial^2 x}{\partial t^2}$$

Simulations-ergbnisse

Räumliche Invarianzen

Logikregeln

$$A \wedge B \Rightarrow C$$

Wissensgraphen

Probabilistische Zusammenhänge

Menschliches Feedback

Algebraische Gleichungen — **Vorwissen** — Menschliches Feedback

Grad der Formalisierung

Je formaler das Wissen repräsentiert ist, umso einfacher kann es in ML integriert werden.

Informiertes Maschinelles Lernen

Integration von Differentialgleichungen

Ausgangssituation

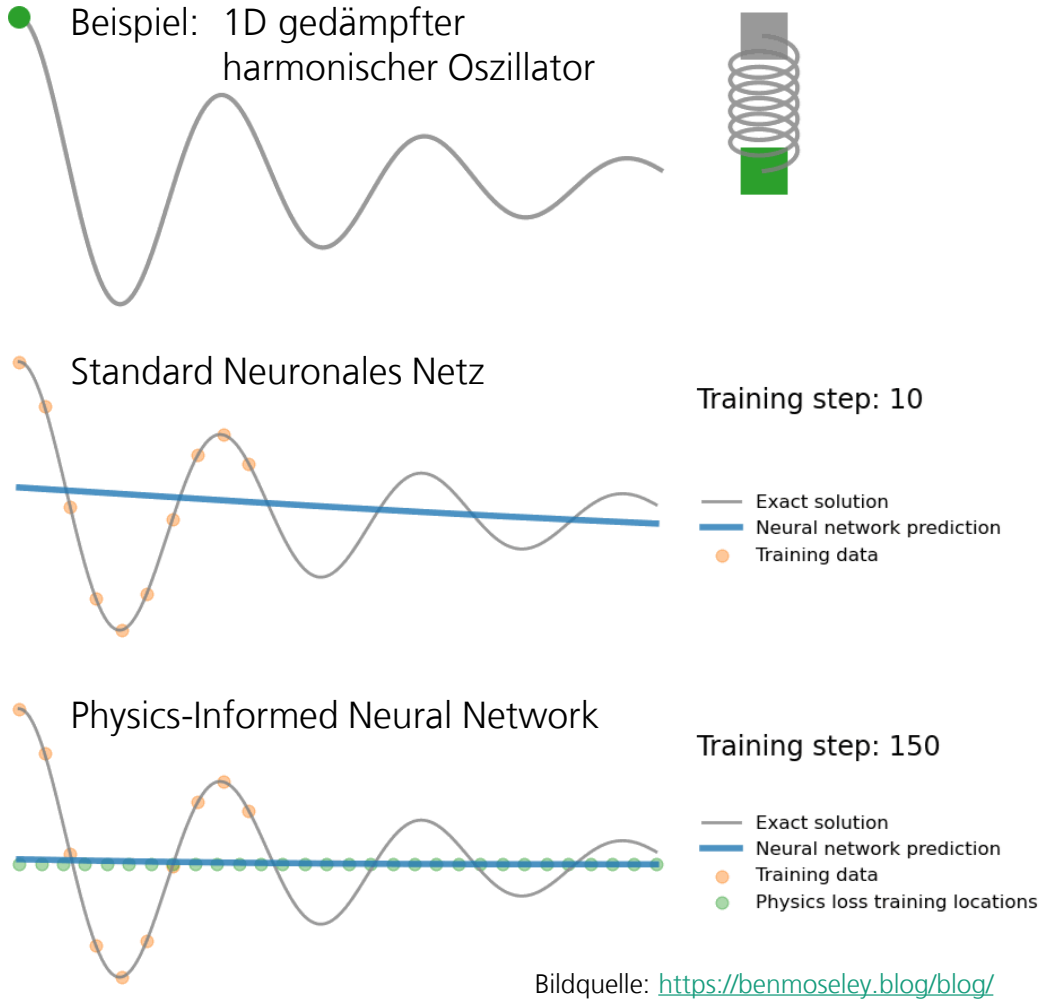
- **Bisher:** Datengetriebene Modellierung des dynamischen Systems
- **Allerings:** Domänenwissen oft in Form mathematischer Modelle (**Differentialgleichungen, DGL**) vorhanden
- Wie kann man Vorwissen nutzen? → **Physik-informiertes Maschinelles Lernen**

Physik-Informierte Neuronale Netze (PINN)

Physics-Informed Neural Network (PINN)

Einführung

- PINNs[2] werden trainiert, um **praktikable Lösung** $\tilde{\boldsymbol{x}}(t)$ für eine gegebene DGL zu approximieren.
- PINNs nutzen automatische Differentiation und die Fähigkeit neuronaler Netze, **jede beliebige Funktion zu approximieren** (Universelles Approximationstheorem).
- Kann verwendet werden, um Parameter einer DGL zu identifizieren, indem parallel zum Training des neuronalen Netzwerks nach einem passenden Parametersatz $\boldsymbol{\theta}^*$ gesucht wird.



Bildquelle: https://benmoseley.blog/blog/

[2]Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, Journal of Computational Physics 2019.

