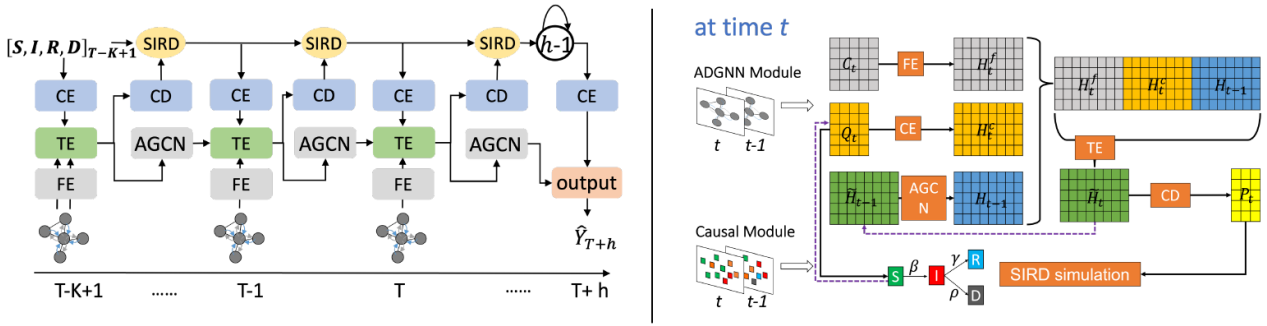
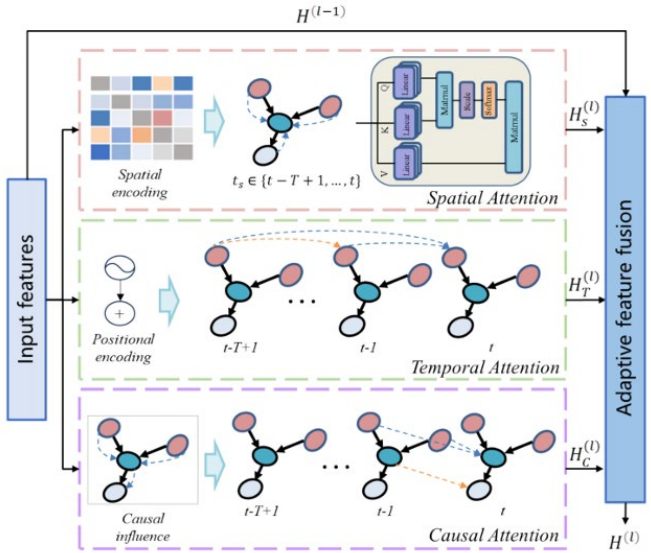
Causal-Based Graph Neural Networks for Predictive Maintenance

Description

- Causal-based Graph Neural Networks are a promising deep learning approach for modeling spatial–temporal dependencies in sensor networks while respecting the causal structure of manufacturing systems. By combining causality with graph architectures, they can strengthen predictive maintenance and improve system reliability by revealing true cause–effect relationships rather than spurious correlations.



Wang, Lijing, et al. "Causalgnn: Causal-based graph neural networks for spatio-temporal epidemic forecasting." Proceedings of the AAAI conference on artificial intelligence.



Zheng, Shuwen, et al. "Causal graph-based spatial–temporal attention network for RUL prediction of complex systems." Computers & Industrial Engineering 201 (2025): 110892.

Causal-Based Graph Neural Networks for Predictive Maintenance

Tasks

- Discuss how to incorporate causal prior knowledge into Graph Neural Networks (GNNs).
- Implement a Causal Neural Network Framework for Predictive Maintenance.
- Show how embedding this knowledge impacts the model's performance.
- Propose potential improvements or optimizations to current methods.
- Discuss the challenges in implementing causality with NNs in a manufacturing environment.

