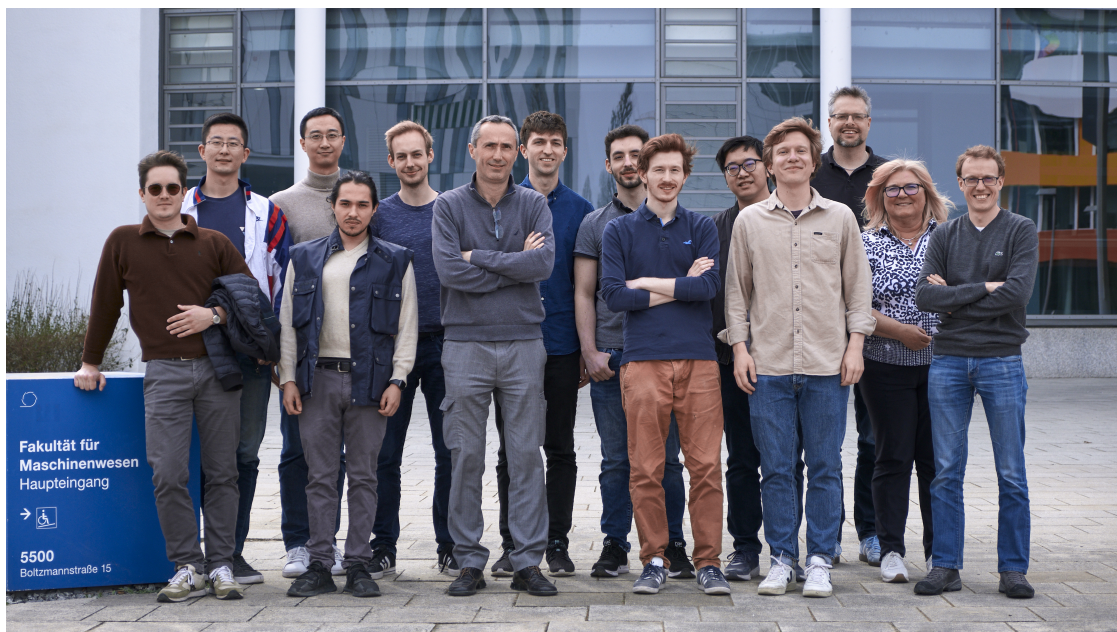


# Chair of Cyber-Physical Systems in Production Engineering: Annual Report 2024

March 31, 2025



# 1 Introduction

**“Designing smart, predictable, and high-performance embedded solutions for next generation Cyber-Physical Systems.”**

Modern Cyber-Physical Systems (CPS) are the next generation of engineered systems in which computing, communication, and control technologies are tightly integrated. Applications include system automation, the Internet of Things (IoT), smart buildings, smart manufacturing, smart cities, digital agriculture, robotics, and autonomous vehicles. The chair of Cyber-Physical Systems in Production Engineering was founded in September 2018.

In 2024, the research activities of the chair focused on the following topics: a) designing and implementing novel resource management policies for embedded real-time systems running on high-performance heterogeneous platforms, b) developing new reinforcement learning architectures and algorithms for CPS, and c) developing synthetic training paradigms for 6D pose recognition in robotic automation. Members of the chair were involved in the peer review process of several international conferences/journals in real-time embedded systems and CPS, including RTSS 2024, RTAS 2025, RTAS 2024, ECRTS 2024, ETFA 2024, RTCSA 2024, DAC 2024, DATE 2025, NeurIPS 2024, ICLR 2025, ICCPS 2024, OSPERT 2024, JRWRTC 2024, IROS 2024, ITSC 2024, as well as Journal of IEEE Transactions on Computers, IEEE Transactions on Parallel and Distributed Systems, IEEE Systems Journal, and IEEE Access.

## 2 Predictable and high-performance resource management of CPS on heterogeneous platforms

The widespread use of artificial intelligence (AI) algorithms in many Cyber-Physical Systems (CPS), such as autonomous cars, drones, and smart robots, has driven the integration of specialized hardware accelerators (e.g., GPUs, FPGAs, TPUs) on high-performance multiprocessor boards. Towards ensuring safety and real-time requirements, these heterogeneous multiprocessor systems-on-chips (MPSoC) pose unprecedented challenges. In fact, the implementation of complex CPS using these platforms generates increasing volumes of real-time (e.g., imaging) data flows, causing the hardware memory hierarchy (the DRAM, the interconnect, and the cache hierarchy, especially the last level cache shared among multiple cores) to become a bottleneck and a source of temporal unpredictability. This phenomenon is further aggravated by the presence of accelerators that can independently access memory with high-bandwidth requests. Traditional techniques to allocate and optimize the execution of real-time tasks on safety-critical CPS do not consider the heterogeneity of the computing elements and the complexity of MPSoCs' memory hierarchy. In addition, classical task models widely adopted in the real-time scheduling domain fail to capture the parallelization and heterogeneous computing needs of the new workloads.

On the application and deployment side, our research considers optimization and scheduling techniques to hide the complexity of the configuration space from the integrators while enforcing isolation and ensuring the real-time properties of the workload. Problems under analysis are: the schedulability of DAG tasks leveraging techniques and observations from the graph domain and reinforcement learning [58]; partitioning strategies to schedule gang tasks [54, 57]; the response time analysis of global fixed-priority preemptive scheduling of constrained-deadline tasks running on a uniform multiprocessor where each processor can be characterized by a different speed [53]; the optimization of shared resources such as caches while simultaneously providing real-time guarantees [56].

At the integration level, developers and integrators face the challenge of finding the right trade-offs between selecting the appropriate scheduling policies, assigning real-time tasks to (heterogeneous) cores, selecting the size of cache partitions, and determining adequate bandwidth to allocate to each communication resource. Our work tackles these challenges with techniques that the industry can rapidly adopt, and that aim to practically simplify the deployment of real-time workload on MPSoC without sacrificing either predictability or performance. On the platform side, we research, develop, and evaluate techniques to restore isolation and temporal predictability of safety-critical software. We specifically target solutions (see [74, 73, 34, 79]) that prove to perform well “in practice”, and we focus our integration efforts at both Operating System (OS) and Hypervisor levels. Hypervisors (see [1, 39, 46, 21, 20]) have become the de-facto industry standard to ensure isolation in certified and partitioned safety-critical systems but do not provide satisfactory isolation and predictability when contention at the cache, interconnect, and DRAM levels are considered. To facilitate the evaluation and the adoption of these techniques by the industry, in addition to publications (see [27, 80, 7, 30, 28]), we actively participate to the development of open source hypervisors (see [6, 45]) to make the designed techniques not only readily available, but also supported by an active community. On these topics, we also actively collaborate with highly skilled international teams [38, 70], which pursue objectives close to ours. We additionally actively develop open-source real-time frameworks (see [42, 61]) to improve the interoperability and exchange of results among international research groups.

In the remainder of this section, we present more details on our recent works [27, 30, 58, 57, 54]. These papers are the results of our research efforts in 2024 toward predictable and high-performance resource management for CPS on heterogeneous platforms.

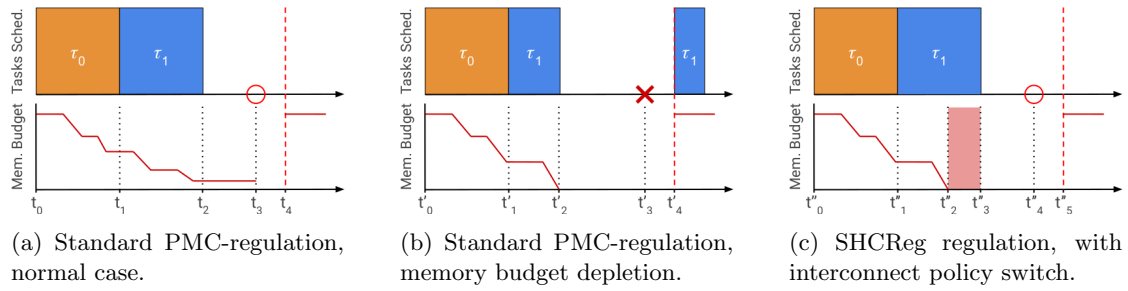


Figure 1: Example scenario of a PMC-regulated CPU, where an increased memory consumption causes  $\tau_1$  to miss its deadline.

## 2.1 Mixed-Criticality Task-based Isolation [27]

Mixed-criticality systems (MCS) have emerged as a crucial area of research in the development of cyber-physical systems (CPS). These systems integrate tasks with varying levels of importance, such as safety-critical and non-critical functions, into a single platform. The coexistence of such tasks poses significant challenges to ensure that safety-critical tasks maintain predictable behavior while non-critical tasks maximize resource utilization.

