International Journal on Advances in Intelligent Systems

2025 vol. 18 nr. 1&2

The *International Journal on Advances in Intelligent Systems* is Published by IARIA.
ISSN: 1942-2679
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Intelligent Systems
Volume 18, Number 1 & 2, 2025

International Journal on Advances in Intelligent Systems
Volume 18, Numbers 1 & 2, 2025

CONTENTS

pages: 1 - 10
The Smart Highway to Babel: the Coexistence of Different Generations of Intelligent Transport Systems and Conventional Drivers
Oscar Amador Molina, Halmstad University, Sweden
Felipe Valle, Halmstad University, Sweden
Nicholas Sjögren, Halmstad University, Sweden
Duc Huy Vu, Halmstad University, Sweden
María Calderón, Universidad Politécnica de Madrid, Spain
Ignacio Soto, Universidad Politécnica de Madrid, Spain
Manuel Urueña, Universidad Internacional de La Rioja, Spain

pages: 11 - 21
Exploring Deep Learning Techniques for Artist and Style Recognition in the Paintings-100 Dataset
Erica M. Knizhnik, Lake Forest College, United States
Brian Rivera, Lake Forest College, United States
Atreyee Sinha, Edgewood College, United States
Sugata Banerji, Lake Forest College, United States

pages: 22 - 31
A Lego-inspired, Modular Agent-Based Metasystem for Emergency Departments
Francisco Mesas Cervilla, Escuelas Universitarias Gimbernat (EUG), Computer Science School, Universitat Autonoma de Barcelona, Sant Cugat del Valles, Barcelona, Spain, Spain
Manel Taboada, Escuelas Universitarias Gimbernat (EUG), Computer Science School, Universitat Autonoma de Barcelona, Sant Cugat del Valles, Barcelona, Spain, Spain
Dolores Isabel Rexachs del Rosario, Computer Architecture and Operating System Department, Universitat Autonoma de Barcelona, Barcelona, Spain, Spain
Francisco Epelde Gonzalo, Consultant Internal Medicine, University Hospital Parc Tauli, Universitat Autonoma de Barcelona Sabadell, Barcelona, Spain, Spain
Alvaro Wong, Computer Architecture and Operating System Department, Universitat Autonoma de Barcelona, Barcelona, Spain, Spain
Emilio Luque, Computer Architecture and Operating System Department, Universitat Autonoma de Barcelona, Barcelona, Spain, Spain

pages: 32 - 45
Stability of Dynamic Fluid Transport Simulations with Mixing Flows
Mehrnaz Anvari, Fraunhofer Institute for Algorithms and Scientific Computing, Germany
Anton Baldin, PLEdoc GmbH and Fraunhofer Institute for Algorithms and Scientific Computing, Germany
Tanja Clees, University of Applied Sciences Bonn-Rhein-Sieg and Fraunhofer Institute SCAI, Germany
Bernhard Klaassen, Fraunhofer Research Institution for Energy Infrastructures, Germany
Igor Nikitin, Fraunhofer Institute for Algorithms and Scientific Computing, Germany
Lialia Nikitina, Fraunhofer Institute for Algorithms and Scientific Computing, Germany

pages: 46 - 56
Maneuver-Based Decision Making in Autonomous Driving via Reinforcement Learning and Simulation-to-Real Transfer
Xin Xing, Ostfalia University of Applied Sciences, Germany

Morteza Molaei, Ostfalia University of Applied Sciences, Germany
Sebastian Ohl, Ostfalia University of Applied Sciences, Germany

pages: 57 - 67
Optimizing Urban Intersections: Harnessing Visible Light Communication for Smart Traffic Management
Manuel Vieira, ISEL, CTS-UNINOVA and LASI, Portugal
Manuela Vieira, ISEL, CTS-UNINOVA ,LASI and NOVA School of Science and Technology, Portugal
Gonçalo Galvão, ISEL and NOVA School of Science and Technology, Portugal
Paula Louro, ISEL, CTS-UNINOVA and LASI, Portugal
Mario Véstias, ISEL/IPL-INESC-INOV, Portugal
Pedro Vieira, ISEL, and Instituto das Telecomunicações IST, Portugal

pages: 68 - 78
Combinatory Logic as a Model for Intelligent Systems Based on Explainable AI
Thomas Fehlmann, Euro Project Office AG, Switzerland
Eberhard Kranich, Euro Project Office, Germany

The Smart Highway to Babel: the Coexistence of Different Generations of Intelligent Transport Systems and Conventional Drivers

Oscar Amador Molina
School of Information Technology
Halmstad University
Halmstad, Sweden
email: oscar.molina@hh.se

Felipe Valle
School of Information Technology
Halmstad University
Halmstad, Sweden
email: felipe.valle@hh.se

Nicholas Sjögren
School of Information Technology
Halmstad University
Halmstad, Sweden
email: nicsjo21@student.hh.se

Duc Huy Vu
School of Information Technology
Halmstad University
Halmstad, Sweden
email: ducvu21@student.hh.se

María Calderón
Depto. de Ing. de Sistemas Telemáticos
ETSI Telecomunicación
Universidad Politécnica de Madrid
Madrid, Spain
email: maria.calderon@upm.es

Ignacio Soto
Depto. de Ing. de Sistemas Telemáticos
ETSI Telecomunicación
Universidad Politécnica de Madrid
Madrid, Spain
email: ignacio.soto@upm.es

Manuel Urueña
Escuela Superior de Ingenieros y Tecnología
Universidad Internacional de la Rioja
Logroño, Spain
email: manuel.uruena@unir.net

***Abstract*—The gap between technology readiness level in Cooperative Intelligent Transport Systems (C-ITS) and its adoption and deployment has caused a phenomenon where conventional drivers have to coexist with intelligent vehicles. Furthermore, intelligent vehicles are also a heterogeneous fleet of cars with different levels of connection and automation. For the connection part, at least two types of network access technologies have to coexist. Furthermore, for the case of the European Telecommunications Standards Institute (ETSI) Intelligent Transport Systems protocols, work is being completed in Release 2 of the specification while Release 1 deployments are still underway. In the automation side, levels of automation differ from Levels 1–3, when human drivers are needed at least for backup driving tasks, to Levels 4–5, where full automation is in place. This, coupled with industry and consumer trends in the vehicle industry, is bound to cause a scenario where fully C-ITS-enabled vehicles have to coexist with *non*-C-ITS road users and, at the very least, with different versions of C-ITS. In this paper, we analyze this phenomena from the connection side by performance in terms of efficiency and safety of two releases of the ETSI GeoNetworking protocol, and from the automation side, by assessing the coexistence of conventional drivers with fully intelligent vehicles. Our results show that it is partial homogeneity (coexistence of two types of vehicles) that affects safety and efficiency. Finally, we discuss possible paths to tackle the upcoming compatibility and coexistence problems.**

Index Terms*—*Cooperative Connected and Automated Driving*; *Coexistence*; *Contention Based Forwarding*; *ETSI

I. Introduction

The problem of coexistence between different generations of intelligent vehicles is arriving at least in the form of different releases of Layer 3 protocols [1]. This problem might hinder the use of C-ITS to maximize road safety and traffic efficiency, which has been one of the cornerstones upon which future mobility is built. The final stage of Cooperative, Connected and Automated Mobility (CCAM) depends on the presence of C-ITS on all roads and at all times, exchanging information and coordinating their maneuvers [2]. Thus, having intelligent vehicles that cannot talk to each other prevents them, by definition, from cooperating.

The road to CCAM is divided in three different fronts: *connection* (the ability to exchange information through networks), *cooperation* (the protocols that define how intelligent vehicles react to information and each other's actions), and *automation* (the level of human intervention on the driving task). These fronts have particular stages (e.g., levels of automation [3]), but they share common stages, such as the *Days* in Vision Zero [2]. These Days (1–4) are incremental steps toward the realization of full CCAM:

- on Day 1, *awareness* starts, and vehicles share their status using messages like Cooperative Awareness Message

(CAM) and Decentralized Environmental Notification Message (DENM) (i.e., in the framework established by the ETSI);
- on Day 2, *cooperation* starts, and vehicles exchange information from their sensors using, e.g., Collective Perception Messages (CPMs);
- on Day 3, road users communicate their intentions; and
- on Day 4, road users execute coordinated maneuvers.

These days take into account the evolution of technology. For example, in the *connection* front, Day 1 considers the use of Vehicular ad hoc Networks (VANETs) supported on cellular communications (i.e., LTE) or in WiFi (e.g., ETSI ITS-G5, based on IEEE 802.11p). From Day 2 onward, C-ITSs expect the use of evolved technologies (e.g., 5G, 802.11bd, and technologies beyond these two). The choice between cellular or WiFi is the first hurdle towards the harmonic coexistence of different types of intelligent vehicles, and ETSI develops media-dependent protocols for both approaches [4], [5]. Thus, manufacturers and transportation authorities are given the chance to select one or many technologies.

However, industry and consumer patterns are likely to cause a scenario where vehicles that are produced in 2023, with the technological features present this year, will share the road with fully CCAM-enabled vehicles in 2050 [6]. Even now, figures from the industry show that the average age for a vehicle in Europe ranges from 12 to 14.7 years for cars and trucks, respectively, and some countries have even larger mean values [7]. This means that is highly likely to have a fleet with 1) different technological capabilities, and 2) different versions of the same technology.

In this paper, we present the effect of the coexistence of different levels of intelligence in vehicles. First, we assess the coexistece of two versions of one safety-critical protocol: Release 1 of ETSI Contention-Based Forwarding (CBF) [8], and the changes proposed to Release 2, which were originally presented in [9] and [10]. We evaluate efficiency metrics such as the number of transmissions and its variation with larger penetration rates of the newer protocol in scenarios where a message has to be distributed within a Destination Area. Secondly, we evaluate the coexistence of conventional drivers and automated vehicles in a suburban setting. Here, our metrics are safety (number of collisions and emergency breaking events) and efficiency (the effect on the traffic flow). Finally, we discuss the likely scenarios for coexistence and possible compatibility between two versions of one protocol.

The rest of the paper is organized as follows: in Section II, we present the two releases of the ETSI CBF protocol; in Section III we explain briefly the levels of automation expected for Future Mobility; in Section IV, we perform an experimental assessment of the penetration rate of the updated CBF protocol on effectiveness and efficiency; Section V presents a study on the coexistence of different combinations of conventional and highly automated road users; Section VI presents a discussion on scenarios and alternatives to palliate the problem of having a mixed fleet; and finally, conclusions and future work are presented in Section VII.

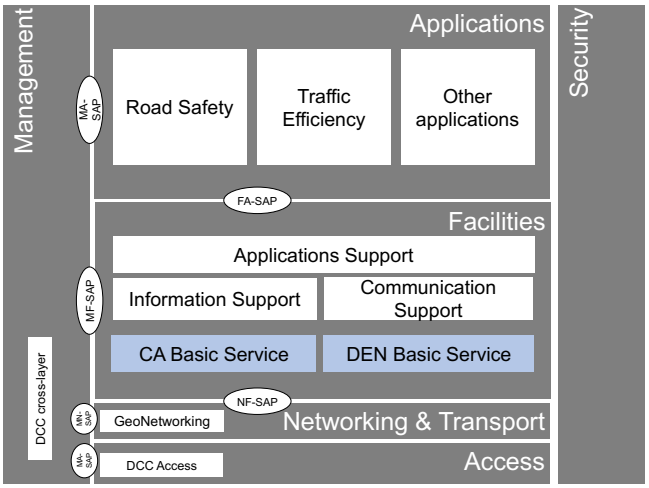


Fig. 1. ETSI ITS Architecture.

II. Connected Vehicles

A. ETSI ITS Architecture

Fig. 1 shows the layers and entities of the ETSI ITS architecture. At the very top, the Application layer hosts systems that pursue the goals of all C-ITSs — road safety and traffic efficiency — as well as other functions (e.g., related to infotainment). These applications are supported by the Facilities layer, e.g., by safety-critical Day 1 services like the Cooperative Awareness (CA) and Decentralized Environmental Notification (DEN) basic services. These services exchange messages with other nodes (vehicles and the infrastructure) that allow applications fulfill their roles: for example, a DENM warns road users about roadworks ahead of the road, and an application can suggest or take a new route.

Messages are generated by services at the Facilities layer and then get sent down the stack to the Networking & Transport layer. Depending on the use case and requirements from applications, a message can be broadcast to neighbors one hop away (i.e., Single-Hop Broadcasting (SHB)), or towards a specific area of interest (Destination Area). The latter is achieved through GeoNetworking [8]. In either case, packets are encapsulated and sent down to the Access layer for transmission.

The Access layer executes Medium Access Control as well as Congestion Control functions. This layer accommodates both WiFi-based and cellular-based access technologies. For the case of WiFi-based access (i.e., ETSI ITS-G5), channel occupation (i.e., Channel Busy Ratio (CBR)) is measured at this layer and, using Decentralized Congestion Control (DCC) [11], each station calculates the share of the medium it can use, which ranges from 0.06% to 3% of the medium, or a message rate between 1 and 40 Hz. This means that, even in extremely low congestion conditions, consecutive messages must wait in the DCC queues for at least 25 ms between each dequeuing. From these queues, frames are then sent to the

Enhanced Distributed Channel Access (EDCA) queues where they wait for their time to contend for access to the medium.

The road a message takes from generation to transmission and the possible bottleneck or sinkhole effects that different phenomena, e.g., at the Access layer, can have on protocol performance is accounted for by ETSI protocols. E.g., a CAM can only be generated if the message rate is less or equal to the one allowed by DCC. However, the appearance of new services and the expected effect of having a high number of nodes in proximity of each other has prompted the research community to study these effects continuously [12].

B. GeoNetworking in ETSI ITS

Routing protocols in conventional computer networks rely on Layer 3 addresses to send data between hosts in remote locations. This is typically achieved through IP addressing. In the context of VANETs, where use cases sometimes require the dissemination of information to a given area, geographical awareness is required for a routing protocol. Hence, GeoNetworking functionalities are included, e.g., in the Networking & Transport layer of the ETSI ITS protocol stack [8]. GeoNetworking allows for messages to reach a Destination Area without the need of maintaining a record of the network addresses of nodes in that area, which would be difficult due to the dynamic nature of vehicular networks.

ETSI defines mechanisms to broadcast information to a geographical Destination Area when:

- the source is outside the Destination Area and the message has to arrive in it (e.g., using Greedy Forwarding or CBF); or
- the message originates from or arrives into the Destination Area and is disseminated using CBF or Simple Forwarding.

Non-Area mechanisms are out of the scope of this paper, but we can summarize Greedy Forwarding as a mechanism where each hop selects its farthest known neighbor and determines it as the next hop toward the Destination Area. These type of mechanisms have been widely studied, and the ETSI-defined version of Greedy Forwarding is evaluated in-depth in [13] and [14].

Regarding Area forwarding mechanisms, Simple Forwarding can be described as a brute-force mechanism where every node that receives a message forwards it immediately (i.e., simple flooding). CBF, on the other hand, makes receivers start a *contention* timer that is proportional to their distance from the last hop before they decide to forward the message. If they listen to a forwarding while they are waiting for their timer to expire, they cancel the timer and drop their copy of the packet.

1) Inefficiencies in Release 1 of ETSI CBF: Efforts from the research community have evaluated the performance of ETSI CBF. While the theoretical frame which supports CBF makes it more optimal than, e.g., simple forwarding, the way it interacts with other layers in the ETSI ITS architecture causes phenomena that affect its efficiency.

The interaction between ETSI CBF and the DCC mechanism at the Access layer causes an undesired effect: even if the CBF timer expires, and the decision to forward the packet is made, if there is congestion in the channel or if another packet has just been transmitted, the forwarding is stopped at the DCC queues (for ETSI ITS-G5) or the scheduler (for C-V2X). This means that the actual transmission may not occur when CBF has decided, and this phenomenon can occur in any station, so even if a copy of the message is received during contention, it is not guaranteed that it comes from an optimal forwarder. Furthermore, Release 1 of ETSI GeoNetworking relies Duplicate Packet Detection (DPD) to CBF, so, if a backlogged message from a DCC-affected forwarder is received at a neighbor which had already forwarded or even cancelled its copy will enter the loop once again.

2) ETSI CBF Release 2: The issues with DPD and the effect of DCC on Release 1 for ETSI CBF had been studied widely in the literature [13], [15], [16]. Yet, it was the work in [9] and [10] that was presented to ETSI as a change request that was iterated and matured before it reached the necessary consensus to be Release 2 of ETSI CBF.

The differences in Release 2 of Area CBF are:

- The inclusion of DPD inside the CBF algorithm.
- Interfacing with the cross-layer DCC mechanism to offer awareness of the time before DCC allows the next transmission, and account for it when calculating the contention timer (optional for cellular-based communications).
- A procedure to determine if a copy received during contention actually comes from a better forwarder.
- An updated timer formula to account for receptions beyond the maximum expected distance.

However, since Release 2 services might have different requirements and characteristics, it is not clear if Release 1 nodes will be able to receive messages originating from Release 2 nodes, even for safety-critical applications. If this is the case, and nodes executing Release 2 of ETSI CBF coexist with nodes executing Release 1, there might be effects on awareness and efficiency metrics. In Section IV, we evaluate these effects in Area CBF in a highway scenario.

III. Automated Vehicles

The Dynamic Driving Task (DDT), whether it is performed by a conventional driver or an automated vehicle, refers to the to operational and tactical functions. Operational functions consist of motion (e.g., steering and brake/throttle control, for lateral and longitudinal motion respectively). Tactical functions relate to Object and Event Detection and Reaction (OEDR), e.g., following a route, avoiding obstacles and risks. With automation, several sub-tasks of the DDT are offloaded to the automated driving system (ADS). The level of automation of an ADS is defined by the tasks it can offload from human operators [3]:

- Level 1: the ADS controls **either** longitudinal **or** lateral motion. The rest of the DDT remains under control of the driver in the vehicle.
- Level 2: the ADS controls **both** longitudinal **and** lateral motion. The rest of the DDT remains under control of the driver in the vehicle.
- Level 3: the ADS controls all of the DDT within an Operational Domain (e.g., only in highways). The driver in the vehicle is required for fallback (e.g., when the ADS fails to solve a problem).
- Level 4: the ADS controls all of the DDT and only requires human input to solve situations it cannot handle by itself (e.g., tactical operations).
- Level 5: the ADS controls all of the DDT and only asks for authorization from a human operator to perform tactical operations.

In theory, Levels 4–5 do not need a driver to be present in the vehicle, and only require humans to control the vehicle's motion. These operators can potentially be away from the vehicle [17]. This implies that, even from Level 3, the driving style of a vehicle depends also on how they are programmed to react in a given scenario. Just as human drivers have different *driving styles* [18] depending on their goals, automated vehicles can potentially be programmed to become more or less compliant [19], thus, to be more competitive or more cooperating.

This means that, even if vehicles are fully connected and automated and can understand each other, the way they react to events can vary greatly even within the CCAM fleet. Just as the problem of coexistence between solely conventional vehicles with drivers of different styles, this might become an issue with different penetration rates, heterogeneous levels of automation, and varying responses to similar stimuli.

In Section V, we assess the coexistence of conventional drivers and full-CCAM vehicles. In order to perform a better analysis, the CCAM fleet is homogeneous within itself. In Section VI, we discuss in more detail what implications CCAM fleet heterogeneity has in the ability for agents to cooperate.

IV. Experimental Evaluation of Coexistent Releases of Network Layer Protocols

A. Simulation Scenario

We evaluate the effect of different ratios of nodes executing Releases 1 and 2 of ETSI CBF in a highway scenario where a vehicle is stationary on the shoulder of a road. It starts sending DENMs [20] at 1 Hz with a Destination Area covering 4 km of a road with 4 lanes per direction. The vehicular density is 30 veh/km on each lane. We take measurements for 30 s after a warm-up period of 120 s. We evaluate:

1) **Packet-delivery Ratio (PDR):** the number of successful individual receptions of a message in the Destination Area divided by the total number of vehicles in the area at the time of DENM generation.
2) **Number of transmissions:** how many transmissions (i.e., from the source and forwarders) have occurred.

Our toolkit consists of the OMNET++-based simulator Artery [21], which implements the ETSI ITS protocol stack using Vanetza and Veins [22] for the physical model of ETSI ITS-G5. Mobility is controlled by SUMO [23]. A set of vehicles execute ETSI CBFRelease 1 [8], and an increasing number of vehicles (see the penetration rate parameter) execute the improvements included in Release 2 as described in [10]. In our set-up, and due to the nature of the message (i.e., Road Hazard Warning (RHW)), we consider Release 2 and Release 1 messages to be mutually understandable. The rest of the parameters are specified in Table I.

TABLE I
Simulation Parameters: Heterogeneous Connectivity

Parameter	Values
Access Layer protocol	ITS-G5 (IEEE 802.11p)
Channel bandwidth	10 MHz at 5.9 GHz
Data rate	6 Mbit/s
DCC	ETSI Adaptive DCC
Transmit power	20 mW
Path loss model	Two-Ray interference model [24]
Maximum transmission range	1500 m
CAM packet size	285 bytes
CAM generation frequency	1–10 Hz (ETSI CAM [25])
CAM Traffic Class	TC2
DENM packet size	301 bytes
DENM Traffic Class	TC0 (Source) and TC3 (Forwarders)
DENM lifetime	10 s
DPL size	32 packet identifiers per Source
Default Hop Limit	10
Rel. 2 penetration rate	0, 25, 50, 75, 100%
SUMO car-following model	Krauss (default settings)

B. Results

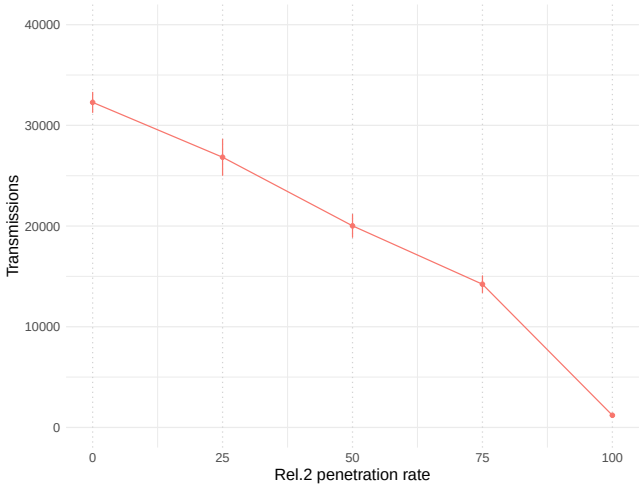


Fig. 2. Number of transmissions in different Release 2 penetration rates.

Fig. 2 shows the effect of even a minority portion of nodes executing a non-optimized protocol. There is beyond an order of magnitude in executed transmissions between the 0% and the 25% penetration rate for Release 2. From there, there is a linear increase until the almost 30:1 ratio between Releases 1 and 2 in line with the results in [9] and [10].

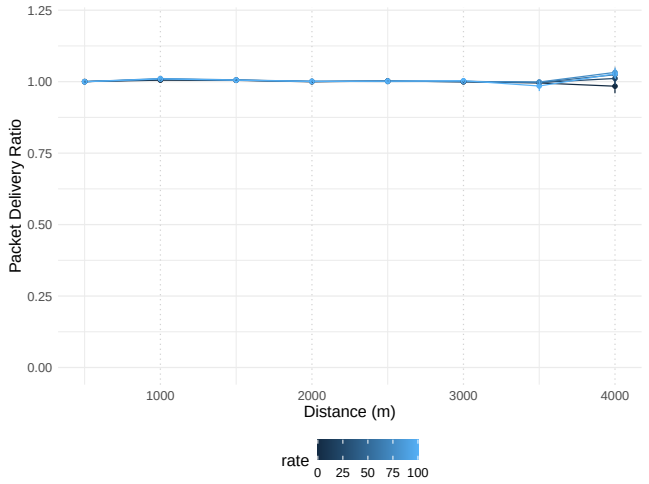


Fig. 3. Packet Delivery Ratio for different Release 2 penetration rates.

However, this issue is not reflected in awareness. Fig. 3 shows the PDR over the distance in the 4 km-long Destination Area. Lines overlap for most of the distance, up to the last segment where they fan out in favor of higher penetration rates. However, this phenomenon is due to an unbalance in the turnover rate (i.e., the ratio between vehicles entering and exiting the Destination Area after the DENM was generated). These extra vehicles are accounted for since the message is still within validity, and it is relevant to newcomers into the Destination Area.

The main takeaway of this experiment is that, as long as Releases 1 and 2 GeoNetworking messages are mutually intelligible, there is an effect on efficiency but not in safety (for the case of multi-hop DENMs from a single source). However, inefficient forwarding will occupy the medium with unnecessary repetitions of messages. Thus, in scenarios where there is more than one source trying to disseminate safety-critical messages, unnecessary transmissions are bound to cause collisions or, at least, to block access to the medium for more necessary messages waiting to be forwarded. Further work needs to be performed on how non-mutually intelligible messages affect performance, since Release 1 is likely to reach higher PDR using brute force, while Release 2 will either yield access to the medium or might find a path to transmit immediately. What is sure is that, in that scenario, safety will be compromised.

V. Experimental Evaluation of Coexistece between Conventional and Automated Road Users

A. SUMO simulation scenario

We use a map of the City of Halmstad, in Sweden. Fig. 4 shows the road section we chose — roundabout in a suburban section of the town where pedestrians, cyclists, buses, and passenger vehicles have to interact with each other. The full description of the map and its characteristics can be found in [26].

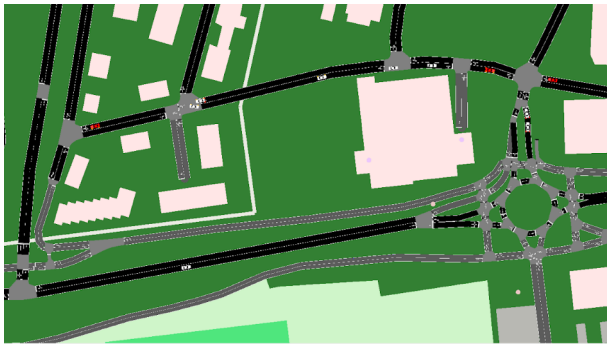


Fig. 4. SUMO map of the suburban road in Halmstad

The challenge to simulate different types of road users in SUMO comes from the very nature of the simulator — vehicles follow a modelled behavior within certain limits defined by parameters. One of these SUMO parameters, for example, is *driver imperfection*. This parameter that goes from 0–1, defines how a vehicle will act to keep speed limits, minimum distances, acceleration and deceleration. By default, it is set to 0.5, while a value of 0 indicates a perfect driver. Thus, one can assume that a fully automated vehicle would have a perfect driver, which is the assumption we follow in one part of our experiments and also in [26].

However, the literature offers following models that emulate the behavior of cooperative intelligent vehicles. The authors in [27] have implemented a car-following model that identifies different phases in driving and relies on experimental data. This offers an opportunity to test out how the SUMO default model compares to a data-driven model based on cooperative vehicles.

Thus, our simulation will allow us to 1) compare the models to simulate CCAM in SUMO, and 2) to measure the effect of intra- and inter-model heterogeneity on mobility metrics, which are:

1) **Road Safety:** collisions, emergency braking events.
2) **Traffic Efficiency:** average speed, time loss.

The rest of the simulation parameters are included in Table II.

TABLE II
Simulation Parameters: Conventional and Automated Driving

Parameter	Values
Number of vehicles	200
Vehicle flow	Poisson, average 0.2
Number of pedestrians	29
Pedestrian flow	0.01
Runs (time)	100 (3000 s)
Model for Automated Driving	Krauss (perfect driver), CACC
Model for Conventional Driving	Krauss (0.5 imperfection)
Automation penetration rate	0, 25, 50, 75, 100%

B. Results

This set of experiments offer a chance of measuring two types of heterogeneity: intra-model (when we use perfect and

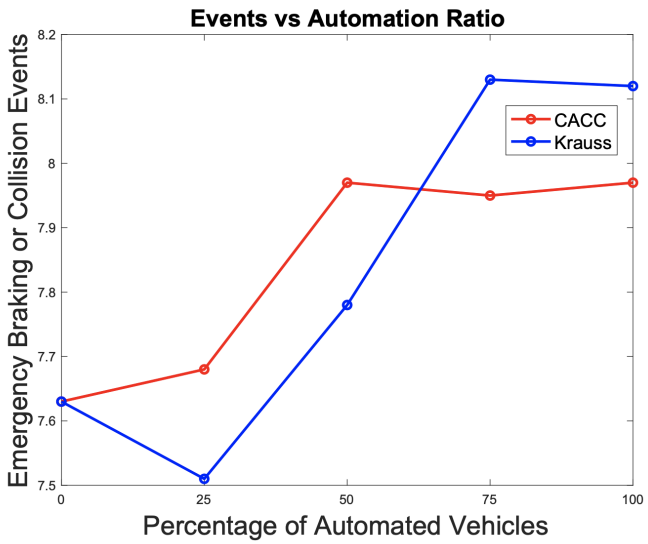


Fig. 5. Safety Events (emergency braking, collisions) for different penetration rates

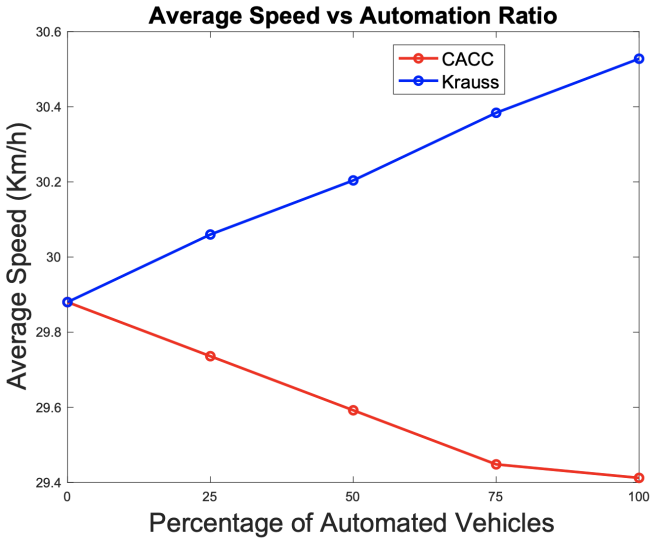


Fig. 6. Average speed for different penetration rates

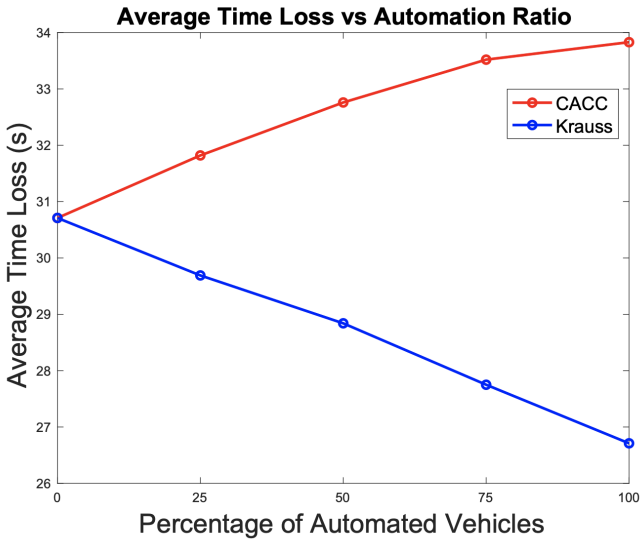


Fig. 7. Average time loss for different penetration rates

imperfect drivers within the Krauss model), and inter-model (when we use the CACC model from [27] and the Krauss model with default settings for automated and conventional drivers respectively). We denote these two measurements as Krauss and CACC in our figures.

Fig. 5 shows the average of the sum of emergency braking and collision events. There is a zone of instability when the penetration rate increases. We attribute this to previous findings in the literature [19], [28]: it is not only the proportion of intelligent (or compliant) vehicles that affects safety, but also the order in which they are placed in the flow. Even one single non-compliant driver can affect a large number of more-compliant, more-cautious drivers and push them into emergency braking events or even into a collision. The effect is however, different in the inter-model comparison. with the mixes showing different types of variations and Krauss increasing linearly in the region between 25 and 75% of automation penetration.

In terms of efficiency, Figs. 6 and 7 show the effect of increasing the number of perfect drivers (for Krauss) and CCAM vehicles (for CACC). In here, the effect of inter-model homogeneity is clearly visible. To help us situate, the Krauss line shows vehicles following the same strategy, and increasing the proportion of vehicles that are able to stick to that strategy increases efficiency (higher speeds, less time losses). On the other hand, the CACC line shows how partially-compliant vehicles affect the efficiency of the whole C-ITS since highly-compliant vehicles are more cautious by definition (i.e., they aim to complete a trip as safely as possible, even if there is a trade-off in time consumed). The difference, however, between a fully-perfect driver fleet and a fully cooperative fleet is of slightly over 1 km/h.

In conclusion, we consider that 1) heterogeneity will affect C-ITSs and that the problem will only get worse in reality since both Cs and the A in CCAM can potentially represent different generations of communication technologies, diverging cooperation strategies, and varying automation levels; and 2) that inter-model heterogeneity is a better representation of these diverging strategies and approaches to the DDT. However, further work needs to be done in order to represent full CCAM, i.e., by being able to represent different levels of compliance, and how compliance can be hindered by unreliable communications.

VI. Discussion

In this section, we discuss the upcoming challenges to Future Mobility stemming from fleet heterogeneity. First, we present a study of how the coexistence of two different releases of a protocol, one being an incremental improvement of the

other, affects efficiency. For our case, packets were compatible, and Release 1 nodes could understand Release 2 messages and vice versa. However, the road to full CCAM is long, and this might not be the case even in the near future. Secondly, we present a discussion on the actual possibility for full cooperation in highly-automated scenarios.

A. The upcoming Tower of Babel

Vehicles equipped with ETSI ITS nodes are already on the road communicating with large deployments. Just in the first three quarters of 2023, more than 250,000 C-ITS-equipped Volkswagen ID. cars were delivered [29]. These cars can communicate with each other, with other ETSI ITS-compatible vehicles, and with current deployments such as the one covering the entire Austrian motorway network [30].

However, these vehicles and deployments all use Release 1 services. While some Release 2 features are software-dependant, e.g., new services such as the Vulnerable Road User awareness (VA) basic service, and can be installed during car services or using over-the-air updates, some others will likely require a deeper update (e.g., compatibility with Multi-channel Operation (MCO)).

While backwards-compatibility is a common issue in computer networks, the characteristics of the vehicular market and industry make it especially more difficult. This is one of the first cases where a massive number of *legacy* nodes will likely share spaces with nodes up to 20 years more modern [7]. This will create a scenario where pockets of segregated nodes are bound to destabilize the system, at the very least make it more inefficient, while compromising efficacy and safety.

1) Past experiences with backwards compatibility: One example of backwards compatibility is the jump between Transport Layer Security (TLS) 1.2 and 1.3. The 1.3 version was released in RFC 8446 in August 2018 [31]. Its benefits over past versions have been widely studied [32], but there are known examples of problems with its adoption [33].

The main problem with TLS 1.3 is *protocol ossification*. This phenomenon occurs when deployed equipment (e.g., middleboxes) does not recognize new protocols or even extensions to known protocols that were released after they were installed. This causes them to interrupt packets that are valid, but unrecognizable for the middlebox.

The solution for TLS 1.3, and for other examples of ossification, was to encapsulate new messages so that the *wire image* of the packets is acceptable for older middleboxes. This could be a path to follow with safety-critical messages exchanged by nodes executing different releases of ETSI ITS.

At the Access layer, 802.11p (upon which ETSI ITS-G5 is based) and its evolution 802.11bd are somewhat compatible. One of the main differences between 802.11bd and 802.11p is the channel bandwidth — 20 MHz up from 11p's 10 MHz. However, 11bd can also work in 10 MHz, and does so if it detects nodes using only 10 MHz, thus, falling back into 11p when needed. However, this approach might not be efficient in Future Mobility scenarios, when 11p's channel capacity might not be able to accommodate the myriad of applications that will try to use the medium.

The foreseeable scenario if nodes cannot process packets from newer releases (i.e., if Release 1 nodes cannot handle Release 2 GeoNetworking traffic) can cause a disruption in Non-Area GeoNetworking [8] if Greedy Forwarding is used. Since it is likely that beacons (e.g., CAMs) will always be compatible, a Release 2 node can select a Release 1 neighbor as the next hop for a message. The next hop will not process the message, and thus it will not reach the Destination Area, since there are no fallback nodes in ETSI Greedy Forwarding. This phenomenon can be avoided, for example, using CBF, where multiple nodes become the next hop and contend to forward the message, increasing the chances of nodes from both releases hearing the forwarded message. Further work will address the impact of this phenomenon on Non-Area forwarding.

2) Nodes with different technologies: In the network side, even at Day 1, there is an identified risk of *non-interoperability* [34]. Since ETSI ITS is media-independent, it does not mandate that one access technology shall be used. Thus, there are vehicles and road-side equipment that use, e.g., LTE or 802.11p. ETSI recognizes the scenario and proposes co-existance methods [35] where, for example, vehicles using different technologies share the time domain. This means that cellular-based nodes occupy the C-ITS band for a fraction of the time and WiFi-based nodes use it for the complement. This, however, is not full inter-operability, since nodes using different access technologies will not "listen" to each other, and this approach compromises every metric: efficiency (diminishing the amount of resources), efficacy (messages are not delivered to all connected road users), and thus, safety.

Further work has to be performed within the research and industry communities to 1) determine whether WiFi and cellular can possibly inter-operate, and 2) whether inter-operability is possible, search for a path to evolve in a way that newer versions of access technologies account for older nodes. One possible approach is to adopt approaches such as Software-Defined Radio (SDR), which would allow equipment to be updated over the air as long as hardware supports newer features, such as different modulation and coding schemes.

This phenomenon will be aggravated when technologies from different Days coexist. For example, a *legacy* node that cannot interpret or even receive intention-sharing or maneuver-coordination message exchanges will likely affect the way CCAM-enabled vehicles converge to a solution. Once again, this will affect traffic efficiency and might hinder road safety. Further work is being performed to assess the effect of a mixed fleet in the optimal performance in CCAM.

For the specific phenomenon in this work, the differences between ETSI CBF Releases 1 and 2 are purely software-based. There is no need for extra fields in the headers, or new values in the existing fields. The main differences come in what the algorithm does with information it already used, namely, the position vector from the last hop and the source. It also uses an existing interface to the Management

entity to consult the cross-layer DCC mechanism and account for transmission rate control information when calculating a contention timer (although this feature is optional).

We foresee two simple solutions: 1) existing equipment that is able to receive an update adopts Release 2, or 2) Release 2 GeoNetworking messages are encapsulated as Release 1, as was the case for TLS 1.3. Both approaches will ensure safety in given scenarios, but approach 1 guarantees more efficiency, and thus, more availability of resources for other applications.

B. (Non) Cooperative Connected and Automated Mobility

Cooperation, and cooperative games specifically, is widely studied in the context of autonomous systems. Coming from mathematics and with applications in social analysis, game theory is divided into two categories: cooperative and non-cooperative games. The difference between these two resides in the ability of agents to exchange goals and tactics [36]. Therefore, communication is paramount, since cooperation relies in the quality and availability of information about other agents. In vehicular communications, this is reflected on how Day 4 (coordination) is an incremental step over Day 3 (intention sharing).

However, even if communications are 100% reliable (likely at the expense of efficiency, as we confirmed in our experiments), there will be differences in reactions to events not only between human-driven and automated vehicles, but likely among the fully automated fleet. Even if the final goal of all C-ITSs is to ensure road safety and traffic efficiency, the strategies and tactics might differ depending on human-influenced factors, not only humans-in-the-loop, but also the influence of decision makers in the design of ADSs.