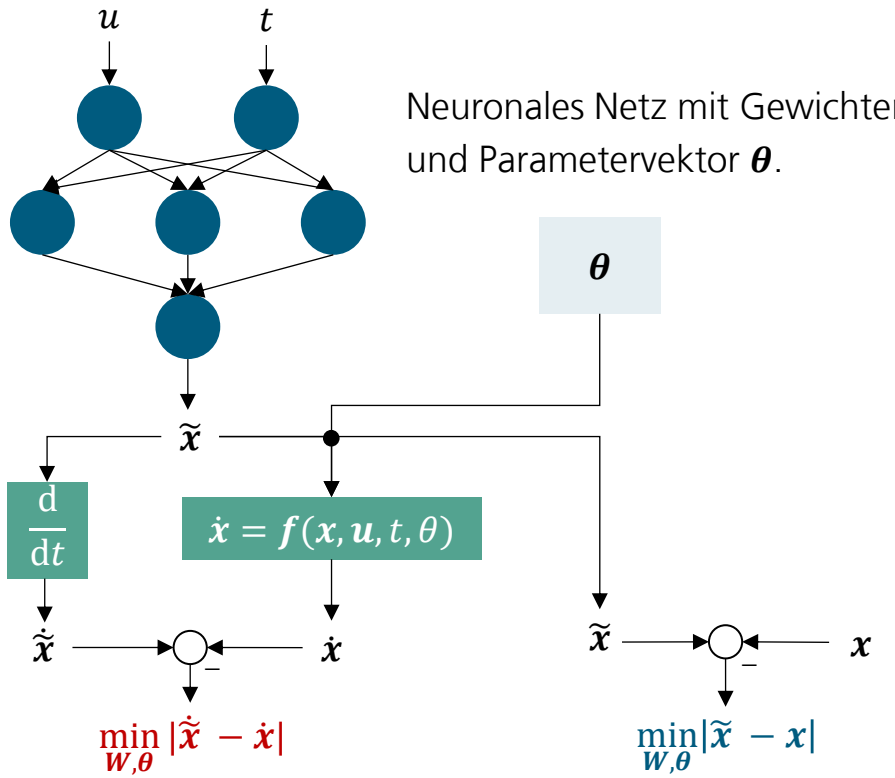
Physik-Informierte Neuronale Netze (PINN)

Physics-Informed Neural Network (PINN)

Einführung

- PINNs[2] werden trainiert, um **praktikable Lösung** $\tilde{\boldsymbol{x}}(t)$ für eine gegebene DGL zu approximieren.
- PINNs nutzen automatische Differentiation und die Fähigkeit neuronaler Netze, **jede beliebige Funktion zu approximieren** (Universelles Approximationstheorem).
- Kann verwendet werden, um Parameter einer DGL zu identifizieren, indem parallel zum Training des neuronalen Netzwerks nach einem passenden Parametersatz $\boldsymbol{\theta}^*$ gesucht wird.

Aber: Was ist mit nicht beobachtbaren Zuständen und Unsicherheiten (Systemrauschen, verrauschte Daten)?



[2]Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, Journal of Computational Physics 2019.

Kalman-Bucy-Informed Neural Network (KBINN)

Lernen aus verrauschten Daten

💡 Kernidee

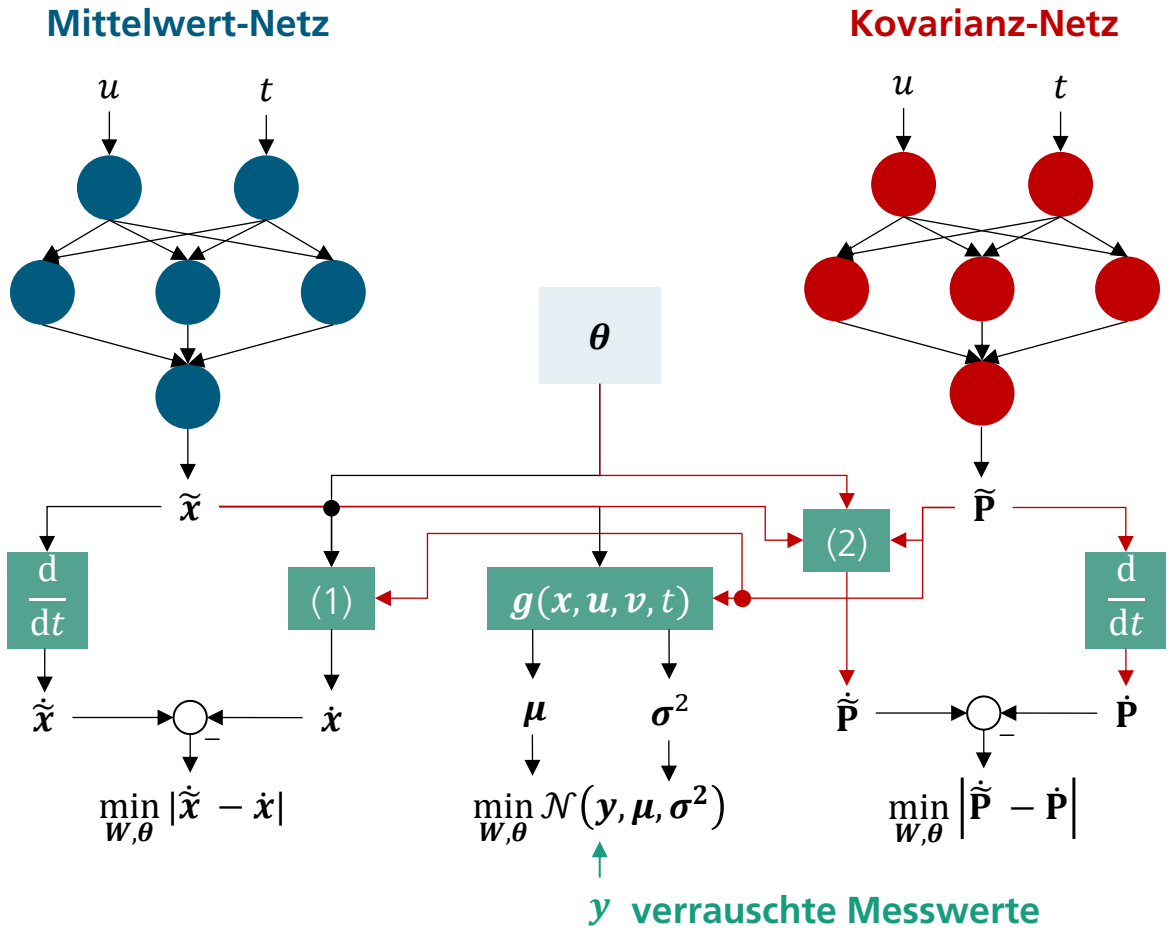
- Nutze **erweitertes Kalman-Bucy-Filter**