Whether in the compute or memory domain, the mixed-critical tasks are traditionally isolated from one another using bandwidth servers/regulators. Well-known and accepted solutions such as the Constant Bandwidth Server (CBS) and Memguard, respectively, come to mind. However, their combination and interplay remain uncharted territory.

As we discovered, the lack of coordination between the two mechanisms leads to poor handling of *memory overload* conditions. These conditions correspond to all scenarios where, despite being eligible for scheduling at the OS level, high-criticality tasks are blocked by the memory bandwidth regulation implemented in the hypervisor.

The example depicted in Figure 1 perfectly illustrates multiple scenarios that can arise during the system’s lifetime. In this example, the considered system is composed of one low- and one high-criticality task (respectively  $\tau_0$  and  $\tau_1$ ) scheduled on one core using a CBS server for each task. The common PMC budget assigned to the core is determined beforehand via profiling and the addition of a fixed *safety margin*, which is common practice in industrial applications. While in Figure 1a,  $\tau_1$  is able to complete on time, in Figure 1b suffers from a *memory overload* as it experiences extra blocking due to the lack of sufficient *memory budget* caused by an unexpected increased memory consumption from  $\tau_0$ . Such an increase can be due to changing computational needs requiring additional memory accesses (for example, object detection or object tracking in an almost empty street vs. a crowded intersection). We note that such an increase in memory consumption cannot be determined a priori without resorting to very pessimistic overestimations.

The primary objective of [27] is to achieve robust isolation between tasks of different criticalities, both in terms of computational resources and memory usage. This isolation ensures that high-criticality tasks are not adversely affected by low-criticality tasks, even in scenarios of unexpected resource demand or system overload. We propose a hardware/software co-design methodology named Mixed-Criticality Task-Based Isolation (MCTI) to address the aforementioned challenge, and we evaluate an early prototype. MCTI proposes to handle *memory overloads* such as the one presented above via a simple set of rules that adequately instrument well-known and available tools. In the example illustrated in Figure 1c, upon the detection of an overload, the critical task  $\tau_1$  is allowed to bypass the normal PMC-regulation mechanism, serving  $\tau_1$  critical memory access. At the same time, SHCReg programs a smart interconnect to





2. Instead of *indirectly* monitoring a core’s memory bandwidth between the last-level cache and the memory controller through performance monitoring counters (PMCs), we let the FPGA’s *snoop* on the memory traffic directly through its cache coherent interconnect. This improves the temporal resolution of the memory bandwidth regulation, as the FPGA is able to observe all memory transactions with their addresses in a cycle-accurate manner in real-time. This extends from our earlier work on CAESAR [47].
3. Direct access to the low-level debug infrastructure of the SoC from the FPGA allows us to halt and resume cores much faster than in MemPol. This improves the regulation performance by an order of magnitude and overcomes the problems of setpoint overshooting observed in MemPol.

Overall, MemCoRe improves upon MemPol’s regulation by an order of magnitude and enables a novel *fine-grained spatiotemporal nanosecond-scale bandwidth regulation*.

### 2.3 Edge Generation Scheduling for DAG Tasks Using Deep Reinforcement Learning [58]

Current real-time applications in the automotive, avionics, and industrial domains realize their functionalities through complex chains of intercommunicating tasks. For example, [2, 24] present recent driving assistance and autonomous driving applications where data is processed through multiple periodically-activated steps, from sensory data acquisition (*e.g.*, Lidar and cameras) to actuation commands (*e.g.*, braking and steering). Such applications—including their execution and precedence-constraint requirements—are modeled using directed acyclic graph (DAG) tasks.

In [58], we study the problem of scheduling a non-preemptive DAG task on a multicore platform with identical processors. Drawing from graph theory, we develop a novel schedulability test based on the key observation that a DAG whose *width* is not greater than the number of available processors and whose *length* is less than or equal to its deadline is schedulable. We classify such a DAG as a *trivially schedulable* DAG and show that any DAG is schedulable *if and only if* it can be converted into a trivially schedulable DAG by adding edges. In addition, we show that a trivially schedulable DAG task can be dispatched via *global* and *partitioned* strategies. While global dispatching strategies usually require prioritized queues for ready jobs, we show that prioritization is not needed when dispatching a trivially schedulable DAG task because a ready job is guaranteed to have an idle processor available for execution. For partitioned dispatching strategies, the paths covering a DAG can simply be assigned to processors in the order of the precedence constraints.

To test whether a DAG task is schedulable, we then focus on the problem of adding appropriate edges to convert it into a trivially schedulable DAG task without violating its original constraints. To this end, we propose the *Edge Generation Scheduling* (EGS) framework that attempts to make a DAG task trivially schedulable by iteratively adding appropriately chosen edges. If EGS succeeds in reducing the width to the number of processors while maintaining the length less than or equal to the deadline, the original DAG task is guaranteed to be schedulable. The EGS framework shifts the complexity of solving the DAG scheduling problem to the problem of selecting *the best* edges to add to a DAG to make it trivially schedulable. We exploit topological and temporal graph properties to limit the search space when adding edges and propose a deep reinforcement learning (DRL) approach to learn an edge generation policy. In particular, we use the DRL algorithm Proximal Policy Optimization (PPO) [49] and the graph representation neural network architecture Graphormer [72] that is well suited for solving this class of problems. A high-level illustration of the proposed EGS framework is presented in Figure 3.

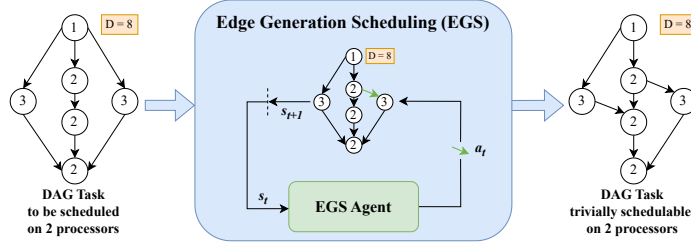


Figure 3: The Edge Generation Scheduling Framework.

By combining the proposed EGS framework and the edge generation policy learned by the developed DRL, we derive a concrete DAG scheduling algorithm called EGS-PPO and evaluate it against state-of-the-art DAG scheduling heuristics. Our results show that EGS-PPO consistently outperforms the other approaches by requiring fewer processors to schedule the same DAGs. Additionally, EGS with a random edge generation policy can achieve results similar to the state-of-the-art, highlighting the significance of the EGS framework. We also compare EGS-PPO against an optimal mixed-integer linear programming (MILP) baseline on small DAG tasks. EGS-PPO consistently outperforms other approaches, reducing the difference from the optimal solution by a factor of three to five.

In summary, our paper [58] makes the following contributions:

1. We present a new schedulability test (trivial schedulability) for DAG tasks based on observations from the graph domain;
2. We propose a novel DAG scheduling framework (EGS) that minimizes processor usage by iteratively generating edges;
3. We formulate the edge generation problem as a Markov decision process (MDP) and develop a deep reinforcement learning (DRL) agent to learn an effective edge generation policy for EGS;
4. We evaluate the effectiveness of the proposed EGS framework and DRL algorithm by comparing it with exact solutions and state-of-the-art DAG scheduling algorithms through extensive experiments on synthetic DAG tasks.

## 2.4 Strict Partitioning for Sporadic Rigid Gang Tasks [54, 57]

Gangs are fine-grained parallel jobs that have to start at the same time and execute in parallel across multiple processing units. Gang scheduling has been widely used in high-performance computing [43, 76], distributed cloud [40], and containers [12]. In recent years, it has also been increasingly gaining popularity in embedded systems. In particular, gang scheduling is well-suited for emerging deep-learning applications [18, 8] deployed on highly parallel hardware accelerators [51, 71] that require significant computational resources and may involve complex inter-task communication and dependencies.

Most of the approaches for gang scheduling in real-time systems are *global* [18, 17, 35, 33, 55, 36, 41, 32, 22, 23, 15], wherein tasks are not pinned to any particular processor and can start execution on any available processors. However, global gang scheduling has several limitations. First, global gang scheduling usually suffers from large migration costs, especially when parallel tasks are deployed on hardware accelerators where the migration and setup times are usually high (*e.g.*, *FPGA* dynamic partial reconfiguration [9] or *Edge TPU* model parameter loading [55, 25]).

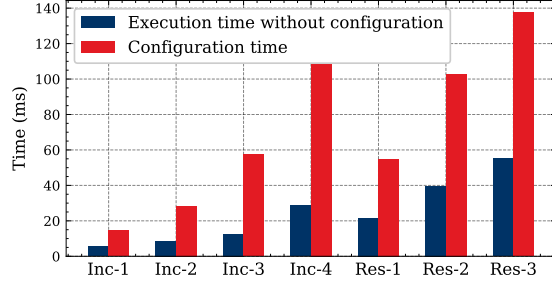


Figure 4: Configuration time vs. net execution time of DNN inferences on 16 Edge TPUs. “Inc-1,2,3,4” denote Inception-v1 [60], v2 [13], v3 [13], v4 [59], respectively; “Res-1,2,3” denote ResNet-50,101,152 [26], respectively.