In a parallel way on how humans have different driving styles [18], brands also have *personalities* that are reflected in marketing and positioning strategies. In conventional driving, manufacturers have models that drive into different styles and demographics, which are correlated according to the work in [18]. A vehicle that is designed to trigger thrilling emotions will likely allow a driver to operate at the edge of the rules (e.g., speed limits). On the other hand, a vehicle that is designed to tap into the demographic seeking for safety and control will trade-off power for stability. Brands, consequently, try to appeal to these different personalities at every stage of the product's life cycle: design, manufacturing, marketing and positioning, selling, and maintenance.

Therefore, it is safe to assume that these trends will continue when manufacturers adopt Levels 4–5 of automation, when the DDT is performed by the ADS in its entirety. This means that automation will have a *personality* or a *driving style* that matches its branding and its intended market. Thus, strategies to reach the final goal of C-ITSs will also be affected by these marketing decisions.

Regarding the effect of personality in mediation and conflict management, the Thomas-Kilmann conflict model categorizes responses to stimuli depending on two dimensions of a person's style: assertiveness and cooperativeness. Fig. 8 shows our adaptation of the Thomas-Kilman model where we rename

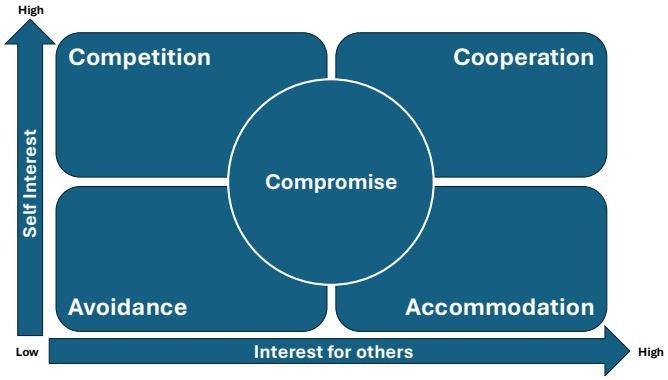


Fig. 8. Adaptation of the Thomas-Kilmann conflict model

assertiveness as "Self Interest" and cooperativeness as "Interest for Others". This allows us to refer to the top right quadrant as cooperation instead of collaboration. Regardless of these semantic changes, it is clear from the model that the way an agent is *programmed* to balance self interest and collective interests will affect the manner in which it will react to stimuli such as learning other agents' strategies.

Let us imagine a trio of vehicles in a road segment. They are all heading in the same direction for a considerable time, and they are all CCAM-enabled. The first one is a heavy-duty vehicle (HDV), the second one is a family-oriented passenger car, and the last one is a sports car. They all know each other's intentions (drive straight for several kilometers), and they can form a platoon. This would allow the smaller vehicles to draft safely behind the HDV and save energy. However, the sports car might not be programmed to save fuel and decides not to enter the platoon.

Other scenarios, where vehicles are not programmed to act assertively, might also cause efficiency problems. We see this in the results of our experiments, where the full CACC scenario drives at slower speeds than scenarios with conventional drivers or even the Krauss scenario with only perfect drivers. This last scenario could reflect the cooperation quadrant of the conflict model, since everyone has the same strategy and is programmed to drive at the maximum allowed velocity, while the CACC algorithm trades off efficiency for safety.

Finally, and outside of the context of technology itself, it is this sort of stakeholder decisions that prevent the arrival of even Level 3 automation into the market. As explored in [17], liability is a major issue in the deployment of fully automated vehicles. In case of an accident, the owner, the car manufacturer, and even product and services providers could be liable. However, we see this also as an opportunity to reach agreements and regulations that would force these *brand personalities* be overriden in cooperative mobility scenarios.

VII. Conclusions and Future Work

In this extension to the work in [1] not only we present the study of the coexistence of two releases of a GeoNetworking protocol in the context of ETSI ITS — Releases 1 and 2 of

ETSI CBF, but also on how automated vehicles coexist with conventional drivers. So, we explore the phenomenon of heterogeneity and coexistence in the networking and automation domains of C-ITSs.

In the networking part, we have proved that, as long as releases are compatible and nodes can understand each other, safety metrics stay high even if resource efficiency is compromised. Then, we presented a discussion of possible settings that are likely to happen when Future Mobility is completely mature (i.e., Day 4 of Vision Zero), where a Tower of Babel scenario might occur, and road users are segregated into pockets of nodes *speaking* different *languages* (and some not *speaking* at all). Even when the first C in CCAM stands for *cooperative*, this cooperation is not likely to occur when agents are not able to hear and understand each other.

For the mix of automated and conventional driver, we found that even if automation is homogeneous, the presence of conventional drivers in the mix alters safety and efficiency metrics. Furthermore, homogeneity does not guarantee more efficiency, and it also depends on automation strategies. We show that when automation pushes for higher speeds, it is also prone to more safety-related events. However, when its strategy is to prime safety, it compromises efficiency, and then the average speed of the system with full automation is slightly decreased from that with zero automation.

Future work includes a more in-depth analysis of the effect of *multi-modal road users* (e.g., disconnected users, legacy fleet) in the optimal performance of the CCAM-enabled fleet (i.e., connected and automated vehicles). For this, work in simulation setups is currently being performed, in order to account for network reliability issues in cooperation.

ACKNOWLEDGMENT

This work was partially supported by SAFER in the project "Human Factors, Risks and Optimal Performance in Cooperative, Connected and Automated Mobility", the Knowledge Foundation in the project "SafeSmart – Safety of Connected Intelligent Vehicles in Smart Cities" (2019-2024), and the EL-LIIT Strategic Research Network in the project "6G wireless" – sub-project "vehicular communications".

REFERENCES

[1] O. Amador, I. Soto, M. Calderon, and M. Urueña, "The smart highway to babel: the coexistence of different generations of intelligent transport systems," *VEHICULAR 2024, The Thirteenth International Conference on Advances in Vehicular Systems, Technologies and Applications*, 2024.

[2] European Commission (EC), "Towards a European road safety area: policy orientations on road safety 2011-2020," July 2010.

[3] SAE, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," SAE International, Standard, Apr. 2021.

[4] European Telecommunications Standards Institute (ETSI), "Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4. Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 2: Media-dependent Functionalities for ITS-G5," ETSI TS 102 636-4-2 - V1.4.1, February 2021.

[5] ——, "Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4. Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 2: Media-dependent Functionalities for LTE-V2X," ETSI TS 102 636-4-3 - V1.2.1, August 2020.

[6] D. Milakis, M. Snelder, B. Arem, B. Wee, and G. Homem de Almeida Correia, "Development and transport implications of automated vehicles in the Netherlands: Scenarios for 2030 and 2050," *European Journal of Transport and Infrastructure Research*, vol. 17, pp. 63–85, 01 2017.

[7] European Automobile Manufacturers' Association, "Average age of the EU vehicle fleet, by country," accessed: 2024-01-31. [Online]. Available: https://www.acea.auto/figure/average-age-of-eu-vehicle-fleet-by-country/

[8] European Telecommunications Standards Institute (ETSI), "Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 4. Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality," ETSI EN 302 636-4-1 - V1.4.1, January 2020.

[9] O. Amador, M. Urueña, M. Calderon, and I. Soto, "Evaluation and improvement of ETSI ITS Contention-Based Forwarding (CBF) of warning messages in highway scenarios," *Vehicular Communications*, vol. 34, p. 100454, 2022, accessed: 2024-01-31. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209622000018

[10] O. Amador, I. Soto, M. Calderon, and M. Urueña, "Studying and improving the performance of ETSI ITS contention-based forwarding (CBF) in urban and highway scenarios: S-FoT+," *Computer Networks*, vol. 233, p. 109899, 2023, accessed: 2024-01-31. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128623003444

[11] European Telecommunications Standards Institute (ETSI), "Intelligent Transport Systems (ITS); Decentralized congestion control mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part," ETSI EN 102 687 - V1.2.1, April 2018.

[12] Ó. Amador Molina, "Decentralized congestion control in etsi intelligent transport systems and its effect on safety applications," 2020.

[13] R. Riebl, *Quality of Service in Vehicular Ad Hoc Networks: Methodical Evaluation and Enhancements for ITS-G5*. PhD Thesis. De Montfort University, 2021. [Online]. Available: https://dora.dmu.ac.uk/handle/2086/21186

[14] V. Sandonis, I. Soto, M. Calderon, and M. Urueña, "Vehicle to internet communications using the etsi its geonetworking protocol," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 3, pp. 373–391, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.2895

[15] S. Kühlmorgen, H. Lu, A. Festag, J. Kenney, S. Gemsheim, and G. Fettweis, "Evaluation of Congestion-Enabled Forwarding With Mixed Data Traffic in Vehicular Communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 233–247, 2020.

[16] T. Paulin and S. Ruehrup, "On the Impact of Fading and Interference on Contention-Based Geographic Routing in VANETs," in *2015 IEEE 82nd Vehicular Technology Conference (VTC Fall)*, 2015, pp. 1–5.

[17] O. Amador, M. Aramrattana, and A. Vinel, "A survey on remote operation of road vehicles," *IEEE Access*, vol. 10, pp. 130 135–130 154, 2022.

[18] O. Taubman-Ben-Ari and D. Yehiel, "Driving styles and their associations with personality and motivation," *Accident Analysis & Prevention*, vol. 45, pp. 416–422, 2012.

[19] A. Sharma, Z. Zheng, J. Kim, A. Bhaskar, and M. M. Haque, "Assessing traffic disturbance, efficiency, and safety of the mixed traffic flow of connected vehicles and traditional vehicles by considering human factors," *Transportation research part C: emerging technologies*, vol. 124, p. 102934, 2021.

[20] European Telecommunications Standards Institute (ETSI), "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service," ETSI EN 302 637-3 - V1.3.1, April 2019.

[21] R. Riebl, H. Günther, C. Facchi, and L. Wolf, "Artery: Extending Veins for VANET Applications," in *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 2015, pp. 450–456.

[22] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2011.

[23] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent Development and Applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, December 2012.

[24] C. Sommer, S. Joerer, and F. Dressler, "On the applicability of Two-Ray path loss models for vehicular network simulation," in *2012 IEEE Vehicular Networking Conference (VNC)*, 2012, pp. 64–69.

[25] European Telecommunications Standards Institute (ETSI), "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specifications of Cooperative Awareness Basic Service," ETSI EN 302 637-2 - V1.4.1, April 2019.

[26] N. Sjögren and H. Vu, "The effect of a mixed-capability vehicular fleet on vulnerable road user safety," 2024.

[27] K. N. Porfyri, E. Mintsis, and E. Mitsakis, "Assessment of acc and cacc systems using sumo," *EPiC Series in Engineering*, vol. 2, pp. 82–93, 2018.

[28] L. An, X. Yang, and J. Hu, "Modeling system dynamics of mixed traffic with partial connected and automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 755–15 764, 2022.

[29] Volkswagen Group, "Volkswagen Group delivers 45 percent more all-electric vehicles in first 9 months," accessed: 2024-01-31. [Online]. Available: https://www.volkswagen-newsroom.com/en/press-releases/volkswagen-group-delivers-45-percent-more-all-electric-vehicles-in-first-9-months-17771/

[30] C-ITS Deployment Group, "ASFINAG, Managing everyday traffic situations more easily with C-ITS / Car2X," accessed: 2024-01-31. [Online]. Available: https://c-its-deployment-group.eu/news-events/news/2022-07-27-asfinag-c-its-imagevideo

[31] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, Aug. 2018. [Online]. Available: https://www.rfc-editor.org/info/rfc8446

[32] O. Levillain, "Implementation flaws in tls stacks: Lessons learned and study of tls 1.3 benefits," in *Risks and Security of Internet and Systems*, J. Garcia-Alfaro, J. Leneutre, N. Cuppens, and R. Yaich, Eds. Cham: Springer International Publishing, 2021, pp. 87–104.

[33] S. Lee, Y. Shin, and J. Hur, "Return of version downgrade attack in the era of tls 1.3," in *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 157–168. [Online]. Available: https://doi.org/10.1145/3386367.3431310

[34] C-ROADS, "C-ROADS platform position paper on 5.9 GHz band," accessed: 2024-01-31. [Online]. Available: https://www.c-roads.eu/fileadmin/user_upload/media/Dokumente/C-Roads_Position_paper_on_59GHz_final.pdf

[35] European Telecommunications Standards Institute (ETSI), "Intelligent Transport Systems (ITS); Pre-standardization study on co-channel co-existence between IEEE- and 3GPP- based ITS technologies in the 5 855 MHz - 5 925 MHz frequency band ," ETSI TR 103 766, September 2021.

[36] Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge: Cambridge University Press, 2009.

Exploring Deep Learning Techniques for Artist and Style Recognition in the Paintings-100 Dataset

Erica M. Knizhnik*†, Brian Rivera*‡, Atreyee Sinha§, and Sugata Banerji*¶
*Department of Mathematics and Computer Science, Lake Forest College, Lake Forest, IL, USA.
†Email: knizhnikem@lakeforest.edu
‡Email: riverab@lakeforest.edu
§Computing and Information Sciences, Edgewood College, Madison, WI, USA. Email: asinha@edgewood.edu
¶Corresponding author. Email: banerji@lakeforest.edu

***Abstract*—Painting classification is a challenging interdisciplinary research problem in computer vision. With more fine-art paintings being available in the form of high-resolution digital scans, the development of effective classification algorithms has become vital. Such algorithms would have numerous applications, including but not limited to museum curation, several different industries, painting theft and forgery investigation, and art education. While some progress has been done in this field, accurately identifying the painter or the artistic style from the painting remains a complex task. Towards that end, we present an enhanced image dataset comprising high-resolution painting images from 100 diverse artists across 14 distinct styles. This dataset builds upon the Painting-91 dataset originally created by Khan et al. Our main contributions in this work are three-fold. First, we improve the older dataset by correcting errors, enhancing image resolution, and expanding it with more images, artists, and styles. Second, we perform an extensive evaluation of this newly constructed Paintings-100 dataset using several different convolutional neural network (CNN)-based classification techniques for both artist and style recognition tasks. Finally, we explore the different stylistic characteristics that the networks focus on to recognize the specific artists and styles of paintings, and demonstrate that our proposed and improved dataset is more suitable for patch-based models than the earlier published Painting-91 dataset due to larger image resolutions.**

***Keywords*—painting classification; image dataset; style classification; artist classification; CNN ensemble.**

I. Introduction

The current work expands upon our previous work [1], where we present a new high-resolution dataset of paintings, and explore image classification on it.

In the last decade, a significant quantity of artwork has been digitized. That fact, combined with the substantial progress in the area of computer vision, has opened up the interesting research area of automated painting classification [2]. Automated painting classification can be broken down into two subtasks: artist identification and style categorization. The former task involves identifying a painting as the work of a specific artist, while the second task involves labeling paintings by art movement or style. Identifying artists and styles in fine-art paintings has numerous applications in several industries, such as tourism and movie-making, art education, and investigation of art forgery (though the system proposed here does not claim to be suitable for this last application). For instance, a user may take the photo of a painting or reproduction somewhere and want to know more about the painting, such as the artist's name and style. A system that can autonomously classify art is, therefore, of great interest. However, both tasks pose

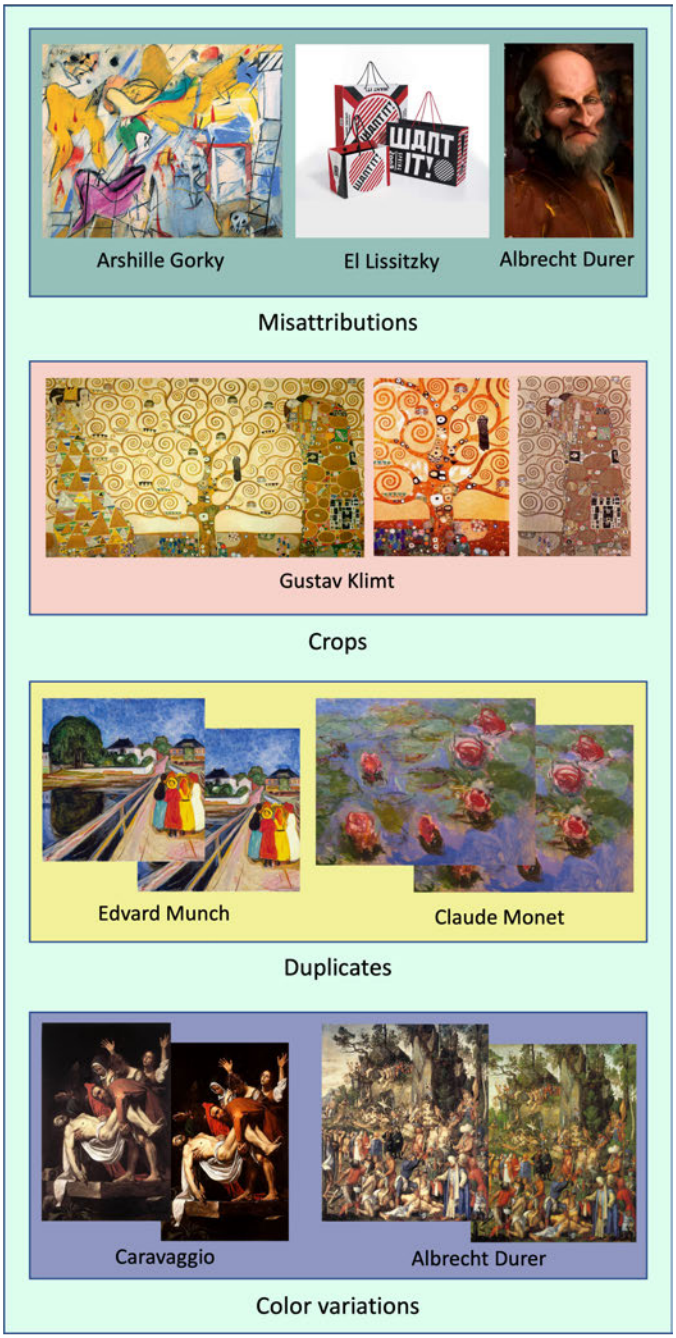


Figure 1. Some errors that exist in the old Painting-91 dataset.

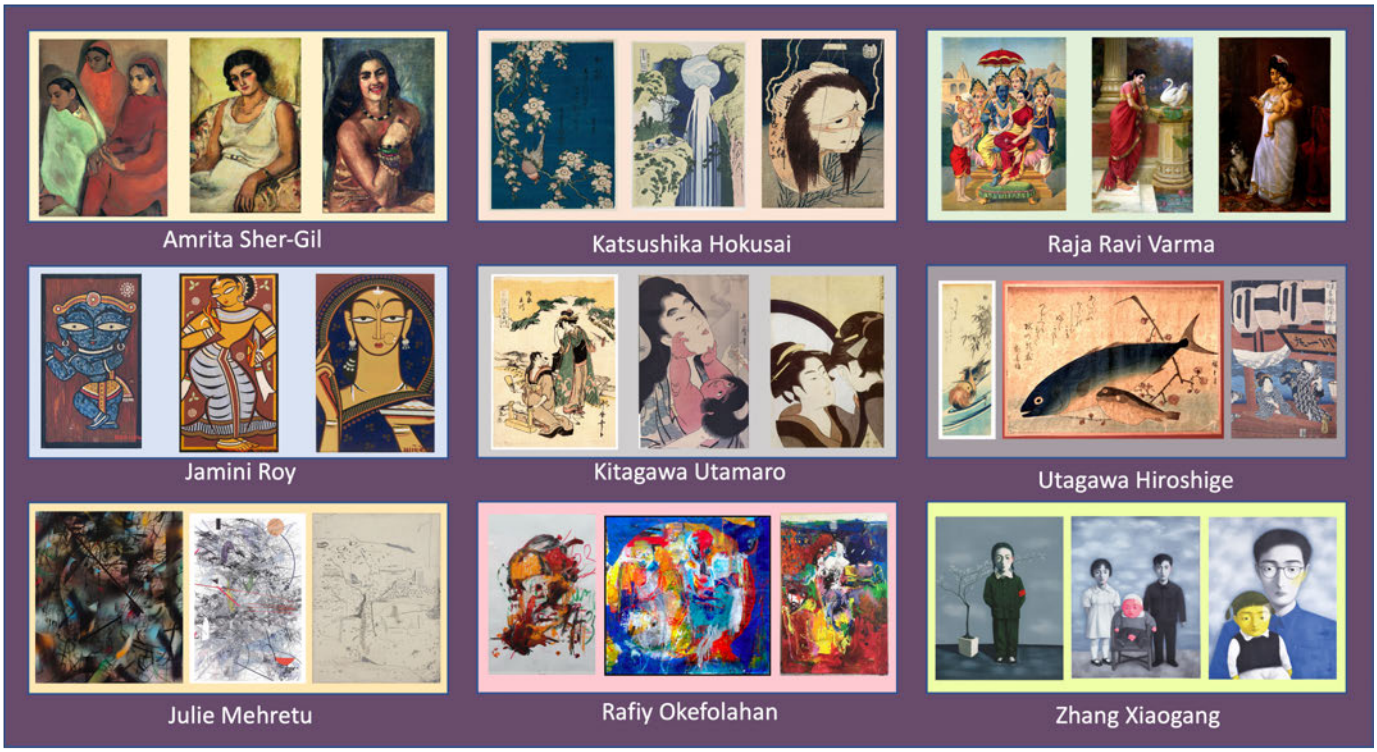


Figure 2. Some paintings of the newly added nine artists that are included in the Paintings-100 dataset.

significant challenges due to the complexities of artistic styles, subjectivity in the interpretation of paintings, varied image quality, lack of fine details, and context of the visual images due to the presence of stylistic variations that can occur even within a single artist's work [3] [4] [5].

In the current work, our contribution is threefold: first, we do a detailed discussion of the new Paintings-100 image dataset curated by us [1], highlighting our improvements to the Painting-91 Dataset [3]. Second, we have done an extensive evaluation of this rich and varied dataset using several convolutional neural network (CNN)-based methods on whole images as well as random image patches for both the artist and style classification tasks, finally showing that an ensemble of multiple models performs best. Last, but not the least, we attempt to identify the salient features of some of the style classes by examining the CNN response maps, and also identify the points of confusion between classes.

The rest of this paper is organized as follows. Section II lists some of the other work in this field. Section III and its subsections outline in detail the construction of the new dataset used in this work. Section IV describes our proposed methodology in detail, and Section V reports the classification performance on this dataset, the experiments performed, the results obtained, and discusses the outcomes. Finally, we list our conclusions and directions for future research in Section VI.

II. Related Work

Painting styles encompass the unique techniques, methods, and characteristics artists use to express themselves. Over the last two decades, computer vision, machine learning, and artificial intelligence have been successfully applied to analyzing and interpreting fine-art paintings and drawings [6] [7] [8] [9], offering innovative tools for art experts and scholars. Unlike artist categorization, which focuses on individual artists, style classification recognizes that multiple painters can share a common style, making it a distinct and challenging problem [3].

The traditional computer vision techniques for image classification use either color [10], shape [11] or texture [12] features. But feature extraction also results in the loss of semantic information from the painting image, thus increasing the challenge of identifying its style [13]. Techniques such as detecting and recognizing the artist's signature are not universally applicable due to the signature often getting cropped out of digital reproductions. That is why in recent years, computer vision researchers have explored the painting style recognition problem using CNNs, which are better at preserving semantic information. One of the major challenges of being able to effectively use CNNs for painting classification is the need for large hand-labeled datasets [5]. The limited availability of training data has led to a reliance on pre-trained models. Thus, instead of training a neural network from scratch, existing approaches either fine-tune pre-trained models, utilize them for feature extraction, or opt for non-neural network-based

methods altogether.

In [5], the authors explored the applicability of CNNs for art-related image classification tasks by performing extensive CNN fine-tuning experiments and consolidating the results for five different art-related classification tasks. They also showed that fine-tuning networks pre-trained for scene recognition and sentiment prediction produced better results than those pre-trained for object recognition, demonstrating the effectiveness of leveraging scene and sentiment knowledge for style recognition. Rodriguez et al. [7] used a weighted sum of the individual-patch classification outcomes to provide the final stylistic label of the analyzed painting. Lately, [14] employed a framework to compare the performance of six pre-trained CNN architectures (Xception, ResNet50, InceptionV3, InceptionResNetV2, DenseNet121, and EfficientNet B3) for style classification using transfer learning, and studied the effect of different optimizers with learning rates on each model.

In our previous work [15], we explored the use of pre-trained CNN models as a feature extraction tool for painting classification. Some of the popular painting datasets that are available publicly for artist and style classification include the Painting-91 dataset by [3], the WikiArt dataset [16], and the Painting dataset consisting of ten classes of fine-art paintings from the PASCAL VOC [17]. But even though these datasets exist, the number of hand-labeled paintings available for effectively using CNNs is very limited [5]. To that end, we have worked on expanding the existing Painting-91 dataset [3] to construct a bigger dataset called the Paintings-100 dataset [1]. While constructing this dataset, we worked on improving the existing Painting-91 dataset [3] to not only include newer artists and painting styles, but also carefully remove different mis-attribution and other human errors that existed in that dataset. Some of these errors are shown in Figure 1. We also enhanced image resolutions from the previous dataset, and augmented certain artist categories, which had fewer images in the previous dataset, with more images. Finally, we did extensive experiments with several CNN models to address both the artist classification and style classification tasks.

III. Dataset Construction

When we worked on [15], we realized that the images in the original Painting-91 dataset [3] are too small for learning meaningful features using deep learning. While trying to replace the images with their high-resolution versions, we found several kinds of human errors and other limitations in the original dataset which needed to be fixed. These issues, some of which are shown in Figure 1, are described in the subsections below, along with the improvements made by us.

A. Low Resolution

This was the main motivation for constructing the new dataset. The mean size of an image in the Painting-91 dataset is 268×263 pixels. These dimensions are smaller than the input sizes of many modern CNN models. So, to improve the quality of the data, we started replacing the images with high-resolution versions downloaded from the Internet via Google Reverse Image Search [18]. We were successful in this task for about 97% of the images, but we also ran into other errors as detailed next.

Figure 3. Some paintings of the 14 different style categories that are included in the Paintings-100 dataset.

B. Mis-Attributions

These are images labeled with a painter's name that are not painted by that painter. Some of these mis-attributed images are deliberate attempts to copy the attributed painter's style, some are created using image editing software by making collages of existing paintings, and some others have simply been downloaded from a source on the Internet, which also had the wrong label.

C. Duplicates

Several of the images in most artist classes are duplicates of other images also in the class. The number of images per class varies from 30 to 51, which is already very small for training deep learning models, and the presence of duplicate and mislabeled images further reduces this number. For instance, the painter class Hieronymus Bosch has 50 paintings, out of which 25 are duplicates (exact or slightly variant copies), and

TABLE I
New artists whose paintings were added to the dataset, along with their nationality and style.

Artist	Nationality	Style
Amrita Sher-Gil	Hungarian-Indian	Several
Jamini Roy	Indian	Indian folk art
Julie Mehretu	Ethiopian American	Several
Katsushika Hokusai	Japanese	Ukiyo-e
Kitagawa Utamaro	Japanese	Ukiyo-e
Rafiy Okefolahan	Cape Verdean	Contemporary multimedia
Raja Ravi Varma	Indian	Indian realism
Utagawa Hiroshige	Japanese	Ukiyo-e
Zhang Xiaogang	Chinese	Surrealism

a further 5 are wrongly attributed, thus bringing the actual number of usable images down to 20.

D. Cropped Images

These are images which show only part of a painting, the whole of which may or may not be present in the dataset. Since the overall composition bears as much information about a painter's identity or style as details do, just having a small cropped portion of a painting in the dataset is not ideal.

E. Color Variations

These are also copies of other images in the dataset. However, instead of being exact duplicates, these images have a different color palette. There is no way of knowing which of the copies has a more accurate color palette, and so, color cues lose their significance in classification. To further complicate matters, some artists (such as Andy Warhol) themselves produced multiple copies of the same painting with slight differences in details and color, which count as different images in the dataset.

F. Lack of Diversity

While the original dataset contains an impressive collection of works from 91 painters and 13 style categories, this collection focuses exclusively on Europe and the Americas. There are no painters representing the rich artistic heritage of Asia and Africa. This is not exactly an error, but an omission in the dataset that needed to be addressed for overall improvement.

G. Improvements

We took several steps to address the above issues. First, we replaced most images with their high-resolution versions wherever such a version was available in the public domain. The mean image size in the new dataset is $1,523 \times 1,493$ pixels. This amounts to a 32-fold increase in the number of pixels per image, on average. Second, we replaced wrongly labeled images with their correct counterparts, or new images by the same artist. Third, wherever possible, we also added new paintings to all artist categories that had less than 50 paintings. Fourth, we reduced the number of duplicates by replacing them with new paintings wherever possible. Last, but not the least, we added 50 paintings each by 9 more painters spanning a diverse array of styles representing Asian and African art (shown in Figure 2 and Table I). This makes our new Paintings-100 dataset a more diverse, inclusive and representative database of fine-art paintings. The presented Paintings-100 dataset has $5,357$ images which is an impressive 25% increase from the $4,266$ image Painting-91 dataset.

We also added the style movement Ukiyo-e, into this new collection for style classification task. Examples of paintings from each of these 14 style classes is shown in Figure 3. Table II displays a list of the various painting styles used by the different artists that are included in the Paintings-100 dataset.

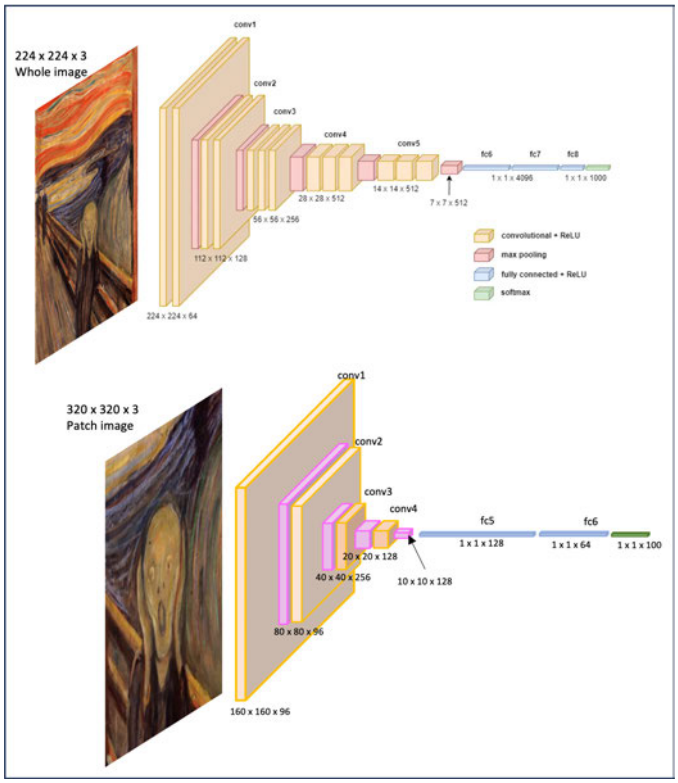


Figure 4. For artist classification task, we used both whole images as well as random patches from the images to feed into different CNN models.

IV. Methodology

The original Painting-91 dataset, and by extension, the proposed Paintings-100 dataset, are both designed for two classification tasks. These tasks are artist classification and style classification. The first task is straightforward as every image has an artist class label, and the artist classes are roughly equal in size. For the second task, the dataset contains 14 style class labels in addition to the 100 artist class labels. This is a slight increase from 13 style classes in the Painting-91 dataset (Ukiyo-e is the new style class introduced). Each style class contains works from more than one artist, but not all artists have a style class label [3]. In the current work, we have analyzed the dataset with respect to both these problems.

A. Artist Classification

While CNN-based models have largely outperformed other techniques for various classification tasks, the artist classification problem is somewhat challenging for these models. This is mainly due to two reasons. Firstly, deep learning is data-hungry, and very few artists manage to paint more than a few dozen completed paintings in their lifetime. This severely limits the images available for training. Secondly, CNN models take fairly low-resolution images as their input. This means, we either need to downsample the images and lose all detail, or crop the images and lose all sense of composition and context. Since neither solution was fully acceptable to us on its own, we decided to use a bit of both.

TABLE II
DIFFERENT PAINTING STYLES INCLUDED IN THE PAINTINGS-100 DATASET.

Style	Artists	# Images
Abstract Expressionism	Jackson Pollock, Mark Rothko, Willem De Kooning	167
Baroque	Caravaggio, Diego Velazquez, Jan Vermeer, Nicolas Poussin, Peter Paul Rubens, Rembrandt Van Rijn	304
Constructivism	El Lissitzky, Kazimir Malevich, Wassily Kandinsky	153
Cubism	Fernand Leger, Georges Braque, Piet Mondrian, Picasso	157
Impressionism	Claude Monet, Edgar Degas, Edouard Manet, Pierre-Auguste Renoir	205
Neo-classical	Jacques-Louis David, Jean-Auguste-Dominique Ingres	106
Pop Art	Andy Warhol, David Hockney, Roy Lichtenstein	153
Post Impressionism	Amedeo Modigliani, Georges Seurat, Paul Cezanne, Paul Gauguin, Vincent Van Gogh	255
Realism	Camille Corot, Gustave Courbet, James McNeill Whistler, Jean Francois Millet, Raja Ravi Varma	256
Renaissance	Raphael, Sandro Botticelli, Titian	172
Romanticism	Caspar David Friedrich, Eugene Delacroix, Francisco De Goya, John Constable, Joseph Mallord William Turner, William Blake	310
Surrealism	Georgia Okeefe, Joan Miro, Max Ernst, Rene Magritte, Salvador Dali, Zhang Xiaogang	314
Symbolism	Gustave Moreau, Gustav Klimt	105
Ukiyo-e	Katsushika Hokusai, Kitagawa Utamaro, Utagawa Hiroshige	150

1) Preprocessing: To address the problem of too few images and too much detail, we used an ensemble of multiple CNN models that use both downsampled whole images and full-size patches cropped out of the high-resolution images. These patches were randomly selected square patches of size 224×224 pixels or larger. In both cases (downsampled whole image and cropped patches), we used 24 images per class with augmentation (variations created by slight rotation, translation, shear, scaling, and horizontal mirroring) for training, 6 per class for validation, and the rest for testing. The whole and cropped images were histogram normalized and preprocessed for their corresponding CNN models.

However, using only whole images for training poses another problem. Even though each style class has three or more artists, the total number of training images is still quite low for training CNNs properly. Because of this, we use image augmentation techniques to increase the size of our training set. We use translation, rotation, shear, zoom and horizontal flip operations on our images for augmentation. The images are also resized to 224×224 pixels. Finally, each image is passed through a preprocessing function specific to each pretrained network before passing through the network itself.

2) Model Selection: Classifying whole images and classifying patches are two different problems. For classifying the patches, we designed our own CNN from scratch and trained it using 25 random square patches from each training image. For the whole image classification, we fine-tuned the VGG-16 network [19] trained on the ImageNet image dataset [20] since we had far fewer images. These two models are shown in Figure 4. The models were chosen empirically. We used decision fusing based on the labels predicted by the two models.

Although the style classification task takes the same input as the artist recognition task, it has a different output. Specifically, here the challenge lies in comprehending the artistic style of the artwork, which is often more complex and subtle than merely recognizing the painting's content or the artist. For this task, we use deep learning as well. The styles present in our dataset are Abstract expressionism, Baroque, Constructivism, Cubism, Impressionism, Neo-classical, Pop art, Post-impressionism, Realism, Renaissance, Romanticism, Surrealism, Symbolism and Ukiyo-e. Each of these styles offers a unique perspective on how artists interpret the world and their experiences. For example, abstract expressionism that flourished in the mid-twentieth century, emphasizes spontaneous, automatic, or subconscious creation, with Jackson Pollock and Mark Rothko as key figures. On the other hand, Ukiyo-e is a genre of Japanese art that flourished from the 17th to the 19th centuries, admired for its beauty, craftsmanship, and cultural significance. All the style categories being used

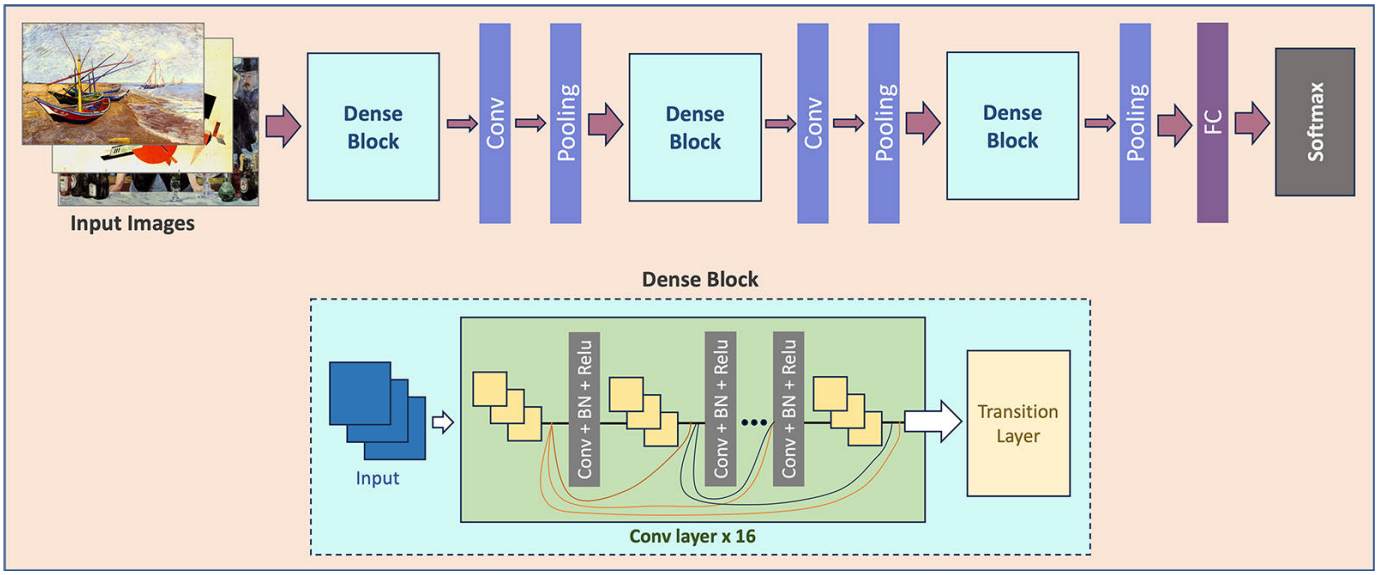


Figure 5. The internal architecture of DenseNet201 [21].