(1) $\dot{\tilde{\boldsymbol{x}}}(t) = f(\tilde{\boldsymbol{x}}, u, 0, t, \theta) + \mathbf{K}(t) \cdot \big(\boldsymbol{y}(t) - g(\tilde{\boldsymbol{x}}, u, 0, t)\big)$

(2) $\dot{\tilde{\mathbf{P}}}(t) = \mathbf{A}(t)\tilde{\mathbf{P}}(t) + \tilde{\mathbf{P}}(t)\mathbf{A}(t) - \mathbf{K}(t)\mathbf{C}(t)\tilde{\mathbf{P}}(t) + \mathbf{G}(t)\mathbf{Q}(t)\mathbf{G}^{\mathrm{T}}(t)$

- Exakte Lösung von (1) und (2) nur in Spezialfällen
  → Trainiere je ein neuronales Netz für Mittelwerte (**Mittelwert-Netz**) und Kovarianzen (**Kovarianz-Netz**).
- Beide Netze und die DGL-Parameter $\boldsymbol{\theta}$ werden wie ein normales PINN trainiert.

Aber: Was, wenn DGL nicht vollständig bekannt ist?



Tobias Nagel and Marco F. **Huber**: *Kalman-Bucy-Informed Neural Network for System Identification*. Proceedings of the 61st IEEE Conference on Decision and Control (CDC), 2022.