Figure 4 illustrates the large configuration overhead of seven representative deep neural networks (DNNs) on an AI Accelerator card integrated with 16 Edge TPUs. In contrast to global scheduling, partitioned scheduling reserves processing units for each model so that the model weights can be cached in the internal memory statically, eliminating the need for frequent and costly memory allocation during runtime. Second, global gang scheduling is burdened with interference overestimation resulting from the difference in gang tasks’ parallelism levels (*e.g.*, finding the set of interfering jobs that may exactly fit on the available processors [35] and unbounded priority-inversion in non-preemptive gang scheduling [16]). Meanwhile, partitioned scheduling can avoid these issues by isolating tasks with large parallelism variance.

While partitioned scheduling can help reduce runtime overheads and isolate interference overestimation, the underlying task allocation is equivalent to the bin-packing problem [48] and is hence highly intractable (NP-hard in the strong sense [31]). Moreover, partitioned systems of sequential tasks can be easily analyzed using exact uniprocessor schedulability tests. However, for parallel tasks, more complicated tests may be necessary to account for the anomalies that may occur in partitioned gang systems where different gang task groups share some processors [69].

To address these challenges, our paper [57] proposes a simple yet effective method, named *strict partitioning*, for scheduling rigid gang tasks (*i.e.*, gang tasks with fixed parallelism levels) on identical multiprocessor platforms. It builds disjoint partitions of tasks and processors so that the tasks in different partitions do not interfere with each other. Within the boundaries of each partition, tasks can run under any type of online scheduler. Moreover, strict partitioning tries to group tasks with similar parallelism levels onto the same partition so that uniprocessor scheduler (*e.g.*, *Deadline Monotonic (DM)* [5], *Earliest Deadline First (EDF)* [37]) and exact schedulability tests are applicable. Specifically, in [57], we

- Propose a new *strict partitioning* strategy for scheduling rigid gang tasks;
- Propose a *first-fit decreasing volume (FFDV)* heuristic to partition rigid gang tasks and multiprocessor platforms;
- Present two strict partitioning variants **SP-U** and **SP-G** by combining FFDV with different online schedulers. For **SP-U**, we prove utilization bounds. For **SP-G**, we improve FFDV to achieve better performance;
- Evaluate the proposed strategy and algorithms by comparing them with state-of-the-art preemptive and non-preemptive gang scheduling techniques on synthetic task sets and a case study based on Edge TPU benchmarks.

### 3 Deep Reinforcement Learning for Cyber-Physical Systems

Deep reinforcement learning (DRL) is a promising class of learning algorithms to tackle complex optimization problems for control and planning of Cyber-Physical Systems through interactions with the environment alone. The recent advance in DRL enables robots to master complicated tasks with impressive performances, e.g., locomotion, autonomous driving, and robotic manipulation. However, the training of the DRL agents is typically sampling inefficient and unsafe during the exploration. Moreover, the learned agents are parameterized with deep neural networks, which are hard to predict and verify, imposing safety risks for the deployment of physical systems.

Our research in this direction focuses on integrating domain knowledge into the data-driven DRL. One aspect is a residual policy diagram, which consists of a model-based policy and a DRL policy. Such a residual diagram can take advantage of both model-based policies and data-driven DRL: the model-based policy can guide the exploration of DRL agents during training and regulate the behavior of the DRL agent; meanwhile, the DRL agent learns to effectively deal with the uncertainties and compensate for the modeling errors faced by the model-based policies. Our recent publication investigated the potential of residual DRL for improving the safety assurance and performance of DRL-enabled cyber-physical systems. Another aspect is integrating the knowledge of the problem’s geometry information into the learning algorithm to solve map-based path-planning problems. By exploiting the invariance and equivariance properties of the task, we show that the proposed algorithm outperforms multiple policy learning baselines regarding sampling efficiency and performance.

#### 3.1 Physics-Regulated Deep Reinforcement Learning [11]

Machine learning (ML) technologies have been integrated into autonomous systems, defining learning-enabled autonomous systems. These have succeeded tremendously in many complex tasks with high-dimensional states and action spaces. However, the recent incidents due to the deployment of ML models overshadow the revolutionizing potential of ML, especially for safety-critical autonomous systems. Developing safe ML is thus more vital today. In the ML community, deep reinforcement learning (DRL) has demonstrated breakthroughs in sequential decision-making in broad areas, ranging from autonomous driving to games. This motivates us to develop a DRL-based safe learning framework for achieving safe and complex tasks of safety-critical autonomous systems.

Despite the tremendous success of DRL in many autonomous systems for complex decision-making, applying DRL to safety-critical autonomous systems remains a challenging problem. It has a deep root to the action policy of DRL being parameterized by deep neural networks (DNN), whose behaviors are hard to predict and verify, raising the first safety concern. The second safety concern stems from DRL’s purely data-driven DNN for powerful function approximation and representation learning of action-value function, action policy, and environment states. Specifically, recent studies revealed that purely data-driven DNN applied to physical systems can infer relations violating physics laws, which sometimes leads to catastrophic consequences (e.g., data-driven blackout owing to violation of physical limits).

To address the aforementioned safety concerns, we propose the **Phy-DRL**: a physics-regulated deep reinforcement learning framework with enhanced safety assurance. Depicting in Figure 5, Phy-DRL has three novel (invariant-embedding) architectural designs:

- **Residual Action Policy**, which integrates data-driven-DRL action policy and physics-model-based action policy.

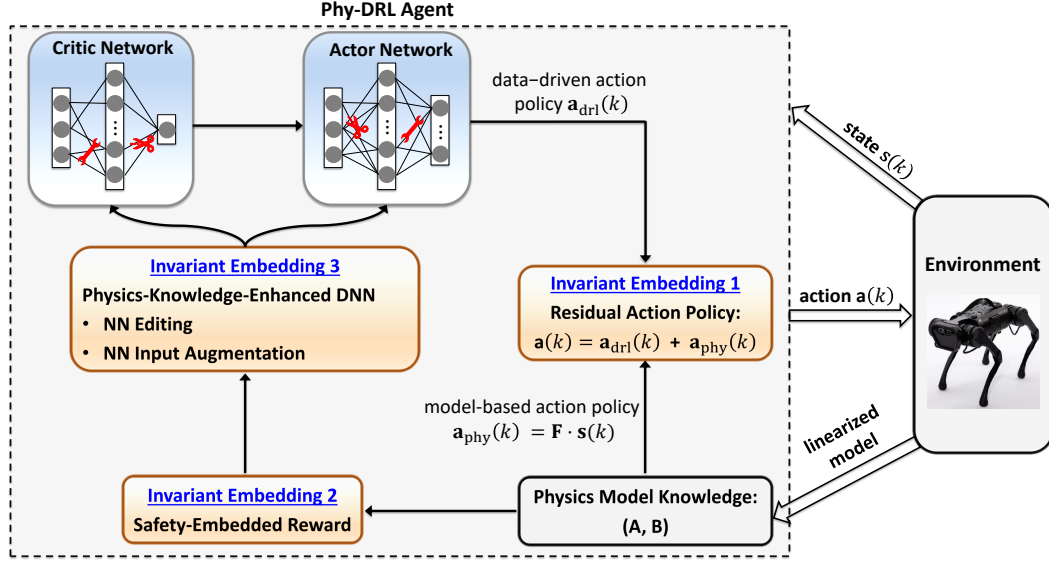


Figure 5: The plot shows the diagram of the proposed Phy-DRL framework. It consists of a real plant, a physics-model-based controller, a DRL algorithm of *actor-critic* architecture, and a physics-model-guided reward module. The final control command is computed by taking the summation of the action generated from the model-based controller and the action output from the actor-network of DRL. The states, control actions, and rewards computed from the Physical-Model-Guided Reward module are saved as training data for optimizing the critic and actor networks.

- **Safety-Embedded Reward**, in conjunction with the Residual Action Policy, empowers the Phy-DRL with a mathematically provable safety guarantee and fast training.
- **Physics-Knowledge-Enhanced Critic and Actor Networks**, whose neural architectures have two key components: i) NN input augmentation for directly capturing hard-to-learn features, and ii) NN editing, including link editing and activation editing, for guaranteeing strict compliance with available knowledge about the action-value function and action policy.

As shown in Figure 5, Phy-DRL employs an analyzable and verifiable model-based action policy, offering fast training toward safety guarantee. Meanwhile, the linear model knowledge (leveraged for computing model-based policy) works as a model-based guidance for constructing the safety-embedded reward for DRL towards a mathematically provable guaranteed safety. Lastly, the proposed NN editing guarantees the strict compliance of critic and actor networks with partially available physics knowledge about the action-value function and action policy. The experiments on simulated cart-pole and quadruped robots have demonstrated that Phy-DRL offers higher sampling efficiency for policy learning and improves safety assurance in various testing scenarios.