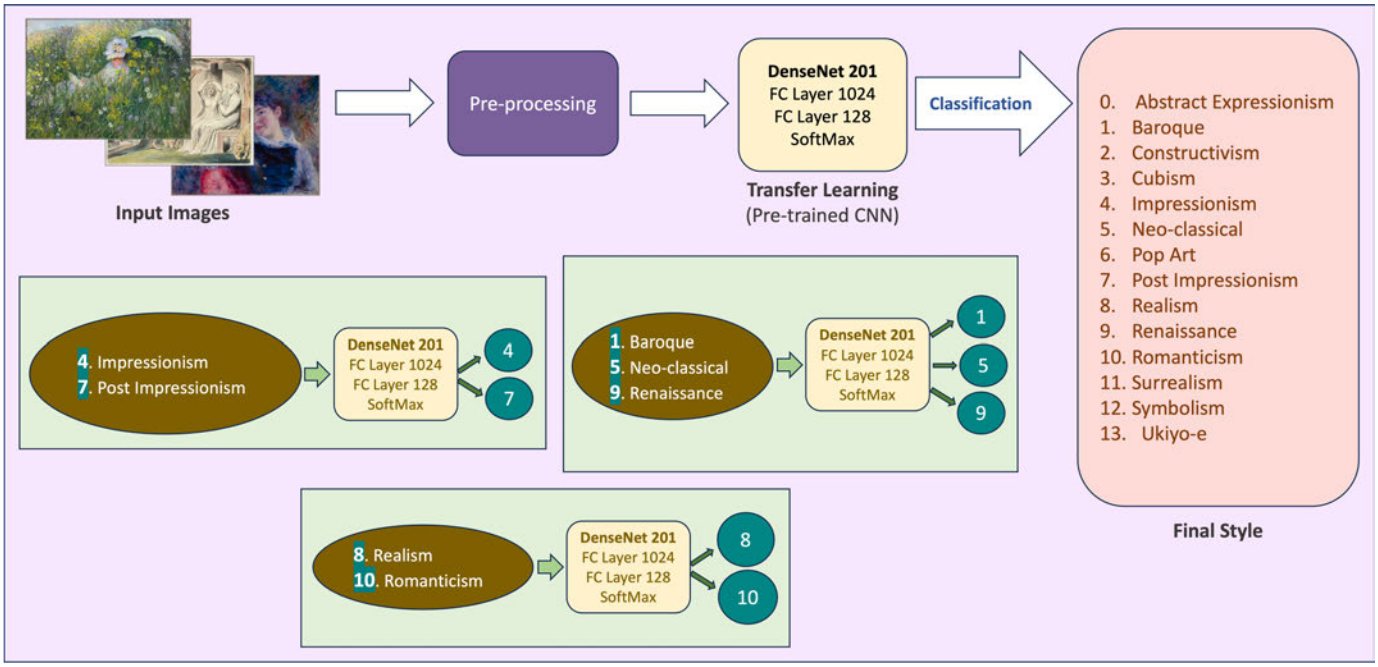


Figure 6. Our proposed method for style recognition. We employ an ensemble of four DenseNet201 models to do style classification in two stages.

for this work are listed in Table II. It should be noted that only 55 out of the 100 artists are included in this task since some of the other artists painted multiple styles, or were sole representatives of their style.

B. Style Classification

In this section, we discuss our experiments regarding the style classification task and our results in detail.

1) Preprocessing: While addressing the artist recognition problem previously, we found that dividing the image into small patches and using them for training the classifier worked well [1]. However, that technique did not work well with the style recognition task. Our intuitive understanding of this is, the style class of an image is much more dependent on the whole image rather than finer details. That is why, our models cannot reliably learn the style patterns with small patches of the images.

However, using only whole images for training poses another problem. Even though each style class has three or more artists, the total number of training images is still quite low

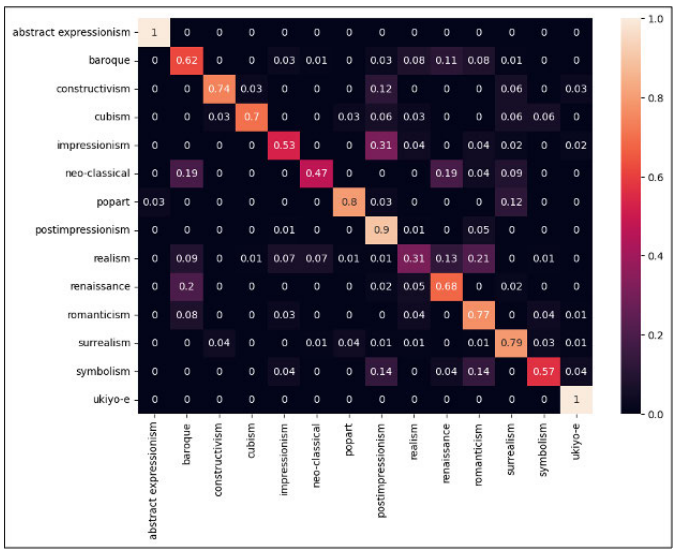


Figure 7. Confusion Matrix of Style Classification using DenseNet201. The rows indicate actual class labels while the columns indicate predicted class labels.

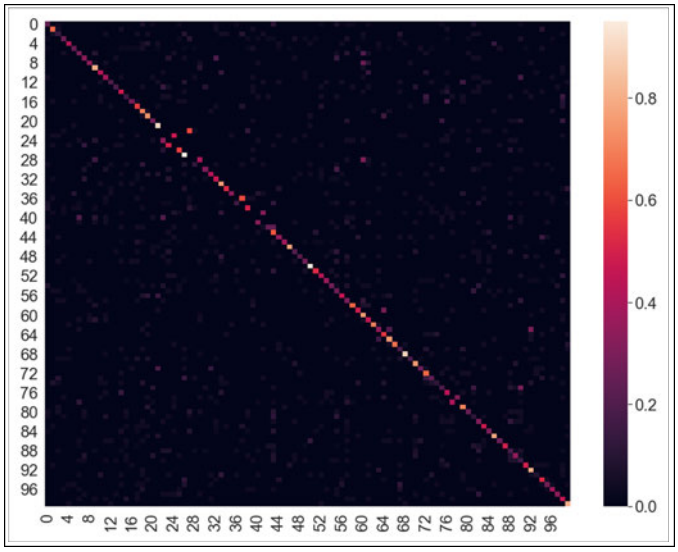


Figure 8. The confusion matrix for the artist classification experiment using the combined decision of two CNN models. The rows indicate actual class index values while the columns indicate predicted index values.

for training CNNs properly. Because of this, we use image augmentation techniques to increase the size of our training set for fine-tuning pre-trained networks (as detailed in the next section). We use translation, rotation, shear, zoom and horizontal flip operations on our images for augmentation. The images are also resized to 224×224 pixels. Finally, each image is passed through a preprocessing function specific to each pre-trained network before passing through the network itself.

2) Model Selection: The lack of labeled training images that makes painting classification so challenging for CNNs is somewhat less acute for style classification, but it is still very much present. The limited number of training images makes this problem particularly suited for transfer learning. For this work, we test five well-known CNN architectures on our data. Out of the five, three performed well on the style classification problem. All CNN models were pre-trained on ImageNet data [20].

Since the number of images, even after augmentation, is not sufficient to train a deep neural network from scratch, we did not create our own model for this task. We tried several pre-trained CNN models and compared their performance. The models that performed reasonably well were the VGG16 network [19], the DenseNet121 network [21], and the DenseNet201 network [21]. The tested models that did not perform well were the InceptionV3 [22] and the EfficientNetB3 [23] models. Their performances are detailed in Section V. Since the DenseNet201 was our best-performing model, we selected this model for all further classification experiments.

DenseNet201 [21] is a deep convolutional neural network with 201 layers, where each layer is connected to every other layer in a feed-forward manner. It connects all its 201 layers directly, without skipping any connections. This allows each layer to learn not just from the previous layer, but also from all the layers that came before, mitigating the vanishing gradient problem and enhancing feature propagation. Additionally, it promotes feature reuse while achieving a compact architecture with a reduced parameter count. The internal architecture of this network is shown in Figure 5.

3) Ensemble-based Classification: Now, we will discuss the ensemble-based classification method shown in Figure 6. The confusion matrix of the style classification task as done by the DenseNet201 model is shown in Figure 7. As can be seen in the matrix, there are several areas of inter-class confusion. The biggest of these is between Impressionism and Post-impressionism. Other large confusion rates are between Realism and Romanticism, and between Baroque, Neo-classical and Renaissance. To handle these particularly difficult classification problems, we train three more DenseNet201 models. The first of these is trained only on Impressionism and Post-impressionism images, the second only on Baroque, Neo-classical and Renaissance images, and the third only on Realism and Romanticism images. While testing, we first pass each test image through the network trained to classify between all 14 classes. If the output label is one of the classes with high confusion, that image is passed through the CNN model specialized for that class and the output label from this second model is assigned to it. So, if the first model outputs one of the labels shown in the green circles in Figure 6, the image in question is passed through a second CNN model which assigns the final label to it. This leads to considerable improvement in the classification accuracy as detailed in Section V-B.

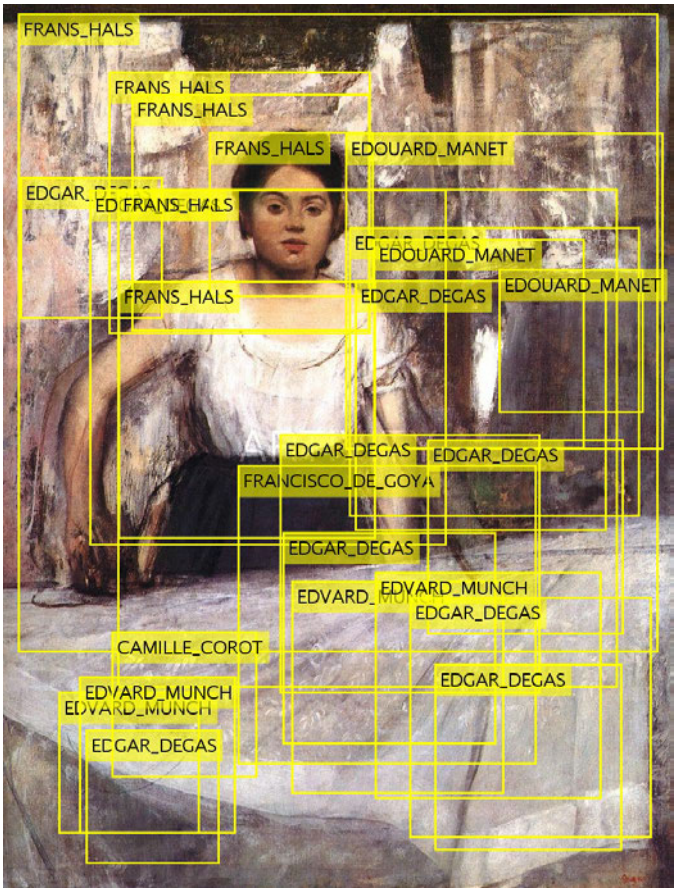


Figure 9. Painting by Edgar Degas. When the whole image is used for artist recognition, the CNN identified it as a Frans Hals painting, whereas by using random patches, it is correctly classified as an Edgar Degas artwork.

V. Experiments and Discussion

In the following subsections we describe the two sets of experiments that we performed on the Paintings-100 dataset. These two sets of experiments were done for the artist recognition and style recognition tasks, respectively.

A. Artist Classification Experiments

For the artist classification task, our initial results were promising, with the patch-based model yielding a 32% accuracy on the test set, the whole image model yielding 33%, and the fused accuracy at 38%. The confusion matrix for this result is shown in Figure 8. Figure 9 illustrates the effectiveness of such a fusion. In this example, although the whole image classifier predicts the label to be Frans Hals, different patches vote for different labels and the true artist, Edgar Degas, gets the most votes.

We also did a visualization of the responses from the last convolutional layer of our patch-image classifying CNN using the Grad-CAM technique [24]. This "heatmap" analysis highlights the regions of an image that are key identifiers for artist recognition. While this is a work in progress, the results demonstrated in Figure 10 show some of the characteristics

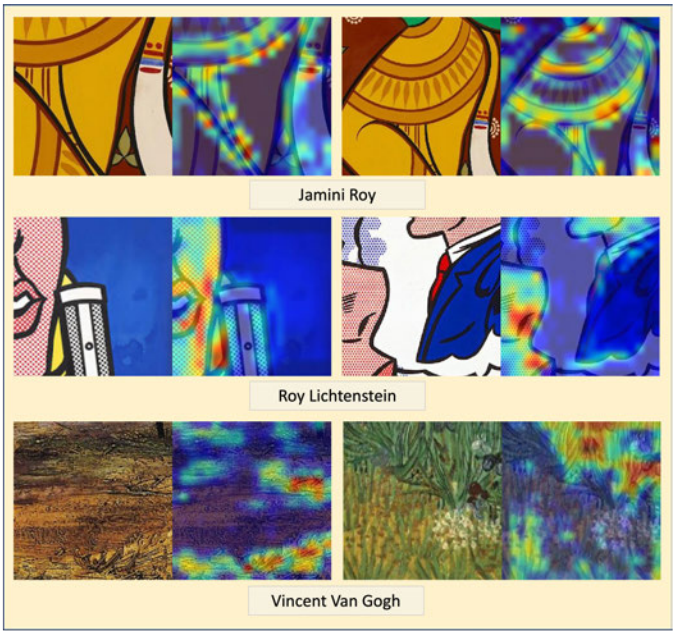


Figure 10. A few paintings and their Grad-CAM response maps showing regions of interest for artist recognition as detected by the CNNs.

of artists that the network can identify correctly. For example, bold outlines are a signature characteristic of Indian painter Jamini Roy and these outlines are highlighted in the topmost example in Figure 10. Similarly, dotted patterns and certain kinds of brush strokes are recognized as characteristic features of Roy Lichtenstein and Vincent Van Gogh, respectively.

B. Style Classification Experiments

For the style classification task, we first ran the same experiment once for each CNN architecture that we tested. This was a single 14-class classification of all style images. Out of the total number of images shown in Table II, we used 80% from each class for training and the rest for testing. 80% of the 80% used for training are used for true training and the other 20% are used for validation. The models that performed reasonably well on this experiment were the VGG16 network [19], the DenseNet121 network [21], and the

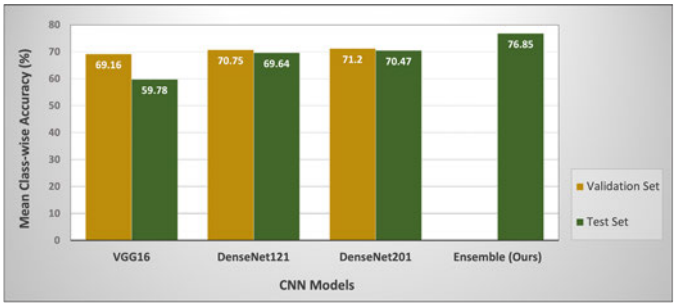


Figure 11. A comparison of the validation and test set accuracies of the three different CNN architectures that we tested, along with the ensemble accuracy on the test set.

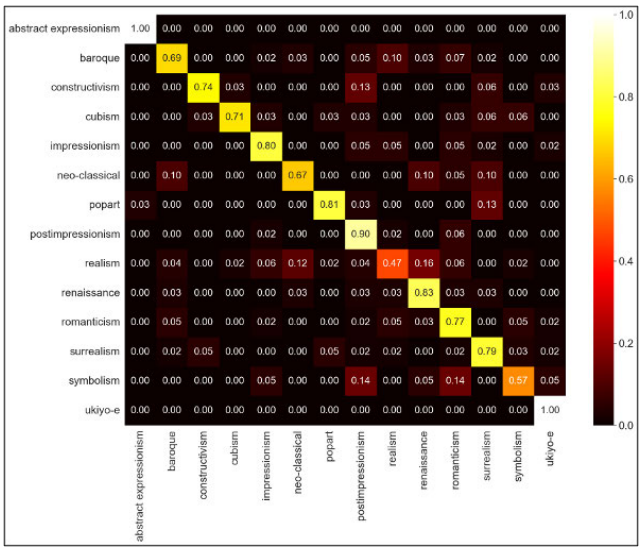


Figure 12. Confusion Matrix of Style Classification using our ensemble-based method. The rows indicate actual class labels while the columns indicate predicted class labels.

DenseNet201 network [21]. Since the DenseNet201 with a validation accuracy of 71.20% was the best performer, we proceeded to the next stage with this as our primary architecture. The validation and test set classification accuracies of all the models we tested are shown in Figure 11. It should be noted that the validation accuracy is not shown for our proposed ensemble model since we are combining the decisions of different trained models to get this result, and the concept of validation is not meaningful here.

For the next stage, we trained three more DenseNet201 networks. The first was trained on Impressionism and Post-impressionism images and gave us an accuracy of 92.16%. The second model was trained on three classes, namely, Baroque, Neo-classical, and Renaissance, and gave us a validation accuracy of 77.42%. The last model was trained on images from two style classes - Realism and Romanticism, and yielded a validation accuracy of 82.22%.

Subsequently, we created a two-level ensemble of CNN models as described in Section IV and passed all test images through this ensemble. This method gave us a test accuracy of 76.85%, which was an improvement of about 6% over a single DenseNet201 handling all 14 style categories on the test set. It should also be noted that this result is even higher than the results shown by [14] on the Painting-91 dataset [3] which contains one style class and five artists less (for this specific problem) than our Paintings-100 dataset. A comparison of the class-wise classification accuracies using the proposed ensemble method can be found in Figure 12.

Next, we used the Grad-CAM method [24] to view the response maps of a classification network. For this experiment, we used the VGG16 network instead of the DenseNet201 since VGG16 is a linear model and easier to combine with Grad-CAM. A small sample set from the results is shown in

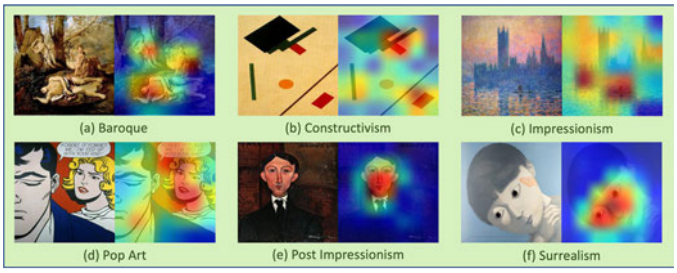


Figure 13. A few paintings and their Grad-CAM response maps showing regions of interest for style recognition as detected by the CNNs.

Figure 13. This gives us some insight into what the networks are looking for to correctly classify the images. For instance, in Figure 13(a), the network clearly recognizes the Baroque style by focusing on the three human figures in the painting. In (d) and (e), the network focuses on the faces of the human figures to recognize their respective styles, while in (f), the focus is primarily on the eyes of the subject. These black eyes are a defining characteristic of surrealist painter Zhang Xiaogang, and the network learns to recognize them during training.

Figure 14 shows some of the confusing images that were misidentified by the system. But it is easy to see that these images are actually confusing to label. While (a), (c), and (f) are labeled as Surrealism, Cubism, and Constructivism, respectively, all of them look somewhat similar to the post-impressionist pastoral landscapes by Van Gogh or Paul Gauguin. Similarly, (b), (d) and (e) have features of the style they are labeled as by the network, along with the style they are originally annotated with.

Finally, we wanted to see how good our model was in

Figure 14. Some confusing images that were misclassified for style recognition as detected by the CNNs.

Figure 15. A few samples of images from outside this dataset, whose styles were accurately predicted by our system.

recognizing the styles of images from outside the dataset. To test this, we created a second small dataset of 140 images (10 from each style class). All of these images were by artists who were not among our style recognition training data. In fact, most of these artists are not even in the Paintings-100 dataset. Our system performed quite well with this completely unseen data as well, predicting 44.28% of the styles correctly. Some of the correctly predicted images from these external images are shown in Figure 15, and some of the wrongly classified images are shown in Figure 16. It should be noted that classifying painting style often involves a subjective decision and the same painting may sometimes reflect the properties of two or more different styles, which makes classifying the work of unseen artists very challenging.

VI. Conclusion and Future Work

In this work, we expanded our experimentation upon the large scale diverse high-resolution image dataset that we recently presented for artist and style classification. While this was based on the existing Painting-91 dataset, the improvements were significant enough for the Paintings-100 dataset to be considered a new dataset. We have explored both the artist recognition and the painting style classification problems by conducting extensive experiments using several CNN architectures, and found that ensembles of CNN models showed more promising results for both tasks. As the experiments show, our proposed ensemble methods perform better than any one single CNN model tested by us. Some of these methods cannot be applied on the original Painting-91 dataset because of low-resolution images. The focus of our work was exploring the suitability of the newly introduced Paintings-100 dataset for the artist and style classification problems, and we can safely say that it is indeed suitable for these tasks.

There are many different ideas that we would like to try out on this dataset in the near future. Currently, we are selecting the patches for the artist recognition task randomly. In future, we want to try selecting patches with face detectors and object detectors to see how that affects our results. Photographic conditions such as ambient lighting and camera model create big differences in the color maps of the digitized paintings. We plan to use some color normalization techniques to reduce the effect of photographic conditions on the paintings. In a later work, we would like to extend this work by including other painting datasets and other CNN models, since the generalization performance of our method still has room for improvement. We would also like to expand upon the response maps portion of this work to better understand and explain the functioning of our models.

References

[1] E. Knizhnik, B. Rivera, A. Sinha, and S. Banerji, “Paintings-100: A diverse painting dataset for large scale classification,” in The Ninth International Conference on Advances in Signal, Image and Video Processing (SIGNAL 2024), 2024, pp. 12–15.

[2] D. Zhu, “The application of computer image processing technology in the fine arts creation,” in 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA), 2014, pp. 790–792.

[3] F. S. Khan, S. Beigpour, J. V. de Weijer, and M. Felsberg, “Painting-91: A large scale database for computational painting categorization,” Machine and Vision Application (MVAP), vol. 25, no. 6, 2014, pp. 1385–1397.

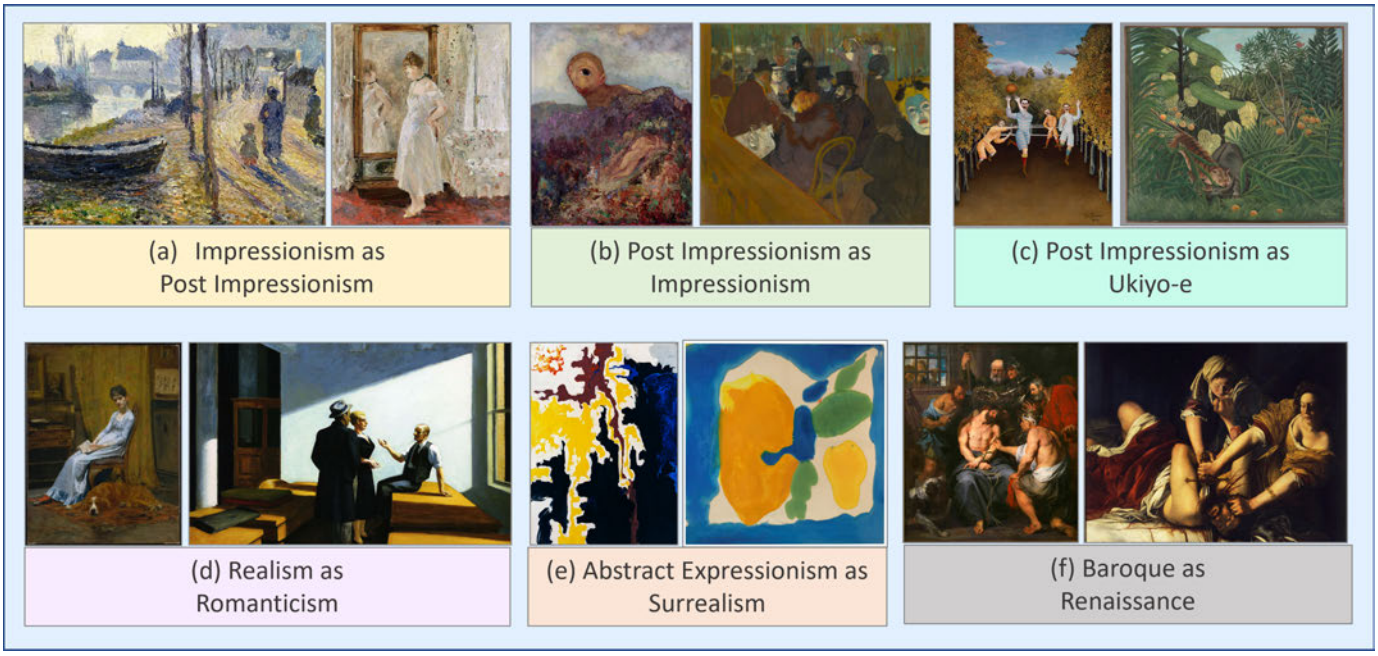


Figure 16. A few samples of images from outside this dataset, whose styles confused our system, leading to wrong predictions.

[4] J. M. Silva, D. Pratas, R. Antunes, S. Matos, and A. J. Pinho, "Automatic analysis of artistic paintings using information-based measures," Pattern Recognition, vol. 114, 2021, p. 107864.

[5] E. Cetinic, T. Lipic, and S. Grgic, "Fine-tuning convolutional neural networks for fine art classification," Expert Systems with Applications, vol. 114, 2018, pp. 107–118.

[6] R. S. Arora and A. Elgammal, "Towards automated classification of fine-art painting style: A comparative study," in Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), 2012, pp. 3541–3544.

[7] C. S. Rodriguez, M. Lech, and E. Pirogova, "Classification of style in fine-art paintings using transfer learning and weighted image patches," in 2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS), 2018, pp. 1–7.

[8] S. Smirnov and A. Eguizabal, "Deep learning for object detection in fine-art paintings," in 2018 Metrology for Archaeology and Cultural Heritage (MetroArchaeo), 2018, pp. 45–49.

[9] C. Sandoval, E. Pirogova, and M. Lech, "Two-stage deep learning approach to the classification of fine-art paintings," IEEE Access, vol. 7, 2019, pp. 41 770–41 781.

[10] I. Nunez-Garcia, R. Lizarraga-Morales, U. Hernandez-Belmonte, V. jimenez Arredondo, and A. Lopez-Alanis, "Classification of paintings by artistic style using color and texture features," Computación y Sistemas, vol. 26, 12 2022.

[11] R. G. Condorovici, C. Florea, and C. Vertan, "Automatically classifying paintings with perceptual inspired descriptors," Journal of Visual Communication and Image Representation, vol. 26, 2015, pp. 222–230. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1047320314001977

[12] Q. You, J. Luo, H. Jin, and J. Yang, "Robust image sentiment analysis using progressively trained and domain transferred deep networks," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29, 09 2015.

[13] Q. Zhao and R. Zhang, "Classification of painting styles based on the difference component," Expert Systems with Applications, vol. 259, 2025, p. 125287. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424021547

[14] B. Menai and M. Babahenini, "The effect of optimizers on CNN architectures for art style classification," International Journal of Computing and Digital Systems, vol. 31, 2023, pp. 353–360.

[15] S. Banerji and A. Sinha, "Painting classification using a pre-trained convolutional neural network," in Proceedings of the 2nd Workshop on Computer Vision Applications (WCVA) at the Tenth Indian Conference on Vision, Graphics and Image Processing (ICVGIP), 2016, pp. 168–179.

[16] B. Saleh and A. Elgammal, "Large-scale classification of fine-art paintings: Learning the right metric on the right feature," International Journal for Digital Art History, no. 2, Oct. 2016, pp. 70–93.

[17] E. J. Crowley and A. Zisserman, "In search of art," in Workshop on Computer Vision for Art Analysis, ECCV, 2015, pp. 54–70.

[18] "Google reverse image search," https://www.google.com/imghp, accessed: May 29, 2025.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[20] J. Deng et al., "Imagenet: A large-scale hierarchical image database," in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[21] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818–2826.

[23] A. Batool and Y.-C. Byun, "Lightweight efficientnetb3 model based on depthwise separable convolutions for enhancing classification of leukemia white blood cell images," IEEE Access, vol. 11, 2023, pp. 37 203–37 215.

[24] R. R. Selvaraju et al., "Grad-cam: Visual explanations from deep networks via gradient-based localization," in 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 618–626.

A Lego-inspired, Modular Agent-Based Metasystem for Emergency Departments

Francisco Mesas*, Manel Taboada*, Dolores Rexachs†,
Francisco Epelde‡, Alvaro Wong†, and Emilio Luque†
*Escuelas Universitarias Gimbernat (EUG), Computer Science School, Universitat Autonoma de Barcelona,
Sant Cugat del Vallès, Barcelona, Spain
{francisco.mesas,manel.taboada}@eug.es
†Computer Architecture and Operating System Department, Universitat Autonoma de Barcelona, Barcelona, Spain
{dolores.rexachs,emilio.luque,alvaro.wong}@uab.cat
‡Consultant Internal Medicine, University Hospital Parc Tauli, Universitat Autonoma de Barcelona Sabadell, Barcelona, Spain
fepelde@tauli.cat
https://webs.uab.cat/hpc4eas/

***Abstract*—Emergency Departments (EDs) are complex systems that require coordination of medical personnel and resources to manage situations effectively. This research addresses the basic principles for designing a modular system that allows the creation of computational models to improve service quality using available resources. Based on the accumulated knowledge of experts in the field of ED, the modular system ensures that each component accurately reflects the particular features present in various emergency healthcare settings, ensuring its adaptability. By applying Agent-Based Modeling and Simulation (ABMS), an analysis of the agents involved—such as patients, doctors, resources, and computer systems—is considered. ABMS, known for its ability to individually adapt to each agent, allows the design of customized environments that meet the unique needs of various regions and healthcare structures. Inspired by the modularity and versatility of Lego® blocks, this ABMS system seeks to transform a monolithic approach into an adaptable tool. Through the introduction of a metasystem, modular agent definitions, and a testing platform, the proposed "agent box" facilitates the construction and validation of computational models. These enhancements ensure easier adaptation, scalability, and robustness, potentially improving emergency care quality and facilitating strategic decision making in this critical service.**

Keywords-Agent-Based Modeling and Simulation (ABMS); Emergency Department (ED); Emergency Healthcare Systems; Modular System; Automated Testing; Model Validation; Decision Support Systems (DSS); Healthcare Simulation.

I. Introduction

The Emergency Departments (EDs) currently face an increasingly complex landscape due to the saturation experienced in recent years, a phenomenon that highlights both the growing demand for emergency medical care and the need to provide quick and efficient responses in a pressured environment [1][2].

Simulation stands out as a compelling tool in the context of EDs, allowing us to perform analyses of hypothetical scenarios through "what if" questions [3]. This technique enables anticipation and preparation against potential adverse situations, helping to improve response capacity to the increasing demands these services may face, especially in critical situations such as pandemics or flu outbreaks, which have recently tested their capacity [4]. For example, through simulation, it is possible to assess the impact that increasing patients' arrival in the ED would have on waiting times and service quality, thus allowing us to devise effective strategies to reduce saturation and ensure adequate care.

In the realm of EDs, simulation techniques are crucial for the analysis of complex processes. Among these, Discrete Event Simulation (DES) and Agent-Based Modeling and Simulation (ABMS) stand out for their effectiveness. DES focuses on the analysis of discrete events over time, allowing evaluation of how each event impacts the flow and operation of the emergency system. It enables us to understand sequences and resource use, but might not capture all human interactions. In contrast, ABMS offers a more dynamic and detailed perspective by modeling the behavior and interactions between multiple individual agents, such as patients, doctors, and nurses, as well as their environment. One of the important characteristics of ABMS is the "emergent properties", in other words, "the higher-level system properties emerge from the interactions of lower-level subsystems (Agents)", making it the ideal choice according to various studies [5][6].

It is important to differentiate our simulation system from machine learning and artificial intelligence techniques, which often operate as "black boxes" [7]. These systems depend on opaque algorithms and large volumes of data to "learn". We work on another aspect, where we collaborate with experts to precisely define the variables that influence the system and the behavior of each element. In this way, we can understand the operation of the simulator with total transparency, as if we had a "God mode" that lets us to interpret each result and understand the reason for each event.

The variability in the operation of EDs is clearly manifested in the differences in regulatory systems and certifications, e.g., in the field of phlebotomy, we observe a regulatory divergence between the United States and Spain [8]. In the former, certification is a mandatory requirement, while in the latter, it is not required. When considering the implementation of simulation techniques to improve EDs, these structural and regulatory variations must be taken into account. Therefore, it is necessary to adapt simulation solutions to the specific

characteristics of each emergency system.

Models and simulators developed up to date by the Research Group of the Universitat Autonoma de Barcelona (UAB) "High Performance Computing For Efficient Applications and Simulation" (HPC4EAS) and other researchers operate in a monolithic manner, which creates certain limitations in terms of adaptability. A monolithic system, by definition, is one in which different components of the software are tightly integrated or unified into a single program developed for a specific case, which can complicate its adaptation to new contexts. Faced with this situation, two initial solutions are presented: modify the existing monolithic model to adapt it to new needs, despite the difficulties this may entail, or develop a new simulator from scratch.

Given the application of the ABMS concept in these systems and inspired by the modularity and versatility of Lego® blocks, a third proposal emerges: to disaggregate those monolithic simulators to create an "agent box." This box would contain all agents that could be involved in the ED, including medical personnel, patients, administrative staff, and physical resources. This strategy allows the simulator to be fluidly adapted to different ED environments and also to expand the agents and their interactions within the system, a solution that will allow handling the complexity of these environments.

To achieve this, agents must be individually defined and modularized into separate files or libraries, facilitating reuse and simplifying future modifications. In addition, incorporating an external Python-based metasystem to automate testing and validation will guarantee modularity, early error detection, and robustness of each component. This structured approach, complemented by a clear separation of agent behavior from their interactions, ensures greater scalability and adaptability, thus supporting continuous improvement and expansion of the ED simulator.

The remainder of this article is structured as follows: Section II provides a concise summary of the previous works carried out by the HPC4EAS group; Section III examines the fundamental properties of the proposed metasystem; Section IV reviews similar research, Section V explains how the metasystem is working and the changes to make it, Section VI decomposes the actual and specific agents in the simulator, and analyzes the role of the agents, and Section VII describes conclusions and future plans for the research work.

II. Previous Work

This section presents the results of projects carried out by HPC4EAS, research group from the Department of Computer Architecture and Operating Systems at the Universitat Autònoma de Barcelona (UAB). The work has been conducted in collaboration with the staff of the ED at Sabadell Hospital (Corporació Sanitària Parc Taulí), a reference center in the Catalan health system. In addition, various studies related to the topic are integrated.

The research group has developed both a conceptual model and a computational model (we can consider that the simulator is the implementation of the computational model) that utilizes the ABMS technique, distinguishing between active and passive agents. Active agents are capable of making decisions and acting autonomously, representing individuals, such as doctors, nurses, and patients, who interact and respond to the dynamics of the ED. On the other hand, passive agents do not take initiatives on their own but are essential for executing predetermined processes and enabling interactions, such as hospital information systems, communication networks, and laboratory services. These agents interact within a virtual environment that simulates the areas and processes of an ED, managing different levels of urgency and priority in patient treatment. The interaction between these agents and the modeled environment allows for the replication of the particularities of a real emergency service [9].

The project has evolved through several key phases, starting with the development of a conceptual model derived from a meticulous analysis of the elements of the ED, including the triage system that stratifies urgency into five severity levels, specifically the Manchester Triage System [10], with level I being the most critical and level V the least. In addition, to mapping other operational aspects and examining the interactions among agents to reproduce the system behavior, the simulator also distributes patients in the ED into two zones, Zone A and Zone B, according to this severity classification, assigning patients with levels I to III to Zone A for priority care, while those with less severity, levels IV and V, are placed in Zone B, designed for less urgent situations. This segmentation is important for the management of patient flow [11].

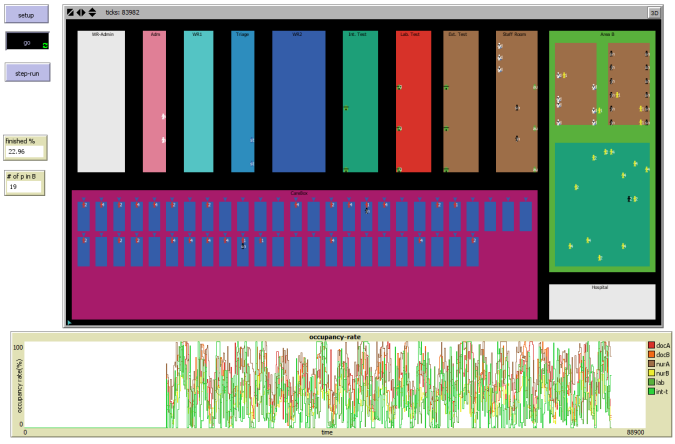


Figure 1. Simulator of the Sabadell Hospital ED, created with NetLogo.

After establishing the conceptual model and understanding the mechanisms of the ED operation, the next step was the creation of the computational model. This model translates the theory and observations of the conceptual model into algorithms and data structures that can be processed by computer systems. In this phase, the behaviors of both active and passive agents are programmed, and the interaction rules and operational procedures, such as the triage system, are encoded. The goal of the computational model is to faithfully

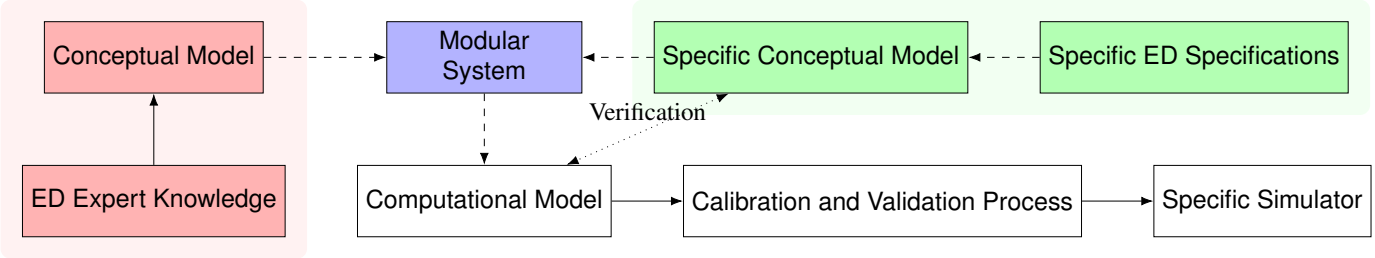


Figure 2. Diagram of the design process of a simulator using a modular system for a specific ED.

reflect the dynamics of a real ED, allowing the simulation of different scenarios and their possible outcomes as can be seen in Figure 1. This model becomes a sophisticated tool to predict the behavior of the ED at Sabadell Hospital. This scenario was represented using NetLogo software [12], a modeling environment designed for ABMS, which provides the possibility to accurately design and simulate the operations of a hospital ED.

The system also incorporates stochastic elements, such as variability in patient arrival times and treatment durations, to account for the unpredictability of ED operations. Furthermore, the simulation outcomes align closely with actual data, providing insights into potential bottlenecks or inefficiencies in resource allocation. By testing different scenarios, such as increasing staffing levels during peak hours or reallocating resources between zones, the simulator serves as a decision support system for hospital managers.

In the work conducted by various members of the research group, the simulator has been adapted and applied to analyze how to optimally use the limited resources available in the ED [13], to generate information about specific scenarios that, while possible, rarely occur in reality [14], and thus learn about the best way to manage them, or also to analyze, model, and simulate the transmission of the Methicillin-resistant Staphylococcus Aureus (MRSA) virus [15], and its effects on the operation of the ED, in order to explore the potential benefits of adopting preventive measures.

III. General Characteristics of the Metasystem: Lego® System

Building on existing work and advancements in the simulation and modeling of EDs using ABMS techniques, we propose the creation of a metasystem, named the Lego® System. This system aims to manage the modularity of ABMS to develop an adaptable simulation environment.

The metasystem will originate from a conceptual model developed with the collaboration of ED specialists and the disaggregation of current simulators, which will facilitate the definition of standard modules that can be used in various health environments. This will allow for an efficient transition from a specific conceptual design to a computational configuration within the metasystem when it is necessary to develop a computational model for any ED. With the computational model ready, the necessary calibration and validation process must involve the use of specific data that the hospital can offer, and discussions should be held with them to determine the available data to guide this calibration to conclude with a specific simulator. The calibration process uses data that the hospital can provide, such as the number of patients arriving per hour, the approximate average time required to perform certain tasks, the number of nurses, doctors, and so on. With all this information, high-performance computing is used to test the different combinations and see which one is closest to reality, and whether it is a valid enough approximation. It is important to understand that large volumes of data are not used, nor is data that cannot be understood. The entire simulation process is based on the experience and knowledge of ED experts. In Figure 2, the process of the actual ED is detailed. The section to be analyzed is highlighted in red, while the specific areas of an ED intended to be modeled with the metasystem are highlighted in green. Dotted lines denote the new flow of requirements for the Modular System in blue, which produces a Computational Model.

The goal is to develop a platform that facilitates the creation of computational models of EDs, through an interface functionally similar to Lego®, which allows us to work with "blocks". These blocks represent the various agents and processes involved in the operation of ED and are designed to be customizable. Flexibility is a key point; the system must allow for the combination of blocks in multiple ways, thus adapting to the operational particularities of various EDs. For example, it is possible to explore the impact of variations in staff roles, e.g., analyzing the consequences of assigning more or fewer responsibilities to a nurse or simulating scenarios where another team member assumes these tasks. With monolithic systems, such adaptations are costly; consequently, when different hospitals need to be analyzed, it becomes expensive.