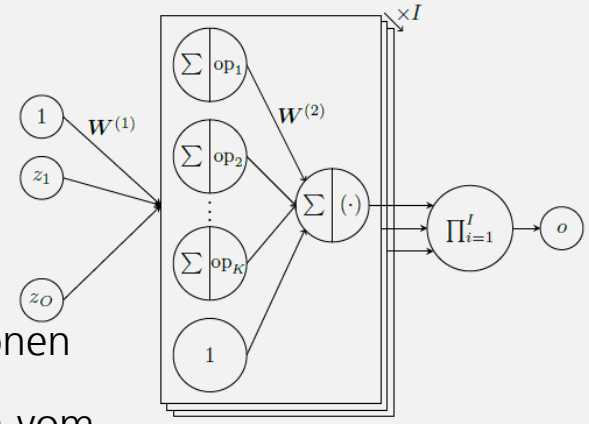
Fraunhofer IPA

ODE Learner

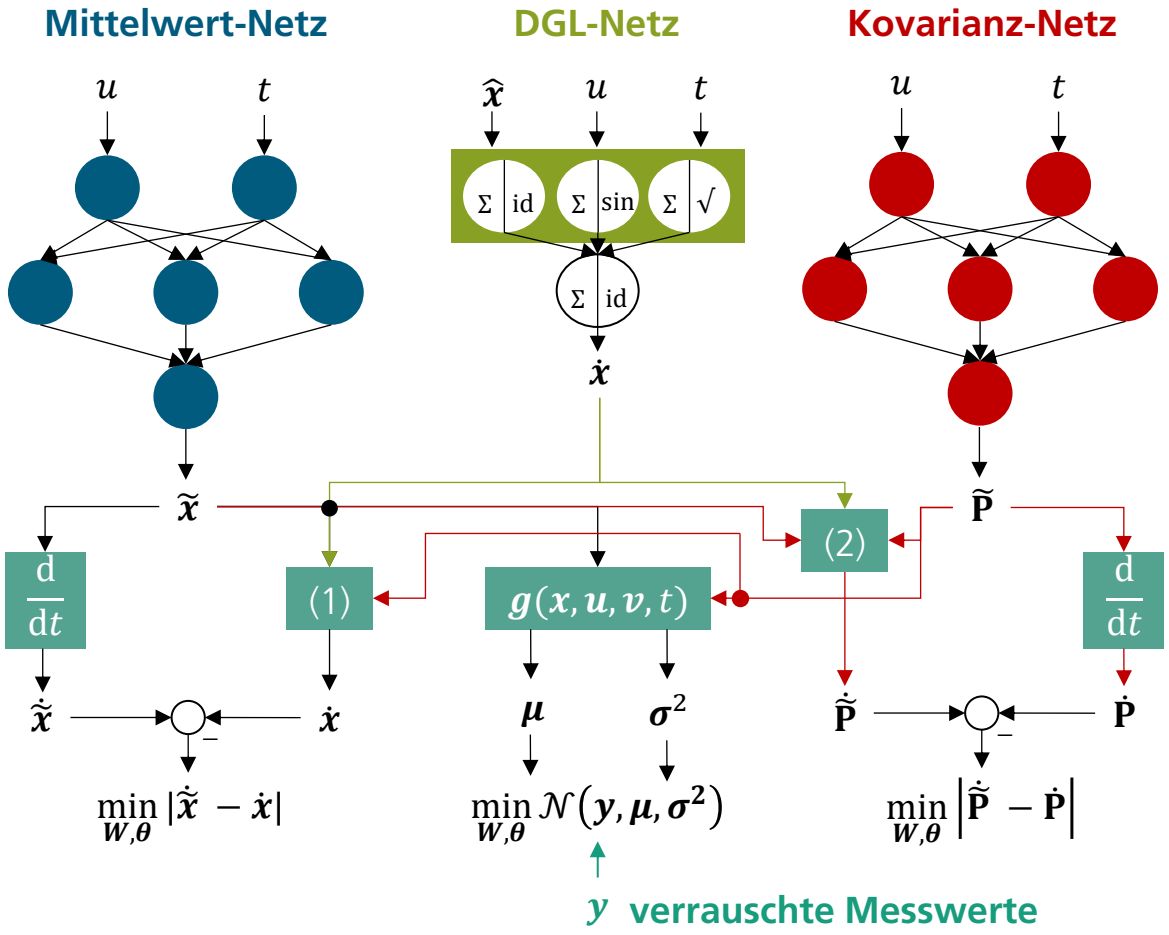
Identifizieren von DGLs aus Daten

Kernidee

- Einführen eines neuronalen Netzes zum Lernen (Identifizieren) von Teilen oder einer ganzen DGL (**DGL-Netz**)
- DGL-Netz nutzt spezialisierte **Operator-Neuronen** anstelle normaler Aktivierungsfunktionen



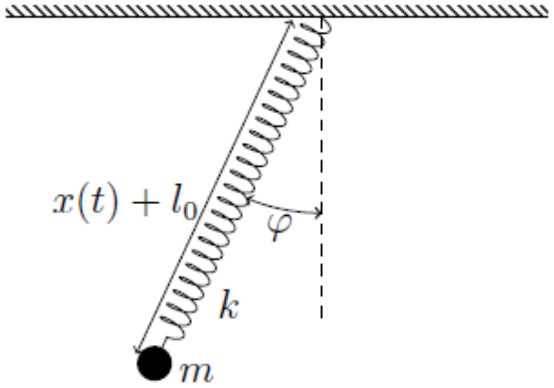
- Operatoren werden vom Nutzer definiert und erlauben das Einbringen von Vorwissen über das System



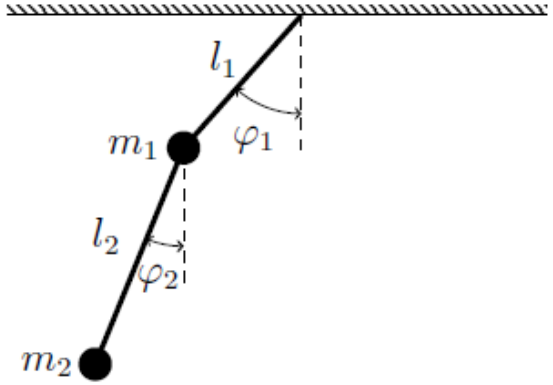
Tobias Nagel and Marco F. **Huber**: *Identifying Ordinary Differential Equations for Data-efficient Model-based Reinforcement Learning*. 2024 International Joint Conference on Neural Networks (IJCNN), 2024.