Desired Requirements

- Experience in Python, Deep Learning Frameworks (e.g., TensorFlow,PyTorch)

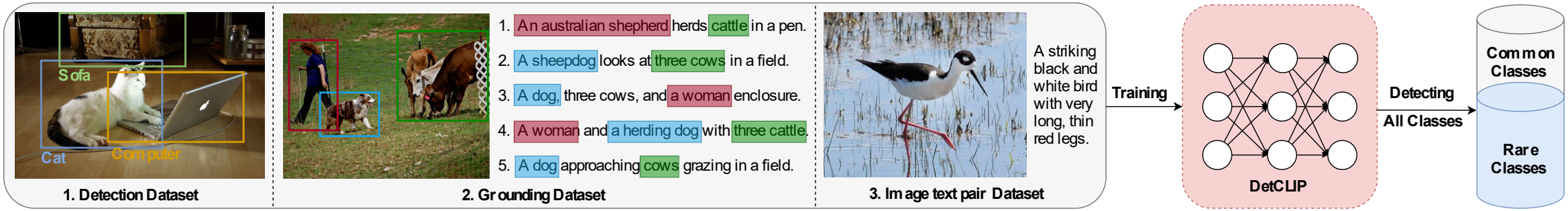
Supervisor

- Shahenda Youssef, MEng. shahenda.youssef@iosb.fraunhofer.de

Open-Vocabulary Object Detection

Description

- Open-vocabulary object detection (OVOD) aims to localize and recognize objects described by natural language, instead of relying on a fixed set of labels. By coupling visual representations with language, OVOD can respond to flexible text prompts and generalize to novel categories, reducing dependence on exhaustive annotations. This is increasingly important for dynamic, real-world scenarios where taxonomies evolve and long-tail concepts matter.
- The objective of this course is to implement a demonstrator with state-of-the-art open-vocabulary object detector. Different approaches will be investigated and tested for this purpose. A particular focus is put on performant deployment.



Yao, Lewei, et al. "Detclip: Dictionary-enriched visual-concept paralleled pre-training for open-world detection." *Advances in Neural Information Processing Systems* 35 (2022): 9125-9138.

Open-Vocabulary Object Detection

Tasks

- Investigate state-of-the-art approaches for open-vocabulary object detection
- Evaluate object detectors for a set of tasks/datasets
- Deploy with TensorRT and optimize model latency (Quantization, Sparsity, etc.)
- Design and implement a demonstrator

Desired Requirements

- Experience in Python, Linux, PyTorch
- Knowledge of object detection / computer vision; familiarity with vision–language models is beneficial
- Basic experience with containers, version control, and profiling tools is a plus

Supervisor

- Stefan Wolf, M.Sc. stefan.wolf@iosb.fraunhofer.de
- Dr.-Ing. Andreas Specker andreas.specker@iosb.fraunhofer.de

High Performance Gaussian Mixture Fitting

Description

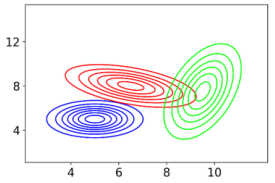
Just like Fourier series that are basis functions for periodic functions, Gaussian mixtures are Basis functions for probability density functions making them invaluable in probabilistic artificial intelligence. The expectation maximization and its full Bayesian counterpart are often times used to find the parameters of a Gaussian mixture that explains a particular set of data. There exist implementations of the said algorithm but are not scalable which is a limitation due to the contemporary datasets and dimensions associated with them. The scalability problem can be remedied using ideas and technologies in high performance computing and we aim to incorporate a couple of them.

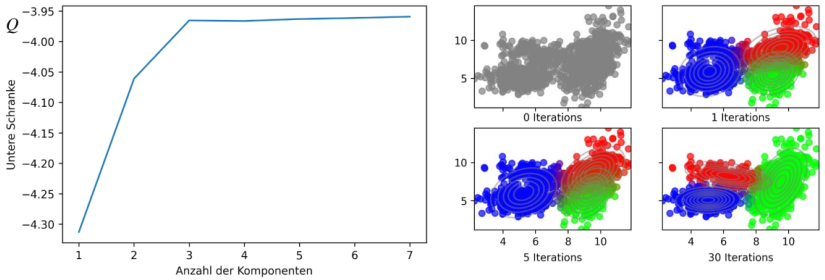
Beispiel: EM-Algorithmus für Gauß'sche Mixturen, $K = 3$