To carry out the disaggregation of these components, it is important to analyze the state variables that will characterize the different agents, as well as define how transitions between these states will occur. In this context, three main categories are established: two corresponding to active agents and one to a passive agent, which will allow us to explore differences in their operation.

Among the active agents, we find common elements that all of them share, such as:

- **Identifier:** Each agent has a unique identifier that allows the system to recognize it in each temporal iteration.

- **Location:** Records the current location of the agent in the ED, which can vary from admission to the treatment area or specific tests.
- **Action:** Agent actions, such as waiting to be called, receiving instructions, or moving between different areas of the ED. These actions will vary by agent.

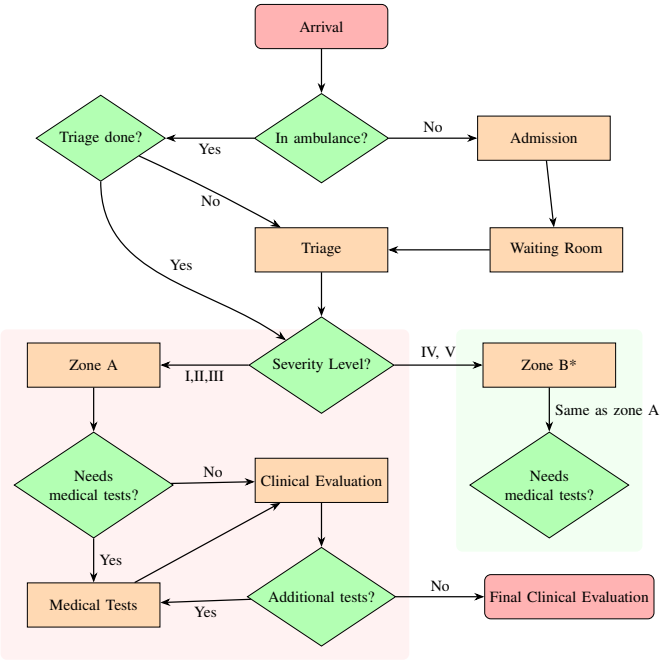


Figure 3. Diagram of the process patients go through in the ED.

For the particular case of patients, there are complex state variables and transitions. We can distinguish three specific state variables; personal details, priority level, and communication level. Patients are recognized as one of the most crucial agents in the ED. Their **personal details**, such as age, gender, culture, and religion, are collected and considered to provide tailored treatment. The assignment of a **priority level** based on triage determines the urgency of medical attention, while the **communication level** between the patient and the ED staff is an indicator of the effectiveness of the interaction.

The diagram showed in Figure 3 illustrates the process a patient undergoes upon arrival at an ED. It begins with their arrival, a critical point where their unique identifier is assigned, and their initial location or time of arrival is recorded. If they arrive in a medicalized ambulance, triage has already been conducted on-site; otherwise, if they arrive on their own or in a nonmedicalized ambulance, the process starts with their admission.

Priority level assignment occurs during triage, guiding the patient through the system to either to treatment areas, a separated zone (Zone B in the figure) with one specific waiting room and attention boxes for less severe cases (patients with priority level IV or V) or directly to a carebox (Zone A) for patients with more critical conditions (patients with priority level I, II or III).

The level of communication is important at each stage, from assessing whether medical tests are needed for taking decisions about additional treatments. An evaluation cycle of treatment and possible re-evaluation continues until a resolution point is reached: the patient is discharged or further measures are taken based on their needs.

Each step of the process reflects the interaction between the patient's state variables and the actions of the ED system.

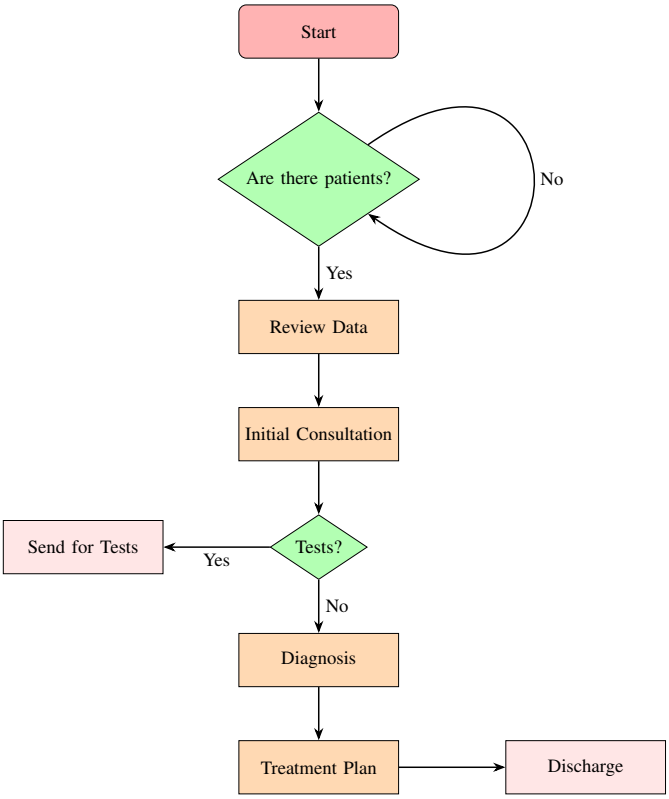


Figure 4. Diagram of the process that doctors undergo in an ED.

Continuing with our investigation of active agents in the ED, doctors are a key player whose state variables reflect their role in the care environment. In contrast to patients, doctors' actions are defined by a series of clinical steps and defined tasks that are dictated by the patient's priority level. While specifics of each patient's condition are important for individual care, they are less important for large-scale simulation purposes, where the emphasis is on overall patient flow and resource allocation.

Doctor's actions in the ED range from being inactive, which could mean waiting for the next patient, to more interactive actions, such as asking a patient to come forward, requesting detailed information, making a preliminary diagnosis, and ordering specific tests or treatments. A doctor may also be in an active waiting phase, awaiting the results of tests they have ordered, then making decisions based on those results, such as ordering the patient's discharge from the ED or making a final diagnosis to be entered into the Computer System, as evidenced in Figure 4.

The level of experience of the doctor, classified as low, medium, or high, influences their actions, and is a critical component that impacts the efficiency of work within the ED. A highly experienced doctor may be able to make quicker diagnoses or handle more complex cases in less time. For this reason, the metasystem incorporates a state variable to manage such issues. This is not reflected in the schema because the process remains the same; however, it depends on the state of each agent.

The operation of the Information System (IS) in an ED is essential for efficient and accurate care. It is part of an interactive process where the key decisions that the IS makes are in response to the received requests. Initially, the system checks for pending requests and, based on this, proceeds to obtain reports, register patients, and issue medical alerts. Decisions about whether patient data already exist lead to further actions, such as registering new data or adding them to the existing system. The workflow facilitates the processing of information and the continuous updating of medical records.

As a passive agent, the IS depends on interactions with active agents, such as the medical or administrative staff of the ED, to change state. The system's propensity for errors is categorized into low, medium, and high levels, which can affect the operability of the ED.

The IS, as a passive agent within the ED, plays a significant role in coordinating between the different components of the healthcare system. The ability to process and issue information accurately is necessary to maintain a smooth workflow and to ensure that patients receive the necessary care at the appropriate time. It is a component that supports all the operations of the ED, from admission and triage to the patient being discharged.

The interaction between doctors and patients, mediated by the information system, is a delicate dance of consultations, diagnostics, and decisions that advance the patient through the care process, as reflected in the discussed figures.

IV. Related Work

The adaptability of simulation models to various health systems seeks to improve EDs. This flexibility will allow the implementation of the proposed modular metasystem, which can be adjusted to the specifics of different emergency care settings.

There are initiatives by research groups that have used simulation to enhance the effectiveness of EDs. The 3S Research Group and the Shelford team in England have conducted simulations at the University Hospital of Dublin [16] and in specific cases of the ED in London [17] respectively, offering valuable reference models for our proposal.

Moreover, it is important to analyze health systems in their social and economic context, as factors such as funding and access to health services vary significantly between countries [18].

Analyzing how health systems function provides a more global perspective, it is necessary to evaluate the different health models found in each country. According to the World Health Organization, there are four main models [19] that have their distinctive characteristics regarding funding, management, and coverage.

The Beveridge model, implemented in countries like Spain, Portugal, and Finland, is characterized by its funding through income taxes, with the government assuming total control of healthcare management and providing universal coverage. This approach contrasts with the Bismarck model, prevalent in Austria, Germany, and Switzerland, where funding comes from mandatory contributions to social insurance funds. Although the state regulates healthcare entities, coverage depends on the individual's employment status, and copayments are included for certain services.

On the other hand, the National Health Insurance Model, found in Japan, Canada, and South Korea, combines elements of both previous models that offer universal and equitable coverage, regardless of employment affiliation [20]. This model allows a greater choice of healthcare providers. Lastly, the Out-of-Pocket model is distinguished by the absence of collective funding, leaving individuals to face healthcare costs without a financial safety net, which limits universal access to health.

Each model reflects a different philosophy regarding the role of government, individual responsibility, and the principles of social solidarity. While the Beveridge and National Health Insurance models focus on universal coverage guaranteed by the state, the Bismarck and Out-of-Pocket models present a more segmented or individualized approach to healthcare coverage, which causes different types of ED operations in each case. These differences are reflected in Table I.

These factors can lead to different roles and internal functioning aimed at optimizing available resources. For example, the approach to phlebotomy in the United States, where there is specific training for this skill, differs from other countries with different training approaches, such as in Spain, where nurses are responsible for this process. With the new modular "Lego®" system, the need to adapt the simulator to these variations is no longer a problem, as the modules can be customized and reconfigured to reflect any health system.

There are tools seeking something similar like VisualizER, a DES tool that exemplifies how simulation can be applied to optimize EDs [21]. Although it allows effective simulation of emergency operations, it does not offer the capability to model the individual behavior of agents, which is a crucial component for anticipating unforeseen events.

Our proposal for an ABMS-based metasystem advances beyond existing DES solutions by leveraging the advantages of ABMS for creating modules that allow the result to emerge from the individual interaction of agents. This feature enables understanding and managing the often unpredictable dynamics of EDs, thus providing an adaptable system for healthcare professionals.

V. Operation of the metasystem

In the metasystem for modeling EDs, it is crucial to have an interface or set of tools that facilitate customization of the system to the specific needs of different hospital environments.

TABLE I. COMPARISON OF HEALTHCARE MODELS

Model	Funding	Control and Management	Coverage and Features
Beveridge	Income taxes	Government	Universal, public
Bismarck	Social insurance	State regulates	Employment-dependent, copayments
National Insurance	Taxes and insurances	Mixed	Universal, greater choice of providers
Out-of-Pocket	Private	Individual	Limited access, no financial protection

This functionality allows users to manipulate and redefine the stages and agents involved in the process easily.

Each component of the health system, represented by an agent, can be selected, configured, and placed within a workflow. The proposal is to drag and drop components, thus modeling the flow of the care process according to the criteria of each ED. Through this interface, for example, a new triage procedure specifically designed to respond to pandemic emergencies could be integrated, adjusting the metasystem to reflect changes in protocols. To achieve this, it is necessary to establish a basic form of communication between agents through primitives that are easily interchangeable among them and capable of adaptation. Examples of such primitives include conversing and utilizing objects, which are essential for defining each agent's own internal mechanism.

There will always be a need for a series of forms or commands that allow specifying and modifying the properties and behaviors of each agent. This functionality is relevant when wanting to add a new agent, e.g., a 'pandemic triage agent.' Here, the person in charge has the opportunity to access a library of predefined agents and select the one that fits their needs. Subsequently, the functions of this agent can be customized by adjusting parameters and behaviors.

In the event that a necessary agent is not predefined, tools are provided for users to create one from scratch. This allows the system to be adaptable, enabling each healthcare center the ability to mold the metasystem to their operational reality.

Figure 5 shows the structure of an ED. At the bottom, the set of "agent box" can be observed, a collection of roles and functions from which one can choose to assign to the different phases of the care process. For example, during the admission phase, a distinction is made between a process for a public ED and a private one. In the triage phase, a nurse specialized in this task is required. However, if the situation demands the incorporation of a triage nurse with greater experience due to an increase in the complexity of cases or the need to expedite the process, this new type of professional could be added. This process would be carried out by duplicating the configuration of the existing triage nurse and adjusting her parameters of behavior and performance according to the additional experience she brings to the process.

Consider the scenario where an ED in Spain is public, in such a case, this specific setting can be selected to work within the system. Similarly, settings for other stages, such as Triage, Waiting Room, and Performing Additional Tests can be customized to specify the capabilities and processes for each element of the system. This customization process allows the system to transition smoothly from the general agent-based configuration shown with the box to a more specific configuration that can be shown with the selected agent boxes in the diagram of the Figure 5.

This tailored approach ensures that each component of the ED could operates optimally according to the defined roles and requirements, enhancing both efficiency and patient care.

VI. DECOMPOSING THE ACTUAL SIMULATOR

This section gives a detailed discussion of the design ideas, modular structure, and important implementation details used in the creation of the ED simulator in NetLogo. The goal is to show how an ABMS methodology combined with the simple programming environment of NetLogo results in a versatile and reliable system for simulating and evaluating ED activities.

The current configuration of each agent in the NetLogo model, such as a patient or a member of the medical staff, contains a *incoming-queue* variable that keeps track of all messages received to that agent. This queue-based system is the foundation for how agents coordinate actions and exchange information. This method ensures that all communication remains within the local queue of each agent, preventing direct connections between agents and preserving a loosely connected architecture. The simulation can handle messages more simply and scale to accommodate more actors by breaking interactions up into discrete time increments.

The simulator's communication infrastructure will become more modular and adaptable in the future, keeping the queue-based method while adding additional features like a "guarded-send" mechanism that will handle the cases in which a recipient is no longer active or is temporarily unavailable, preventing silent failures or lost messages and keeping the main idea with the "Lego®-style" modularity principle, the new system will allow for interchangeable communication styles (e.g., point-to-point messaging, publish-subscribe broadcasts) so that components can be replaced or upgraded independently. This would allow agents, in high-traffic simulations, to prioritize messages and filter messages by content so important information is handled first. This layer, an optional layer, allows agents to "subscribe" to particular events; these might include lab results or patient arrivals. These enhancements aim to preserve the original simulator's advantages—loose coupling, transparent debugging, and straightforward implementation—while offering a more dependable platform for simulating increasingly complex ED scenarios.

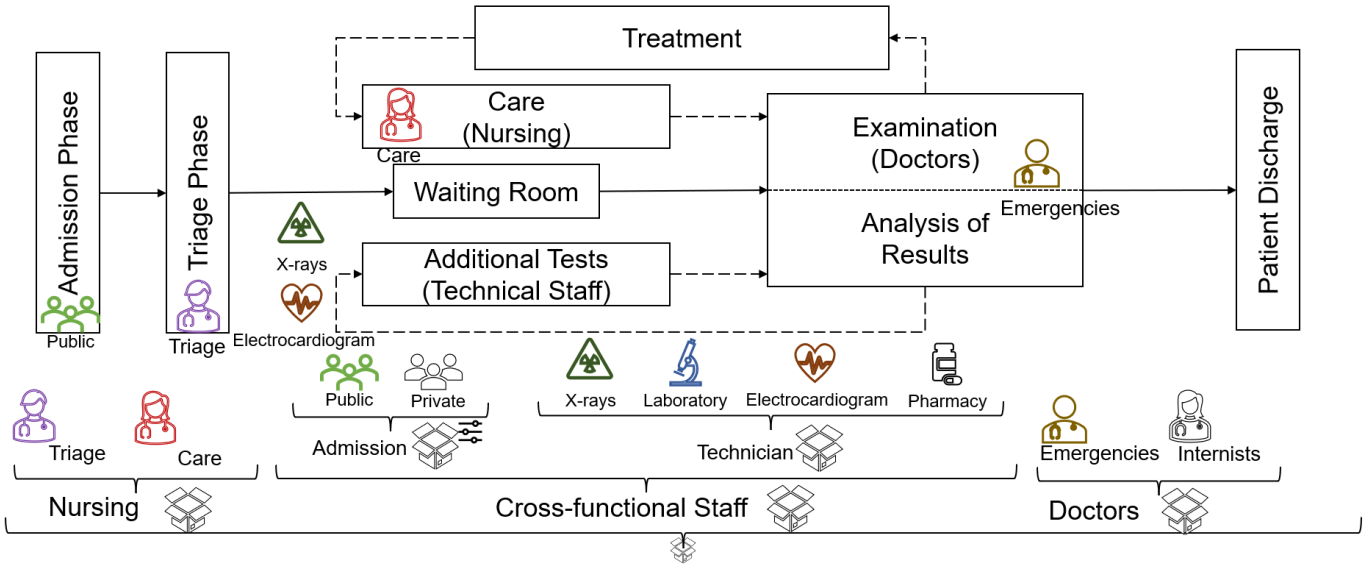


Figure 5. Example of Modules Utilized in an ED.

A. Role of Agents in an ED

The functioning of an emergency department is a complex process that involves teams and resources in coordination, well-timed, and appropriate care for patients. Table II shows the various ED agents and their relation to the whole functioning within the context of the actual simulator.

Admissions staff are those who first report on the arrival of patients as they come in the ED, documenting necessary information about such a patient. This is an important step, as it forms the basis for triage and further care. Once these patients have been registered, the Triage Nurses take on the process of ensuring proper use of resources and timely care of patients. At this stage, your symptoms are evaluated and the prioritization of the treatment order is created.

It is designed based on a digital platform for data management and real-time communication that forms the very basis of the ED: the Information System. It interlinks various roles within the ED, with a view to managing patient flow, improving resource tracking, and assisting well-informed decision making.

The doctors and nurses have their zones of operation in the ED. Physicians in Area A usually see patients who have conditions that are urgent but not life-threatening, while physicians in Area B are more concerned with patients whose cases are less serious or highly specialized. Similarly, nurses in each area provide care specific to the needs of the area, such as medication administration, assisting with procedures, and coordinating activities within the zone.

Key facilities such as careboxes and testing rooms play a critical role in diagnosing and treating the patient. Careboxes refer to individual facilities for patients, where clinical diagnosis and care would be provided; the Test Room is a specialty room intended to house specific tests and examinations of the patient that will be performed, such as imaging and laboratory tests. Each one of these facilities helps ensure that delays do not occur, allowing timely delivery of care.

Auxiliary Staff play a supporting role for the ED nurse in managing all behind-the-scenes logistical work of moving patients around, replenishing supplies, among others. This complements the care process at each juncture where waiting occurs to facilitate patient flow.

Ambulances are the modes of transport that meet the needs within the ED, especially in the transport of critically ill patients and the transfer of patients to specialized care facilities. Outside the ED, Hospitals play an important role in providing extended care for patients who require inpatient services. However, the availability of beds and policies regarding admissions may directly impact ED operations.

B. Metasystem for NetLogo

The current simulator developed with NetLogo has certain characteristics that make new adaptations complex. One of them is the limitation of having all the code in a single file, which makes it very difficult to adapt new agents or change interactions due to the interdependence generated in the code. While it is true that the tool works effectively with ABMS principles, for complex systems it is easy for the code to become disorganized and monolithic.

If we look again at the bottom of Figure 5 we see how these "boxes" and these agents could be the structure of the system, where we can configure the corresponding agents in each box. This concept, if we transfer it to the computing world and specifically to the programming world, we could consider that we have a series of libraries or folders that contain the specifications of each of the elements, in this case the agents. It is clear that the agents will continue to have an interdependence between them because they have

TABLE II. DESCRIPTION OF AGENTS IN AN EMERGENCY DEPARTMENT

Agent	Description
Admissions	Staff in charge of receiving new arrivals and collecting the necessary information to start each patient's journey in the ED. They ensure that everyone is documented appropriately.
Triage Nurses	Personnel who perform quick assessments on patients upon arrival, determining treatment priority based on symptoms and severity. Their evaluations guide the sequence of subsequent treatments.
Information System	A central platform that keeps patient information updated, handles staff requests, and monitors resource usage in real-time. It smooths communication among various roles in the ED.
Patients	Individuals with varying medical conditions who come to the ED. They go through the processes of admission, triage, testing, treatment, and, if necessary, further hospitalization or discharge.
Doctors in Area A	Physicians responsible for handling moderate to urgent cases. They diagnose patients, establish treatment plans, and collaborate with nurses, technicians, and other support staff as necessary.
Doctors in Area B	Physicians stationed in a different section of the ED, typically managing lower-acuity or specialized groups of patients, but with the same responsibilities of evaluation and care coordination.
Nurses in Area A	Nurses who carry out direct patient care in a designated zone, administering medications, assisting with procedures, and ensuring a smooth overall workflow.
Nurses in Area B	Nurses dedicated to a separate section of the ED, responsible for patient care tasks similar to those in Area A but often focusing on less critical cases.
Careboxes	Treatment cubicles or rooms where patients receive clinical assessments and interventions. Each carebox is typically operated by a specific nurse or care team.
Test Rooms	Specialized areas (imaging, laboratory, etc.) where patients undergo diagnostic procedures. Efficient management of these rooms helps reduce testing delays and bottlenecks.
Auxiliary Staff	Personnel handling vital support functions, such as transferring patients from one location to another, restocking supplies, and assisting in non-clinical tasks.
Waiting Rooms	Spaces where patients remain before receiving care or while awaiting results. Proper supervision here is essential for patient comfort and timely redirection to the next step.
Ambulances	Vehicles dedicated to bringing critical cases to the ED and transporting patients out when they require specialized services elsewhere. Their availability can significantly impact system flow.
Hospitals	Broader facilities that provide inpatient care for individuals requiring extended treatment beyond the immediate scope of the ED. Bed availability and admission policies often influence ED throughput.

to communicate, but in this way the essence of an ABMS is preserved, with strictly the definitions of the agents and without unrealistic adaptations for the functioning of the system, thus being definitions faithful to reality.

Furthermore, it is important to differentiate the internal behavior of the agent and the interactions with other agents. In order to respect the proposed system in which everything is segmented by pieces, it is necessary to create an independent file that defines the interactions that these agents have to do and that is easy to adapt to new systems.

To integrate all the information that NetLogo can interpret, a program is needed to consolidate all the files. This process, known as flattening, transforms the source code by eliminating nested control structures and reducing it to a sequence of simple program statements. Although the literature, such as László [22], applies this technique in the context of obfuscation to hinder reverse engineering, in our case it is used to unify the code and facilitate its interpretation.

C. Testing Metasystem

Each module of the metasystem, organized in independent directories, requires a specific set of tests to validate its individual behavior before integrating it into the full simulation. To facilitate this task that is not yet implemented, Python is proposed as an external environment that manages and automates unit tests on components developed in NetLogo.

The procedure will consist of preparing specific scenarios using Python scripts that invoke NetLogo in headless mode (without a graphical interface). The initial data for the test will be generated from Python in standard format (e.g., JSON or CSV) and loaded into NetLogo, where the specific module will be executed isolated from the rest of the system using mocks when they are necessary. Finally, the results generated by NetLogo will be retrieved again by Python, allowing automatic validations to be performed through unit tests.

This modular approach facilitates early error detection and ensures that each component meets the defined requirements, significantly reducing the complexity of maintenance and final integration of the system.

VII. CONCLUSION AND FUTURE WORK

Simulation in EDs is greatly beneficial in addressing the increasing complexity and saturation these services currently experience. The ability to analyze problematic situations in advance through the simulation of hypothetical scenarios allows EDs to prepare and respond effectively to adverse situations, especially in critical contexts, such as pandemics or disease outbreaks. Simulation not only improves response capacity to growing demand, but also contributes to strategic planning of EDs.

ABMS stands out as the appropriate tool for simulating EDs, surpassing DES in terms of the ability to model the complexities of such systems. ABMS, with its "emergent properties", allows for a detailed representation of interactions among multiple agents, such as patients, doctors, and nurses, and their environment, capturing the essence of human processes.

The development of simulators using ABMS represents a significant advance that allows models to be adapted to different EDs. The transition from monolithic models to a

LEGO-type modular system, known as an "agent box," facilitates the adaptation and expansion of simulators to meet various configurations and needs of ED. This modularity allows efficient customization and reconfiguration, reflecting any health system and its operational particularities. Furthermore, reorganizing the monolithic code structure of the simulator into modular files for individual agents, combined with the proposed flattening process, will facilitate easier adaptation and scalability of the simulator. The introduction of an external Python-based metasystem for automated testing will ensure modularity and robustness, allowing early detection of errors.

This simulation proposal differs from other solutions, such as DES and tools like VisualizER, in its focus on agent adaptation and modeling. Through the "emergent properties" of ABMS, it is possible to model individual behavior and interactions between agents, a crucial component for managing the often unpredictable dynamics of EDs. This provides an adaptable system for healthcare professionals, allowing for more effective ED management.

However, there are limitations and potential future directions for the expansion of this technology. One is the number of predefined modules in the "agent box," which could be addressed by creating a common repository where modules adapted to new needs and contexts are shared and updated. In addition, expanding the use of modular systems in EDs to other healthcare and geographic contexts could provide valuable information and improve the efficiency of EDs worldwide. Moreover, the test and validation of the metasystem proposal have to be done.

In conclusion, the proposal of an ABMS-based metasystem for ED simulation contributes to better understanding and management of these services. Through the ability to model the complexity of human interactions, this technology opens new possibilities to prepare EDs for current and future challenges. The evolution towards modular systems and collaboration in the development of modules can further enhance simulation capabilities, offering continuous improvement of EDs.

In the future, the Delphi method, a process used to arrive at a group opinion or decision by surveying a panel of experts [23], will be necessary to build a comprehensive conceptual model, develop the meta-model and a comprehensive testing metasystem. This analysis will involve multidisciplinary collaboration with clinical expertise and the use of ABMS.

Acknowledgments

This research has been supported by the Agencia Estatal de Investigación (AEI), Spain and the Fondo Europeo de Desarrollo Regional (FEDER) UE, under contracts PID2020-112496GB-I00 and PID2023-146978OB-I00.

References

[1] F. M. Cervilla *et al.*, "Metasystem for modeling emergency departments," in *Proceedings of the Sixteenth International Conference on Advances in System Modeling and Simulation (SIMUL 2024)*, Venice, Italy: IARIA, Sep. 2024, pp. 44–50, ISBN: 978-1-68558-197-8.

[2] M. Samadbeik *et al.*, "Patient flow in emergency departments: A comprehensive umbrella review of solutions and challenges across the health system," *BMC Health Services Research*, vol. 24, no. 1, p. 274, 2024, ISSN: 1472-6963. DOI: 10.1186/s12913-024-10725-6.

[3] F. McGuire, "Using simulation to reduce length of stay in emergency departments," in *Proceedings of Winter Simulation Conference*, 1994, pp. 861–867. DOI: 10.1109/WSC.1994.717446.

[4] M. E. Reschen *et al.*, "Impact of the covid-19 pandemic on emergency department attendances and acute medical admissions," *BMC Emergency Medicine*, vol. 21, no. 1, p. 143, 2021. DOI: 10.1186/s12873-021-00529-w.

[5] T. Monks *et al.*, "Strengthening the reporting of empirical simulation studies: Introducing the stress guidelines," *Journal of Simulation*, vol. 13, no. 1, pp. 55–67, 2019. DOI: 10.1080/17477778.2018.1442155.

[6] M. Yousefi, M. Yousefi, and F. S. Fogliatto, "Simulation-based optimization methods applied in hospital emergency departments: A systematic review," *SIMULATION*, vol. 96, no. 10, pp. 791–806, 2020. DOI: 10.1177/0037549720944483.

[7] A. Barredo Arrieta *et al.*, "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, pp. 82–115, 2020, ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2019.12.012.

[8] Piazza *et al.*, "It's Not Just a Needlestick: Exploring Phlebotomists' Knowledge, Training, and Use of Comfort Measures in Pediatric Care to Improve the Patient Experience," *The Journal of Applied Laboratory Medicine*, vol. 3, no. 5, pp. 847–856, Mar. 2019, ISSN: 2576-9456. DOI: 10.1373/jalm.2018.027573.

[9] M. Taboada, E. Cabrera, F. Epelde, M. L. Iglesias, and E. Luque, "Agent-based emergency decision-making aid for hospital emergency departments," *Emergencias*, vol. 24, pp. 189–195, 2012.

[10] J. M. Zachariasse *et al.*, "Validity of the manchester triage system in emergency care: A prospective observational study," *PLOS One*, vol. 12, no. 2, e0170811, 2017. DOI: 10.1371/journal.pone.0170811.

[11] L. Zhengchun, D. Rexachs, F. Epelde, and E. Luque, "A simulation and optimization based method for calibrating agent-based emergency department models under data scarcity," *Computers and Industrial Engineering*, vol. 103, pp. 300–309, 2017. DOI: 10.1016/j.cie.2016.11.036.

[12] Center for Connected Learning and Computer-Based Modeling, Northwestern University, *Netlogo*, https://ccl.northwestern.edu/netlogo/, Retrieved: July, 2024.

[13] E. Cabrera, M. Taboada, M. L. Iglesias, F. Epelde, and E. Luque, "Optimization of healthcare emergency departments by agent-based simulation," *Procedia CS*, vol. 4, pp. 1880–1889, Dec. 2011. DOI: 10.1016/j.procs.2011.04.204.

[14] E. Bruballa, M. Taboada, E. Cabrera, D. Rexachs, and E. Luque, "Simulation as a sensor of emergency departments: Providing data for knowledge discovery (work-in-progress paper)," in *The Sixth International Conference on Advances in System Simulation (SIMUL 2014)*, Nice, France: IARIA, Oct. 2014, pp. 209–212. DOI: 10.13140/2.1.2015.2322.

[15] C. Jaramillo, M. Taboada, F. Epelde, D. Rexachs, and E. Luque, "Agent based model and simulation of mrsa transmission in emergency departments," *Procedia Computer Science*, vol. 51, pp. 443–452, 2015, International Conference On Computational Science, ICCS 2015, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2015.05.267.

[16] W. Abo-Hamad and A. Arisha, "Simulation-based framework to improve patient experience in an emergency department," *European Journal of Operational Research*, vol. 224, no. 1,

pp. 154–166, 2013, ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2012.07.028.

[17] T. Godfrey *et al.*, "Supporting emergency department risk mitigation with a modular and reusable agent-based simulation infrastructure," in *2023 Winter Simulation Conference (WSC)*, IEEE, 2023, pp. 162–173. DOI: 10.1109/WSC60868.2023.10407894.

[18] O. O. Oleribe *et al.*, "Identifying key challenges facing healthcare systems in africa and potential solutions," *International Journal of General Medicine*, vol. 12, pp. 395–403, 2019. DOI: 10.2147/IJGM.S223882.

[19] H. K. Mitonga and A. P. K. Shilunga, "International health care systems: Models, components, and goals," in *Handbook of Global Health*. Cham: Springer International Publishing, 2020, pp. 1–20. DOI: 10.1007/978-3-030-05325-3_60-1.

[20] C. Cuadrado, F. Crispi, M. Libuy, G. Marchildon, and C. Cid, "National health insurance: A conceptual framework from conflicting typologies," *Health Policy*, vol. 123, no. 7, pp. 621–629, Jul. 2019, Epub 2019 May 18, ISSN: 1872-6054. DOI: 10.1016/j.healthpol.2019.05.013.

[21] ER-Visualizer, *Visualizer: Emergency room visualization tool*, https://github.com/ER-Visualizer/Visualizer, Retrieved: July, 2024.

[22] T. László and Á. Kiss, "Obfuscating C++ programs via control flow flattening," in *Proceedings of the 10th Symposium on Programming Languages and Software Tools (SPLST 2007)*, Dobogókő, Hungary, Jun. 2007, pp. 15–29.

[23] J. de Meyrick, "The Delphi method and health research," *Health Education*, vol. 103, no. 1, pp. 7–16, Feb. 2003, ISSN: 0965-4283. DOI: 10.1108/09654280310459112.

Stability of Dynamic Fluid Transport Simulations with Mixing Flows

Mehrnaz Anvari
Fraunhofer Institute for Algorithms and Scientific Computing SCAI
Sankt Augustin, Germany
email: Mehrnaz.Anvari@scai.fraunhofer.de

Anton Baldin
PLEdoc GmbH and Fraunhofer Institute SCAI
Sankt Augustin, Germany
email: Anton.Baldin@scai.fraunhofer.de

Tanja Clees
University of Applied Sciences Bonn-Rhein-Sieg and Fraunhofer Institute SCAI
Sankt Augustin, Germany
email: Tanja.Clees@scai.fraunhofer.de

Bernhard Klaassen
Fraunhofer Research Institution for Energy Infrastructures IEG
Bochum, Germany
email: Bernhard.Klaassen@ieg.fraunhofer.de

Igor Nikitin
Fraunhofer Institute for Algorithms and Scientific Computing SCAI
Sankt Augustin, Germany
email: Igor.Nikitin@scai.fraunhofer.de

Lialia Nikitina
Fraunhofer Institute for Algorithms and Scientific Computing SCAI
Sankt Augustin, Germany
email: Lialia.Nikitina@scai.fraunhofer.de

***Abstract*—This work presents physically based simulation of energy distribution and substance composition for dynamic fluid transport problems. The main addressed problem is the stability of the algorithms for solving the resulting systems of differential-algebraic equations. The challenges encountered include system degeneration, the appearance of stochastic degrees of freedom, jumps in thermodynamic functions during phase transitions, and proper scaling of equations. The proposed solution is the identification and optimal configuration of solver parameters, strongly affecting the stability and speed of the simulation. Such parameters include regularizing and weighting constants, dimensioning of dynamic terms and startup procedure, the size of the integration step and their total number. The main output of the paper is the optimal choice of these parameters that allows to speed up significantly the dynamic simulation of fluid transport for realistically large network scenarios.**

Keywords-simulation and modeling; mathematical and numerical algorithms and methods; mixing flows; pipeline fluid transport; stability.

I. Introduction

This work extends the results of our conference paper [1], which considered modeling of mixing flows in dynamic fluid transport simulation. The extension includes a more precise implementation of heaters and coolers, as well as a detailed stability analysis of dynamic simulation with mixing flows.

The contributions of the study: this paper continues a series of our works on modeling of fluid transport networks. Previous works presented stationary [2] and dynamic [3] modeling of fluid transport networks limited to a single chemical composition and constant temperature. In addition, some aspects of stationary modeling of mixing fluids of different compositions and/or temperatures were considered in [4]. In this paper, flow mixing modeling will be considered in more detail, with special emphasis on the thermodynamic layer of the model. In particular, dynamic mixing equations and algorithms for their solving will be presented. The developed approach is implemented in our Multi-phYsics Network Simulator (MYNTS) [5], which is used to solve actual transport scenarios for natural gas [6], hydrogen [7], carbon dioxide [8], water [9] and other fluids.

State-of-the-art: fluid transport modeling is based on the conservation of mass flows in the form of dynamic Kirchhoff equations; Darcy-Weisbach pipeline pressure drop formula, with empirical friction term by Nikuradse [11] and Hofer [12]; equation of state computation by simplified analytical models by Papay [13], Peng-Robinson [14] and Soave-Redlich-Kwong [15] or more complex ISO-norm models AGA8-DC92 [16] and GERG2008 [17]–[19].

A number of previous studies [20]–[26] considered modeling of pipeline fluid transport, both at the universal mathematical level [20], and in various application scenarios. Such scenarios include transport of natural gas [21] [23], steam transport in oil refineries [22], carbon dioxide transport [24]–[26]. All these works are characterized by the presentation of transport equations as laws of conservation of mass, momentum and energy. In the presence of various substances, conservation of molar flows is added, while the general relations of thermodynamics of open systems [27] regulate the relations of energy and temperature.

The main problem: a common drawback of existing solutions is the closed nature of modeling within blackbox systems. If it is necessary to change the modeling, modify or introduce new equations and variables, the system must be reprogrammed. In addition, existing systems experience difficulties in solving large realistic network problems in the presence of numerical instabilities. The novelty of our approach consists in transparent modeling, where the user can freely change the equations and experiment with different forms of representing physical processes in fluid transport networks. We also pay special attention to the stability and performance of solution algorithms, which is especially important for realistic scenarios with a large number of elements.

The aim of this work: to extend transparent and numerically stable modeling to mixing flows present in realistic fluid transport scenarios. In our early works [2], [4]–[10] an

implementation for a stationary solver was considered. The main strategy for ensuring stability was a gradual sophistication of the modeling, from a pure pipe system with linear equations for control elements, constant temperature and fluid composition, to a full problem with nonlinear control elements and physical distribution of temperature and fluid composition. At each step, the solution was used as a starting point for the next step. The disadvantage of this approach is that simplified modeling does not always yield a physical solution and sometimes gives a bad starting point for the next iterations. Also, theoretically, direct solution of stationary equations does not always yield a limit point of a stable attractive type, it can also yield a repulsive or saddle point. Dynamic modeling automatically finds stationary points of the attracting type, and can also have richer asymptotics, including runaways, limit cycles and random behavior. All this means that the dynamic solver is advantageous, also for solving stationary problems. The key point of our research is to understand how to use the dynamic solver most optimally, in a stable operation mode. Previously [3] we studied only the pressure-massflow subsystem, with constant chemical composition and temperature. Now we study the stability of the dynamic solver for the full system, including mixing flows and temperature modeling.

In this work, Section II presents the modeling of mixing flows incorporating molar and temperature relationships. Section III describes the numerical experiments performed using the developed methods. Section IV considers extended modeling of heaters and coolers. In Section V, extended stability analysis of the full dynamical solver is performed. Section VI summarizes the main results and conclusions of the work.

II. Modeling of mixing flows

This section describes the details of modeling of mixing flows, consisting of modeling fluid molar composition and temperature distribution.

A. Molar fluid composition

A fluid transport network is described by a directed graph consisting of nodes and edges connecting them. The graph is described by an incidence matrix I_{ne}, in which each edge e has nonzero entries for the nodes n that this edge connects; -1 for the node that edge comes from, $+1$ for the node that edge enters. Mixing fluid flows are described by following equations

$$V_n \partial \rho_n / \partial t = \sum_e I_{ne} m_e, \quad (1)$$

$$V_n \partial (\rho_n \mu_n^{-1}) / \partial t = \sum_e I_{ne} m_e \mu_e^{-1}, \quad (2)$$

$$V_n \partial (\rho_n \mu_n^{-1} x_n) / \partial t = \sum_e I_{ne} m_e \mu_e^{-1} x_e, \quad (3)$$

where V_n is the volume assigned to the node; ρ_n represents the mass density at the node; t denotes time; the sum applies to all edges adjacent to the node; m_e is the mass flow in an edge, considered positive if the direction of flow coincides with the direction of the edge, and negative otherwise; $\mu_{n/e}$ is the molar mass assigned to both the node and the edge; $x_{n/e}$ are the mole fractions of the components that make up the fluid.

Physically, the above equations describe various conservation laws. In particular, (1) is the dynamic Kirchhoff equation and describes the conservation of mass. Here, $V_n \rho_n$ on the left side, with V_n representing a time-independent volume, describes the mass of fluid in the node. The sum on the right side accounts for the mass flow into the node, minus the flow out. Equation (2) describes the conservation of the total molar amount of a fluid, where $V_n \rho_n \mu_n^{-1}$ represents the number of moles in a node, and the sum on the right side is the total molar flow in the node. Finally, (3) describes the molar conservation for each component, $V_n \rho_n \mu_n^{-1} x_n$ represents the number of moles of a given component in a node, and the sum is the molar flow of that component. Equations (1) and (2) are valid in the absence of chemical reactions between the components of the fluid.

The x-vector may also include other quantities to which linear molar mixing applies, such as the molar heat value H_m, and linear approximations (T_c, P_c) used in certain equations of state for critical temperature and critical pressure, among others. Alternatively, such quantities can be calculated in post-processing as a linear combination over the molar composition. Explicit inclusion in the mixing equation allows these quantities to be calculated even when the determination of molar composition is disabled.