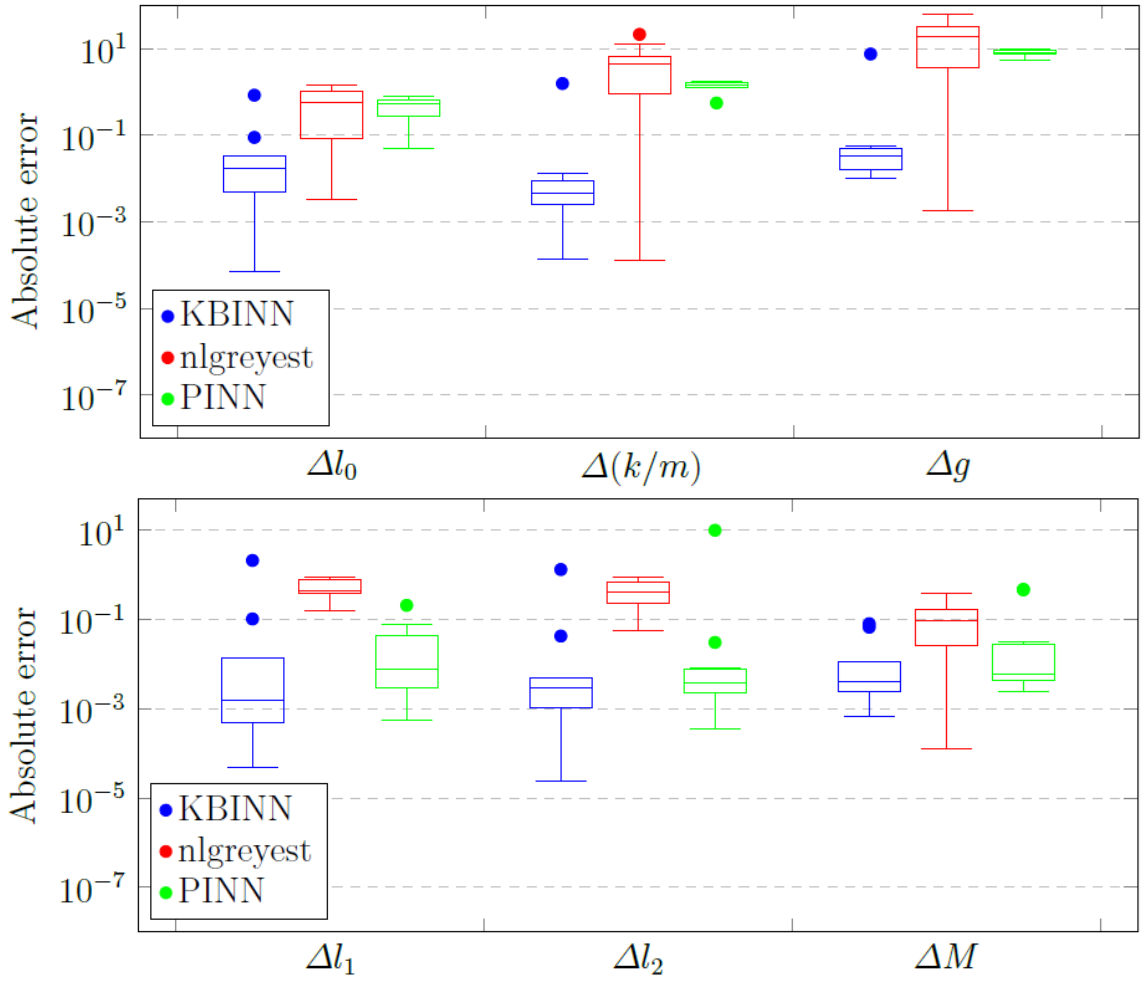
Ergebnisse
Simulationsdaten

Elastisches Pendel



Doppelpendel

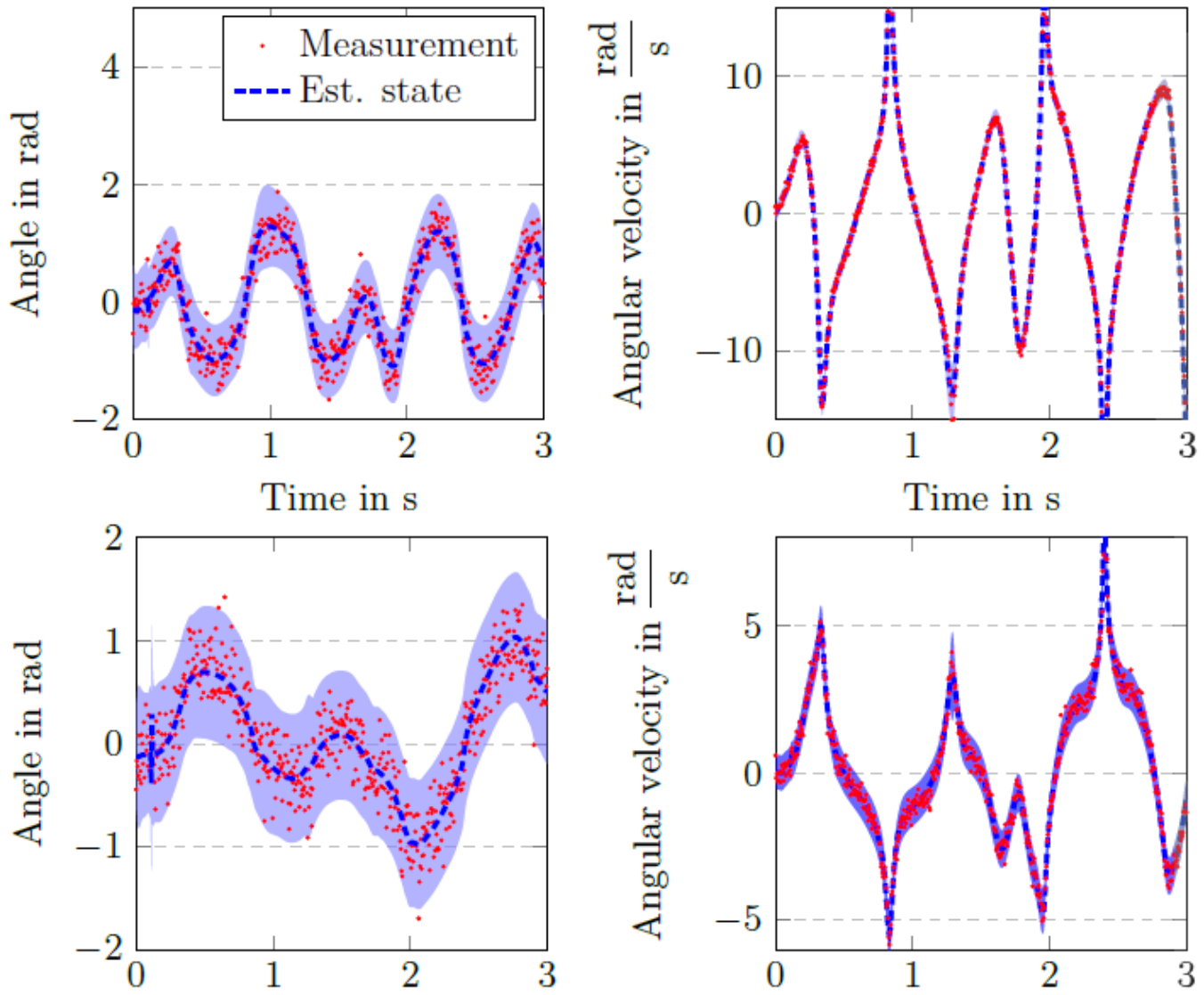




- KBINN erzielt bei identifizierten Parametern hohe Genauigkeiten
- Das gilt auch, wenn …
  - … die **DGL imperfekt ist**, d.h. wenn DGL mit fehlenden Termen (Dämpfung) bereitgestellt wird
  - … der **Zustand ist nicht vollständig beobachtbar**, bspw. wenn nur die Auslenkung gemessen wird

Ergebnisse
Doppelpendel

- Oben: Winkel φ_1 (links) und Winkelgeschwindigkeit $\dot{\varphi}_1$ (rechts) des ersten Pendels