### 3.2 Residual Reinforcement Learning for High-Performance Cyber-Physical Systems [68]

The domain of autonomous racing provides a challenging test bed for real-world applications of cyber-physical systems. In the F110 racing series [44], RC cars of 1/10th-scale race autonomously against the clock, pushing the car’s physics to its limits when racing. The interactive decision-making in multi-agent autonomous racing offers insights valuable beyond the domain of self-driving cars. Mapless online path planning is particularly of practical appeal but poses a challenge for safely overtaking opponents due to the limited planning horizon.

Accordingly, we introduced RaceMOP [68], a novel method for mapless online path planning designed for multi-agent racing of F1TENTH cars. Unlike classical planners that depend on predefined racing lines, RaceMOP operates without a map, relying solely on local observations to overtake other race cars at high speed. As shown in Figure 6, our approach combines an artificial potential field method as a base policy with residual policy learning to introduce long-horizon planning capabilities. We advance the field by introducing a novel approach for policy fusion with the residual policy directly in probability space, building on our previous work on residual policy learning [67]. Our proposed method aggregates the learned residual action  $\mu_{R,\theta}$  with the base action  $a_B$  by means of a *truncated* Gaussian distribution  $\mathcal{N}(\mu, \sigma, c^-, c^+)$  [10] using

$$\mu = a_B + \alpha \cdot \mu_{R,\theta} \text{ and } \sigma = \sigma_{R,\theta}, \quad (1)$$

with a learned, state-independent standard deviation  $\sigma_{R,\theta}$ . The truncation interval  $[c^-, c^+]$  of the *truncated* Gaussian ensures that sampled actions are bounded  $a_t \in [c^-, c^+]$ , while the probability mass gets correctly redistributed for values outside the truncation interval. Note that for this distribution, the mean  $\mu$  is also the distribution’s mode. Using other distributions can either lead to a biased mapping of the base action when fusing before the probability function or wrong gradient information when clipping to a bounded action space is required after sampling. Using the truncated Gaussian overcomes both limitations.

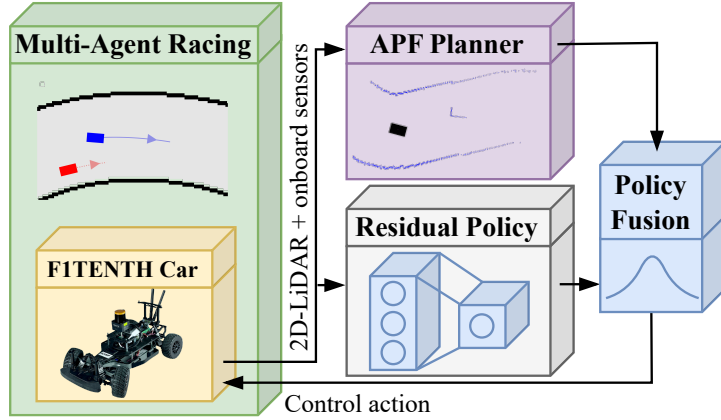


Figure 6: Our method, named RaceMOP, is a novel mapless online path planner for multi-agent racing that uses only local observations. This method fuses an APF planner with a learned residual policy for simulated F1TENTH cars.

Our experiments for twelve simulated racetracks validate that RaceMOP is capable of long-horizon decision-making with robust collision avoidance during overtaking maneuvers. RaceMOP demonstrates superior handling over existing mapless planners while generalizing to unknown



racetracks, paving the way for further use of our method in robotics. Our evaluation compares the performance of RaceMOP against the pure artificial potential field planner to demonstrate the effectiveness of the residual policy learning approach.

First, it can be seen that RaceMOP improves the base policy w.r.t. all performance metrics:

- Lab times are decreased by 8.65 % to  $I_T = 52.41$  s averaged over all training racetracks.
- Despite the faster lap times, RaceMOP manages to decrease unsuccessful overtaking substantially to 0.33 % crashes per attempt, i.e., there is approx. one crash for 300 successful overtaking maneuvers.
- RaceMOP increases environment crashes  $I_E$  for two tracks but reduces them when the artificial potential field planner struggled before.

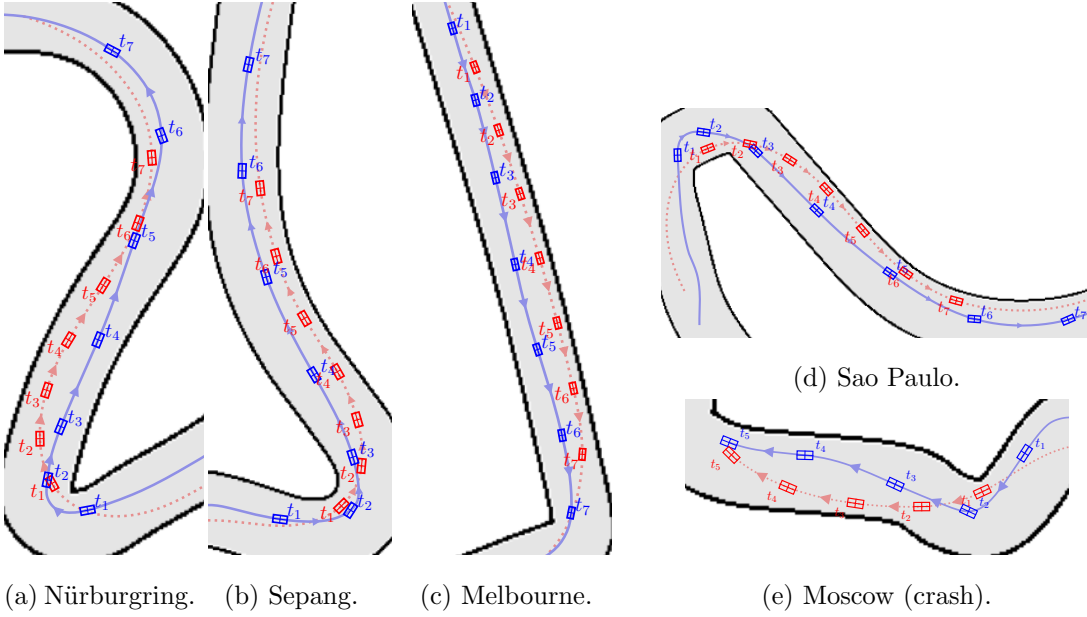


Figure 7: Exemplary overtaking maneuvers of RaceMOP for five different, replicated real-world racetracks where the ego vehicle (blue, full line) overtakes the opponent (red, dashed line), showing various strategic behaviors. Discrete timesteps  $t_1, \dots, t_7$  of the vehicle’s pose are given every 0.5 s.

We discuss RaceMOP’s behavior with five examples. Our analysis of various scenarios and racetracks shows that RaceMOP’s advantage is a robust overtaking behavior at curves. Typically, RaceMOP will approach a curve fast, break if the opponent switches sides when cutting the curve, and then accelerate quickly to pass the opponent. This behavior is effective, affirming that RaceMOP is capable of long-horizon reasoning from local observations where other approaches typically require map data.

Figure 7 visualizes overtaking maneuvers of raceMOP for different driving scenarios on different racetrack maps:

- **Inside Overtaking (Figure 7 (a) and (b)):** The ego vehicle waits with the attempt as the opponent cuts the corner. After the apex, the overtaking becomes feasible as a gap opens at the inside, and the ego accelerates to overtake the opponent at a safe distance.

- **High Velocity Overtaking (Figure 7 (c)):** The ego uses its velocity advantage to overtake the opponent before a  $90^\circ$  curve by breaking late, requiring a careful estimation of its physical driving limit and the opponent’s trajectory to finish the maneuver in time.
- **Outside Overtaking (Figure 7 (d)):** The opponent follows the wall in a curve closely. RaceMOP has learned to wait first and then takes advantage of this behavior, passing the opponent on the outside to finish the maneuver.
- **Crash (Figure 7 (e)):** This unsuccessful overtaking happens at a difficult section where RaceMOP first waits with the overtake. It then starts overtaking but incorrectly interprets the opponent’s behavior, who cuts the corner. As the opponent closes in, the ego is not fast enough to finish the maneuver, leading to a crash.

Our results demonstrate that RaceMOP clearly outperforms comparable, map-less planners for training and test racetracks, respectively. We find that a key component of RaceMOP’s strong performance is the novel method for fusing the base action of the artificial potential field planner with the learned residual action. Moreover, RaceMOP’s advantage is its ability to make long-horizon decisions despite lacking map data, where naive planners fail. Our future work includes testing RaceMOP against various opponent strategies and eventually transferring our method to a real-world race with multiple F1TENTH cars.