The conservation equations of type (1)–(3) are standard, can be found in a textbook, e.g., eq. (4.1) in [27]. Now we will rewrite them in a more convenient form, resolved with respect to derivatives:

$$V_n \rho_n \partial \mu_n^{-1} / \partial t = \sum\nolimits'_e I_{ne} m_e (\mu_e^{-1} - \mu_n^{-1}), \quad (4)$$

$$V_n \rho_n \mu_n^{-1} \partial x_n / \partial t = \sum\nolimits'_e I_{ne} m_e \mu_e^{-1} (x_e - x_n), \quad (5)$$

$$\sum\nolimits'_e = \sum\nolimits_{e, I_{ne} m_e > 0}, \quad (6)$$

where the sum is taken over the flows incoming to the node. To prove it, it is necessary to perform the differentiation in (2) and take into account (1), which will result in (4), in which the sums are taken over all flows, incoming and outgoing. Further, if one takes into account that μ_e^{-1} for an outgoing flow is equal to μ_n^{-1} at a node, the sum can be reduced to the incoming flows. The proof for (5) is similar. The condition of equality of mixed quantities in the node and in the outgoing flow can also be used to reduce the total number of variables. Namely, one can completely eliminate the variables in the edge e, replacing them with the values in the upstream node n', $\mu_e^{-1} \to \mu_{n'}^{-1}$, $x_e \to x_{n'}$. When time derivatives are set to zero, these equations are reduced to stationary formula (see eq. (13) in [4]).

Boundary conditions: $\mu = \mu_{set}$, $x = x_{set}$ are fixed to the specified values in the network entry nodes. The system of (4)–(6) and boundary conditions is closed. Its stationary part on the right side of the equations is non-degenerate if all nodes are connected to at least one entry node in the upstream direction. A complete dynamical system can be non-degenerate even if

this rule is violated, for example, if all flows are zero. In this case, the dynamic term ensures the preservation of the transported quantities, keeping them at the starting values.

Startup algorithm: at entry nodes, the transported values are initialized to set values to satisfy the boundary conditions. In all other nodes, values are initialized to default values, which are either specified by the user or averaged over all set values. As a part of the general procedure [3], the initial pressures are set to a constant, the initial flows are set to zero and all fluid composition-dependent quantities, such as density ρ, are calculated from the appropriate equations of state. This procedure provides a smooth startup, with all equations initially satisfied. Then, fluid starts to propagate from entries to the neighbor nodes with growing massflow, replacing default values with current ones.

V_n-definition: in accordance with the discretization scheme formulated in [3], each pipe contributes half of its volume to the end nodes, and all other elements contribute a nominally specified volume V_0.

Linearity of the system: with known m-flows, the μ^{-1}-subsystem (4) is linear; also, for known m and μ^{-1}, the x-subsystem (5) is linear. This property is convenient for controlling convergence, since each linear subsystem in the non-degenerate case is solvable in one iteration. The following algorithm is used to integrate the equations.

Algorithm (simulation workflow):

```
init;
repeat{ mumix; xmix; Tmix; PM; t+=dt; }
```

Here, `init` represents the initialization of all variables according to the startup algorithm described above. `mumix` is the solution of the μ^{-1}-subsystem, `xmix` is the solution of the x-subsystem, `Tmix` is the solution of the temperature subsystem formulated below, and `PM` is the solution of the pressure-massflow subsystem as formulated in [3]. In this way, it is possible not only to find the dynamic evolution of the system, but also to determine the stationary solution. For the last goal, it is necessary to integrate the system with as large steps as possible until stationarity is achieved. The most stable method suitable for this purpose is time discretization of the implicit Euler type: $\partial v/\partial t \to (v - v_{prev})/dt$, for all dynamic variables v, where v_{prev} is the value from the previous step, dt is the integration step. For a detailed study of dynamic processes, more sophisticated finite-difference schemes [28] [29] can be used.

B. Temperature modeling

The starting point is the law of conservation of energy for open systems (see, for example, eq. (4.14) in [27]):

$$V_n \partial(\rho_n \mu_n^{-1} U_n)/\partial t = \textstyle\sum_e I_{ne} m_e \mu_e^{-1} H_e, \qquad (7)$$

where U is the molar internal energy, $H = U + P\mu/\rho$ is the molar enthalpy, and P is the pressure. The equation is similar to the conditions of molar mixing in (3). The difference is that the derivative of the nodal internal energy is on the left side, and the total enthalpy flow in the node is on the right side. Physically, with each flow, internal energy is introduced into the node, as well as the work of the fluid against the pressure in the node. This work can be combined with internal energy, giving enthalpy on the right side of the equation. On the left side, under the derivative, there is still nodal internal energy. In general case, other terms can be present in the conservation law, vanishing for simple mixing in the node. In particular, no additional work is performed in the node, and due to the assumed absolute thermal insulation of the node, heat transfer becomes zero. Possible processes with additional work and heat transfer are assigned to special edge elements and are described below.

We rewrite equation (7) as follows:

$$V_n \rho_n \mu_n^{-1} \partial H_n/\partial t - V_n \partial P_n/\partial t = \\ = \textstyle\sum_e' I_{ne} m_e \mu_e^{-1} (H_e - H_n), \qquad (8)$$

the derivation is similar to (5), also here the nodal internal energy is re-expressed in terms of enthalpy and pressure in the node.

Boundary conditions: $H = H_{set}$, enthalpy is fixed to the specified value in entry nodes. Alternatively, one can use the condition $T = T_{set}$, which fixes the temperature at the entry nodes.

In addition, according to eq. (4.14) in [27], gravitational and kinetic terms can be added to the internal energy and enthalpy: $H \to H + \mu g h + \mu v^2/2$, where g is the acceleration of free fall, h is the height, and v is the speed of translational motion of the fluid. To calculate the kinetic term, one needs to know the diameter, which is not available for all types of elements. For example, a compressor is a very complex structure to be described by a single diameter. Also, at nodes where many edges join, complex internal motion occurs, which does not coincide with the simple translational motion described by a kinetic term with a single diameter. On the other hand, for the transport of gases, the kinetic term is usually significantly less than the internal energy, for translational velocities significantly lower than the speed of sound. In our simulation, we made it possible to optionally turn off the kinetic term in the temperature equations.

In (8), H_n represents the nodal value, and H_e represents the edge downstream value. The difference from x-mixing is that here the edge downstream value in the general case cannot be replaced by the upstream nodal value, since there are elements that change the enthalpy value. The system cannot be reduced to a purely nodal one; in addition, the system also includes the temperature T of the fluid.

HT-constraint:

$$H = H_{mod}(P, T, x), \qquad (9)$$

$$H = H_{mod}(P, T_{prev}, x) + c_p(T - T_{prev}), \qquad (10)$$

where H_{mod} is the thermodynamic model for enthalpy, $c_p = \partial H_{mod}/\partial T$ is the molar heat capacity calculated at point (P, T_{prev}, x). Equations (9)–(10) and (H, T) variables are introduced per node and edge.

The first equation relates enthalpy and temperature according to the thermodynamic model used. We use GERG2008 [17]–[19] as a concrete implementation of such relation. For software-technical reasons, it cannot be used directly; its call once per internal iteration produces too many total calls of GERG2008 module, resulting in significant slowdown. In addition, the equation is nonlinear, violating the desired linearity property of the Tmix subsystem. The second equation is a linearization of the first, it can be used in internal iterations, with a less frequent update of the coefficients. When using the workflow formulated above, (m, P, ρ) in all mix phases are considered as fixed parameters, updated in PM-phase. For H_{mod} and c_p, updates occur immediately before the start of the Tmix phase.

Default element equation:

$$H_e = m_e > 0?H_{n1} : H_{n2} \tag{11}$$

formulates isenthalpic process [27], where the edge enthalpy is taken from the upstream node, similar to x-mixing. In this and further equations, the edge e goes from node n_1 to node n_2, conditions are written in C-notation: *x?y:z = if(x) then y; else z*. This model is applied to the most of element types, in particular, to valves, regulators, resistors and shortcuts; while the exceptional types are listed below.

Pipe equation:

$$(m_e > 0?(H_{n1} - H_e)\mu_{n1}^{-1} : (H_{n2} - H_e)\mu_{n2}^{-1})|m_e| = \\ = \pi D L c_{ht}(T_e - T_{soil}), \tag{12}$$

the change of enthalpy over the pipe is equal to a heat exchange with the soil, eq. (33.3) in [30]. Here T_{soil} is soil temperature, D is pipe diameter, L is pipe length, and c_{ht} is heat transfer coefficient. The pipe should have sufficiently fine subdivision to model the heat exchange appropriately.

Compressor equation:

$$m_e > 0?(T_e - T_{n1}((|P_{n2}/P_{n1}|^{(\kappa-1)/\kappa} - 1)/\eta + \\ +1)z_{n1}/z_{n2}) : (H_e - H_{n2}) = 0, \tag{13}$$

for positive flow, the change of temperature is described by eq. (38.51) in [30], or a similar formula (eq. (13-31)) without z-correction from [31]; otherwise, isenthalpic process is used. Here κ is isentropic exponent, η is efficiency, z is compressibility factor. This basic model is designed for gas transport, while for liquids, e.g., CO_2 pumps, customer-specific models can be used.

Coolers and heaters:

$$m_e > 0?(A_{set} > 0?(T_e - T_{set}) : (H_e - H_{n1})) \\ : (H_e - H_{n2}) = 0, \tag{14}$$

at the simplest modeling level, we implement these elements by clamp formulas: $T_e = \min(T_{n1}, T_{set})$ for coolers and $T_e = \max(T_{n1}, T_{set})$ for heaters. These formulas are piecewise-linear. Their linearization leads to the common formula above and the active set flag described by the following algorithm.

Algorithm (active set):

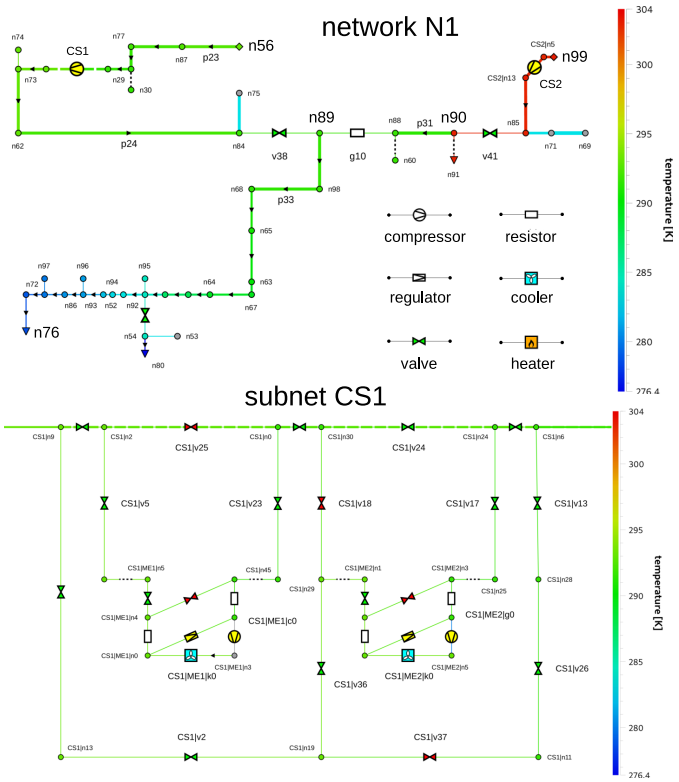


Figure 1. Test network N1.

```
cooler:
  if(Aset==1&&He>Hn1) then Aset=0
  if(Aset==0&&Te>Tset) then Aset=1
heater:
  if(Aset==1&&He<Hn1) then Aset=0
  if(Aset==0&&Te<Tset) then Aset=1
```

Here $A_{set} = 1$ corresponds to an active mode, $A_{set} = 0$ to a standby mode. The algorithm is applied after Tmix-phase, its convergence is tracked.

III. Numerical Experiments

We performed a series of simulations on networks of different complexity levels to study in detail the effects of flow mixing, integration stability, and iteration convergence.

N1 network: the network shown in Figure 1 contains 100 nodes, 111 edges and is used for numerical experiments with the transport of natural gas and hydrogen. Detailed settings of supplies in the considered scenario are presented in Table I. Selected time discretization is $dt = 3 \cdot 10^4 s$, $nsteps = 100$. The network has a simple Y-shaped topology, with two supply nodes n99_gm and n56_gm, as well as a mixing node n89, where the flows from the supplies come together, and the rest of the network, ending with the most distant exit node n76.

Figure 2a shows the evolution of inverse molar mass. Figure 2b presents molar heat value, and Figure 2c demonstrates molar fraction of CH_4, representative for chemical composition in the considered test scenario. In all these plots, the

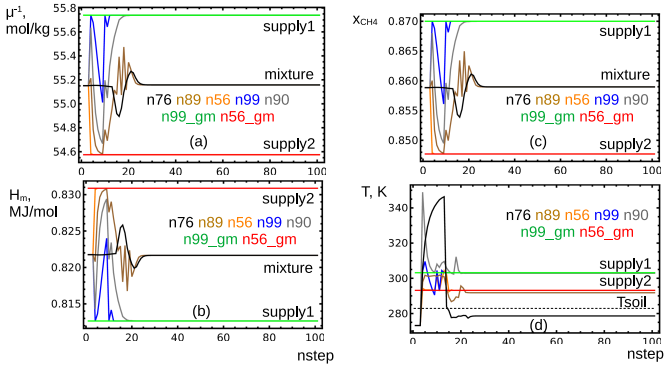


Figure 2. Simulation results (see text for details).

TABLE I
SUPPLY SETTINGS IN VARIOUS SCENARIOS

scenario	entry	composition	temperature
N1 nat.gas	n99_gm	87% CH_4, 1% C_2H_6, 1% C_3H_8, 1% CO_2, 10% N_2	$303.15K$
N1 nat.gas	n56_gm	85% CH_4, 3% C_2H_6, 1% C_3H_8, 1% CO_2, 10% N_2	$293.15K$
dyn-pipe H_2	n0000	95% H_2, 5% N_2	$313.15K$
dyn-pipe CO_2	n0000	95% CO_2, 3% N_2, 2% O_2	$313.15K$

TABLE II
TESTING VARIOUS IMPLEMENTATIONS OF HEATERS ON N85 NETWORKS SET

implementation of heaters	num. of divergent cases
disabled	3
local	0
nonlocal	85
joined	2

values in supply nodes n99_gm and n56_gm are kept constant at set values. In stationary solution, the simple topology of the network leads to a single mixed state, formed in node n89 and propagated downstream to the rest of the network. In the evolution, the values in all nodes tend either to supply values or to this mixed state. Interestingly, in the startup of the evolution, the curves perform several large oscillations between the boundary states, before they relax at the stationary state. This happens due to a complex distribution of flows at the startup phase.

Note that the graphs Figure 2a and Figure 2c have an identical shape, and Figure 2b has the same shape vertically reflected. This happens because there are only two supplies in the network, and the default composition is a linear combination of them. As a result, the trajectory of the system in x-space is limited to a 1-dimensional subspace. Graphs Figure 2a-c are projections of this trajectory to different directions and therefore have the same shape.

Figure 2d shows temperature dependence in selected nodes. During startup evolution, strong heating occurs due to the inverse Joule-Thomson (JT) effect and the influence of the $\partial P/\partial t$-term in (8). With further evolution, the temperature in nodes close to supplies tends to the corresponding constant temperature values of the incoming fluid. In more detail, in the considered scenario, after each supply there is a compressor station, the outlet temperature of which is regulated by a cooler. The outlet temperature of the cooler is set to the same value as that of the corresponding supply. The temperature in network nodes remote from the supply tends to a constant value, slightly below $T_{soil} = 283.15K$, due to the influence of the JT-effect.

N85 networks set: contains 85 realistic natural gas networks, obtained for benchmarking from our industrial partner. The networks are highly resolved, containing up to 4 thousands of nodes each. We used these networks for numerical experiments testing the stability of simulation with a different implementation of heaters. Unlike coolers, which usually control their own output temperature, heaters must control the temperature in an adjacent element, the regulator. In dynamic formulation of the problem, especially at low flows, heaters do not have time to regulate their temperature in order to constantly ensure the set temperature values in the regulator. This leads to divergences. We have tested several options for implementation of heaters, shown in Table II. For disabled heaters, 3 scenarios out of 85 are divergent. For the most stable implementation option, when heaters control their own local temperature, all scenarios are convergent. If the heaters try to control the temperature nonlocally, in the attached regulators, all scenarios diverge, making such implementation impossible. For our selected option, the heaters are joined with regulators, the unified element controls its own output temperature, 2 scenarios out of 85 are divergent, slightly better than the complete disabling of the heaters.

Hydrogen and carbon dioxide pipelines: this is one of our standard test cases, $L = 150km$, $D = 0.5m$ horizontally laid pipeline, transporting gaseous H_2 or CO_2 in liquid or supercritical phase. The case supports variable spatial discretization, for the considered scenario selected to $nsubdiv = 50$. Time discretization is the same as for N1 network. Supply setting is presented in Table I. The considered scenario has a single fluid composition and is used mainly for testing of the temperature modeling. The dynamic simulation starts from $T_{soil} = 283.15K$ and a different $T_{set} = 313.15K$ at the pipeline entry. The simulation converges to stationary solution with nearly exponential fall of temperature from T_{set} to T_{soil}. For CO_2, an observed stronger deviation from the exponent is due to JT-effect and the nonlinear enthalpy model.

Convergence of iterations: in our implementation, we use the globally convergent Newton's solver with Armijo line search rule [32], applied at every time step. For linear problems, it just forwards the solution to the underlying sparse linear solver, that for non-degenerate problems converges in 1 iteration. Due to proper initialization, at the first time step all phases converge in 0 iteration, just keeping the starting values. This provides a good method to test that all variables

are correctly initialized. At the second time step, all mix phases also converge in 0 iteration, while in the last PM phase the network filling begins, and PM phase starts to increase its iteration number. For N1 network and H_2/CO_2 pipe scenarios, all mix phases are solved in 1 iteration on intermediate timesteps, as it should be for non-degenerate linear systems; and in 0 iteration at the last timesteps, due to convergence to stationary solution. For large N85 networks, Tmix phase can have intermediately 2-3 iterations, indicating the remaining degeneracy or the disbalance of scaling factors in Tmix system. This effect will be studied in more details in Section V.

The numerical experiments performed show that the primary purpose of this work has been fully achieved, the modeling has been extended to include mixing flows and is working for scenarios of varying complexity. The modeling in our system is presented in open text form, as a list of variables and equations, which both we and the users can freely modify. This distinguishes us from the existing solutions, in which the modeling is usually hardcoded within the system. We also provide numerical stability of the modeling and the solution algorithms, which allows us to solve large realistic scenarios in fluid transport simulation.

IV. Extended modeling of heaters and coolers

The nonlocal control case is especially difficult to model, when the point with the controlled temperature is not at the heater or cooler output, but in another area of the network, for example, when a temperature sensor is placed there. As shown by the numerical experiments, a direct generalization of the control equations to this case is unstable. At the same time, there is a workaround with the transfer of the temperature control function to the element for which the controlled temperature is local. Although this approach works, it would be desirable to obtain a more realistic modeling, in particular, one that reproduces the correct intermediate temperatures between the heater/cooler and the sensor position. In this section, we consider the extension of modeling necessary for this.

The required diagram for the heater is shown in Figure 3a. It consists of three branches: on – the temperature at the controlled point is maintained at the required value: $T_c = T_{set}$, the heater is on: $T_e > T_{n1}$; standby – the temperature at the controlled point exceeds the required value: $T_c > T_{set}$, the heater is off: $T_e = T_{n1}$; an additional max branch is introduced – the temperature at the controlled point is less than the required value: $T_c < T_{set}$, the heater operates at maximum: $T_e = T_{max}$. The reason for introducing an additional branch is that in some cases the set control goal is unachievable. An example is a vanishingly small flow, when the contribution of the heated fluid from the heater has virtually no effect on the temperature at the controlled point. Also, due to a network configuration error, the controlled point may be outside the influence zone of the heater, for example, behind a closed valve. If there is no max branch in the control equation, then, in the case of a decrease in the controlled temperature below

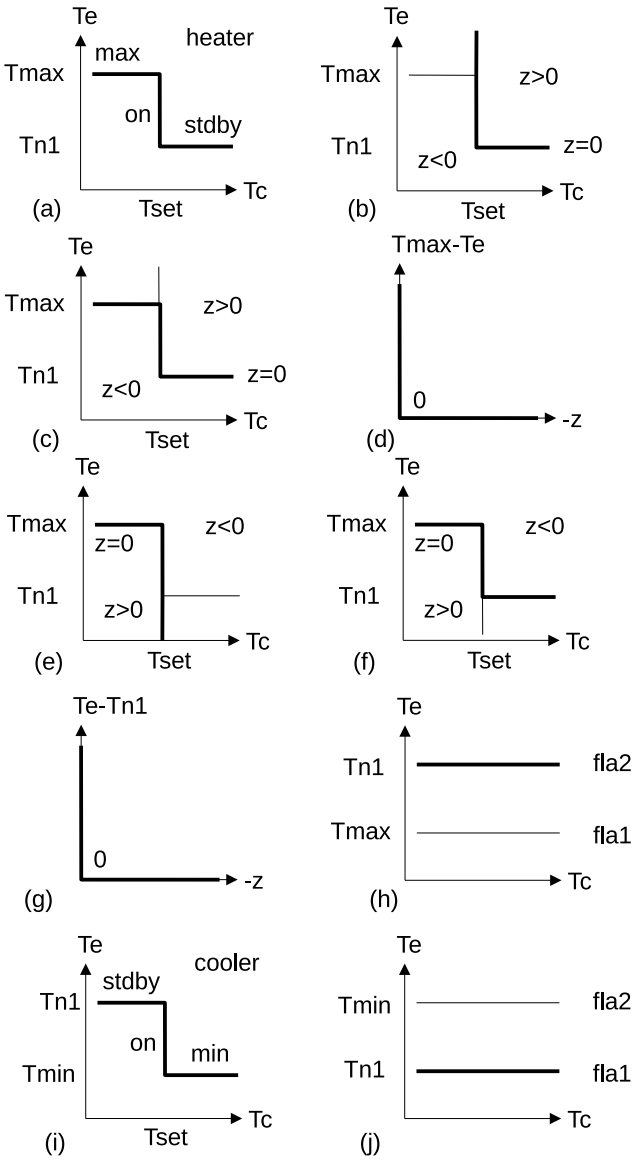


Figure 3. Construction of minmax formulas for heaters and coolers (see text for details).

the required value, the heater will try to heat the fluid more and more, eventually leading to simulation divergence. Introducing the max branch in this case gives a physically reasonable alternative scenario with a limited temperature. For cooler the modification is similar, here the min branch is introduced, as shown in Figure 3i. Extended control equations are as follows:

heater:

$$\max(\min(T_{set} - T_c, T_{max} - T_e), T_{n1} - T_e) = 0, \quad (15)$$

cooler:

$$\max(\min(T_c - T_{set}, T_e - T_{min}), T_e - T_{n1}) = 0, \quad (16)$$

where T_e is the local temperature in the heater/cooler; T_{n1} is the temperature at the heater/cooler inlet; T_c is the temperature at the controlled node/edge, at the sensor location; T_{set} is the

set temperature at that location; $T_{max/min}$ are the temperature limits at the heater/cooler, by default set to $T_{min} = 223.15K$, $T_{max} = 423.15K$. Note that the formulas are now piecewise linear rather than linear. It is not possible to preserve the overall linearity of the simulation, but the new simulation is more stable and does not require convergence of the active set iterations.

We will now provide a detailed derivation of the minmax formulas. Similar formulas are used in other parts of our simulation, and their derivation uses a similar procedure. First, let us consider the heater simulation, represented by the diagram in Figure 3a. Next, in Figure 3b the zero level of the function $z = \min(T_c - T_{set}, T_e - T_{n1})$ is marked with a bold line, dividing the plane into regions of positive and negative values of this function. In Figure 3c we break this line to the desired shape of the diagram, consisting of two pieces:

$$(z \le 0 \& T_e = T_{max}) | (z = 0 \& T_e \le T_{max}). \quad (17)$$

In Figure 3d we use the coordinates $(-z, T_{max} - T_e)$, transform it to the standard representation

$$(-z \ge 0 \& T_{max} - T_e = 0) | (-z = 0 \& T_{max} - T_e \ge 0), \quad (18)$$

equivalent to the equation $\min(-z, T_{max} - T_e) = 0$. After substitutions and algebraic transformations we obtain

$$\min(-\min(T_c - T_{set}, T_e - T_{n1}), T_{max} - T_e) = 0, \quad (19)$$
$$\max(\min(T_c - T_{set}, T_e - T_{n1}), T_e - T_{max}) = 0, \quad (20)$$

hereinafter denoted as formula1.

Alternatively, in Figure 3e, we start constructing the diagram from the other end, considering the zero level of the function $z = \min(T_{set} - T_c, T_{max} - T_e)$; in Figure 3f we obtain the form

$$(z \le 0 \& T_e = T_{n1}) | (z = 0 \& T_e \ge T_{n1}); \quad (21)$$

in Figure 3g in coordinates $(-z, T_e - T_{n1})$ reduced to the standard form

$$(-z \ge 0 \& T_e - T_{n1} = 0) | (-z = 0 \& T_e - T_{n1} \ge 0); \quad (22)$$

or $\min(-z, T_e - T_{n1}) = 0$. Now, after trivial algebra we obtain another formula:

$$\min(-\min(T_{set} - T_c, T_{max} - T_e), T_e - T_{n1}) = 0, \quad (23)$$
$$\max(\min(T_{set} - T_c, T_{max} - T_e), T_{n1} - T_e) = 0, \quad (24)$$

hereinafter denoted as formula2.

It is interesting that these two formulas give an equivalent representation of the diagram shape in Figure 3a, but are not absolutely identical. In the special, physically important case $T_{n1} > T_{max}$, when the input temperature exceeds the maximum limit, these formulas give different results, shown in Figure 3h. Indeed, in formula1:

$$\max(\min(T_c - T_{set}, T_e - T_{n1}), T_e - T_{max}) = 0, \quad (25)$$
$$T_{n1} > T_{max} \Rightarrow T_e - T_{n1} < T_e - T_{max}, \quad (26)$$
$$\text{case1: } T_c - T_{set} \ge T_e - T_{n1}, \quad (27)$$
$$\max(T_e - T_{n1}, T_e - T_{max}) = T_e - T_{max} = 0; \quad (28)$$
$$\text{case2: } T_c - T_{set} < T_e - T_{n1}, \quad (29)$$
$$\max(T_c - T_{set}, T_e - T_{max}) = T_e - T_{max} = 0, \quad (30)$$

case1 and case2 produce the same answer. In formula2:

$$\max(\min(T_{set} - T_c, T_{max} - T_e), T_{n1} - T_e) = 0, \quad (31)$$
$$T_{n1} > T_{max} \Rightarrow T_{n1} - T_e > T_{max} - T_e; \quad (32)$$
$$\text{case1: } T_{set} - T_c \ge T_{max} - T_e, \quad (33)$$
$$\max(T_{max} - T_e, T_{n1} - T_e) = T_{n1} - T_e = 0; \quad (34)$$
$$\text{case2: } T_{set} - T_c < T_{max} - T_e, \quad (35)$$
$$\max(T_{set} - T_c, T_{n1} - T_e) = T_{n1} - T_e = 0, \quad (36)$$

here we also get a horizontal line on Figure 3h, but a different one. Physically, in the special case under consideration, formula1: $T_e = T_{max} < T_{n1}$ leads to the fact that the heater cools the fluid, so here we should choose the answer $T_e = T_{n1}$, described by formula2.

Let's move on to considering cooler, with the diagram shape shown in Figure 3i. Interestingly, it coincides with the diagram for heater, up to the redesignations $T_{n1} \to T_{min}$, $T_{max} \to T_{n1}$. Thus, instead of repeating the derivation, we can make such a redesignation in the answer for heater and obtain two formulas:

formula1:
$$\max(\min(T_c - T_{set}, T_e - T_{min}), T_e - T_{n1}) = 0; \quad (37)$$
formula2:
$$\max(\min(T_{set} - T_c, T_{n1} - T_e), T_{min} - T_e) = 0. \quad (38)$$

For the special case $T_{n1} < T_{min}$, formula2: $T_e = T_{min} > T_{n1}$ would mean that the cooler heats the fluid, so for physical reasons the answer $T_e = T_{n1}$ described by formula1 should be chosen here.

V. Extended stability analysis

Stability analysis of fluid transport simulations was performed in our previous works, for the stationary case in [2], [10], for the dynamic case in [3]. In these works only the PM phase of the simulation was analyzed. Stability analysis for mixing flows modeling will be performed in this section. The main challenge is the configuration of the dynamic solver for solving stationary problems by integrating to a stationary state, while ensuring the stability of the simulation for realistic large-size networks. First, we present the main results for the PM phase, then we move on to the analysis of mixing phases.

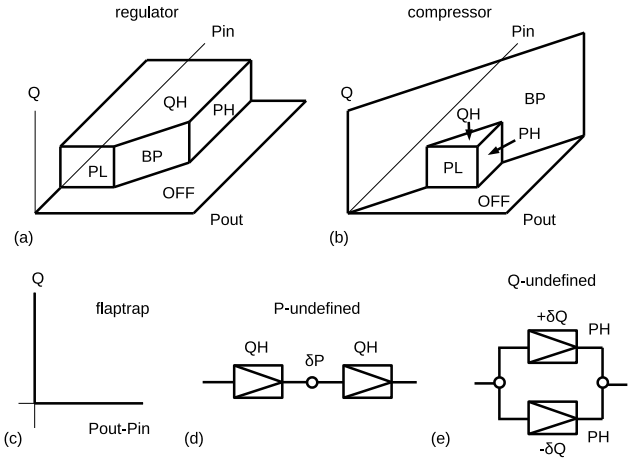


Figure 4. (a)-(c): working diagrams for control elements; (d),(e): possible degenerations of the system; reprinted from [3] by permission (copyright IOP).

PM phase: the main problem for the stability of the simulations is represented by regulators, compressors and flaptraps. The behavior of these elements is given by the diagrams shown in Figure 4a-c. The polyhedral surfaces for regulators and compressors are represented by complex minmax formulas given in [2], the specific form of which is not important for us now. What is important is that combinations of several such elements can be located on certain faces of the surfaces that conflict with each other. For example, in a stationary simulation, two regulators in series on QH-face actually impose the equation $Q = QH$ twice, with one equation wasted, and one unconstrained degree of freedom appears in the system. This degree of freedom corresponds to the undefined pressure at the intermediate point, P-undefined conflict, see Figure 4d. Similarly, two regulators in parallel on PH-face impose the equation $P_2 = PH$ twice, with one equation wasted, and the balance of flows through the regulators is undefined, Q-undefined conflict, see Figure 4e. The described conflicts are not limited to series and parallel connections. The problem is also represented by a long pipe, at the beginning and end of which there are QH-regulators; Y-connection of 3 PH-regulators; conflict between the regulator and Pset/Qset boundary condition at entry or exit; etc. In addition, during the solution process, the working point can change the face on the control diagram, so during the simulation, the described conflicts can spontaneously arise in any part of the network.

For numerical simulations, these conflicts lead to degeneration of the system, the appearance of zero eigenvalues in the Jacobian matrix [2], [10], which leads to divergence of the solver. The general approach to solving this problem is to regularize the equations, to reformulate them as follows:

$$f_{reg} = (1 - \epsilon_s) f + \epsilon_s (P_1 - P_2 - R_s Q) - \epsilon_d \partial m / \partial t, \quad (39)$$

where in the first term f are the original control equations. The second term represents the linear resistor equation, the coefficient $0 \leq \epsilon_s \leq 1$ is chosen so that the regularization can be completely removed at $\epsilon_s = 0$ or, conversely, the control equation can be deformed to a linear resistor at $\epsilon_s = 1$. This type of regularization is static, independent of time derivatives. The third term contains the time derivative and represents dynamic regularization. When choosing the implicit Euler finite difference scheme, this term takes the form $-\epsilon_d (m - m_{prev})/dt$, with $\epsilon_d > 0$. Here Q and m represent the flow in different normalizations and are proportional to each other with a positive coefficient. The common signs in this formula are chosen so that the derivatives of the result with respect to the variables (P_1, P_2, m) have the signature $(+, -, -)$, which, according to [2], is necessary for the convergence of the PM phase of the simulation.

The dynamic term in (39) contains only the m-variable and effectively regularizes only the Q-undefined conflict. The regularization of the P-undefined conflict is performed by the dynamic term $V_n \partial \rho_n / \partial t$ in the Kirchhoff equation (1). This term is able to describe the evolution of the density and the associated pressure even in situations where the control equations do not capture them. The regularizing parameter here is the nodal volume $V_n > 0$, which can also be replaced by one freely adjustable value $V_1 > 0$, without changing the stationary result.

In practice, the use of static regularization leads to the undesirable effect of shifting the solution from the faces of the control equation, violating the control conditions $Q = QH$, $P = PH$. These violations are controlled by the regularizing parameter ϵ_s; for small values, the equation is too singular to solve, and for large values, the physically desirable conditions will be violated. As a tradeoff value, we chose $\epsilon_s = 10^{-3}$, corresponding to 0.1% violation of the control equations and an acceptable level of convergence of the simulations. In the case of divergences, if the cause can be traced back to the control equations via residuals, the user is advised to increase the parameter to $\epsilon_s = 10^{-2}$.

For dynamic regularization, the time derivatives vanish as the stationary solution is reached. Therefore, the dynamic regularizers are switched off in the stationary limit, and no violations of the control equations occur. The limiting factor here is the too slow convergence of the solution for large values of the regularizer. Also, the equations include combinations of ϵ_d/dt, V_1/dt, so for integration with a large step, it is also necessary to artificially increase the regularizing parameters. In our numerical experiments, we varied the described parameters in wide limits and investigated their influence on the simulation stability.

The choice of regularizing parameters was carried out on large natural gas simulations of the N85 type described above and is illustrated by the graphs in Figure 5. At first, we included only the PM phase and investigated its stability separately. Figure 5a shows the idealized case of $\epsilon_s = 1$, when all control equations are replaced by linear resistors. In this experiment, all nodal volumes were also replaced by a single value of V_1. As a result, very fast collective convergence of all simulations below the nominal value $res = 1\%$ is obtained. This numerical experiment shows that in the PM phase we

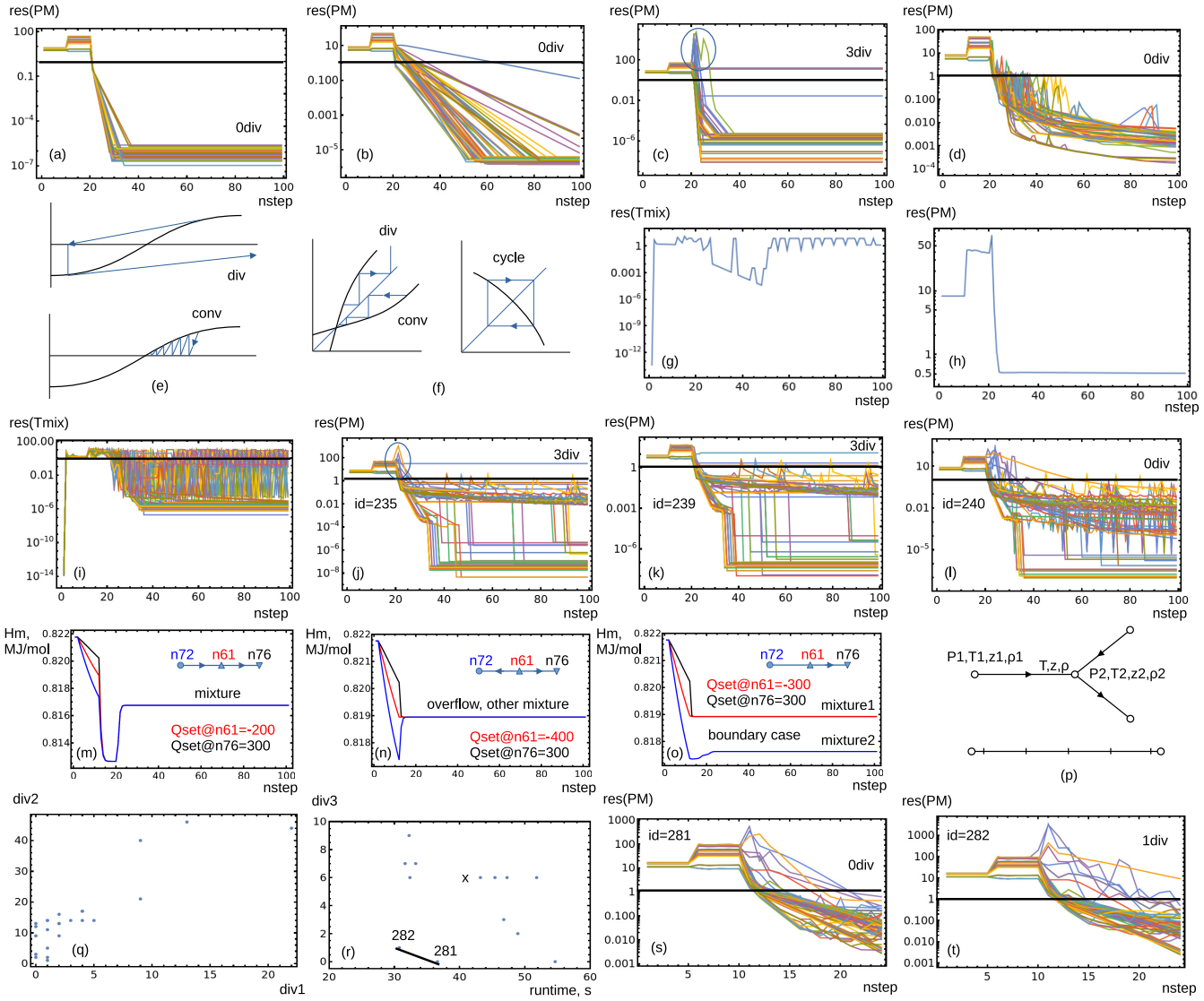


Figure 5. Extended stability analysis (see text for details).

have taken all causes of divergence under control. In Figure 5b we set the nodal volumes to their actual values. The result is still acceptable, but the convergence rate varies from one test case to another. This indicates the advantage of choosing one value for all nodal volumes. In Figure 5c we chose the values $\epsilon_s = 10^{-3}$, $\epsilon_d = 30bar/(kg/s^2)$, $V_1 = 300m^3$, $dt = 3 \cdot 10^5 s$. The configuration is still acceptable, with only 3 out of 85 cases diverged. In this figure, the two initial plateaus correspond to the starting procedure [3] of changing the boundary conditions, first raising all Psets from the starting one to the desired values, then all Qsets. In the second part, we also made a continuous deformation of the regularizing parameter from $\epsilon_s = 1$ to $\epsilon_s = 10^{-3}$. At the end of this interval, the system approaches a singularity, so the characteristic residual peaks are visible in the figure. In Figure 5c, we changed $\epsilon_d = 3 \cdot 10^3 bar/(kg/s^2)$, $V_1 = 30m^3$, and as a result, all simulations went below the nominal threshold. This time, the convergence is slower, but the peak after the starting procedure that generated divergences has disappeared.