- Unten: Winkel φ_2 (links) und Winkelgeschwindigkeit $\dot{\varphi}_2$ (rechts) des zweiten Pendels
- Messwerte (rote Punkte) und Zustandsschätzungen durch KBINN (blaue Linie) des Doppelpendels
- Da KBINN eine **stochastische DGL** lernt, sind auch Unsicherheiten mittels Varianzen quantifiziert (blaue schattierte Fläche)



Fraunhofer IPA

Ergebnisse

Beispiel: Kaskadierte Tanks

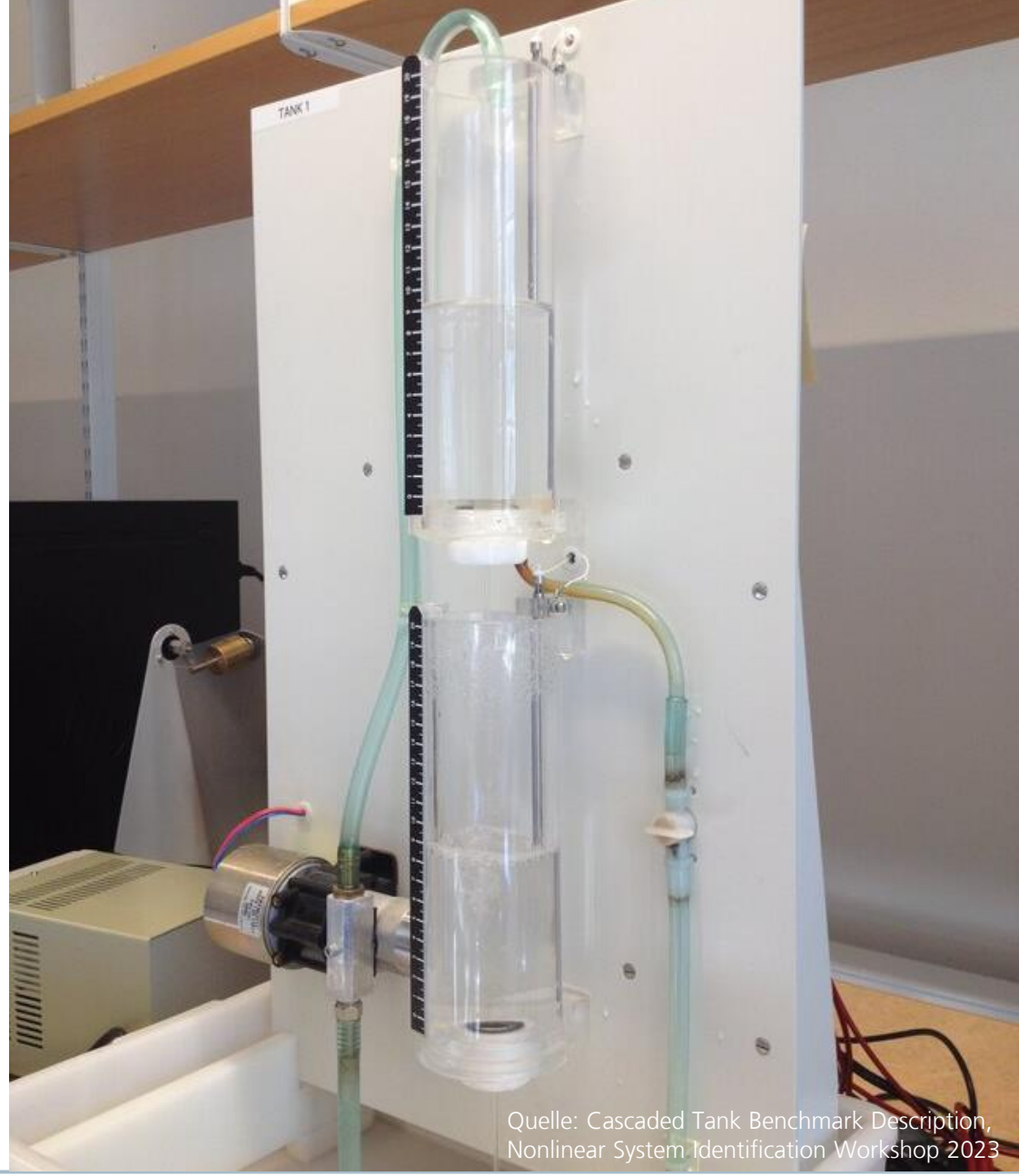
Zwei vertikal angeordnete Wassertanks mit einem Reservoir. Der Eingang des Systems besteht aus einer Wasserpumpe, die Wasser in den oberen Tank pumpt. Messgröße des Systems ist die Füllstandshöhe des unteren Tanks.