### 3.3 Equivariant Ensembles and Regularization for Reinforcement Learning in Map-based Path Planning [65]

Reinforcement learning (RL) is a rapidly advancing methodology for learning policies through interactions with environments, as it promises to address complex real-world problems that were previously unsolvable. Its application breadth is continuously widening with applications in control, planning, and general optimization domains. Many of these environments contain symmetries that could be exploited for improved training efficiency, robustness, and performance. Symmetries in the environments result in equivariance (see Figure 8 for an example) and invariance properties of the optimal policy and value function, respectively. Therefore, imposing these properties on the learning components should be beneficial.

Inspired by the success of convolutional neural networks that make neural networks equivariant to translations, earlier research focused on designing neural networks that are equivariant or invariant to symmetry transformations. However, this endeavor can be daunting as it limits the neural network components to only small subsets of equivariant and invariant layers, opposing the trend toward the ever-increasing complexity of neural networks for performance improvements. Recently, related research has shifted towards adding a regularization term to the training loss that nudges the networks towards equivariance and invariance without constraining the network design choices.

In this work, we incorporate symmetries into the learning process by constructing equivariant policies and invariant value functions without the need for special neural network designs. We introduce *equivariant ensembles* that average over the networks’ outputs for all symmetry transformations. We prove that policy and value ensembles are equivariant and invariant, respectively, and show how they enrich the gradients in policy optimization algorithms such as proximal policy optimization (PPO) [49]. We further use regularization to push the individual components toward the ensembles, adding inductive bias.

To showcase the benefits of the equivariant ensembles and regularization, we evaluate their performance in a challenging, long-horizon, map-based planning application, the unmanned aerial vehicle (UAV) Coverage Path Planning (CPP) problem [62]. In this case study, the environment

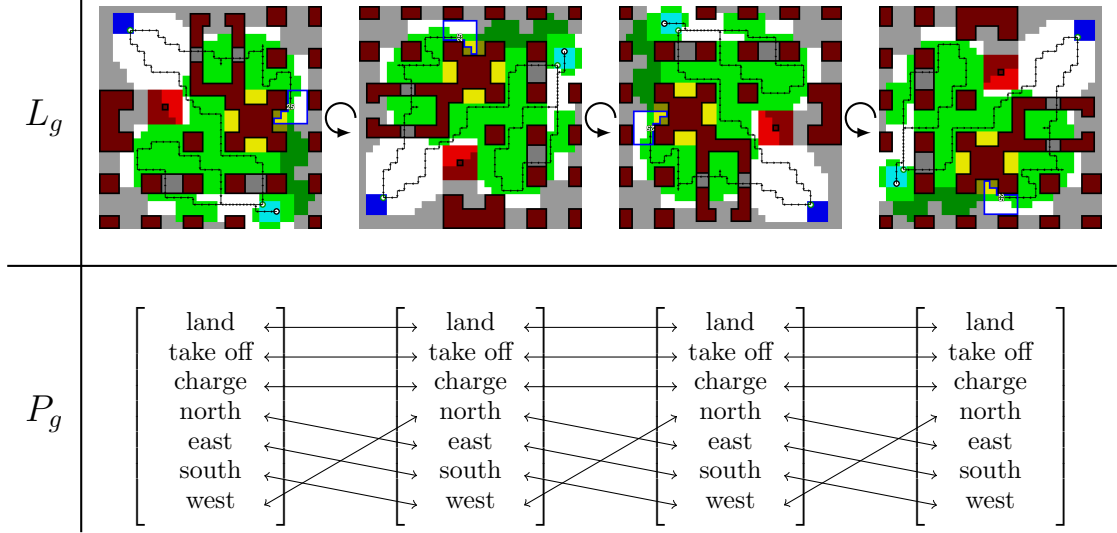


Figure 8: Visualization of the equivariances in a UAV coverage path planning application showing the input and output transformations  $L_g$  and  $P_g$  for all rotations in  $G$ .

state can be represented as a map ([63]), and rotational symmetries can be exploited, as visualized in Figure 8. The results show that the ensemble makes the policy equivariant and that combining the ensemble and regularization improves performance significantly. We further show that regularization on the policy does not guarantee equivariance, which should be considered when regularizing the value estimate towards invariance.

### 3.4 Learning to Generate All Feasible Actions [64]

Modern cyber-physical systems (CPS) are becoming increasingly complex, motivating the use of data-driven techniques such as reinforcement learning (RL) for control and planning. However, ensuring constraint satisfaction remains a key challenge in many applications, as traditional RL methods typically require systematic constraint violations during training, which can be computationally prohibitive or unsafe.

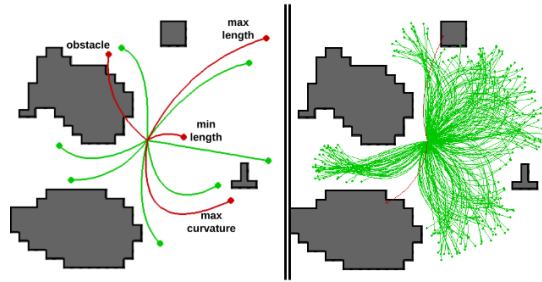


Figure 9: Example spline generation task showing examples of feasible (green) and infeasible (red) splines on the left and sampled splines from a trained feasibility policy on the right. The example shows how the proposed feasibility policy can suggest a wide range of feasible actions (splines) in constrained tasks.

Existing approaches, such as action rejection, resampling, and projection, primarily focus on guaranteeing safety but do not improve the learning efficiency of RL agents. In contrast, this work introduces *action mapping*, a novel framework that explicitly separates feasibility learning from objective optimization. The feasibility policy learns to generate all feasible actions, which can then be used by an objective policy to optimize task performance while ensuring constraint satisfaction.

The feasibility policy is trained as a distribution matching problem, where it learns to sample uniformly from the feasible action space. We derive gradient estimators for different f-divergences, enabling the training of generative models that capture the full diversity of feasible actions. The approach is evaluated in three scenarios: an illustrative example with disconnected feasible regions, a robotic path planning task where feasible trajectory segments must be generated, and a robotic grasping simulation where grasp poses are constrained by object geometry.

Results demonstrate that the proposed feasibility policy successfully generates diverse and well-distributed feasible actions, significantly improving learning efficiency in constrained environments. Figure 9 illustrates this in a spline generation task, where the feasibility policy successfully samples a wide range of feasible trajectory segments while avoiding infeasible ones. By explicitly learning feasibility as a separate step, action mapping facilitates the development of RL frameworks that are both safe and sample-efficient.

## 4 Computer Vision for Robotic Manipulation

Vision-based perception provides promise for robots to understand working scenarios in complex manipulation tasks thereby enhancing overall task success rates. However, training learning-based computer vision components need a huge amount of labeled data, which is expensive in terms of time and labor cost. Our research leverages modern simulators to generate synthetic data for training. We aim to systematically examine the efficacy of synthetic training paradigms in preparing robotic systems for real-world manipulation tasks. Building on the existing foundation, the Chair of Cyber-Physical Systems in Production Engineering developed an integrated baseline 6D pose vision system and a Blender pipeline for generating synthetic data. These advancements have facilitated new methods for robotic grasping and manipulation. With a data generation pipeline and a 6D Pose Estimation pipeline, the foundation for further research has been set.



Figure 10: From left to right: the five selected objects photographed, photo-realistically rendered, depth rendered and automatically generated grasp poses for the duck.

These advancements represent a significant step forward in the application of 6D pose estimation to robotics, particularly in the context of real-world scenarios and varying environmental conditions, as shown in Figure 11.

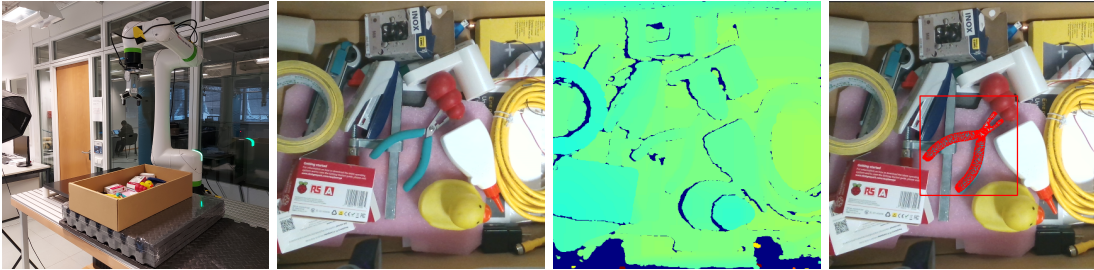


Figure 11: The figure shows real-world grasping experiments. The experimental setup, RGB view, depth view, and the predicted pose of pliers on RGB are shown from the left to the right.