Mixing phases: instabilities are present only in the Tmix phase, the others work without problems. Stabilization can be done using dynamic regularization

$$f_{reg} = f + \epsilon_H \partial H/\partial t \tag{40}$$

with the coefficient $\epsilon_H > 0$, when choosing the sign for the original equation $\partial f/\partial H > 0$. Due to the identity $\partial H/\partial T = c_p > 0$, which relates this derivative to the heat capacity, the regularizing term can be reexpressed in via temperature: $\epsilon_H \partial H/\partial t \rightarrow \epsilon_T \partial T/\partial t$, with a new regularizing parameter $\epsilon_T > 0$. A static regularizing term can also be added to this expression, for example, $\epsilon_s(T - T_{soil})$. This term can lead to physically undesirable effects, for example, a temperature of T_{soil} can be established at the output of a low-flow

regulator, despite the existing thermal insulation. Therefore, if dynamic regularization works, we try to refrain from using static regularization. Note that for moderate-sized systems, the described simulations very rarely lead to divergences and can be used directly. For large systems, such as the N85 set used in our tests, the problems are potentiated, and the simulations require special stabilizing measures. Below, we will analyze in detail the available equations and the instabilities associated with them.

Enthalpy mix equation: it already has time derivatives and does not require additional regularization. When switching off the dynamic terms in (8), the remaining stationary system can be degenerate. The problem occurs for $m = 0$, in particular, at the starting conditions. Also, since in this equation only flows entering a node are included in the sum, the problem occurs for all subgraphs not connected to Tset-nodes in the upstream direction. Physically, this singularity means a T-undefined state in the stationary limit for such subgraphs. The dynamic terms resolve this ambiguity, however, for small V_n/dt the regularization is weak, the system matrix is close to singular. An important tuning factor is the nodal volume. It is also possible to equip both dynamic terms in (8) with free coefficients. This allows one to further strengthen the contribution of the $\partial H_n/\partial t$ term, as well as to weaken or disable the $\partial P_n/\partial t$ term. According to our numerical experiments, removal of $\partial P_n/\partial t$ term improves overall stability.

Temperature equation: when using the linearized version of the simulation, equation (10) has a new type of problem. In fact, it describes a Newton iteration in the temperature variable. Although Newton's method is used in the inner iteration, at each integration step, it has been specially stabilized there [32], while the outer iteration described by (10) is unstabilized. It is widely known that the unstabilized Newton's method produces divergences. As shown in Figure 5e above, the sequence of tangents to the curve may go to infinity if the starting point is chosen poorly. The simplest way to overcome this problem is to increase the slope of the lines above the tangent position, which is equivalent to introducing a coefficient $H = H_{mod} + c_1 c_p (T - T_{prev})$, $c_1 > 1$. As shown in Figure 5e below, this can enforce convergence. Although the convergence rate of such an iteration may be slower than Newton's, it turns out to be more stable. Another way to stabilize is to use the original nonlinear equation (9). In a case when there is a closed analytical formula for this equation, it can be used directly. Of course, this will lead to nonlinearity of the Tmix phase and an increase in the number of internal iterations for its solution, the advantage of this approach is better stability of the simulation.

Compressors: in equation (13) a problem similar to the temperature equation arises. In this equation, there is a strong coupling with the PM-phase, in particular, through the z-coefficients present in it. An increase in T_e at a given iteration leads to an increase in z_{n2} at the next iteration, which through the formula (13) triggers a decrease in T_e at the next iteration. Under strong coupling, this iteration sequence can loop or diverge. Figure 5f illustrates the possible behavior of a one-dimensional iteration, showing prototypical examples of instability. The simple solution proposed in [4] consists in introducing a weighting procedure: $T = T_{eq} w + T_{prev}(1 - w)$, with a constant $0 \leq w \leq 1$. In this case, the new value of the variable is not taken directly from the equation, but is weighted with the previous iteration. In practical applications, this approach allows stabilizing looped or diverging iterations that arise due to strong coupling. After rewriting the weighting procedure as the equation $(T - T_{eq})w + (T - T_{prev})(1 - w) = 0$ and comparing the stabilizing terms $(T - T_{prev})/dt \sim \partial T/\partial t$, it becomes clear that the weighting method is completely equivalent to both dynamic regularization and the stabilization of the Newton iteration presented above, up to a redefinition of the coefficients. Another method for stabilizing the compressor equation is to substitute analytical expressions for z-coefficients, if any, into (13).

Coolers and heaters: equations (15)-(16) have problems similar to compressors. For example, if the heater was in standby at the previous iteration and at the controlled point T_c becomes slightly less than T_{set}, then the heater goes into max mode. If the flow through the heater is small, this may lead to a small increase in T_c over T_{set}, and the heater is forced to return to standby. This may lead to iteration loops. The solution here is also dynamic regularization or the equivalent weighting procedure.

Pipes: equation (12) already contains a regularizing term of the static type $\sim (T - T_{soil})$, so the temperature modeling of pipes is stable. A necessary condition is the presence of a physically reasonable heat exchange coefficient.

Default element equation: in the simple-looking equation (11) the strongest instability is located. When passing through the value $m_e = 0$, the edge enthalpy H_e jumps between the nodal values H_{n1} and H_{n2}. Changes in the sign of the flow can occur both at intermediate steps and at the end of integration. A specific example is small numerical fluctuation of the flow in network sections with zero stationary flow. In this case, a unique situation arises when in the final, physically stationary state there are randomly fluctuating variables of undamped amplitude. The jumps are experienced by both the variables themselves and by the residuals of equations defining them, see Figure 5g, which shows the residual of Tmix phase for one scenario N85.1. The residuals use the maximum norm over the equations, as a result of this definition, the jumps can be separate or merging into a plateau. At the same time, the residual of the PM phase shown in Figure 5h does not have such jumps, and repeats the shape of the residual of the pure PM phase shown in Figure 5c.

A detailed analysis shows that the stochastic edge degrees of freedom (H_e, T_e) formed in the system decouple from the nodal (H_n, T_n). Indeed, coupling is carried out by means of equation (8), in which H_e are multiplied by m_e. Thus, the jumps of H_e at $m_e = 0$ are suppressed. The PM phase includes only the nodal values of T_n, so the stochastic degrees of freedom are decoupled from the PM phase as well. In practice, the Tmix phase residual shown in Figure 5i for the entire N85 set is so noisy that it becomes unusable. The PM

phase residual, Figure 5j, can be used as successfully as for the pure PM phase. Indirectly, the PM residual also controls the nodal values of the Tmix phase, via the strong coupling T_n/P_n in the equations of state. The edge values of the Tmix phase undergo jumps around $m_e = 0$, which arise due to their definition as edge downstream values and are not physically important. Thus, our current recommendation is to ignore the Tmix residual and use only the PM residual to control the convergence of the simulation.

Looking at this issue in even more detail, jumps occur in all edge equations where the separation into $m_e > 0$ and $m_e < 0$ branches is used, and they are also suppressed by the m_e-factor in the nodal coupling. The introduction of branches is necessary, otherwise degeneracies arise in the system. As an example, consider the compressor equation $T_e = T_{n1}a$, $a > 1$, in the stationary limit. For $m_e > 0$, the outlet temperature is further transferred to $T_{n2} = T_e$ via the nodal coupling (8). Negative flow through the compressor is possible due to ϵ_s-regularization for infeasible solutions, both at the intermediate and final integration steps. If the compressor equation remains the same for $m_e < 0$, then nodal coupling will lead to $T_{n1} = T_e$, an overdetermined equation on T_e, and no condition on T_{n2}. As a result, the stationary system will become degenerate, and the stationary solver will diverge. There are additional regularizers for the dynamic solver, but their efficiency will be reduced if they have to suppress a more degenerate stationary system. Introducing the isenthalpic branch into the equations ensures non-degeneracy of the system, and it also generates jumps in the solution. Note that suppressing jumps in the edge equations by introducing dynamic damping or weighting procedures does not work here, it only reduces the amplitude of the jumps by a factor of w. The value $w = 0.5$ is practically acceptable for stabilization in our numerical experiments; for smaller values, the convergence of integration becomes too slow.

Figure 5j shows the PM residual for simulations with values $\epsilon_s = 10^{-3}$, $\epsilon_d = 30bar/(kg/s^2)$, $V_1 = 0.3m^3$. Characteristic is the loss of the collective convergence property, which was present for pure PM simulations. This property is a consequence of single-phase modeling, in which the convergence of the solution at the previous iteration leads to convergence at the next one, with small variations due to small dynamic terms. In the full simulation, mix phases are involved in the iterative process, and the convergence of the outer iterative loop is decisive for the convergence of the simulation. In Figure 5k, we increased $\epsilon_s = 10^{-2}$, which resulted in the absence of the residual peak at the end of the startup procedure, which also led to better stability of the inner iterations and a decrease in runtime. In Figure 5l, with $\epsilon_s = 10^{-3}$, $\epsilon_d = 3 \cdot 10^3 bar/(kg/s^2)$, $V_1 = 30m^3$ were increased. Here, as for the pure PM simulation, convergence became slower, but the stability of the simulation has been improved.

Phase transitions: should be considered, in particular, for CO2 transport [8]. The problem is the presence of a jump in the function $W(T)$ for pure substances or a rapid change in this function in the presence of small impurities. This leads to the failure of the Newtonian method, both in internal and external iterations. Dynamic regularization or weighting do not help here. In this case, jumps also occur in nodal variables, propagate to the PM phase and break the convergence of the simulation altogether. Usually, scenarios without phase transitions are considered in applications, CO2 is transported in a liquid/supercritical dense phase or in a gaseous phase. In the absence of phase transitions, the simulation does not have problems of the described type. In order to prevent phase transitions also for all intermediate states on the integration path, the simulation should be started with (P_{start}, T_{start}) values in the region of the expected solution.

FE-nodes: Figures 5m-o show the behavior of FE-nodes, Qset-supplies without specified mix quantities. For such supplies, the mix quantities are assumed to be taken from the incoming flow. The experiments are done on N1 test network. In Figure 5m, a normal operation is shown, where the added flow is less than for a downstream exit, and the mix quantities are taken from the incoming flow. Figure 5n shows an overflow scenario, where the added flow prevails, and there are no incoming, but only outgoing flows. In this case, in stationary problem, the mix value is undefined. The dynamic modeling has the time-derivative term, making the problem non-degenerate even in this case. The resulting mix values are defined by the history of integration. Typically they remain at the starting default values, different from the mixed state of the normal operation mode. Figure 5o shows a boundary case, when the added flow exactly equals the exit flow. In this case, two different mixed states are formed.

Downstream mismatch: in the PM phase there is a problem of a different type, see Figure 5p. The upper part of the figure shows a pipe, with nodal values of pressure, temperature, compressibility and density (P_1, T_1, z_1, ρ_1) and (P_2, T_2, z_2, ρ_2). Consider the section of the pipe immediately adjacent to the downstream node. By continuity, the pressure at this point coincides with the nodal P_2. Otherwise, the pressure jump would create a non-zero force that would act on a vanishingly small mass of the section and lead to infinite acceleration. The temperature in the section, however, may differ from the nodal one, due to a possible inflow of fluid of a different temperature into the node. Compressibility and density depend on temperature and may also not coincide with the nodal values. As already mentioned, PM modeling uses only nodal values for the mentioned quantities. In particular, the PM equation for pipes includes their nodal average. The described mismatch can lead to a local variation of the result near the downstream node. One possible solution would be to introduce edge quantities (T, z, ρ) and a state equation relating them. In fact, this is not a very good idea, since these quantities have random jumps around $m = 0$, and the stochastic behavior would penetrate into the PM phase. Another, simpler solution is shown in the lower part of the figure. To improve the accuracy of the simulation, long pipes should be split into smaller ones. This procedure can include two short segments, say $\Delta L = 1m$, at the beginning and end of the pipe. As a

result, the downstream mismatch problem will be concentrated in these segments. At the same time, since the pressure drop on short pipe segments is negligible, the influence of the described problem on the result will be excluded. In addition to pipes, the problem can occur in compressors (13), if there is z-correction in their equation. On the other hand, in real scenarios, a cooler is usually installed immediately after the compressor, and the described problem does not arise. When using stand-alone compressors or pumps, it is recommended to insert a short section of pipe immediately after them to avoid possible downstream mismatch.

Scaling: Newton's method, in particular the stabilization algorithms [32] used in it, are sensitive to the scaling of equations. For optimal operation of these algorithms, all our equations were scaled so that their variation in the working region of the variable change was of the same value, nominally chosen as 100 units. As a result of such normalization, the residuals of the equations become dimensionless quantities measuring the current absolute value of the equation as a percentage of its variation in the working region. Further, the residuals are maximized over the equations and characterize the convergence of the solution phases. For a detailed characterization of the convergence, two residuals are introduced, for the inner and outer iterations. The residual at the end of the inner Newton iteration measures the convergence of each integration step. The residual at the beginning of the inner Newton iteration measures the convergence of the integration steps to a stationary solution. When the stationary solution is reached, the variables begin to converge to constant values, and the equations also stop changing. In this case, the residual at the beginning of the inner iteration becomes small, ideally less than the stop-criterion $tol = 10^{-5}\%$, so that the inner iterations should not even start, or less than the acceptable threshold $tol_2 = 1\%$.

Clamping: is another technically necessary procedure. In all equations, such quantities as $v = (P, T, z, \rho)$ must be clamped into the physical domain of change: $v \to \min(\max(v, v_{min}), v_{max})$. In the process of solving, such quantities may go beyond the physical domain, for example, become negative. This may happen for infeasible problems that have no solution in the physical domain, as well as for stationary feasible problems at intermediate iterations. According to the general strategy [2], we maintain convergence of the solver in these domains as well, to ensure stability and localize possible infeasibility. As an example, consider the mixing equation (4) with a dynamic term $\sim \rho_n \partial \mu_n^{-1}/\partial t$. If ρ_n becomes negative during the solution, this term will effectively undergo a time reversal, which will immediately lead to divergence of the integrator. Clamping ρ_n into the positive domain solves the problem. Clamping should be carefully introduced into all equations, however, one should not overdo it. Consider the compressor equation $T_e = T_{n1}a$, $a > 1$. Here one can enter clamping to the T_{n1}, a, or a combined $T_{n1}a$ term. One cannot enter clamping to the T_e term, since this equation is the definition of T_e. In the case of T_e clamping, when it is triggered, the T_e dependence drops out of the equation, which will lead to degeneration. We also experimented with introducing clamping to all T variables not in the equations, but between the integration steps. At first glance, this eliminates the need to introduce T-clamping in numerous equations. However, this leads to a deeper problem. If at the current integration step the solution of the equations is located outside the T-clamps, and a clamp is applied before the next step, then the starting point of the next step will no longer satisfy the equations. This can increase the residual and unnecessarily trigger additional iterations. Therefore, we recommend avoiding the use of clamps and any solution-modifying algorithms between the integration steps.

TABLE III
FINE-TUNING PROCEDURE

id	div1	div2	div3	runtime, s
235	3	14	3	73
239	1	5	3	67
240	0	3	0	78
258	4	14	3	47
259	5	14	7	32
260	1	14	6	43
261	9	21	15	34
262	4	17	2	73
263	2	13	6	72
266	1	14	6	32
267	4	17	6	52
268	4	17	9	32
269	2	13	6	47
270	2	13	7	33
271	2	16	6	45
275	0	9	0	55
276	1	2	2	49
277	13	46	16	109
278	2	9	11	38
279	1	1	11	31
280	22	44	58	44
281	**0**	**13**	**0**	**37**
282	**0**	**12**	**1**	**31**
283	9	40	17	56

TABLE IV
FINE-TUNING RESULTS

par	id=281	id=282
n	25	25
dt, s	$6 \cdot 10^4$	$6 \cdot 10^3$
t_1, s	$3 \cdot 10^5$	$3 \cdot 10^4$
t_2, s	$6 \cdot 10^5$	$6 \cdot 10^4$
t_{end}, s	$1.5 \cdot 10^6$	$1.5 \cdot 10^5$
ϵ_s	10^{-3}	10^{-3}
$\epsilon_d, bar/(kg/s^2)$	30	30
V_1, m^3	0.3	0.3
w	0.5	0.5

Fine-tuning: after we have found parameter values with satisfactory convergence characteristics, see Figure 5j-l, we fine-tune the parameters to achieve optimal runtime while maintaining acceptable stability. To do this, we introduce the following characteristics: div1 – number of cases with divergent Newton iteration at the last integration step; div2 – number of cases with divergent Newton iteration at any integration step; div3 – number of cases that do not reach

stationarity after integration, at nominal level $res = 1\%$. The runtime value is averaged over all cases from the N85 set, convergent or not. Simulations were performed on i7-14700K CPU computer. The values of div1-3 and runtime should be minimized. As the analysis shows, the values div1-2 are correlated with each other, see Figure 5q, they are also weakly correlated with div3. Also, the value div3 is weakly anticorrelated with runtime. For the analysis, one graph Figure 5r is sufficient, representing numerical experiments in coordinates (runtime,div3). The best solution marked with a line in the figure marks the tradeoff boundary, the Pareto front, on which these criteria cannot be simultaneously reduced. The characteristics of the stationary simulator are marked with a cross in the figure. It is evident from the graph that the dynamic solver clearly overperforms the stationary one.

In greater detail, the characteristics of fine-tuning runs are given in Table III. The first three lines (235-240) correspond to the configuration of Figure 5j-l. Next, we chose the point (235) and optimized the dynamic schedule described by three parameters (n, dt, t_1), the number of integration steps, the step size, and the time of the first startup phase. The dependent parameters are $(t_2 = 2t_1, t_{end} = n\,dt)$, the time of the second startup phase, and the total integration time. At the beginning (258-259), we decreased n from the starting value $n = 100$ to $n = 50, 25$, which corresponds to a shortening of the integration interval t_{end} with fixed (dt, t_1). Then (260-261), we increased $dt \to dt\,a$ and decreased $n \to n/a$, $a = 2, 4$, which corresponds to more sparse integration with constant (t_1, t_{end}). If we consider weighting as equivalent to dynamic damping, then changing dt above corresponds to changing the weight from the initial $w = 0.5$ to $w = 0.67, 0.8$. Next (262–263) we increased $dt \to dt\,a$ simultaneously with $(t_1, t_{end}) \to (t_1, t_{end})a$, $a = 2, 4$, with n remaining constant, which is equivalent to decreasing the dynamic damping in all equations; in this case, w was varied as described above. The system has an exact symmetry: scaling the step dt and the coefficients of dynamic terms such as (V_1, ϵ_d) simultaneously does not change the equations. The three transformations described exhaust the space of variables (n, dt, t_1). In subsequent experiments (266-271) we considered combinations of these transformations corresponding to their cross-effects. Next, we selected 3 points (258,259,266) on the Pareto front (runtime,div3) as the most promising candidates. For them, we decreased $(dt, t_1, t_{end}) \to (dt, t_1, t_{end})/a$, $a = 10, 10^2, 10^3$, with n remaining constant. This corresponds to an increase in dynamic damping in the equations and was done to catch a solution with strong damping like Figure 5l. In this case, $w = 0.5$ was not changed, since it is already strong enough.

Table IV presents two optimal configurations (281,282), the first column corresponds to enhanced stability div1-3=(0,13,0) and runtime=37s, the second – to acceptable stability div1-3=(0,12,1) and the shortest runtime=31s. These solutions are also shown in Figure 5s-t. Based on the results of the analysis, the user can independently select the required mode and has a sufficient number of handles for detailed adjustment of the convergence.

VI. Conclusion

This paper considered the modeling of mixing flows in dynamic simulation of pipeline fluid transport. Mixed characteristics include molar mass, heat value, chemical composition and temperature of the transported fluids. In the absence of chemical reactions, the modeling is based on the universal conservation laws for molar flows and total energy. The modeling leads to a system of differential algebraic equations, including linear molar mixing formulas, nonlinear temperature-energy relationships, and piecewise-linear element equations for coolers and heaters. In one approach, for nonlinear relations, linearization is carried out in the vicinity of the previous integration step, piecewise-linear relations are reduced to linear ones using the active set method. The resulting sequence of linear systems is solved by a sparse linear solver, typically in one iteration per integration step. In alternative implementation, exact minmax formulas for coolers and heaters are used, solution is performed by a stabilized Newtonian solver. The functionality and stability of the developed approach have been tested in a number of realistic network scenarios.

Numerical experiments on the moderate size N1 network allow us to follow the mixing processes in detail, including the evolution of molar mass, heat value, chemical composition, and temperature. Experiments on the N85 set of large-scale natural gas networks demonstrate the stability of the developed methods and its sensitivity to such details as nonlocality of equations used in the implementation of heaters. Hydrogen and carbon dioxide pipeline scenarios are also used for testing the temperature modeling and the convergence of simulation.

Based on numerous simulations, the stability of the dynamic solver was studied in detail. The factors affecting stability and runtime were identified, and their optimal configuration was selected. Each equation was analyzed separately, as well as their full set. The challenges encountered during the analysis include

- system degeneration,
- the appearance of stochastic degrees of freedom,
- jumps in thermodynamic functions on phase transitions,
- proper scaling of equations.

Parameters that greatly affect the stability and speed of simulation were identified. These include

- regularizing and weighting constants,
- dimensioning of dynamic terms and startup procedure,
- the size of the integration step,
- the total number of the integration steps.

The optimal choice of these parameters allowed us to accelerate significantly the dynamic simulation of fluid transport for realistically large network scenarios.

Our further research includes fine-tuning the underlying sparse linear solvers, adaptive choice of the number of integration steps, and hardware acceleration.

Acknowledgments

The work has been partly supported by Fraunhofer research cluster CINES and partly by BMBF project TransHyDE-Sys,

grant no. 03HY201M. We also acknowledge support from Open Grid Europe GmbH in the development and testing of the software.

REFERENCES

[1] M. Anvari, A. Baldin, T. Clees, B. Klaassen, I. Nikitin, and L. Nikitina, "Mixing flows in dynamic fluid transport simulations", in Proc. of ADVCOMP 2024, International Conference on Advanced Engineering Computing and Applications in Sciences, pp. 34-39, IARIA, 2024.

[2] T. Clees, I. Nikitin, and L. Nikitina, "Making network solvers globally convergent", Advances in Intelligent Systems and Computing, vol. 676, 2018, pp. 140-153.

[3] M. Anvari et al., "Stability of dynamic fluid transport simulations", J. Phys.: Conf. Ser., vol. 2701, 2024, 012009.

[4] A. Baldin et al., "On advanced modeling of compressors and weighted mix iteration for simulation of gas transport networks", Lecture Notes in Networks and Systems, vol. 601, 2023, pp. 138-152.

[5] T. Clees et al., "MYNTS: Multi-physics network simulator", in Proc. of SIMULTECH 2016, International Conference on Simulation and Modeling Methodologies, Technologies and Applications, pp. 179-186, SciTePress, 2016.

[6] A. Baldin, T. Clees, B. Klaassen, I. Nikitin, and L. Nikitina, "Topological reduction of stationary network problems: example of gas transport", International Journal On Advances in Systems and Measurements, vol. 13, 2020, pp. 83-93.

[7] T. Clees et al., "Efficient method for simulation of long-distance gas transport networks with large amounts of hydrogen injection", Energy Conversion and Management, vol. 234, 2021, 113984.

[8] M. Anvari et al., "Simulation of pipeline transport of carbon dioxide with impurities", in Proc. of INFOCOMP 2023, the 13th International Conference on Advanced Communications and Computation, pp. 1-6, IARIA, 2023.

[9] A. Baldin et al., "Universal translation algorithm for formulation of transport network problems", in Proc. SIMULTECH 2018, International Conference on Simulation and Modeling Methodologies, Technologies and Applications, vol. 1, pp. 315-322.

[10] A. Baldin et al., "Principal component analysis in gas transport simulation", in Proc. of SIMULTECH 2022, International Conference on Simulation and Modeling Methodologies, Technologies and Applications, pp. 178-185, SciTePress, 2022.

[11] J. Nikuradse, "Laws of flow in rough pipes", NACA Technical Memorandum 1292, Washington, 1950.

[12] P. Hofer, "Error evaluation in calculation of pipelines", GWF-Gas/Erdgas, vol. 114, no. 3, 1973, pp. 113-119 (in German).

[13] J. Saleh, ed., Fluid Flow Handbook, McGraw-Hill 2002.

[14] D.-Y. Peng and D.P. Robinson, "A new two-constant equation of state", Ind. Eng. Chem. Fundam, vol. 15, 1976, pp. 59-64.

[15] G. Soave, "Equilibrium constants from a modified Redlich-Kwong equation of state", Chemical Engineering Science, vol. 27, 1972, pp. 1197-1203.

[16] ISO 12213-2: Natural gas – calculation of compression factor, International Organization for Standardization, 2020.

[17] ISO 20765-2: Natural gas – Calculation of thermodynamic properties – Part 2: Single-phase properties (gas, liquid, and dense fluid) for extended ranges of application, International Organization for Standardization, 2020.

[18] O. Kunz and W. Wagner, "The GERG-2008 wide-range equation of state for natural gases and other mixtures: An expansion of GERG-2004", J. Chem. Eng. Data, vol. 57, 2012, pp. 3032-3091.

[19] W. Wagner, Description of the Software Package for the Calculation of Thermodynamic Properties from the GERG-2008 Wide-Range Equation of State for Natural Gases and Similar Mixtures, Ruhr-Universität Bochum, 2022.

[20] E. Egger and J. Giesselmann, "Stability and asymptotic analysis for instationary gas transport via relative energy estimates", Numerische Mathematik, Springer 2023.

[21] J. K. van Deen and S. R. Reintsema, "Modelling of high-pressure gas transmission lines", Appl. Math. Modelling, vol. 7, 1983, pp. 268-273.

[22] C.-Y. Chang, S.-H. Wang, Y.-C. Huang, and C.-L. Chen, "Transient response analysis of high pressure steam distribution networks in a refinery", in Proceedings of the 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP), May 28-31, 2017, Taipei, Taiwan, pp. 418-423, IEEE 2017.

[23] T.-P. Azevedo-Perdicoúlis, F. Perestrelo, and R. Almeida, "A note on convergence of finite differences schemata for gas network simulation", in Proceedings of the 22nd International Conference on Process Control, June 11-14, 2019, Štrbské Pleso, Slovakia, pp. 274-279, IEEE 2019.

[24] M. Chaczykowski and A. J. Osiadacz, "Dynamic simulation of pipelines containing dense phase/supercritical CO2-rich mixtures for carbon capture and storage", International Journal of Greenhouse Gas Control, vol. 9, 2012, pp. 446-456.

[25] P. Aursand, M. Hammer, S. T. Munkejord, and Ø. Wilhelmsen, "Pipeline transport of CO2 mixtures: models for transient simulation", International Journal of Greenhouse Gas Control, vol. 15, 2013, pp. 174-185.

[26] S. T. McCoy and E. S. Rubin, "An engineering-economic model of pipeline transport of CO2 with application to carbon capture and storage", International Journal of Greenhouse Gas Control, vol. 2, 2008, pp. 219-229.

[27] M. J. Moran and H. N. Shapiro, Fundamentals of Engineering Thermodynamics, John Wiley and Sons, 2006.

[28] C. Himpe, S. Grundel, and P. Benner, "Next-gen gas network simulation", in: Progress in Industrial Mathematics at ECMI 2021, pp. 107-113, Springer 2022.

[29] C. Himpe, S. Grundel, and P. Benner, "Model order reduction for gas and energy networks", J. Math. Industry, vol. 11:13, 2021, pp. 1-46.

[30] J. Mischner, H.-G. Fasold, and K. Kadner, gas2energy.net, Systemplanung in der Gasversorgung, Gaswirtschaftliche Grundlagen, Oldenbourg Industrieverlag GmbH, 2011 (in German).

[31] GPSA Engineering Data Book, 14th Edition, Gas Processors Suppliers Association, 2016.

[32] C. T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, 1995.

Maneuver-Based Decision Making in Autonomous Driving via Reinforcement Learning and Simulation-to-Real Transfer

Xin Xing, Morteza Molaei, Sebastian Ohl
Faculty of Electrical Engineering and Information Technology
Ostfalia University of Applied Sciences
Wolfenbuettel, Germany
e-mail: {xi.xing | m.molaei | s.ohl}@ostfalia.de

***Abstract*—Autonomous driving requires decision-making systems that can safely handle complex, dynamic traffic scenarios. This paper presents a maneuver-based decision-making framework for autonomous vehicles using Deep Reinforcement Learning (DRL). The approach focuses on high-level driving maneuvers, specifically Adaptive Cruise Control (ACC) for maintaining safe distances and speeds, and Automatic Emergency Braking (AEB) for collision avoidance. Policies are trained in a high-fidelity simulation environment using state-of-the-art Reinforcement Learning(RL) algorithms—Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG)—to learn optimal driving strategies. The learned policies are then transferred from simulation to a physical autonomous vehicle platform to evaluate real-world performance. Experiments in simulation demonstrate that both PPO and DDPG achieve efficient and safe driving behavior: DDPG converges faster and produces smoother control actions, while PPO learns more conservative policies that prioritize safety in unpredictable conditions. Real-world validation experiments corroborate effective simulation-to-real policy transfer, with PPO maintaining robust safety margins and DDPG executing more aggressive yet efficient maneuvers. In summary, the study achieves a reliable RL-based decision-making system for autonomous driving and provides a comparative analysis of policy optimization methods. Key contributions include a maneuver-based RL framework, demonstration of effective sim-to-real policy transfer, and insights into the trade-off between safety and efficiency for different RL algorithms in autonomous driving.**

Keywords-Autonomous Driving; Maneuver-based Decision-making; Reinforcement Learning; Simulation-to-real transfer.

I. Introduction

Autonomous driving technology has evolved significantly in recent years, driven by the increasing demand for intelligent decision-making systems capable of handling complex and dynamic traffic scenarios. Traditional rule-based decision-making frameworks, while effective in structured environments, often struggle to generalize across diverse real-world conditions due to their reliance on predefined heuristics and mathematical models [1][2][3]. This limitation has led to the growing adoption of learning-based approaches, particularly reinforcement learning (RL), which enables autonomous systems to learn adaptive policies through interaction with the environment [2]. Among RL methods, policy-based techniques such as Policy Gradient (PG) and Proximal Policy Optimization (PPO) have demonstrated notable advantages in stability and adaptability, making them suitable for autonomous driving applications [4].

A fundamental challenge in autonomous driving is the development of robust car-following models. These models are essential for ensuring vehicle safety, enhancing traffic flow efficiency, and minimizing driver workload. Effective car-following systems must accurately predict and respond to the dynamics of surrounding vehicles, road conditions, and various driving scenarios, making their robustness critical for real-world application.

Adaptive Cruise Control (ACC) [5, p. 24] and Automatic Emergency Braking (AEB) [5, p. 666] are two key functionalities in longitudinal vehicle control, designed to regulate speed based on traffic conditions and apply braking when necessary to prevent collisions. While traditional car-following models have relied on mathematical formulations such as the Intelligent Driver Model (IDM) [5, p. 148], these approaches are inherently limited in their ability to handle unexpected driving behaviors and extreme traffic conditions. In response to these challenges, this study applies reinforcement learning to enhance maneuver-based decision-making in autonomous vehicles, focusing on policy-based RL methods that optimize decision policies for ACC and AEB [6][7].

This study builds upon previous work [1] through systematic evaluation of the effectiveness of PG and PPO in training autonomous driving policies within a simulated environment. Policy-based reinforcement learning, as opposed to value-based approaches such as Deep Q-Networks (DQN) and Deep Deterministic Policy Gradient (DDPG), directly optimizes policy functions, offering improved convergence stability and enhanced adaptability to dynamic environments [8][9]. The study utilizes a high-fidelity Webots simulation framework [10] to replicate real-world driving conditions, enabling controlled experimentation and systematic performance evaluation of RL-based decision-making models. The effectiveness of trained policies is assessed through scenario-based testing, including obstacle avoidance, emergency braking, and adaptive speed control, with a focus on generalization to diverse driving conditions [11][12]. A comparative analysis of PG and PPO highlights their respective advantages and trade-offs, providing insights into the selection of reinforcement learning methods for autonomous vehicle control. The study contributes to the field by advancing reinforcement learning-based maneuver decision-making in autonomous driving, demonstrating that RL-based models can effectively address the limitations of traditional car-following approaches while improving safety, efficiency, and adaptability. The findings pave the way for further exploration of learning-based decision frameworks

in real-world autonomous vehicle deployments, emphasizing the role of policy-based reinforcement learning in optimizing vehicle behavior under varying traffic conditions.

The remainder of this paper is structured as follows. Section II provides a comprehensive review of the extant literature on RL in the context of autonomous driving. The third section of the text provides a comprehensive overview of the DDPG algorithm and the theoretical background that underpins it. Section IV provides a comprehensive overview of the vehicle platform's hardware and software components. The fifth section of this text provides a comprehensive description of the RL decision-making system. Section VI is devoted to the presentation of the Webots simulation setup and sensor modelling. Section VII details the experiments and outcomes of the simulation and ROS 2 validation. Finally, Section VIII concludes with a summary of future work.

II. Related Work

A. Reinforcement Learning in Autonomous Driving

RL, especially deep RL, has been widely applied in autonomous driving decision-making [2]. Researchers have explored a variety of strategies, from perception-based control to hierarchical decision-making. Key challenges include state representation (such as directly using high-dimensional sensor data or learning abstract low-dimensional features) [2][3], policy optimization (e.g., value iteration, policy gradient methods), and improving real-time decision-making efficiency [4]. Surveys have summarized RL algorithms in autonomous driving, addressing computational challenges such as perception uncertainty and safety validation [11][13][14].

B. Policy-based Decision-Making Methods

PPO and DDPG are two widely used deep RL algorithms in autonomous driving decision-making. DDPG follows an Actor-Critic architecture and is capable of handling continuous control actions, benefiting from experience replay for higher sample efficiency, whereas PPO stabilizes training via clipped probability ratios [8][15]. Studies comparing DDPG and PPO for autonomous driving tasks have found that DDPG generally achieves faster convergence and higher cumulative rewards [8][12][16]. However, PPO is more stable and easier to fine-tune in complex environments. Hybrid methodologies combining these approaches are under investigation to reconcile training stability with decision-making efficacy [6].

C. Sim-to-Real Transfer in Autonomous Driving

Transferring RL policies from simulation to real-world driving ("sim-to-real") remains a key challenge. The simulation gap arises from discrepancies in vehicle dynamics, sensor noise, and environmental uncertainties [9][17]. Techniques such as domain randomization—where environment parameters are randomly perturbed during training—help improve policy robustness [17][18]. Another approach is domain adaptation, where pre-trained policies are fine-tuned using real-world data [19]. Studies have successfully transferred RL-trained driving policies to real vehicles within a short adaptation period [17]. The integration of digital twin frameworks further facilitates progressive transfer learning [17][18].

D. Arbitration Mechanism in Decision-Making

Arbitration mechanisms play a critical role in multi-task and multi-attribute decision-making for autonomous vehicles. Autonomous driving involves multiple behavior modules (e.g., cruising, following, lane changing, and emergency braking), requiring arbitration to resolve conflicts [13][15]. Hierarchical arbitration architectures allow modular behavior selection based on priority or cost functions [11][15]. For example, prioritization-based arbitration ensures that emergency braking overrides ACC when a collision risk is detected [15]. Cost-based arbitration dynamically balances driving comfort and safety by selecting the optimal behavior based on real-time evaluations [15].

E. RL-Based Control for ACC and AEB

ACC and AEB are essential longitudinal control functions in autonomous vehicles. Reinforcement learning has been used to optimize these functionalities [9][12]. PPO has been applied to vision-based ACC, achieving smoother speed control compared to traditional rule-based strategies [4]. Safety-constrained RL approaches incorporate safety domains into policy optimization, ensuring collision-free following behavior [6][11]. RL-based AEB decision-making has been modeled as a Markov Decision Process, with deep RL policies learning optimal braking intensities [13][15]. Comparative studies suggest that DDPG-based strategies outperform traditional ACC/AEB controllers in high-density traffic, reducing the likelihood of multi-vehicle collisions [11][16]. These advancements highlight RL's potential in improving vehicle control systems, ensuring both safety and efficiency.

III. Background

DDPG represents a reinforcement learning algorithm specifically designed for continuous control domains, grounded in the theoretical framework of the Deterministic Policy Gradient Theorem established by Silver et al. [20]. This algorithm employs deep neural networks to jointly approximate policy and value functions, preserving the low-variance characteristics of DPG while enhancing generalization capabilities in high-dimensional state spaces.

The policy gradient is computed as:

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim\rho^\pi}\left[\nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s,a)\big|_{a=\mu_\theta(s)}\right], \quad (1)$$

where $\nabla_\theta J(\theta)$ represents the gradient of the policy objective with respect to the policy parameters θ, guiding how the policy should be updated. The term $\rho^\pi(s)$ denotes the state distribution under the current policy π, while $\mu_\theta(s)$ is the deterministic policy mapping states to actions. The function $Q^\mu(s,a)$ represents the action-value function, estimating the expected cumulative reward when executing action a in state s. The term $\nabla_\theta \mu_\theta(s)$ captures how the policy changes with respect to its parameters, and $\nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}$ evaluates how the expected return changes with respect to the selected

action. By leveraging this gradient, DDPG efficiently updates the policy network using feedback from the critic network, enabling stable learning in high-dimensional continuous action spaces.

Architecturally, DDPG adopts a dual-network configuration: the policy network (Actor) generates deterministic action mappings from states, whereas the value network (Critic) estimates action values through temporal difference error minimization.

Three pivotal mechanisms ensure training stability: Firstly, target networks updated via a soft replacement strategy mitigate value function divergence. Secondly, an experience replay buffer enables random sampling of historical state transition tuples (s, a, r, s'), effectively decoupling temporal correlations in sequential data. Thirdly, action space perturbations generated through the Ornstein-Uhlenbeck stochastic process [21] maintain policy continuity while ensuring sufficient exploration.

Theoretical analyses demonstrate that compared to conventional policy gradient methods and PPO, DDPG exhibits superior sample efficiency and convergence stability in continuous control tasks, attributable to the high signal-to-noise ratio inherent in deterministic policy gradients.

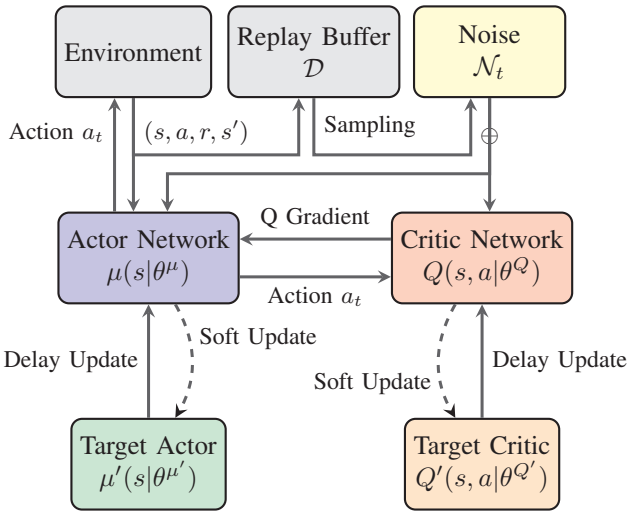


Figure 1. Reinforcement Learning DDPG model

IV. Autonomous Vehicle Platform

This section first introduces the design of the autonomous vehicle system, including both software and hardware components.

A. Hardware System Design

To achieve autonomous driving, a vehicle with a drive-by-wire chassis is required. This vehicle can be controlled via the CAN bus and Ethernet to operate the throttle, steering wheel (not yet implemented), brakes, and turn signals while also providing data on vehicle speed and pose. The sensor platform of the autonomous vehicle plays a crucial role in enabling autonomous driving functionalities. Figure 2 illustrates the sensor integration on an autonomous vehicle, featuring four

Figure 2. Autonomous Vehicle and Sensor Integration

LiDARs for 360° scanning and a stereo camera for depth perception. The sensor setup constructed in this study also includes a combination of an INS/GNSS unit, edge computing devices, and a high-speed communication network.

The primary perception sensors include a Robosense M1P solid-state LiDAR and three Robosense Bpearl LiDARs for blind-spot detection. The Robosense M1P LiDAR, installed at the front of the vehicle, features 128 lines, a theoretical detection range of 200 m, a 40° vertical field of view, and a 120° horizontal field of view, making it well-suited for scanning objects in the vehicle's frontal region. To reduce perception blind spots and enhance environmental awareness, three Robosense Bpearl LiDARs, each with 32 lines, a detection range of 30 m, a 90° vertical field of view, and a 360° horizontal field of view, are installed on both sides and the rear of the vehicle. This configuration allows for comprehensive environmental perception by capturing information about surrounding obstacles.