DGL beschreibt die grundlegende Dynamik des Wasserflusses:

$$\dot{x}_1(t) = -k_1\sqrt{x_1(t)} + k_4 u(t) + w_1(t)\,,$$
$$\dot{x}_2(t) = k_2\sqrt{x_1(t)} - k_3\sqrt{x_2(t)} + w_2(t)$$

Herausforderungen

- Beide Tanks können überlaufen, so dass Wasser in den darunter liegenden Tank bzw. das Reservoir fließt
- Nicht modellierte Sättigungsnichtlinearitäten
- Initialzustände (= Wasserstände beider Tanks) sind unbekannt
- Messwerte der Pumpe sind Werte des D/A-Wandlers



Quelle: Cascaded Tank Benchmark Description, Nonlinear System Identification Workshop 2023

Quelle des Benchmarks: M. Schoukens and J.P. Noël, *Three Benchmarks Addressing Open Challenges in Nonlinear System Identification*, 20th World Congress of the International Federation of Automatic Control, pp.448-453, Toulouse, France, July 9-14, 2017, doi: 10.1016/j.ifacol.2017.08.071.

Fraunhofer IPA

Ergebnisse

Beispiel: Kaskadierte Tanks

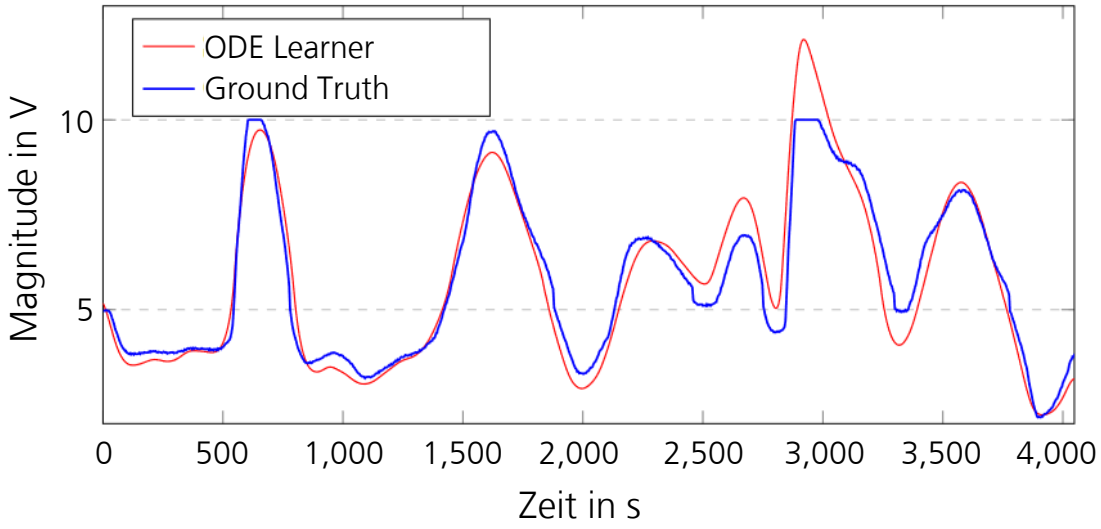
Zwei vertikal angeordnete Wassertanks mit einem Reservoir. Der Eingang des Systems besteht aus einer Wasserpumpe, die Wasser in den oberen Tank pumpt. Messgröße des Systems ist die Füllstandshöhe des unteren Tanks.