### Real Evaluation Dataset

Previous experiments relied on existing State-of-the-Art datasets or qualitative evaluation. However, existing datasets for 6D pose estimation utilizing stereovision have been lacking in the amount of data and quality. We, therefore, introduced an automated way to acquire fully labeled stereovision data, annotated with 6D poses and masks of several different objects [14]. This work is currently under submission and already serves as a validation dataset at our chair (Figure 12). By using a robotic data acquisition system, we captured a large amount of real-world data,

with automatic labeling of the object poses and segmentation masks. Fusing multiple camera views enables us to obtain a high-quality depth reconstruction and circumvent the problems due to the use of an RGB-D camera. The use of high-resolution, dual RGB images allows the exploration of new perception algorithms that do not depend on often unreliable RGB-D cameras (typically used in these applications).

The generalization of the vision models is also the focus of our research, eliminating the need for expensive re-training of the models for each new object. So-called one-shot or zero-shot models can find the pose of novel unseen objects without any need for additional data or training. Future research will investigate the advantages of these techniques.



Figure 12: One sample from our real dataset. From left to right, the annotated RGB image, the depth images of the camera, the segmentation mask, and the high-quality restored depth image are shown for the left camera of the stereo camera.

## **5 Basic Information of the Chair of Cyber-Physical Systems in Production Engineering**

### **Management**

Prof. Dr. Marco Caccamo, Director

[www.mw.tum.de/cps](http://www.mw.tum.de/cps)

[mcaccamo@tum.de](mailto:mcaccamo@tum.de)

Tel: +49 89 289 55170

### **Administrative Staff**

Patrick Meins, Secretary

### **Research Scientists**

- Andrea Bastoni, Dr.
- Alexander Züpke, Dr.
- Harald Bayerlein, Dr.
- Mirco Theile, Dr.
- Daniele Ottaviano, Dr.
- Denis Hoornaert, M.Sc.
- Hongpeng Cao, M.Eng.
- Daniele Bernardini, M.Sc.
- Binqi Sun, M.Sc.
- Raphael Trumpp, M.Sc.
- Lukas Dirnberger, M.Sc.
- Andres Rodrigo Zapata Rodriguez, M.Sc.
- Ashutosh Pradhan, M.Sc.
- Bohua Zou, M.Sc.

### **Research Focus**

- Safety-critical cyber-physical systems
- Real-time systems
- Scheduling and schedulability analysis
- Secure and safe integration of machine learning with CPS
- Reinforcement learning for CPS



## **Competence**

- System-level programming
- Embedded system software design
- Hardware modules design for FPGAs
- Real-time operating systems
- Reinforcement learning for CPS

## **Infrastructure**

- 3 DOF helicopter
- Embedded and FPGA multi-core development platforms
- High-performance servers
- Linear inverted pendulum
- Fused filament fabrication, dual-head 3D printer
- F1/10 autonomous cars
- FANUC CRX and Robco Modular Robot Arm
- Unitree A1 Quadruped Explorer
- Unitree Go2 Quadruped Go2 EDU plus

## **Collaborations**

- University of Illinois at Urbana-Champaign, USA
- University of California, Berkeley, USA
- Boston University, USA
- University of Colorado Boulder, USA
- Indiana University Bloomington, USA
- University of Waterloo, Canada
- Federal University of Santa Catarina, Brazil
- University of Modena and Reggio Emilia, Italy
- EURECOM, Sophia Antipolis, France
- LAAS-CNRS, Toulouse, France
- Université Paris Cité, France
- Nantes Université, France
- Eindhoven University of Technology, Netherland
- The University of Tokyo, Japan
- Technical University of Dortmund, Germany

## Courses

- Concepts and Software Design for Cyber-Physical Systems
- Tutorial Concepts and Software Design for Cyber-Physical Systems
- Advanced Seminar on Safe Cyber-Physical Systems
- PhD-Seminar on Real-Time Cyber-Physical Systems
- Cyber-Physical Systems Lab: Autonomous Applications
- Simplex: Fault-Tolerant Control Strategy for Real-Time Cyber-Physical Systems - Laboratory
- Design and Analysis of Digital Control Systems
- Tutorials on Design and Analysis of Digital Control Systems
- Simulation and Control of Mechanical Systems

## Humboldt Sponsored Research

### Selected Publications 2024

#### Awards

**Best Paper Award:** Coherence-aided memory bandwidth regulation. In Proceedings of 45th IEEE Real-Time Systems Symposium (RTSS), 2024

#### Journal papers

- Binqi Sun, Mirco Theile, Ziyuan Qin, Daniele Bernardini, Debayan Roy, Andrea Bastoni, and Marco Caccamo. Edge generation scheduling for DAG tasks using deep reinforcement learning. *IEEE Transactions on Computers*, 73(4):1034–1047, 2024
- Binqi Sun, Tomasz Kloda, Sergio Arribas Garcia, Giovanni Gracioli, and Marco Caccamo. Minimizing cache usage with fixed-priority and earliest deadline first scheduling. *Real-Time Systems*, pages 1–40, 2024
- Alexander Zuepke, Andrea Bastoni, Weifan Chen, Marco Caccamo, and Renato Mancuso. MemPol: Polling-based Microsecond-scale Per-core Memory Bandwidth Regulation. *Real-Time Systems*, 60, 2024
- Mirco Theile, Daniele Bernardini, Raphael Trumpp, Cristina Piazza, Marco Caccamo, and Alberto L Sangiovanni-Vincentelli. Learning to generate all feasible actions. *IEEE Access*, 2024
- Denis Hoornaert, Golsana Ghaemi, Andrea Bastoni, Renato Mancuso, Marco Caccamo, and Giulio Corradi. Mcti: mixed-criticality task-based isolation. *Real-Time Syst.*, 60(2):328–365, July 2024

## Conference papers

- Binqi Sun, Tomasz Kloda, and Marco Caccamo. Strict partitioning for sporadic rigid gang tasks. In *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 252–264, 2024
- Binqi Sun, Tomasz Kloda, and Marco Caccamo. Response time analysis for fixed-priority preemptive uniform multiprocessor systems. In *36th Euromicro Conference on Real-Time Systems (ECRTS)*, 2024
- Binqi Sun, Tomasz Kloda, Chu-ge Wu, and Marco Caccamo. Partitioned scheduling and parallelism assignment for real-time DNN inference tasks on multi-tpu. In *ACM/IEEE Design Automation Conference (DAC)*, 2024
- Bohua Zou, Binqi Sun, Yigong Hu, Tomasz Kloda, Marco Caccamo, and Tarek Abdelzaher. A performance prediction-based DNN partitioner for Edge TPU pipelining. In *IEEE Military Communications Conference (MILCOM)*, pages 1–6, 2024
- Hongpeng Cao, Yanbing Mao, Lui Sha, and Marco Caccamo. Physics-regulated deep reinforcement learning: Invariant embeddings. In *International Conference on Learning Representations (ICLR)*, 2024
- Mirco Theile, Hongpeng Cao, Marco Caccamo, and Alberto L. Sangiovanni-Vincentelli. Equivariant ensembles and regularization for reinforcement learning in map-based path planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 14164–14171, 2024
- Raphael Trumpp, Ehsan Javanmardi, Jin Nakazato, Manabu Tsukada, and Marco Caccamo. RaceMOP: Mapless online path planning for multi-agent autonomous racing using residual policy learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8449–8456, 2024
- **(Best Paper Award)** Ivan Izhbirdeev, Denis Hoornaert, Weifan Chen, Alexander Zuepke, Y. Hammad, M. Caccamo, and R. Mancuso. Coherence-aided memory bandwidth regulation. In *Proceedings 45rd IEEE Real-Time Systems Symposium (RTSS)*, 2024
- Tobias Betz, L. Wen, F. Pan, G. Kaljavesi, A. Zuepke, A. Bastoni, M. Caccamo, A. Knoll, and J. Betz. A containerized microservice architecture for a ros 2 autonomous driving software: An end-to-end latency evaluation. In *2024 IEEE 30th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2024

## Software Development

- Jailhouse Real-Time. <https://gitlab.com/minervasys/public/jailhouse>
- RT-Bench Framework. <https://gitlab.com/rt-bench/rt-bench>
- Jailhouse Cache-coloring. Jailhouse ML. <https://groups.google.com/g/jailhouse-dev/c/K4rqZxpxa0U>
- Virtio prototype (see [50]). <https://github.com/gschwaer/rt-virtio>
- 6D Pose Estimation Training Pipeline. <https://github.com/HP-CA0/6IMPOSE>
- Synthetic Data Generation. [https://github.com/LukasDb/6IMPOSE\\_Data](https://github.com/LukasDb/6IMPOSE_Data)