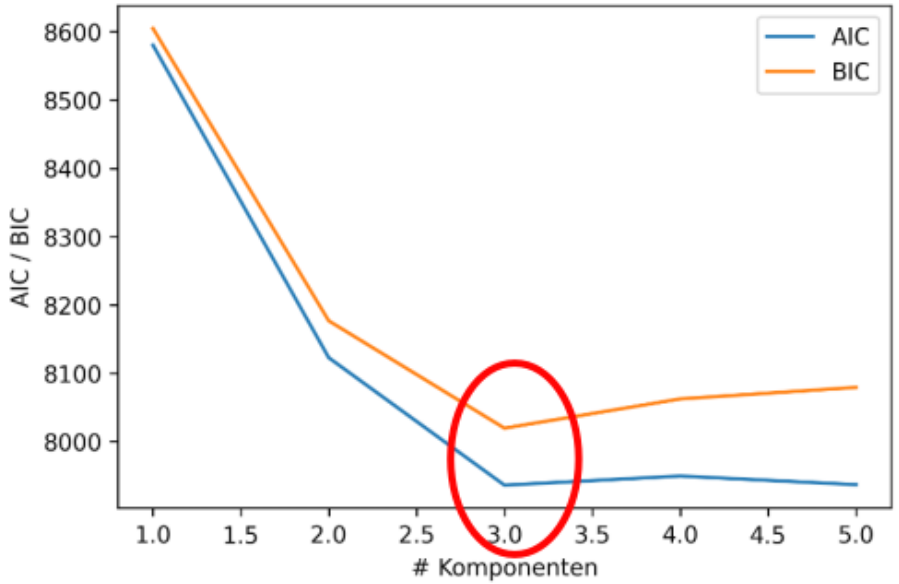
Wahre Verteilung: $\boldsymbol{\mu}_1 = \begin{bmatrix} 5.0 \\ 5.0 \end{bmatrix}$ $\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1.0 & 0 \\ 0 & 0.7 \end{bmatrix}$

$\boldsymbol{\mu}_2 = \begin{bmatrix} 6.5 \\ 8.0 \end{bmatrix}$ $\boldsymbol{\Sigma}_2 = \begin{bmatrix} 2.0 & -0.7 \\ -0.7 & 0.7 \end{bmatrix}$

$\boldsymbol{\mu}_3 = \begin{bmatrix} 9.5 \\ 7.5 \end{bmatrix}$ $\boldsymbol{\Sigma}_3 = \begin{bmatrix} 0.7 & 0.9 \\ 0.9 & 5.0 \end{bmatrix}$







Lehrstuhl für Interaktive Echtzeitsysteme
Institut für Anthropomatik und Robotik

Tasks

- Porting mixture part of the scikit-learn package into a PyTorch based one; capable of GPU acceleration and distributed computing.
  - The final package must be released as a tar ball appropriate for Unix/Linus systems.
  - The release must be done through a gitlab pipeline that does for example the tests, linting, license checking, and security checking.
  - The package documentation must be done automatically and later be deployed into a static website. (sphinx or similar workflow)
  - Benchmarking for speed and accuracy against scikit-learn.

Desired Requirements

- Familiarity with Unix/Linux system.
- Familiarity with Object Oriented Programming.
- Familiarity with Linear Algebra and Optimization
- Google-Fu

Supervisor

- Ali Darijani ali.darijani@iosb.fraunhofer.de

Adaptive Dynamic Programming for Control: A Survey and Recent Advances

Description

Adaptive Dynamic Programming (ADP) is a method used to design controllers for complex systems, especially when finding an exact solution is too difficult. The main problem it addresses is how to optimize the behavior of a system over time, for example, to reach a desired state or balance competing objectives in a game-like scenario. ADP is particularly useful when dealing with large or nonlinear systems, where traditional control methods fail or are too slow. Researchers have developed many algorithms to make ADP efficient, and it can be applied to both continuous-time and discrete-time systems. The motivation for studying ADP is that it allows intelligent control of systems in situations where the environment is complicated, uncertain, or changing, which is very common in engineering, robotics, energy systems, and artificial intelligence applications.

Literatur

Liu, Derong, et al. "Adaptive dynamic programming for control: A survey and recent advances." IEEE Transactions on Systems, Man, and Cybernetics: Systems 51.1 (2020): 142-160.