DGL beschreibt die grundlegende Dynamik des Wasserflusses:

$$\dot{x}_1(t) = -k_1\sqrt{x_1(t)} + k_4 u(t) + w_1(t)\,,$$
$$\dot{x}_2(t) = k_2\sqrt{x_1(t)} - k_3\sqrt{x_2(t)} + w_2(t)$$

Herausforderungen

- Beide Tanks können überlaufen, so dass Wasser in den darunter liegenden Tank bzw. das Reservoir fließt
- Nicht modellierte Sättigungsnichtlinearitäten
- Initialzustände (= Wasserstände beider Tanks) sind unbekannt
- Messwerte der Pumpe sind Werte des D/A-Wandlers



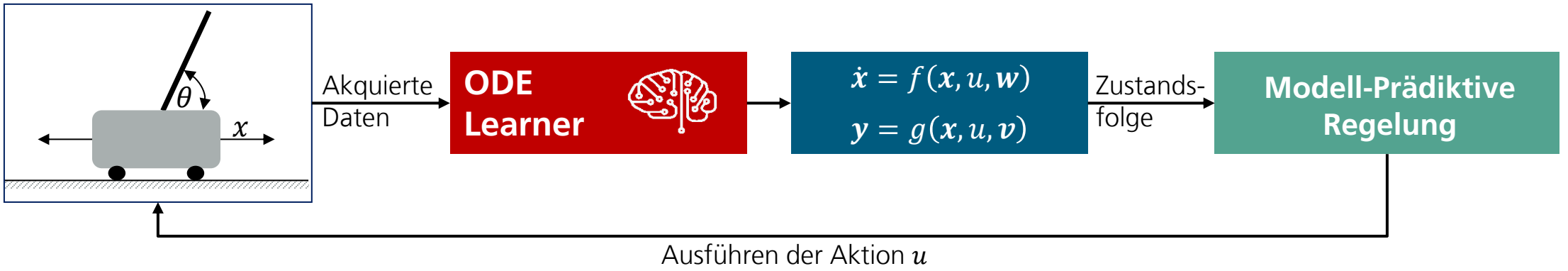
Methode	RMSE in V ↓
ODE Learner	**0.61**
MLP	1.23
SINDy	1.53
Matlab	1.26
INN	0.81

Quelle des Benchmarks: M. Schoukens and J.P. Noël, *Three Benchmarks Addressing Open Challenges in Nonlinear System Identification*, 20th World Congress of the International Federation of Automatic Control, pp.448-453, Toulouse, France, July 9-14, 2017, doi: 10.1016/j.ifacol.2017.08.071.

Fraunhofer IPA

Ergebnisse

Beispiel: Cart Pole



Modell-basiertes Reinforcement Learning

- **Zeitkontinuierliches, partiell beobachtbares MDP (POMDP):** Nur Schlittenposition x und Pendelwinkel ϕ werden gemessen
- **Lernen des Modells:** Nutze ODE Learner
- **Planen der Aktionen:** Nutze modell-prädiktive Regelung zum Aufschwingen, LQG zur Stabilisierung

Video

Aufschwingen des Pendels

Videoquelle: Tobias Nagel, 2024

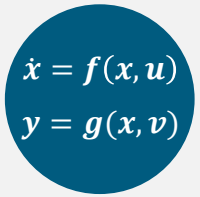
Fraunhofer IPA

Zusammenfassung

Bayes‘sches Maschinelles Lernen

- Online Lernen von Bayes‘schen neuronalen Netzen
- Modell-Prädiktive Regelung mit Bayes‘schen neuronalen Netzen

Physik-Informiertes Maschinelles Lernen



- Systemidentifikation mit physik-informierten neuronalen Netzen
- Model-Prädiktive Regelung mit gelernten Differentialgleichungen

Methoden

- **Kalman Bayesian Neural Network (KBNN)**
  - Quantifizierung von Unsicherheiten in Neuronalen Netzen
  - Erlaubt gradientenfreies Online-Lernen
- **ODE Learner**
  - Kalman-Bucy-Informed Neural Network (KBINN) und DGL-Netz lernen DGL aus verrauschten Messungen
  - Unbeobachtbare Zustände und Einbinden von Vorwissen
- **Grundlage für modell-basiertes Reinforcement Learning zum Regeln dynamischer Systeme**

Ausblick

- **KBNN:** Erweiterung auf unterschiedliche Netz-architekturen, vereinfachte der prioren Verteilung
- **ODE Learner:** Kürzere Trainingszeit, auf PDE erweitern

Vielen Dank für Ihre Aufmerksamkeit!

Prof. Dr.-Ing. Marco Huber

Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA
Tel. +49 711 970 1960
marco.huber@ipa.fraunhofer.de

www.ipa.fraunhofer.de/ai
www.ipa.fraunhofer.de/bildverarbeitung

Fraunhofer IPA

Fraunhofer-Institut für Produktions-technik und Automatisierung IPA