- Robotic Grasping. [https://github.com/LukasDb/6IMPOSE\\_Grasping](https://github.com/LukasDb/6IMPOSE_Grasping)
- Mapless Autonomous Racing. <https://github.com/raphajaner/raceMOP>
- DyPACC-enabled RISC-V core. <https://github.com/denishoornaert/DyPACC-NaxRiscv>

## References

- [1] Siemens AG. Jailhouse hypervisor. <https://github.com/siemens/>. Accessed: 2025-01-29.
- [2] Matteo Andreozzi, Giacomo Gabrielli, Balaji Venu, and Giacomo Travaglini. Industrial Challenge 2022: A High-Performance Real-Time Case Study on Arm. In *Euromicro Conference on Real-Time Systems (ECRTS)*, volume 231, pages 1:1–1:15, 2022.
- [3] ARM. Arm architecture reference manual supplement. memory system resource partitioning and monitoring (MPAM) for Armv8-A. <https://developer.arm.com/docs/ddi0598/latest>.
- [4] ARM. Quality of Service in ARM Systems: An Overview, 2014.
- [5] N.C. Audsley, A. Burns, M.F. Richardson, and A.J. Wellings. Hard real-time scheduling: The deadline-monotonic approach. *IFAC Proceedings Volumes*, 24(2):127–132, 1991.
- [6] Andrea Bastoni. Jailhouse Public Repository with Real-Time Extensions. <https://gitlab.com/minervasys/public/jailhouse>.
- [7] Tobias Betz, L. Wen, F. Pan, G. Kaljavesi, A. Zuepke, A. Bastoni, M. Caccamo, A. Knoll, and J. Betz. A containerized microservice architecture for a ros 2 autonomous driving software: An end-to-end latency evaluation. In *2024 IEEE 30th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2024.
- [8] Jiang Bian, Abdullah Al Arafat, Haoyi Xiong, Jing Li, Li Li, Hongyang Chen, Jun Wang, Dejing Dou, and Zhishan Guo. Machine learning in real-time internet of things (IoT) systems: A survey. *IEEE Internet of Things Journal*, 9(11):8364–8386, 2022.
- [9] Alessandro Biondi and Giorgio Buttazzo. Timing-aware FPGA partitioning for real-time applications under dynamic partial reconfiguration. In *NASA/ESA Conference on Adaptive Hardware and Systems*, pages 172–179, 2017.
- [10] John Burkardt. The truncated normal distribution. *Department of Scientific Computing Website, Florida State University*, 1:35, 2014.
- [11] Hongpeng Cao, Yanbing Mao, Lui Sha, and Marco Caccamo. Physics-regulated deep reinforcement learning: Invariant embeddings. In *International Conference on Learning Representations (ICLR)*, 2024.
- [12] Carmen Carrión. Kubernetes scheduling: Taxonomy, ongoing issues and challenges. *ACM Computing Surveys*, 55(7):1–37, 2022.
- [13] Szegedy Christian, Vanhoucke Vincent, Sergey Ioffe, Shlens Jon, and Wojna Zbigniew. Rethinking the Inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.
- [14] Lukas Dirnberger, Johannes Roser, Daniele Bernardini, Cristina Piazza, and Marco Caccamo. Dropjects: Large-scale stereo vision dataset for zero-shot learning. *Soft Computing (UNDER SUBMISSION)*, 2024.
- [15] Zheng Dong and Cong Liu. Analysis techniques for supporting hard real-time sporadic gang task systems. In *IEEE Real-Time Systems Symposium (RTSS)*, pages 128–138, 2017.

- [16] Zheng Dong and Cong Liu. Work-in-progress: Non-preemptive scheduling of sporadic gang tasks on multiprocessors. In *IEEE Real-Time Systems Symposium (RTSS)*, pages 512–515, 2019.
- [17] Zheng Dong and Cong Liu. A utilization-based test for non-preemptive gang tasks on multiprocessors. In *IEEE Real-Time Systems Symposium (RTSS)*, pages 105–117, 2022.
- [18] Zheng Dong, Kecheng Yang, Nathan Fisher, and Cong Liu. Tardiness bounds for sporadic gang tasks under preemptive global EDF scheduling. *IEEE Transactions on Parallel Distributed Systems*, 32(12):2867–2879, 2021.
- [19] Farzad Farshchi, Qijing Huang, and Heechul Yun. BRU: bandwidth regulation unit for real-time multicore processors. In *IEEE Real-Time and Embedded Technology and Applications Symposium, RTAS 2020, Sydney, Australia, April 21-24, 2020*, pages 364–375. IEEE, 2020.
- [20] Bosch GmbH. ETAS RTA Hypervisor. <https://www.etas.com/en/products/rta-vrte.php>. Accessed: 2021-02-08.
- [21] SYSGO GmbH. PikeOS Hypervisor. <https://www.sysgo.com>.
- [22] Joël Goossens and Vandy Berten. Gang FTP scheduling of periodic and parallel rigid real-time tasks. *arXiv preprint arXiv:1006.2617*, 2010.
- [23] Joël Goossens and Pascal Richard. Optimal scheduling of periodic gang tasks. *Leibniz Transactions on Embedded Systems*, 3(1):04:1–04:18, 2016.
- [24] Arne Hamann, Dakshina Dasari, Falk Wurst, I Saudo, N Capodiecici, P Burgio, and M Bertogna. WATERS industrial challenge. In *Proceedings of the 10th International Workshop on Analysis Tools and Methodologies for Embedded Real-Time Systems (WATERS)*, 2019.
- [25] Changhun Han, Hoon Sung Chwa, Kilho Lee, and Sangeun Oh. SPET: Transparent sram allocation and model partitioning for real-time DNN tasks on edge TPU. In *ACM/IEEE Design Automation Conference (DAC)*, pages 1–6, 2023.
- [26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [27] Denis Hoornaert, Golsana Ghaemi, Andrea Bastoni, Renato Mancuso, Marco Caccamo, and Giulio Corradi. Mcti: mixed-criticality task-based isolation. *Real-Time Syst.*, 60(2):328–365, July 2024.
- [28] Denis Hoornaert, Julian Pritzi, Andrea Bastoni, and Marco Caccamo. A dynamic priority-aware coherent cache architecture for reactive real-time system. In *Proceedings of the 32nd International Conference on Real-Time Networks and Systems, RTNS 2024, Porto, Portugal, November 6-8, 2024*, pages 142–152. ACM, 2024.
- [29] Intel. Resource Director Technology, 2024.
- [30] Ivan Izhbirdeev, Denis Hoornaert, Weifan Chen, Alexander Zuepke, Y. Hammad, M. Caccamo, and R. Mancuso. Coherence-aided memory bandwidth regulation. In *Proceedings 45rd IEEE Real-Time Systems Symposium (RTSS)*, 2024.

- [31] David S. Johnson. Fast algorithms for bin packing. *Journal of Computer and System Sciences*, 8(3):272–314, 1974.
- [32] S. Kato and Y. Ishikawa. Gang EDF scheduling of parallel task systems. In *IEEE Real-Time Systems Symposium (RTSS)*, pages 459–468, 2009.
- [33] Eugene Kim, Jinkyu Lee, Liang He, Youngmoon Lee, and Kang G. Shin. Offline guarantee and online management of power demand and supply in cyber-physical systems. In *IEEE Real-Time Systems Symposium (RTSS)*, pages 89–98, 2016.
- [34] T. Kloda, M. Solieri, R. Mancuso, N. Capodieci, P. Valente, and M. Bertogna. Deterministic memory hierarchy and virtualization for modern multi-core embedded systems. In *2019 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, page 1–14, 2019.
- [35] Seongtae Lee, Nan Guan, and Jinkyu Lee. Design and timing guarantee for non-preemptive gang scheduling. In *IEEE Real-Time Systems Symposium (RTSS)*, pages 132–144, 2022.
- [36] Seongtae Lee, Seunghoon Lee, and Jinkyu Lee. Response time analysis for real-time global gang scheduling. In *IEEE Real-Time Systems Symposium (RTSS)*, pages 92–104, 2022.
- [37] C. L. Liu and James W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM*, 20(1):46–61, jan 1973.
- [38] Renato Mancuso. Boston University. <http://www.bu.edu/cs/profiles/renato-mancuso/>.
- [39] José Martins, Adriano Tavares, Marco Solieri, Marko Bertogna, and Sandro Pinto. Bao: A Lightweight Static Partitioning Hypervisor for Modern Multi-Core Embedded Systems. In Marko Bertogna and Federico Terraneo, editors, *Workshop on Next Generation Real-Time Embedded Systems (NG-RES 2020)*, volume 77 of *OpenAccess Series in Informatics (OASICS)*, pages 3:1–3:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
- [40] Ioannis Moschakis and Helen Karatza. Evaluation of gang scheduling performance and cost in a cloud computing system. *The Journal of Supercomputing*, 59:975–992, 2010.
- [41] Geoffrey Nelissen, Joan Marcè i Igual, and Mitra Nasri. Response-Time Analysis for Non-Preemptive Periodic Moldable Gang Tasks. In *Euromicro Conference on Real-Time Systems (ECRTS)*, pages 12:1–12:22, 2022.
- [42] Mattia Nicolella, Shahin Roozkhosh, Denis Hoornaert, Andrea Bastoni, and Renato Mancuso. RT-Bench: An extensible benchmark framework for the analysis and management of real-time applications. In *Proceedings of the 30th International Conference on Real-Time Networks and Systems (RTNS)*, pages 184–195, 2022.
- [43] John K. Ousterhout. Scheduling techniques for concurrent systems. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, volume 82, pages 22–30, 1982.
- [44] Matthew O’Kelly, Hongrui Zheng, Dhruv Karthik, and Rahul Mangharam. Fltenth: An open-source evaluation environment for continuous control and reinforcement learning. In *NeurIPS 2019 Competition and Demonstration Track*, pages 77–89. PMLR, 2020.
- [45] Xen Project. Xen 4.20 Cache-Coloring. <https://lists.xen.org/archives/html/xen-devel/2024-12/msg01098.html>.