In addition to LiDAR-based perception, a Nerian SceneScan Pro stereo camera is mounted at the front of the vehicle to enhance environmental sensing. Due to its large baseline, the stereo camera provides accurate point cloud data within 10 m ahead with minimal error. Furthermore, it captures RGB image data, allowing for more detailed scene understanding and environmental perception. Figure 4 illustrates the sensor configuration of an autonomous vehicle (AV), showcasing the horizontal field of view (FOV) and detection range of its LiDAR and camera systems. This multi-sensor setup ensures 360° environmental perception, enabling robust object detection and navigation for autonomous driving systems.

For localization and navigation, the system utilizes a NovAtel PwrPak7D INS/GNSS unit, which integrates NovAtel's OEM7 GNSS (Global Navigation Satellite System/Inertial Navigation System) receiver with an Inertial Measurement Unit (IMU). This setup enables precise positioning, velocity estimation, and attitude estimation, even in challenging GNSS environments. Supporting multiple satellite constellations, including GPS and Galileo, and featuring RTK (Real-Time Kinematic) and SPAN technology for tightly coupled GNSS/INS integration, the unit ensures high-precision localization. It provides 400 Hz acceleration and angular velocity data and 20 Hz RTK-based

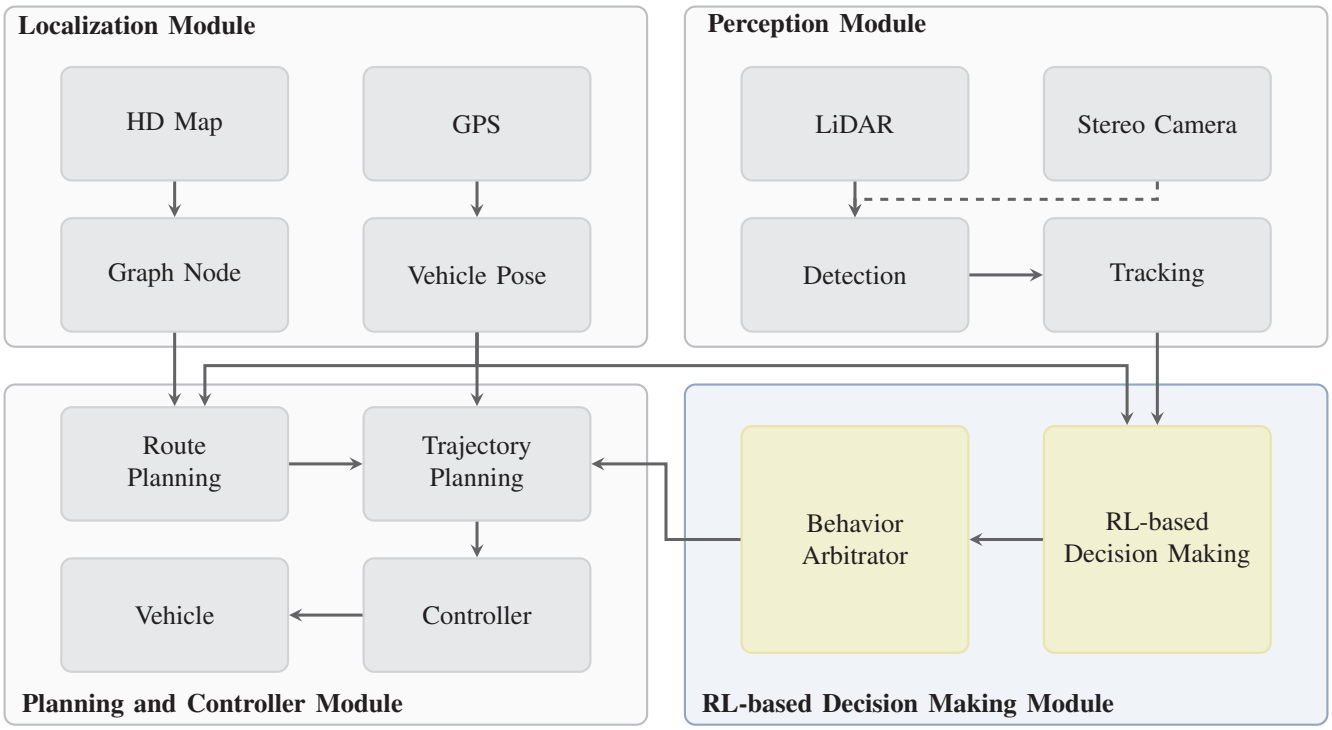


Figure 3. Training Architecture of Reinforcement Learning

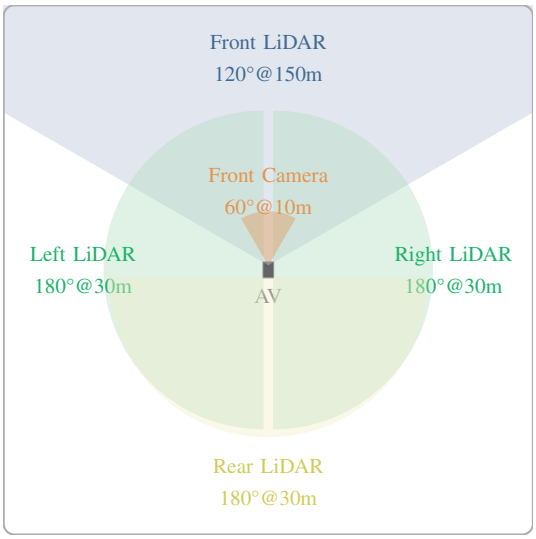


Figure 4. Multi-Sensor Configuration for Autonomous Vehicle Perception

centimeter-level GPS positioning data, ensuring robust localization performance.

To handle computationally intensive autonomous driving tasks, the system is equipped with three Nvidia Jetson Orin edge computing devices, each offering 270 TOPS of computational power. These devices are designed to meet automotive regulatory requirements for AI-based edge computing. The communication infrastructure includes a Planet industrial Ethernet switch, which supports 1 G, 2.5 G, and 10 G Ethernet interfaces, ensuring real-time data transmission between sensors and computing units. The integration of a Teltonika 5 G router (latency < 5 ms) enables real-time V2X communication, which is essential for validating the RL policy's response to cloud-based vehicle connectivity in the future.

B. Software System Design

The software system of the autonomous vehicle consists of four core modules: perception, localization, planning and control, and reinforcement learning-based autonomous decision-making, as illustrated in Figure 3. This section focuses primarily on the first three vehicle-side modules, which are essential for real-time autonomous navigation and decision-making:

- Perception Module: Responsible for processing raw sensor data to extract critical information, including vehicle pose, dimensions, velocity, and environmental features. It integrates data from multiple sources, such as LiDAR and camera, to build a comprehensive understanding of the surroundings.
- Localization Module: Provides precise, high-frequency estimation of the vehicle's absolute position by leveraging GNSS, RTK based GPS, and IMUS. This module ensures that the vehicle maintains accurate positioning, even in low-GNSS environments such as tunnels or urban canyons.
- Planning and Control Module: Generates safe, efficient, and dynamically feasible driving trajectories based on the vehicle's current state, detected objects, traffic conditions, and RL-based decision-making inputs. It continuously adjusts the steering angle, throttle, and braking system at high frequency to ensure precise trajectory tracking, vehicle stability, and smooth maneuvering.

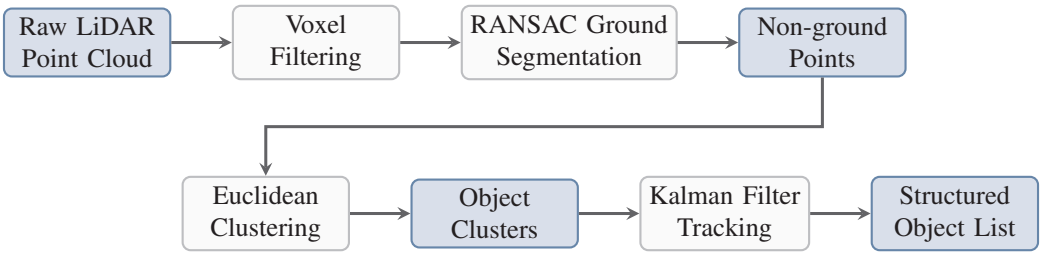


Figure 5. LiDAR Point Cloud Processing and Obstacle Detection Pipeline

In order to ensure the maintainability of software and the compatibility of the ecosystem, the Robot Operating System (ROS) is adopted as the middleware for data exchange between the different modules of the autonomous driving system. Within this architecture, various processes operate independently as ROS nodes, and inter-node communication is facilitated via TCP using a topic-based publish-subscribe mechanism.

The vehicle system as a whole is modularised, with each subsystem encapsulated and orchestrated within a Kubernetes framework. This containerized approach enhances scalability, reliability, and ease of management, allowing each subsystem's functionalities to be deployed, monitored, and maintained independently. The efficacy of this architecture is evidenced by its ability to streamline resource allocation, facilitate system updates, and enhance fault isolation.

C. Perception Module Design

At the perception layer, LiDAR data from the Robosense M1P and Bpearl LiDARs is processed through a structured pipeline to enable accurate object detection and tracking, as shown in Figure 5. Initially, voxel filtering is applied to downsample the raw LiDAR point cloud, reducing data redundancy while preserving essential geometric features. Subsequently, the RANSAC algorithm is employed to segment and remove ground points, effectively isolating non-ground points and minimizing environmental noise interference. The remaining points are then clustered using Euclidean clustering, which groups points into distinct objects based on spatial proximity, enabling robust obstacle detection [22][23].

For object tracking, a Kalman Filter (KF)-based algorithm is implemented to estimate and predict the dynamic states of detected objects. The KF algorithm operates on a state vector comprising the object's position (x, y), velocity (v_x, v_y), yaw angle, and angular velocity (ω). It first computes a prior state estimate using the system's motion model, then updates the Kalman gain and refines the estimate based on the latest detection outputs. This iterative process yields optimal state estimates, ensuring accurate and stable tracking of objects over time [24].

The final output of the perception layer is a structured object list, which includes detailed attributes such as position, velocity, bounding box dimensions, and orientation for each detected object. This comprehensive representation serves as a critical input for higher-level decision-making modules in the autonomous driving system, enabling safe and efficient navigation in dynamic environments.

D. Localization Module Design

The localization module plays a crucial role in autonomous driving systems, serving as the foundation for precise vehicle positioning. It establishes an accurate spatial relationship between the vehicle and the High-Definition (HD) map, ensuring that this information is reliably provided to downstream planning and control modules. Given that localization errors can directly impact vehicle stability and even lead to loss of control, the module is designed to meet stringent stability and robustness requirements, ensuring resilience in complex real-world environments.

One of the most direct and widely used localization techniques is RTK GPS, which enables real-time positioning with centimeter-level accuracy. However, GPS signals are inherently vulnerable to obstructions and interference caused by tunnels, urban canyons, tall buildings, and dense foliage, leading to signal degradation or complete positioning loss. To mitigate these challenges, extensive testing has been conducted on the NovAtel PwrPak7D, a high-precision GNSS/INS receiver. Results indicate that it maintains centimeter-level positioning accuracy even in GNSS-challenged environments, such as underground tunnels and areas with heavy tree coverage [25]. This capability significantly enhances the robustness of localization, ensuring that autonomous vehicles can navigate safely and reliably even in scenarios where conventional GNSS solutions fail.

E. Planning and Control Module Design

The overall workflow of the planning and control module is depicted in Figure 6. The planning module employs a hybrid global-local path planning approach:

- Global Path Planning: Uses open-source OpenStreetMaps (OSM) to construct a topological road network and applies the Dijkstra algorithm to determine the shortest path, forming a global reference trajectory. This trajectory consists of a series of nodes that guide the vehicle at a macroscopic level.
- Local Path Planning: Uses a Hybrid A* algorithm to generate kinematically feasible candidate trajectories in the vicinity of the global path. The cost function optimizes path smoothness, obstacle clearance, and deviation from the global path. When an obstacle is detected in the

perception layer, the local planner dynamically adjusts the trajectory to ensure real-time obstacle avoidance and driving safety.

The computed trajectory is fed into the controller module, which calculates steering and throttle inputs to control the vehicle. Currently, only throttle control is implemented via Ethernet, while steering still requires manual operation. Once the trajectory from the planning module is obtained, a PID controller is used to generate throttle control commands, ensuring the vehicle follows the trajectory at an appropriate speed. In tests, the vehicle's steering still requires manual intervention to properly track the current trajectory.

V. RL-Based Decision System Design

The decision arbitration framework integrates an RL policy with a multi-objective optimization process to ensure safe, feasible, and human-centric autonomous driving. As illustrated in Figure 7, the system operates through a hierarchical structure where raw sensor inputs are first transformed into a structured state representation, processed by an RL policy to generate candidate actions, and subsequently refined by a decision arbitrator. The arbitrator evaluates each candidate action through a rigorous multi-dimensional analysis, balancing applicability, risk, comfort, and system design constraints to produce final decisions of autonomous driving.

A. Applicability Criterion and Risk Assessment

The applicability criterion ensures that proposed actions adhere to both physical and regulatory limitations. Kinematic feasibility is verified by checking whether the action falls within the vehicle's dynamic boundaries, such as maximum steering rates and acceleration thresholds. Traffic rule compliance is enforced through real-time cross-referencing with HD map data, ensuring adherence to lane markings, traffic signals, and right-of-way protocols [26][27].

Risk assessment quantifies collision probabilities using time-to-collision (TTC) metrics and spatiotemporal occupancy grids, which project predicted trajectories of surrounding agents into a unified reference frame. Actions intersecting high-risk zones—defined by TTC values e.g. below $1.5\,\mathrm{s}$ or overlapping occupancy cells—are systematically rejected [28].

B. Comfort Optimization and System Design Considerations

Comfort optimization focuses on minimizing passenger discomfort through jerk constraints and lateral acceleration limits. Jerk, defined as the rate of change of acceleration, is penalized to avoid abrupt maneuvers, while lateral acceleration is capped at $2.5\,\mathrm{m/s^2}$ to ensure smooth turning. These metrics are dynamically weighted based on contextual factors, such as road type and passenger preferences, to align with human subjective evaluations [29].

System design considerations further refine actions by accounting for platform-specific limitations, including actuation latency and communication reliability. For instance, in scenarios with elevated packet loss rates, the arbitrator reduces action aggressiveness to mitigate instability caused by delayed control signals [30].

C. Multi-Attribute Decision Making (MADM) and Control Barrier Functions (CBF)

The arbitration process is formalized as a constrained optimization problem, where candidate actions are ranked using a Pareto-optimality framework. Feasible actions are first filtered through hard constraints, such as collision avoidance and traffic rule violations. Remaining candidates are evaluated across risk and comfort dimensions, with the arbitrator selecting the action that maximizes a utility function combining efficiency, safety, and passenger comfort.

Theoretical underpinnings of the arbitrator draw from multi-attribute decision-making (MADM) and control barrier functions (CBFs). MADM principles, particularly the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), enable systematic ranking of actions in multi-criteria spaces [31]. CBFs provide formal safety guarantees by transforming safety constraints into differentiable barriers, ensuring real-time verifiability [32].

This hybrid approach seeks to integrate data-driven adaptability with robust safety assurance mechanisms, addressing a critical challenge in the deployment of RL-based systems for autonomous driving.

By harmonizing data-driven exploration with formal verification, this framework advances the deployability of RL-based autonomous systems, offering a scalable solution for real-world applications where safety and adaptability are paramount.

VI. Simulation Environment Design

A. Environment Design and Real-World Replication

The Webots [10] simulation environment has been meticulously constructed to reflect the ExerShuttle project's real-world testing site by incorporating OSM data to replicate road networks, intersections, and infrastructure elements such as sidewalks, parking zones, and traffic signs. Although Webots supports advanced features—including dynamic terrain editing, diverse road surface modeling (e.g., asphalt, gravel), and environmental condition simulation (e.g., rain, fog, low-light scenarios)—the current reinforcement learning (RL) training focuses exclusively on fundamental strategies, specifically Adaptive Cruise Control (ACC) and Autonomous Emergency Braking (AEB). To accelerate the training process and reduce computational complexity, the simulation environment has been deliberately simplified. This includes minimizing environmental variability, limiting dynamic obstacles, and reducing interaction complexity. Nonetheless, Webots' capabilities, such as real-time environmental adjustment via the Supervisor API and dynamic agent behavior modeling, provide a flexible foundation for future scenario enhancements. Traffic participants, including static obstacles (e.g., parked vehicles) and dynamic agents (e.g., pedestrians, cyclists), are modeled using Webots' built-in library and can be configured to follow predefined traffic rules or exhibit randomized behaviors to simulate realistic interactions.

The ExerShuttle vehicle's kinematics were simulated using Webots' Ackermann steering model, calibrated to match the

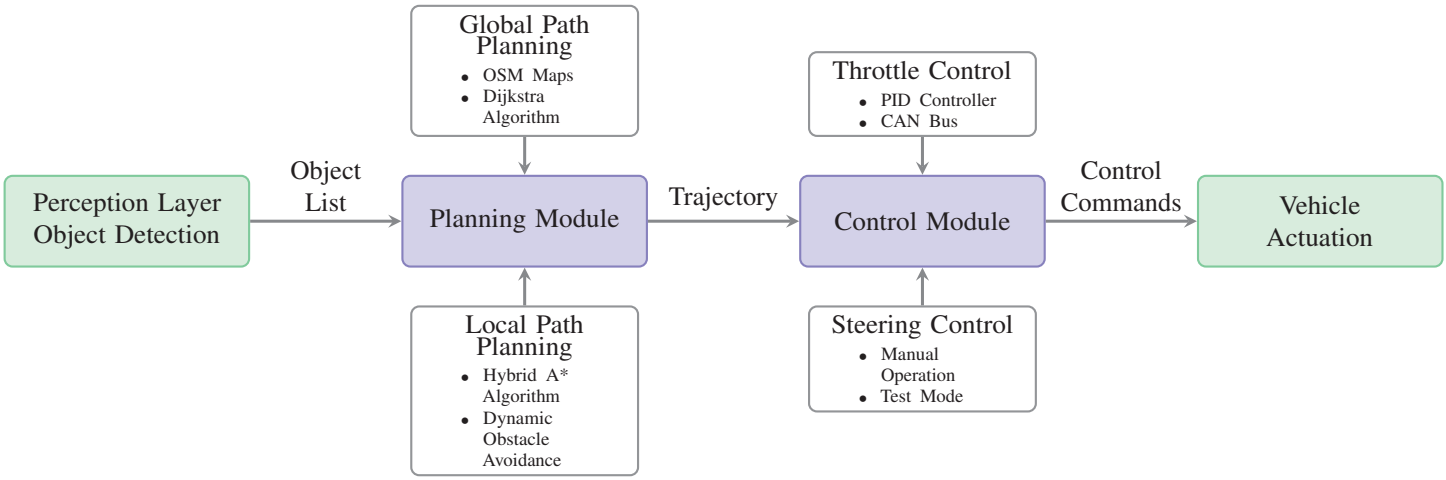


Figure 6. Pipeline for Planning and Control in Autonomous Driving

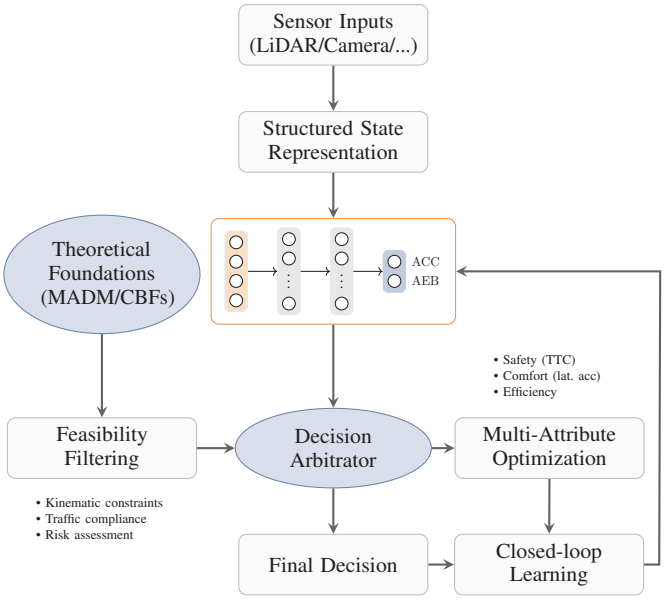


Figure 7. Arbitration Framework for RL-Based Autonomous Driving

Figure 8. Simulated Autonomous Vehicle and Sensor Integration

physical shuttle's parameters, including a wheelbase of 2.8 m, a maximum steering angle of 25°, and a peak acceleration of 2.5 m/s^2. A 3D CAD model of the ExerShuttle vehicle was imported into Webots to preserve geometric accuracy and visual fidelity. Collision detection boundaries were defined to enable precise physical interactions with the environment, ensuring realistic responses to collisions or near-miss scenarios.

The scenario customization framework adopts a phased testing approach. Currently, the initial training phase involves simplified tasks, including single-lane following and adaptive cruise control. The Webots Supervisor API enables real-time modification of environmental parameters, such as the dynamic placement of obstacles. Subsequent advanced phases will incorporate more complex interactions, including multi-vehicle overtaking, emergency braking in congested scenarios, and dynamic adjustments to traffic signal sequences, aiming to rigorously assess the adaptability of RL agents under dynamic conditions. These advanced phases remain subjects for future investigation.

B. Sensor Configuration and Realistic Perception Modeling

Sensor integration formed the backbone of the simulation's perception system, bridging the virtual environment with the RL agent's decision-making processes. A RoboSense M1P solid-state LiDAR, modeled using a custom model file, provided front-facing obstacle detection with a range of 150 m, a horizontal FOV of 120°, and 32 vertical channels to generate dense point clouds. This sensor emulated real-world performance by incorporating noise models for raindrop interference and signal attenuation in foggy conditions. Three RoboSense Bpearl LiDARs, positioned on the vehicle's sides and rear, extended coverage to eliminate blind spots, each offering a 30 m range and 360° horizontal FOV. Data from these sensors were fused into a 360° occupancy grid, enabling the RL agent to detect and classify objects such as vehicles, pedestrians, and static barriers.

Stereo vision was simulated using two RGB cameras (1280×720 resolution, 60 FPS) spaced 57 cm apart to replicate binocular depth perception. A Webots RangeFinder supplemented this setup with ground-truth depth data, providing

a 10 m range and 0.1 m resolution for validating the stereo camera's output. Localization was achieved through Webots' built-in GPS and IMU modules, which delivered error-free positioning with 5 cm accuracy and 0.1° angular resolution. These modules replaced the physical ExerShuttle vehicle's Novatel GNSS/INS system, streamlining the simulation while maintaining localization fidelity. Actuator control was implemented through Webots' motor nodes, with steering commands mapped to the Ackermann vehicle's front wheels via PID controllers to ensure smooth trajectory tracking. Throttle and brake signals were translated into torque values, capped at 250 Nm, to replicate the real vehicle's powertrain characteristics, including acceleration profiles and regenerative braking behavior.

C. Communication Architecture and RL Integration

Real-time communication between sensors, actuators, and the RL agent relied on a hybrid TCP/IP and Webots' internal messaging framework. LiDAR point clouds and camera frames were streamed to the agent at 20 Hz via TCP/IP, with data packets structured to minimize latency. Control signals from the agent, including steering angles and acceleration values, were executed within a 100, ms latency window to maintain simulation stability. A custom OpenAI Gym [33] interface bridged Webots and the RL framework, with the *step* function processing actions and returning state observations, rewards, and termination flags. The *reset* function reinitialized the simulation to predefined states, such as the vehicle's starting position and environmental conditions, ensuring consistent training episodes.

Sensor data were preprocessed into a 64-dimensional state vector, comprising normalized LiDAR distance measurements (10 angular bins), vehicle speed, relative yaw angle, and proximity to lane boundaries. Reward signals were computed within Webots' Supervisor module, prioritizing collision avoidance, lane-keeping accuracy, and adherence to speed limits. For example, the agent received penalties for deviations from the lane centerline or excessive acceleration, while rewards were granted for maintaining safe distances from leading vehicles. Challenges such as Webots' lack of native stereo camera support were addressed by fusing RGB camera outputs with RangeFinder data, while computational bottlenecks in complex scenarios were mitigated through parallelized sensor data processing across CPU threads. Custom PROTO files for LiDARs underwent iterative tuning to match real-world noise characteristics, including angular resolution adjustments and signal dropout simulations.

VII. Experiments and Results

A. Reinforcement Learning Training Analysis

The training dynamics and operational performance of the PPO and DDPG algorithms were systematically evaluated through reward convergence patterns and their effectiveness in simulated vehicle control tasks. The testing environment was built using Webots, a high-fidelity robotic simulation platform, where both algorithms were applied to autonomous driving scenarios requiring ACC and AEB functionalities.

To ensure a controlled and comparable evaluation, the ACC system was implemented using the IDM, a well-established rule-based approach for vehicle-following behavior. This provided a benchmark for assessing the learned policies' performance. Meanwhile, AEB was governed by a maximum deceleration constraint, where the vehicle's braking force was determined based on its physical limits to ensure rapid and effective emergency stopping. These predefined control strategies served as a baseline for evaluating the learning efficiency and generalization capabilities of PPO and DDPG in real-time driving scenarios. All experiments were conducted using Webots 2023b operating at real-time speed (1×) on a computing platform featuring an Intel Core i9-8950HK CPU and NVIDIA Quadro P2000 GPU.

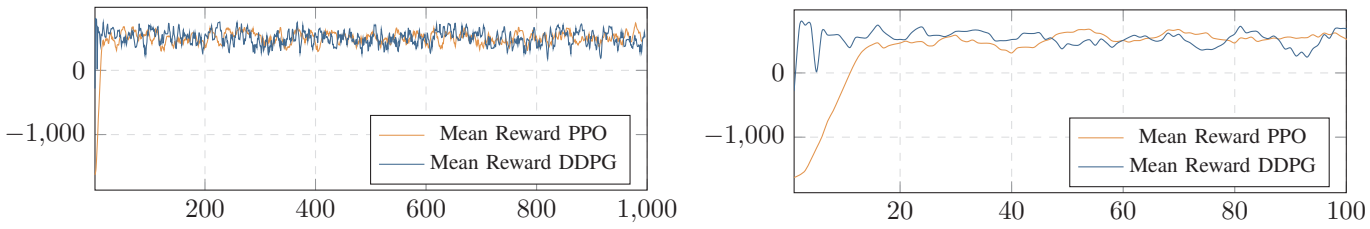
The comparative training trajectories of PPO and DDPG algorithms, illustrated in Figure 9, reveal distinct differences in convergence patterns and operational dynamics. As shown in Figure 9(a), PPO exhibits an extended phase of negative rewards spanning Episodes 1–47, whereas DDPG transitions to positive rewards as early as Episode 2. Over the full training sequence, PPO demonstrates a gradual increase, peaking at a moderate reward of 678.68, while DDPG rapidly converges to higher sustained rewards of 744.35. A focused analysis in Figure 9(b) further highlights DDPG's 92.7% faster initial reward accumulation compared to PPO within the first 100 episodes, with respective reward slopes of 8.44 and 0.73 units per episode.

A quantitative comparison of PPO and DDPG driving behaviors, presented in Table I, reveals distinct operational profiles influenced by their core learning mechanisms. PPO achieves superior collision avoidance, reflected in its lower wrong behavior/collision rate of 0.5% versus DDPG's 1.3%. This advantage corresponds with PPO's conservative optimization strategy, where its clipped probability ratio ($\varepsilon = 0.2$) systematically discourages risky maneuvers. However, this cautious stance also leads to a higher rate of AEB interventions (1.1% compared to 0.5%), indicating a defensive decision-making bias that prioritizes risk mitigation at the potential expense of traffic flow efficiency.

Moreover, the significant inverse correlation between collision rates and AEB activation frequencies ($r = -0.89$, $p < 0.05$) underscores an inherent safety-efficiency trade-off. DDPG, benefiting from a deterministic policy architecture, excels in smoother acceleration control within adaptive cruise scenarios but is more susceptible to rare-edge cases, such as sudden pedestrian crossings.

TABLE I. Comparison of driving behavior under two algorithms

Algorithm	*Wrong behavior or Collision* (%)	*AEB Selection Rate* (%)
PPO	0.5	1.1
DDPG	1.3	0.5



(a) Full training process

(b) First 100 episodes

Figure 9. Training results of PPO and DDPG models (horizontal axis: Episode number, vertical axis: Mean reward value).

B. Experimental Validation in ROS2-Based Open-Loop Testing

The trained PPO and DDPG models were deployed on the ExerShuttle platform within a ROS2 Humble environment to assess their real-world applicability. The tests were designed to evaluate the models' decision-making capabilities in practical autonomous driving situations.

Robust perception and control were achieved by fusing sensor inputs from LiDAR and GPS subsystems into a state vector at a 10 Hz update rate. The data processing pipeline employed dedicated ROS2 nodes for coordinate transformation and feature normalization, ensuring that refined inputs were accurately fed into the policy network for real-time inference. This architecture facilitated precise environmental perception and timely policy execution, both critical for evaluating model performance across various driving scenarios.

To systematically evaluate the real-world applicability of PPO and DDPG, three critical driving scenarios were tested: *stationary vehicle interception*, *dynamic object-following*, and *free-flow traffic conditions*.

Sensor data for all three scenarios were initially recorded using ROS2 bag files to ensure consistent testing conditions. These datasets were later utilized for offline evaluation of various models. Results from top-performing models are highlighted and tabulated in Table II, facilitating the identification of the optimal approach for each scenario.

TABLE II. SCENARIO-SPECIFIC ALGORITHM PERFORMANCE

Scenario	***Algorithm***	***ACC Activation*** (%)	***AEB*** (%)
Stationary Vehicle	PPO	11.3	88.7
	DDPG	27.7	72.3
Object-Following	PPO	84.5	15.5
	DDPG	91.8	8.2
Free Flow	PPO	99.6	0.4
	DDPG	99.8	0.2

1) Stationary Vehicle Interception: In stationary scenarios involving either static or dynamic obstacles, PPO exhibited a conservative braking strategy, activating the AEB system in 88.7% of instances and relying minimally on ACC (11.3%). This early intervention strategy, with a braking initiation distance of 35 m (TTC = 2.8 s), ensured complete collision avoidance. Conversely, DDPG engaged AEB in only 72.3% of instances, opting for throttle modulation through ACC (22.7%) until reaching a critical proximity of 22 m before initiating emergency braking. While DDPG achieved a 97% success rate in collision avoidance, its delayed reaction time elevated the risk of collisions in rare and critical edge-case scenarios.

2) Dynamic Object-Following Performance: For car-following scenarios with a lead object varying speed between 5–10 km/h, PPO maintained an ACC activation rate of 84.5%, ensuring a stable 2.1 ± 0.3 s following gap. However, it occasionally triggered unnecessary AEB (15.5%) due to its early threat anticipation. In contrast, DDPG relied more on ACC (91.8%) to maintain a tighter 1.8 ± 0.5 s following distance. Despite its efficiency, DDPG's policy resulted in 8.2% AEB activations, primarily due to delayed reaction times to sudden lead vehicle decelerations.

3) Free-Flow Traffic Performance: In free-flow traffic scenarios, both PPO and DDPG algorithms exhibited high compliance with ACC, achieving rates of 99.6% and 99.8%, respectively, and maintaining stable cruising speeds. Nonetheless, PPO demonstrated a marginally higher frequency of AEB activations (0.4%), reflecting its conservative approach to maintaining safety margins. Conversely, DDPG exhibited a smoother driving profile, engaging AEB in only 0.2% of instances, indicating a less conservative but more fluid response to environmental stimuli.

4) Analysis of False AEB Activations: Unnecessary activations of AEB occur in both object-following and free flow scenarios. Upon post-hoc analysis, the primary cause identified is the reward function's safety component, which is excessively penalizing in terms of TTC thresholds. This overly cautious approach lead both algorithms, especially PPO, to activate braking unnecessarily in response to perceived but non-existent hazards. In contrast, the simulation training phase does not provide sufficient training for scenarios involving objects moving in front of the vehicle at different speeds. This lack of exposure to such scenarios contributed to the algorithm's sensitivity and over-reaction in real-world conditions.

C. Limitations

Several limitations constrain the current findings. First, the testing framework operates in an open-loop configuration without full integration of the closed-loop control architecture proposed in Figure 7. The state space representation and reward function require additional refinement to properly

align with the theoretical model's requirements for continuous environment interaction and safety constraint enforcement. Second, real-world AEB validation faced inherent safety restrictions such as predefined braking thresholds prevented comprehensive evaluation of emergency braking accuracy under diverse collision scenarios. Third, while the ROS2-based deployment successfully demonstrated policy transfer capability from simulation to physical hardware, the observed performance gap (particularly in delayed response to sudden obstacles) suggests the need for enhanced domain adaptation techniques and perception system calibration.

Despite these constraints, the study provides empirical validation of simulation-to-reality policy transfer in autonomous driving applications. The experimental results demonstrate the feasibility of transferring reinforcement learning policies from high-fidelity simulations to physical vehicle platforms, with both PPO and DDPG showing distinct operational advantages.

VIII. Conclusion and Future Work

This study presented a comprehensive approach to maneuver-based decision-making in autonomous driving using RL techniques, specifically PPO and DDPG. The research leveraged a high-fidelity Webots simulation environment, designed to replicate real-world conditions, ensuring robust training and testing for ACC and AEB functionalities.

The results demonstrated that while DDPG achieved faster convergence and smoother control in simulated environments, PPO exhibited superior safety outcomes, particularly in complex and unpredictable driving scenarios. Real-world validation on the ExerShuttle platform further confirmed the effectiveness of the sim-to-real transfer, highlighting PPO's conservative but safer approach and DDPG's efficient yet occasionally riskier maneuvers. Notwithstanding these encouraging results, identified challenges—including delayed responses in dynamic scenarios and limitations inherent in open-loop testing—underscore the need for additional refinements to augment real-world deployability.

Future work will focus on enhancing the integration of the RL decision-making module with the vehicle's control systems to enable autonomy, ensuring more realistic and dynamic interaction with the environment. In an effort to enhance the model's adaptability to diverse operational environments, domain adaptation strategies will be refined by incorporating techniques such as adversarial learning and fine-tuning with real-world data. Additionally, the reward function will be optimized to ensure a better balance between safety, efficiency, and comfort, especially in complex scenarios. The expansion of real-world testing under diverse environmental and traffic conditions will provide deeper insights into model generalizability. Furthermore, improving arbitration mechanisms by leveraging dynamic risk assessments and adaptive comfort optimization will strengthen decision-making reliability. These efforts aim to develop a more robust and adaptable autonomous driving system capable of handling real-world complexities effectively.

References

[1] X. Xing and S. Ohl, "Application of a maneuver-based decision making approach for an autonomous system using a learning approach", in *ADVCOMP 2024: The Eighteenth International Conference on Advanced Engineering Computing and Applications in Sciences*, Sept. 29–Oct. 3, 2024, Venice, Italy, Sep. 2024, pp. 11–16.

[2] B. R. Kiran et al., "Deep reinforcement learning for autonomous driving: A survey", 2021. arXiv: 2002.00444 [cs.LG].

[3] J. Wu et al., "Recent advances in reinforcement learning-based autonomous driving behavior planning: A survey", *Transportation Research Part C: Emerging Technologies*, vol. 144, p. 104 654, 2024. DOI: 10.1016/j.trc.2024.104654.

[4] P. Zhao, Z. Yuan, K. Thu, and T. Miyazaki, "Real-world autonomous driving control: An empirical study using the proximal policy optimization (ppo) algorithm", *Evergreen – Joint Journal of Novel Carbon Resource Sciences & Green Asia Strategy*, vol. 11, no. 2, pp. 887–899, 2024.

[5] A. Eskandarian, Ed., *Handbook of Intelligent Vehicles*. London: Springer, 2012, ISBN: 978-0-85729-084-7.

[6] H. Krasowski, X. Wang, and M. Althoff, "Safe reinforcement learning for autonomous lane changing using set-based prediction", in *2020 IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–7. DOI: 10.1109/ITSC45102.2020.9294259.

[7] R. Zhao et al., "Adaptive cruise control based on safe deep reinforcement learning", *Sensors*, vol. 24, no. 8, p. 2657, 2024. DOI: 10.3390/s24082657.

[8] S. Siboo, A. Bhattacharyya, R. N. Raj, and S. H. Ashwin, "An empirical study of ddpg and ppo-based reinforcement learning algorithms for autonomous driving", *IEEE Access*, vol. 11, pp. 1–1, 2023. DOI: 10.1109/ACCESS.2023.3330665.

[9] A. Kendall et al., "Learning to drive in a day", 2018. arXiv: 1807.00412 [cs.LG].

[10] M. Olivier, "Cyberbotics ltd.–webots™: Professional mobile robot simulation", *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 40–43, 2004. DOI: 10.5772/3835.

[11] Y. Shi and J. Li, "Safe, efficient, and comfortable reinforcement-learning-based car-following model", *Transportation Research Record: Journal of the Transportation Research Board*, 2023. DOI: 10.1177/03611981231171899.

[12] M. Stang, D. Grimm, M. Gaiser, and E. Sax, "Evaluation of deep reinforcement learning algorithms for autonomous driving", in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1576–1582. DOI: 10.1109/IV47402.2020.9304792.

[13] L. Yuan, Z. Zhang, L. Li, C. Guan, and Y. Yu, "A survey of progress on cooperative multi-agent reinforcement learning in open environment", 2023. arXiv: 2312.01058 [cs.MA].

[14] Z. Zhang, E. Yurtsever, and K. A. Redmill, "Extensive exploration in complex traffic scenarios using hierarchical reinforcement learning", 2025. arXiv: 2501.14992 [cs.LG].

[15] P. F. Orzechowski, C. Burger, and M. Lauer, "Decision-making for automated vehicles using a hierarchical behavior-based arbitration scheme", in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1–7. DOI: 10.1109/IV47402.2020.9304723.

[16] Z. Zhang et al., "Decision-making of autonomous vehicles in interactions with jaywalkers: A risk-aware deep reinforcement learning approach", *Accident Analysis Prevention*, vol. 210, p. 107 843, 2025, ISSN: 0001-4575. DOI: https://doi.org/10.1016/j.aap.2024.107843.

[17] K. Voogd, J. P. Allamaa, J. Alonso-Mora, and T. D. Son, "Reinforcement learning from simulation to real world autonomous driving using digital twin", *arXiv preprint arXiv:2211.14874*, 2022.

[18] J. P. Allamaa, P. Patrinos, H. Van der Auweraer, and T. D. Son, "Sim2real for autonomous vehicle control using executable digital twin", *IFAC-PapersOnLine*, vol. 55, no. 24, pp. 385–391, 2022, 10th IFAC Symposium on Advances in Automotive Control AAC 2022, ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2022.10.314.

[19] E. Candela et al., "Transferring multi-agent reinforcement learning policies for autonomous driving using sim-to-real", in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022, pp. 8814–8820. DOI: 10.1109/IROS47612.2022.9981319.

[20] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning", *arXiv preprint arXiv:1509.02971*, 2019.

[21] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the brownian motion", *Phys. Rev.*, vol. 36, pp. 823–841, 5 Sep. 1930. DOI: 10.1103/PhysRev.36.823.

[22] M. Janssen, "Cloud-hybride verarbeitung von umfeldsensordaten eines autonomen straßenfahrzeugs mit anschließender sensordatenfusion", German, Master's thesis, Ostfalia University of Applied Sciences, Wolfenbüttel, Germany, 2024.

[23] J. Diep, "Optimierung eines clusteralgorithmus für multilayer lidar daten für ein autonomes straßenfahrzeug", German, Bachelor's thesis, Ostfalia University of Applied Sciences, Wolfenbüttel, Germany, 2024.