Adaptive Dynamic Programming for Control: A Survey and Recent Advances

Karlsruher Institut für Technologie

- **Tasks**
  - Understanding the concept of ADP.
  - Trying to improve the literature methods by replacing the direct optimization problem to find the optimal action, exploring m ultiple shooting in ADP, and leveraging neural networks to learn the optimal action.
  - Implementing this improved methods
  - Evaluation
- **Desired Requirements**
  - Experience in Python, Linux, PyTorch
- **Supervisor**
  - Negar Arabizadeh (negar.arabizadeh@kit.edu)

Contrastive Learning as Goal-Conditioned Reinforcement Learning

Karlsruher Institut für Technologie

Description

- Themenbeschreibung

  In reinforcement learning (RL), an agent learns to achieve goals by interacting with an environment. A key challenge is learning good representations of states and goals, which makes learning faster and more stable. Traditional RL often struggles with this, and some methods add extra tricks like data augmentation or auxiliary losses to help.

  This paper shows a different approach: instead of adding extra parts, we can learn representations directly using contrastive learning. In other words, the RL algorithm itself can be designed as a contrastive representation learning method. The key idea is that the inner product of the learned representations corresponds to a goal-conditioned value function, which tells the agent how good it is to be in a state when aiming for a goal. This method works well even without additional tricks and improves performance on many tasks, including offline and image-based tasks.

Literature

Eysenbach, Benjamin, et al. "Contrastive learning as goal-conditioned reinforcement learning." Advances in Neural Information Processing Systems 35 (2022): 35603-35620.

Contrastive Learning as Goal-Conditioned Reinforcement Learning

Tasks

- Understand the method: Learn how contrastive learning works
- Implement the method from the paper
- Compare results: Check if the contrastive RL method helps the agent achieve goals better than standard RL methods.

Desired Requirements

- Experience in Python, Linux, PyTorch

Supervisor

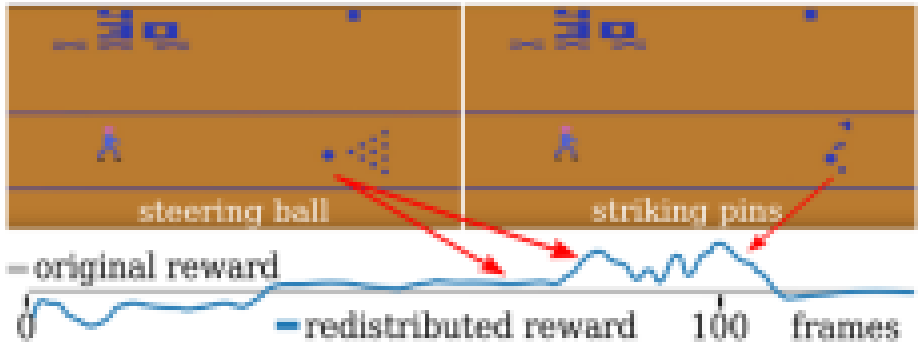
- Negar Arabizadeh (negar.arabizadeh@kit.edu)

Delayed Reward Problems

- **Description**
  - Themenbeschreibung
  - “Delayed reward problems” are a concept from reinforcement learning (RL) and decision-making systems. A delayed reward problem occurs when the consequences of an action are not immediately observed, but appear later in time. That is, an action taken now might only affect the reward (success or failure) after many steps, making it harder for the agent to know which actions were good or bad.

Literature

Arjona-Medina, Jose A., et al. "Rudder: Return decomposition for delayed rewards." Advances in Neural Information Processing Systems 32 (2019).



Arjona-Medina, Jose A., et al. "Rudder: Return decomposition for delayed rewards." Advances in Neural Information Processing Systems 32 (2019).

Delayed Reward Problems

- **Tasks**
  - Understanding the concept of the delayed reward problem
  - What is the state of the art in this area?
  - Design and implement this method according to the literature paper referenced in these slides.
  - Evaluation
- **Desired Requirements**
  - Experience in Python, Linux, PyTorch
- **Supervisor**
  - Negar Arabizadeh (negar.arabizadeh@kit.edu)