- [46] Xen Project. Xen Project Public Repository. <http://xenbits.xen.org/gitweb/?p=xen.git>.
- [47] Shahin Roozkhosh, Denis Hoornaert, and Renato Mancuso. CAESAR: Coherence-aided elective and seamless alternative routing via on-chip FPGA. In *Proceedings 43rd IEEE Real-Time Systems Symposium (RTSS)*, 2022.
- [48] Vivek Sarkar. Partitioning and scheduling parallel programs for execution on multiprocessors. 1 1987.
- [49] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [50] Gero Schwaericke, Rohan Tabish, Rodolfo Pellizzoni, Renato Mancuso, Andrea Bastoni, Alexander Zuepke, and Marco Caccamo. A real-time virtio-based framework for predictable inter-VM communication. In *2021 IEEE International Real-Time Systems Symposium (RTSS)*, 2021.
- [51] Kiran Seshadri, Berkin Akin, James Laudon, Ravi Narayanaswami, and Amir Yazdanbakhsh. An evaluation of edge tpu accelerators for convolutional neural networks. In *IEEE International Symposium on Workload Characterization (IISWC)*, pages 79–91, 2022.
- [52] Parul Sohal, Michael Bechtel, Renato Mancuso, Heechul Yun, and Orran Krieger. A closer look at intel resource director technology (rdt). In *Proceedings of the 30th International Conference on Real-Time Networks and Systems, RTNS '22*, page 127–139, New York, NY, USA, 2022. Association for Computing Machinery.
- [53] Binqi Sun, Tomasz Kloda, and Marco Caccamo. Response time analysis for fixed-priority preemptive uniform multiprocessor systems. In *36th Euromicro Conference on Real-Time Systems (ECRTS)*, 2024.
- [54] Binqi Sun, Tomasz Kloda, and Marco Caccamo. Strict partitioning for sporadic rigid gang tasks. In *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 252–264, 2024.
- [55] Binqi Sun, Tomasz Kloda, Jiyang Chen, Cen Lu, and Marco Caccamo. Schedulability analysis of non-preemptive sporadic gang tasks on hardware accelerators. In *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 147–160, 2023.
- [56] Binqi Sun, Tomasz Kloda, Sergio Arribas Garcia, Giovanni Gracioli, and Marco Caccamo. Minimizing cache usage with fixed-priority and earliest deadline first scheduling. *Real-Time Systems*, pages 1–40, 2024.
- [57] Binqi Sun, Tomasz Kloda, Chu-ge Wu, and Marco Caccamo. Partitioned scheduling and parallelism assignment for real-time DNN inference tasks on multi-tpu. In *ACM/IEEE Design Automation Conference (DAC)*, 2024.
- [58] Binqi Sun, Mirco Theile, Ziyuan Qin, Daniele Bernardini, Debayan Roy, Andrea Bastoni, and Marco Caccamo. Edge generation scheduling for DAG tasks using deep reinforcement learning. *IEEE Transactions on Computers*, 73(4):1034–1047, 2024.
- [59] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI Conference on Artificial Intelligence*, 2017.

- [60] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.
- [61] RT-Bench Team. RT-Bench Framework. <https://gitlab.com/rt-bench/rt-bench>.
- [62] Mirco Theile, Harald Bayerlein, Marco Caccamo, and Alberto L Sangiovanni-Vincentelli. Learning to recharge: UAV coverage path planning through deep reinforcement learning. *arXiv preprint arXiv:2309.03157*, 2023.
- [63] Mirco Theile, Harald Bayerlein, Richard Nai, David Gesbert, and Marco Caccamo. UAV path planning using global and local map information with deep reinforcement learning. In *2021 20th International Conference on Advanced Robotics (ICAR)*. IEEE, 2021.
- [64] Mirco Theile, Daniele Bernardini, Raphael Trumpp, Cristina Piazza, Marco Caccamo, and Alberto L Sangiovanni-Vincentelli. Learning to generate all feasible actions. *IEEE Access*, 2024.
- [65] Mirco Theile, Hongpeng Cao, Marco Caccamo, and Alberto L. Sangiovanni-Vincentelli. Equivariant ensembles and regularization for reinforcement learning in map-based path planning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 14164–14171, 2024.
- [66] Mirco Theile, Hongpeng Cao, Marco Caccamo, and Alberto L. Sangiovanni-Vincentelli. Equivariant ensembles and regularization for reinforcement learning in map-based path planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 14164–14171, 2024.
- [67] Raphael Trumpp, Denis Hoornaert, and Marco Caccamo. Residual policy learning for vehicle control of autonomous racing cars. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–6. IEEE, 2023.
- [68] Raphael Trumpp, Ehsan Javanmardi, Jin Nakazato, Manabu Tsukada, and Marco Caccamo. RaceMOP: Mapless online path planning for multi-agent autonomous racing using residual policy learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8449–8456, 2024.
- [69] Niklas Ueter, Mario Günzel, Georg von der Brüggen, and Jian-Jia Chen. Hard real-time stationary gang-scheduling. In *Euromicro Conference on Real-Time Systems (ECRTS)*, pages 10:1–10:19, 2021.
- [70] Unimore. HiPeRT Lab. <https://hipert.unimore.it>.
- [71] Micaela Verucchi, Gianluca Brilli, Davide Sapienza, Mattia Verasani, Marco Arena, Francesco Gatti, Alessandro Capotondi, Roberto Cavicchioli, Marko Bertogna, and Marco Solieri. A systematic assessment of embedded neural networks for object detection. In *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 937–944, 2020.
- [72] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems (NeurIPS)*, 34:28877–28888, 2021.

- [73] H. Yun, R. Mancuso, Z. P. Wu, and R. Pellizzoni. PALLOC: DRAM bank-aware memory allocator for performance isolation on multicore platforms. In *2014 IEEE 19th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, page 155–166, 2014.
- [74] H. Yun, G. Yao, R. Pellizzoni, M. Caccamo, and L. Sha. Memory bandwidth management for efficient performance isolation in multi-core platforms. *IEEE Transactions on Computers*, 65(2):562–576, 2016.
- [75] Heechul Yun, Waqar Ali, Santosh Gondi, and Siddhartha Biswas. BWLOCK: A dynamic memory access control framework for soft real-time applications on multicore platforms. *IEEE Trans. Computers*, 66(7):1247–1252, 2017.
- [76] Y. Zhang, H. Franke, J.E. Moreira, and A. Sivasubramaniam. Improving parallel job scheduling by combining gang scheduling and backfilling techniques. In *International Parallel and Distributed Processing Symposium (IPDPS)*, pages 133–142, 2000.
- [77] Yanqi Zhou and David Wentzlaff. MITTS: memory inter-arrival time traffic shaping. In *43rd ACM/IEEE Annual International Symposium on Computer Architecture, ISCA 2016, Seoul, South Korea, June 18-22, 2016*, pages 532–544. IEEE Computer Society, 2016.
- [78] Bohua Zou, Binqi Sun, Yigong Hu, Tomasz Kloda, Marco Caccamo, and Tarek Abdelzaher. A performance prediction-based DNN partitioner for Edge TPU pipelining. In *IEEE Military Communications Conference (MILCOM)*, pages 1–6, 2024.
- [79] Alexander Zuepke, Andrea Bastoni, Weifan Chen, Marco Caccamo, and Renato Mancuso. MemPol: Policing Core Memory Bandwidth from Outside of the Cores. In *29th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2023.
- [80] Alexander Zuepke, Andrea Bastoni, Weifan Chen, Marco Caccamo, and Renato Mancuso. MemPol: Polling-based Microsecond-scale Per-core Memory Bandwidth Regulation. *Real-Time Systems*, 60, 2024.