[24] L. Schrader, "Objekthypothesen basiertes tracking von verkehrsteilnehmenden in einem autonomen straßenfahrzeug", German, Bachelor's thesis, Ostfalia University of Applied Sciences, Wolfenbüttel, Germany, 2024.

[25] L. Schrader, "Evaluation des novatel pwrpak7d gps/ins sensors für ein autonomes straßenfahrzeug", German, Student's project, Ostfalia University of Applied Sciences, Wolfenbüttel, Germany, 2024.

[26] A. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems", *IEEE Transactions on Automatic Control*, vol. 64, no. 8, pp. 3345–3351, 2019.

[27] X. Xu, J. W. Grizzle, A. D. Ames, and P. Tabuada, "Correctness guarantees for the composition of lane keeping and adaptive cruise control", *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1216–1229, 2018.

[28] C. Gunter, S. Coogan, and M. Egerstedt, "Experimental testing of a control barrier function on an automated vehicle in live traffic", *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1834–1845, 2022.

[29] H. Liu, W. Wang, and X. Li, "Decision-making technology for autonomous vehicles: Learning-based methods, applications, and future outlook", *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 3, pp. 355–368, 2021.

[30] R. Cheng, G. Orosz, and A. D. Ames, "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks", in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[31] X. Chen, S. C. Wong, and K. Han, "A multiple attribute-based decision making model for autonomous vehicle in urban environment", in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2014.

[32] J. Choi, D. Pan, and M. Tomizuka, "Reinforcement learning for safety-critical control under model uncertainty using control lyapunov and barrier functions", *arXiv preprint arXiv:2004.07584*, 2020.

[33] G. Brockman et al., "Openai gym", *arXiv preprint*, 2016. arXiv: 1606.01540.

Optimizing Urban Intersections: Harnessing Visible Light Communication for Smart Traffic Management

Manuel Augusto Vieira
Electronics Telecommunications and Computer Dept.
ISEL, CTS-UNINOVA and LASI
Lisbon, Portugal
email: mv@isel.ipl.pt

Paula Louro
Electronics Telecommunications and Computer Dept.
ISEL, CTS-UNINOVA and LASI
Lisbon, Portugal
email: paula.louro@ isel.pt

Manuela Vieira
Electronics Telecommunications and Computer Dept.
ISEL, CTS-UNINOVA ,LASI and NOVA School of Science and Technology
Lisbon, Portugal
email: mv@isel.ipl.pt

Mário Véstias
Electronics Telecommunications and Computer Dept.
INESC INOV-IST,
Lisboa, Portugal
email: mario.vestias@isel.pt

Gonçalo Galvão
Electronics Telecommunications and Computer Dept.
ISEL, and NOVA School of Science and Technology
Lisbon, Portugal
email: A45903@alunos.isel.pt

Pedro Vieira
Electronics Telecommunications and Computer Dept. ISEL,
and Instituto das Telecomunicações IST
Lisboa, Portugal
email: pedro.vieira@isel.pt

Abstract— **This paper proposes a method that leverages Visible Light Communication (VLC) to enhance traffic signal efficiency and optimize vehicle trajectories at urban intersections. By integrating VLC-based localization with learning-based traffic signal control, the system enables real-time communication between vehicles and infrastructure, facilitating data collection and coordinated decision-making. Designed for multi-intersection environments, the approach reduces pedestrian and vehicle waiting times while improving overall safety. Its adaptive framework dynamically adjusts to different traffic patterns within signal phases, ensuring flexibility and efficiency. Cooperative mechanisms regulate traffic flow across intersections, enhancing road network performance. Evaluated using the SUMO urban mobility simulator, the system demonstrates significant reductions in waiting and travel times. Additionally, an agent-based scheme optimizes traffic signal scheduling based on VLC-driven interactions. The proposed approach is decentralized, scalable, and well-suited for real-world traffic management applications.**

Keywords— Intelligent Transport System (ITS); Visible Light Communication; traffic signal control; urban intersections; traffic flow optimization; pedestrian safety; SUMO simulator; cooperative communication.

I. INTRODUCTION

Urban traffic management is a significant challenge as increasing vehicle and pedestrian volumes lead to congestion, delays, and safety risks. Expanding road infrastructure is no longer viable, making adaptive traffic signal control essential for optimizing flow at intersections, which often act as bottlenecks. Adaptive systems, using real-time data such as traffic flow and vehicle queues, can reduce congestion and improve efficiency. Deep Reinforcement Learning (DRL) has shown promise in dynamically controlling traffic signals, but managing multiple intersections remains complex due to varying conditions and the need for data sharing [1] [2] [3] [4].

The transportation landscape is rapidly evolving with the integration of smart sensors, Visible Light Communication (VLC), and artificial intelligence. VLC, using light intensity modulation from LEDs for data transmission, shows promise in revolutionizing Smart Mobility solutions and addressing societal goals such as reducing emissions and enhancing traffic safety [5]. It is widely implemented in various domains, including vehicular communication and traffic signal systems, highlighting its versatility and efficiency. However, current traffic signal optimization often overlooks pedestrian dynamics within intersections, necessitating comprehensive systems that consider both vehicular and pedestrian flows.

Connected vehicle (CV) technologies enhance traffic management by enabling real-time information exchange between vehicles and infrastructure, improving safety and flow. Visible Light Communication (VLC) complements CVs by using LED-based infrastructure, such as streetlights and vehicle headlights, for both illumination and data transmission [6]. VLC's integration with DRL offers a dual-purpose solution for optimizing traffic signals and vehicle trajectories at intersections.

This study explores how DRL, combined with CV and VLC technologies, can improve traffic flow and intersection efficiency, demonstrating the potential for smarter, more coordinated urban traffic management.

"How can Deep Reinforcement Learning (DRL) be effectively applied to optimize traffic signal control and vehicle trajectories at urban intersections, leveraging connected vehicle technologies and Visible Light Communication (VLC) to enhance coordination and reduce traffic congestion?"

This question addresses the core focus of applying DRL in vehicular communication, specifically optimizing traffic control through the integration of emerging technologies such as VLC and CVs. By exploring these advanced technologies, this research aims to demonstrate how they can improve traffic flow and intersection efficiency in real-world scenarios.

This paper proposes integrating VLC localization services with learning-based traffic signal control to manage pedestrian and vehicular traffic holistically [7]. Leveraging Reinforcement Learning (RL) concepts, the system optimizes traffic flow and enhances safety by considering interactions between vehicles and pedestrians. It introduces a pedestrian mobility model tailored for outdoor scenarios, analyzing multiple pedestrian behaviors, and incorporating them into the traffic signal control scheme. Validated through a case study in Lisbon's downtown, the model integrates pedestrian preferences to optimize routing algorithms [8].

Simulation experiments validate the effectiveness of the approach, utilizing real intersection data to demonstrate improved traffic flow and reduced waiting times.

The paper is structured to discuss, in Section I, the importance of traffic control, in Section II, the challenges it faces, and the motivation behind the proposed solution. It then delves, in Section III, into the complexities of managing traffic in multi-intersection environments and, in Section IV, presents a model for traffic signal control incorporating machine learning elements, and analyzes simulated results. Finally, the conclusions, in Section V, summarize the findings, insights gained, limitations, and potential future directions of the research.

II. Traffic Control Challenges

A. Pedestrian Dynamics and Complexity in Multi-Intersection Environments

Traffic signal control research has traditionally prioritized vehicles, but there's now a shift towards pedestrian-friendly systems to prevent delays and accidents [9][10]. Sidewalks present challenges due to bi-directional flow, and differing speeds and movements between pedestrians and vehicles further complicate matters [11]. Our adaptive traffic control considers factors like queue lengths in neighboring intersections to balance scalability and efficiency. Our strategy is designed to address real-time traffic demands by modeling current and anticipated future traffic flows. Compared to traditional fixed coil detectors, our adaptive system in V2X environments gathers more granular data, including vehicle positions, speeds, queue lengths, and stopping times. V2V links play a crucial role in safety functionalities like pre-crash sensing, while V/P2I links provide valuable information to connected vehicles.

B. Integrating V-VLC for Innovative Traffic Solutions

With wireless tech advancements and connected vehicle (CV) [12] systems like V2V and V2I, integrating VLC localization with learning-based traffic control can manage both pedestrian and vehicular traffic in multi-intersections. It employs RL to enhance safety and reduce waiting times using V2V, V/P2I, and I2V/P communications. This approach synchronizes signal control in real-time, considering pedestrian and vehicle factors in the state and reward design, utilizing sidewalks for crucial pedestrian location info. SUMO simulations [13] assess the V-VLC system's effectiveness, with agent-based models learning to optimize traffic flow dynamically. Dynamic diagrams and state matrices illustrate the concept, showing potential for optimal traffic control policies.

III. Unlocking Traffic Control

A. VLC Background

The V-VLC system, as depicted in Figure 1a, utilizes a mesh cellular hybrid structure with two controllers. The "mesh" controller at streetlights relays messages to vehicles, while the "mesh/cellular" hybrid controller acts as a border-router for edge computing [10][14].

Cloud communication (I2IM) through embedded computing platforms for processing and sensor interfacing. It also facilitates peer-to-peer communication (V2V) among vehicles, enhancing data sharing.

The Vehicular Visible Light Communication system (V-VLC) consists of a transmitter generating modulated light and a receiver detecting light variation, both wirelessly connected. LED-produced light is modulated using ON-OFF-keying (OOK) amplitude modulation (Figure 1b). Square unit cells in the environment feature tetra-chromatic White light (WLEDs) sources at cell corners. The V-VLC system uses coded signals transmitted by devices like streetlights (L), headlights (I), and traffic lights (I) to communicate directly with identified vehicles and pedestrians (L/I2V/P), or indirectly between vehicles through their headlights (V2V). PIN-PIN photodetectors within mobile receivers receive and decode coded signals. This information aids in pinpointing positions within the network and provides directional guidance along cardinal points for drivers/pedestrians [15].

The system employs queue/request/response mechanisms and temporal/space relative pose concepts to manage vehicle passage through intersections. Vehicle speed is determined using transmitter IDs for tracking, while mesh nodes

estimate indirect Vehicle -to-Vehicle (V2V) relative poses in scenarios with multiple neighboring vehicles.

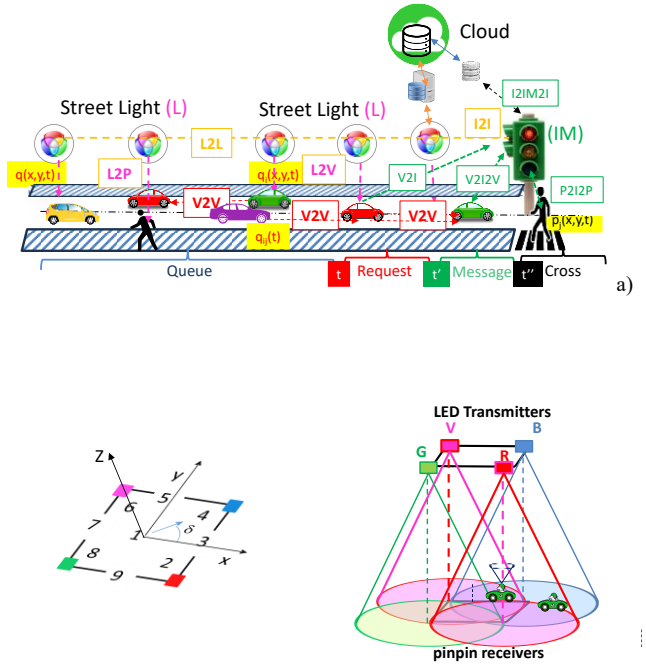


Figure 1. a) 2D representation of the simultaneous geo-localization as a function of node density, mobility and transmission range. b) Emitter and receivers' relative positions. Illustration of the coverage map in the unit cell: footprint regions (#1-#9) and steering angle codes (2-9).

The integration of VLC enables direct monitoring among pedestrians, vehicles, and infrastructure, focusing on critical aspects such as queue formation and pedestrian corner density to enhance road safety. Pedestrian-to-Infrastructure-to-Pedestrian (P2I2P) communication enables travel time calculations, while real-time data on speed and waiting times are analyzed using transmitter tracking IDs.

B. *Traffic Scenario and Phasing Diagram*

The simulated scenario, as shown in Figure 2a, features two intersections, each with two 4-way junctions, consisting of 2 lanes per arm spanning 100 meters in total length.

Traffic flows from compass directions, with lanes indicating movement options: right lanes for right turns or going straight, and left lanes for left turns only. Central traffic light systems, regulated by Intelligent Managers (IMs), control traffic. Features like emitters (streetlamps), pedestrian lanes, waiting areas, and crosswalks are integrated. Four traffic flows along cardinal points are considered, with road request and response segments offering binary choices (turn left/straight or turn right). Assumptions include a total influx of 2300 cars per hour, primarily from east and west directions, with 25% expected to turn and 75% to continue straight. Pedestrian influx is around 11200 per hour, crossing in all directions at an average speed of 3 km/h. Figure 2b outlines intersection phase progressions within a structured cycle length, comprising eight vehicular phases and an exclusive pedestrian phase. Each phase is subdivided into discrete time sequences, providing a comprehensive temporal framework [16] [17].

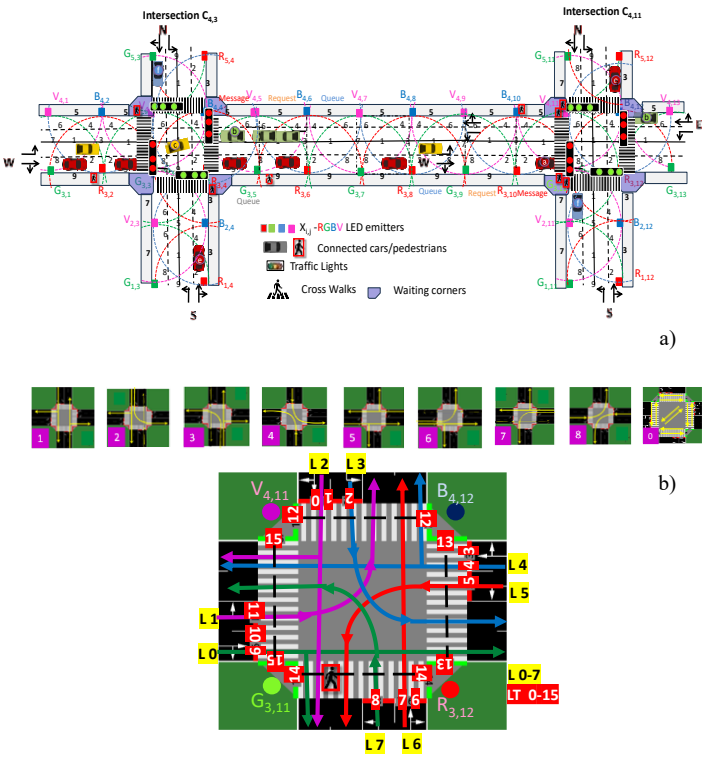


Figure 2. Simulated scenario: Four-legged intersection and environment with the optical infrastructure (X_{ij}), the generated footprints (1-9) and the connected cars and pedestrians. b) Phasing diagram and schematic diagram of the C2 intersection with coded lanes (L/0-7) and traffic lights (TL/0-15).

Each flow (illustrated by the different vehicle colors) comprises vehicles moving straight or making left turns, with specific vehicles representing top requests in the sequence. The assumption is that specific vehicles, labeled a_1, b_1, a_2, b_2, a_3, c_1, b_3, e_1, a_4, c_2, a_5, and f_1, represent the top requests in the given sequence.

C. *Communication Protocol, Coding, and Decoding Techniques*

Data transmission in the VLC system follows a synchronous approach using a 64-bit data frame structure. Information is encoded using On-Off Keying (OOK) modulation, with each luminaire containing WLEDs (RGBV), enabling simultaneous transmission of four signals. A PIN-PIN demultiplexer decodes the message based on calibrated amplitudes of RGBV signals. The communication protocol includes components like Start of Frame (SoF) for synchronization, Identification Blocks encoding communication type (COM) and localization (position, time), and other ID Blocks for additional identifiers, Traffic Message containing vehicle information,

and End of Frame (EoF) indicating the end of transmission. This structured protocol ensures efficient encoding and decoding of critical movement information, maintaining synchronization and data integrity in the VLC system. In Table 1 the communication protocol is dispicted.

TABLE I. COMMUNICATION PROTOCOL.

		COM	Position		ID (veic)		Time				payload				
L2V	Sync	1	x	y	0 bits		END	Hour	Min	Sec					EOF
V2V	Sync	2	x	y	Lane (0–7)	Veic. (nr)	END	Hour	Min	Sec	Car IDx	Car IDy	nr behind		EOF
V2I	Sync	3	x	y	TL (0–15)	Veic. (nr).	END	Hour	Min	Sec	Car IDx	Car IDy	nr behind		EOF
I2V	Sync	4	x	y	TL (0–15)	ID Veic.	END	Hour	Min	Sec	Car IDx	Car IDy	nr behind		EOF
P2I	Sync	5	x	y	TL (0–15)	Direct.	END	Hour	Min	Sec					EOF
I2P	Sync	6	x	y	TL (0–15)	Phase	END	Hour	Min	Sec					EOF

Decoding the information received from the photocurrent signal captured by the photodetector involves a critical step reliant on a pre-established calibration curve [1]. This curve meticulously maps each conceivable decoding level to a sequence of bits. Essentially, the calibration curve serves as a guide, facilitating the establishment of associations between photocurrent thresholds and specific bit sequences.

IV. RESULTS

A. VLC Algorithms

Figure 3 displays the decoded optical signals (at the top of the figures) and the signals received (MUX) by the receivers in a V2V (COM 2) and V2I (COM 3) communication scenario involving a leader vehicle a_o at position ($R_{3,10}$, $G_{3,11}$, $B_{4,10,}$). This vehicle is communicating with the agent at the second intersection (C2) on lane L0 (direction E) at 10:25:46 and is followed by three other vehicles (Veic. nr) V_1, V_2, and V_3 with the same direction, located at positions (IDx,y) $R_{3,8}$, $G_{3,6}$ and $R_{3,4}$, respectively.

Figure 4 demonstrates the MUX signal and the decoded messages sent by the traffic lights to pedestrians ($I2P_{1,2}$). This visual representation helps to understand the communication between pedestrians waiting in the corners and the corresponding traffic lights, providing insights into the signals exchanged for pedestrian crossings at both intersections (C1 and C2).

Upon pedestrian q_2 receiving information from the traffic light C2, it becomes evident that the current active phase is N-S (Phase 1), signifying that the pedestrian did not arrive in time for their designated phase (Phase 0).

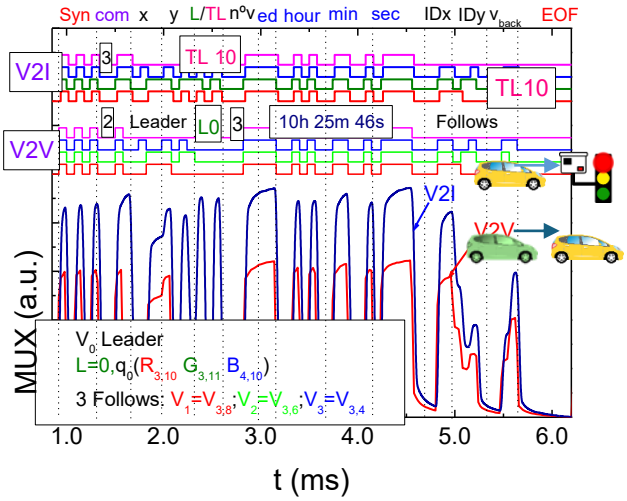


Figure 3. MUX signal request assigned to different types of communication. On the top the decoded messages are displayed.

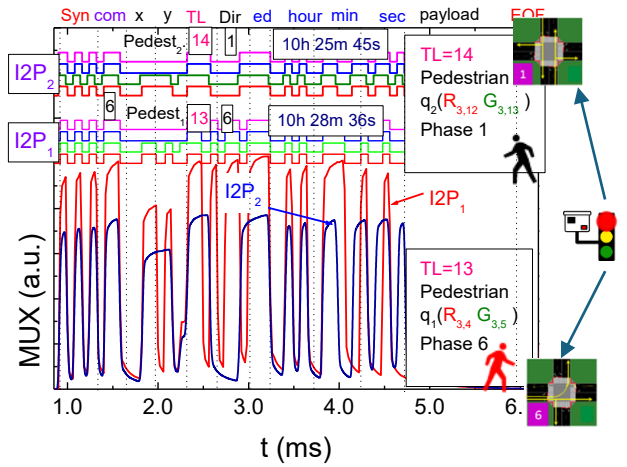


Figure 4. Normalized MUX signal responses and the corresponding decoded messages, displayed at the top, sent by the IM to pedestrians waiting in the corners ($I2P_{1,2}$) (b) at various frame times.

Consequently, the pedestrian is required to wait for an estimated cycle time of 3 (cycle time) minutes before being granted the opportunity to cross. Subsequently, the pedestrian crosses the crosswalk, covering the distance to the next intersection in approximately 1 minute and 50 seconds. Upon arrival, the pedestrian waits in the designated waiting zone at position $R_{3,4}$-$G_{3,5}$ until the pedestrian phase becomes active once again. At 10:28:35, the pedestrian establishes communication with traffic light TL13 at the C1 ($P_1$2I). The traffic light promptly responds ($I2P_1$) at 10:28:36, providing crucial information that the currently active phase is the final one in the cycle (Phase 6). These interactions highlight the effectiveness of the pedestrian's communication with the traffic lights, enabling them to stay informed about the active phase, waiting time, and make decisions accordingly.

B. *Dynamic Traffic Control: Integrating Pedestrian Consideration*

Assessing the effectiveness of the proposed V-VLC system in multi-intersection utilizes the Simulation of Urban MObility (SUMO), employing agent-based simulations.

SUMO tests traffic control algorithms, manages intersections, and oversees pedestrian crossings, mirroring real-world conditions. For data analysis, SUMO collects and analyzes simulation data, including vehicle trajectories, travel times, congestion levels, and pedestrian movements.

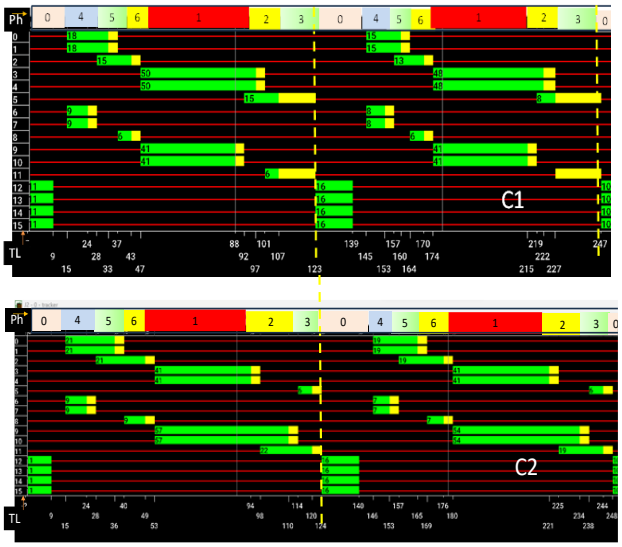


Figure 5. State phasing diagrams for C1 and C2 intersections.

The simulation scenario, adapted to the SUMO simulator, provides insights into traffic light signals and vehicle/pedestrian movements within the terminals. In Figure 5 a state diagram was generated for C2 intersection, incorporating both vehicles in the lanes (2300v/h) and pedestrians (11200 p/h) in the sidewalks during two cycles of 120 seconds. These diagrams offer insights into the dynamic behavior of traffic light signals and carrier/pedestrian movements within the simulated terminals. As can be observed in the diagrams it is possible to distinguish the different cycles that occur during the simulation. It always begins with a pedestrian phase (Phase 0), during which some pedestrians can cross the crosswalk, turning red for pedestrians starting from 11 seconds. Then, phases dedicated to vehicles (Phases 1-8) take place until it concludes at 123 seconds. At this moment, the second cycle begins, with the pedestrian phase becoming active again. The same process repeats until 247 seconds, marking the end of this second cycle and the initiation of a third cycle. These diagrams align with the analysis conducted for pedestrians.

V. INTELLIGENT TRAFFIC CONTROL SYSTEM

With the data collected on vehicles via VLC through the cells in Figure 1, implemented via lamps along the roads as shown in Figure 2, an intelligent traffic system must be developed to optimize traffic flow at intersections. This system utilizes reinforcement learning (RL), a machine learning paradigm where an agent learns to make decisions by interacting with its environment. Agents in RL aim to achieve a goal in uncertain, potentially complex environments by receiving feedback in the form of rewards or punishments. The fundamental idea is for the agent to learn optimal behavior or strategies through trial and error.

At each time step t, the agent receives a state input s_t, based on the observation of the environment and then executes an action a_t, that transforms the state observed to a next state s_{t+1}. Then the reward r_t, a metric that defines how good the action was for the environment, is calculated. In this case, the reward is defined by (1), using the accumulated total waiting time, atw_t, as a metric for vehicles (*veh*) and pedestrians (*ped*). $atwt_t$ and $atwt_{t-1}$ are the accumulated total waiting time of all the cars/pedestrians in the intersection captured respectively at agentstep t and agentstep $t-1$. The weights of the p_{veh} and p_{ped} are set based on the desired priority that the agent should have towards vehicles and pedestrians during network training. The agent will learn a policy that benefits one more than the other, or keeps the system balanced if the weights are equal.

If the agent's behavior leads to positive environmental reward, which indicates that the waiting time is longer in the past, *t-1*, than at the present moment, *t*, then the tendency of producing this behavior by the agent will be strengthened, and vice versa. The goal is to maximize the cumulative discounted reward.

$$r_i = p_{veh}(atwt_{veh,t-1} - atwt_{veh,t}) + p_{ped}(atwt_{ped,t-1} - atwt_{ped,t}) \quad (1)$$

This experience $e_x = (s_t, a_t, r_t, s_{t+1})$ will be stored in the replay memory, to be used in the future to train the agent. The replay memory is a dataset of an agent's experiences $D_t = (e_1, e_2, \dots, e_t)$, which are gathered when the agent interact with the environment as time goes by (t = 1, 2…, n).

To train the agent, the deep Q-Learning technique is employed, leveraging the Q-Learning algorithm [18] [1]. The Q-value represents the expected cumulative reward of taking a particular action in a particular state and following the optimal policy thereafter. These Q-values are predicted by a Neural Network (NN) that takes the state as input and outputs Q-values for each possible action.

The Q-value represents the expected cumulative reward of taking a specific action in each state while following the optimal policy thereafter.

A neural network (NN) predicts these Q-values by taking the state as input and outputting Q-values for each possible action. The state of the environment comprises 100 cells at each intersection, indicating the presence of vehicles or pedestrians. These cells are set to '1' if occupied and '0' if not. Each lane, divided into 10 cells, indicates vehicle movement toward the intersection, with cell sizes increasing farther from the intersection. With 8 lanes per junction, there are 80 vehicle cells per intersection. For pedestrians, only the waiting zones are considered, each divided into 5 cells, totaling 20 pedestrian cells per intersection as draft in Figure 6, where the Flowchart during simulation and training is also displayed.

The neural network's input layer consists of 100 neurons representing the state of the environment. This is followed by

five hidden layers, each with 400 neurons using rectified linear units (ReLUs). The output layer features nine neurons, each representing the Q-values for potential actions. To refine Q-value predictions, a Mean Squared Error (MSE) function quantifies the disparity between predicted and target Q-values, enhancing the learning process. N represents the number of samples stored in memory, and Q_{target} and Q_{pred} denote the target and predicted values, respectively. After each training episode, target Q-values for action-state pairs are calculated based on Equation (2).

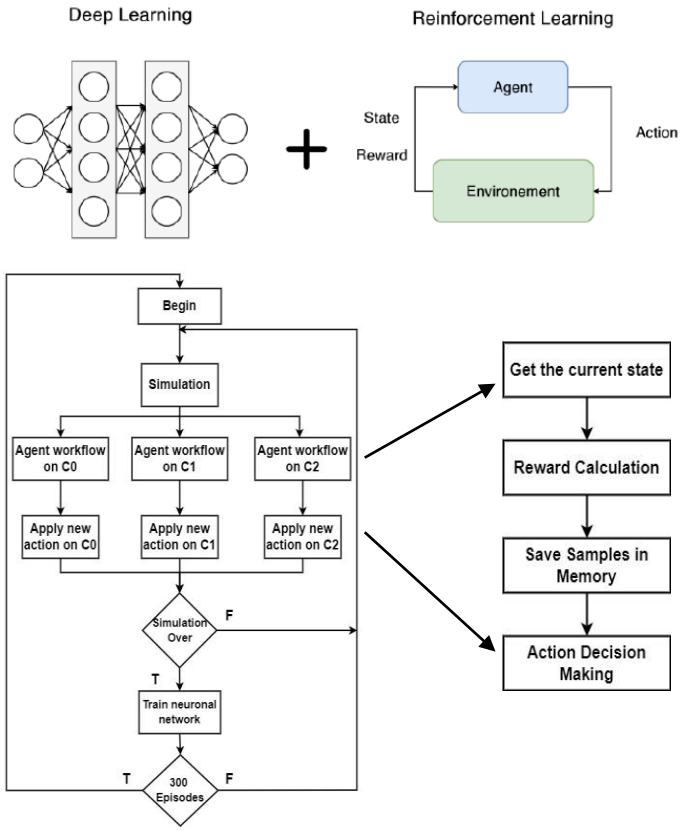


Figure 6. Deep Reinforcement Learning and flowchart during simulation and training .

$$MSE_{Loos} = \frac{1}{N}\sum_{i=1}^{N}\left(Q_{target} - Q_{pred}\right)^2 \quad (2)$$

N is the number of samples stored in memory, and the target and predicted value, Q_{target} and Q_{pred}, respectively. After each episode of training, the target Q-values for action-state pairs are calculated based on (3).

$$Q_{target} = r_t + \gamma . maxQ_{pred}(s_{t+1}, a') \quad (3)$$

The nine Q-values at the neural network's output correspond to the nine actions shown in Figure 7. The agent selects the action that best suits the current traffic situation, without following a predefined order. Conversely, today's dynamic traffic systems at junctions follow a fixed sequence of phases, as shown in Figure 7.

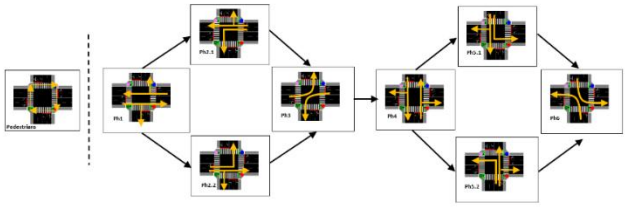


Figure 7. Nine possible actions that can be chosen by the agent.

This can result in activating a phase that does not align with current traffic needs. The next section compares these two systems to highlight their differences and evaluate their effectiveness.

.

VI. SIMULATION RESULTS

A. *Traffic Signal Control Model*

The objective of this section is to compare the *dynamic system* of nine fixed phases and variable timing (Fig. 3c) with the intelligent system where the agent defines the order and timing of phases according to the combined flow of vehicles and pedestrians, *intelligent system*.

In order to implement the system, reinforcement learning (RL) is used, which is a type of machine learning where the agent learns to make decisions through interactions with the environment. RL-based approaches typically consider the traffic flow states surrounding intersections as observable states. The change in signal timing plans is treated as an action, and the traffic control performance as feedback. This section presents the process of constructing an urban traffic control system using the reinforcement learning method [19, 20, 21]. RL enables systems to take actions in a dynamic environment through trial and error methods to maximize rewards based on the feedback generated from taking actions. The main entity of RL is the agent that receives and interprets information from the environment and takes actions. This permits the agent to learn through trial and error.

The primary objective for these agents is to attain a goal within an environment characterized by uncertainty and potential complexity. Feedback, in the form of rewards or punishments, serves as the guiding mechanism for the agent's learning process. The reward function evaluates the difference in accumulated waiting time between the current and previous steps in all lanes, with negative rewards indicating higher waiting times [22].

B. *Training Results*

To evaluate the behaviour of the intelligent traffic control system in relation to pedestrian and vehicle scenarios, a comparison was made with the dynamic traffic control system. The neural network used was trained with a reward system that weighted the waiting times for vehicles (p_{veh}) and pedestrians (p_{ped}) equally, for 300 epochs, each lasting one

hour. Both systems considered the same generation rates for pedestrians and vehicles, totalling 2300 vehicles and 11000 pedestrians in the traffic scenario.

Figure 8 shows the cumulative negative reward from training the network for both agents. Both agents evolved and learned from their traffic experiences throughout the episodes. The curves converged towards less negative reward values, indicating better decision-making over time.

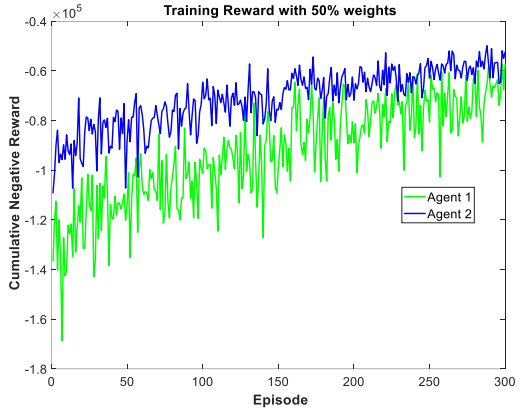


Figure 8. Cumulative Negative reward for both agents in training.

After training the network, tests were conducted to compare both systems under two traffic scenarios representing peak hour conditions. The first scenario involved high vehicle and pedestrian traffic (High-High) with 2300 cars and 11000 pedestrians. The second scenario had high vehicle traffic but low pedestrian traffic (High-Low), with 2300 cars and 5600 pedestrians.

C. *Testing Results – High-High and High-Low scenarios*

Figures 9 and 10 display the number of pedestrians waiting in zones at the two junctions for both traffic scenarios.

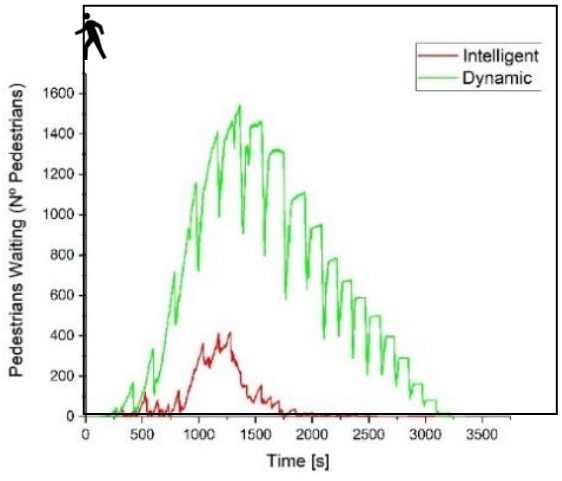


Figure 9. Comparison of the number of pedestrians stopped waiting in both systems for the High-High scenario.

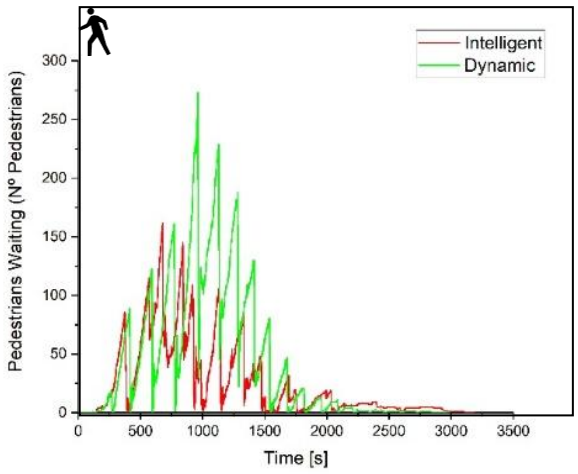


Figure 10. Comparison of the number of pedestrians stopped waiting in both systems for the High-Low scenario.

Figures 11a and 11b illustrate vehicle waiting times under both scenarios.

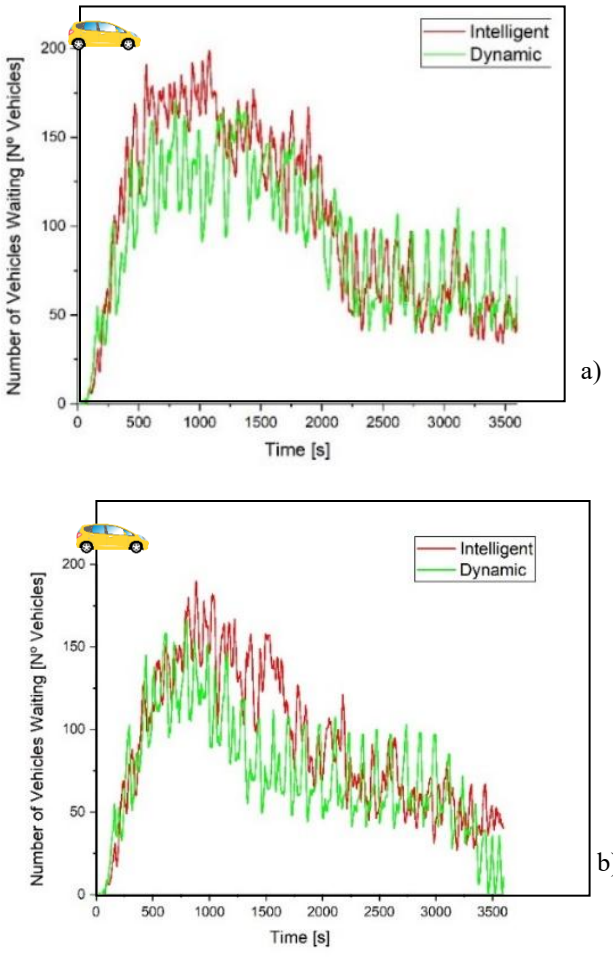


Figure 11. Comparison of the number of cars in the entire environment for both systems for the High-High (a) and High-Low (b) scenarios.

Figure 11a shows that the intelligent system reaches a peak of waiting vehicles between 8 and 15 minutes due to higher pedestrian traffic affecting vehicle flow. In contrast, Figure 11b indicates a peak at around 15 minutes when pedestrian traffic is lower, allowing the system to balance vehicle and pedestrian phases better.

Despite both scenarios having high vehicle traffic, the intelligent system manages fewer waiting vehicles in the low pedestrian scenario. The dynamic system shows consistent behavior with a 120-second cycle time, but the high pedestrian count negatively impacts vehicle dispatch, suggesting that many waiting pedestrians might lead to poor vehicle flow.

D. Testing Results – Phase diagram

The objective is to compare the dynamic system of nine fixed phases and variable timing (Figure 7) with the intelligent system where the agent defines the order and timing of phases according to the combined flow of vehicles and pedestrians.

The intelligent system under consideration was trained using a reward equation that assigned different weights to vehicles and pedestrians, 25% and 75%, respectively, prioritizing pedestrian traffic flow in the environment. Consequently, when pedestrians are requesting to cross pedestrian crossings, the agents prioritize their requests during periods of significant pedestrian traffic ensuring appropriate attention is given. This prioritization naturally impacts vehicle traffic to some extent, but not excessively.

The neural networks for each scenario underwent training with 300 episodes, each lasting 3600 seconds. To characterize the scenarios, several variables related to traffic were employed to assess the system's performance. These variables include queue sizes, where individual intersections in each scenario were analyzed to compare the flow of cars in each. The average queue size for each scenario was also calculated to observe the impact of the number of cars on the environment and the system's response in each case. The average speed of cars was also considered, as vehicle speed provides insights into the fluidity of traffic. Lastly, the number of cars in halting (waiting) was analyzed, providing insights into the impact of the number of vehicles on the environment.

To investigate the behavior of pedestrians in the environment, some variables were considered: average speed of pedestrians and halting. The first allows observing the influence of the cycle durations of each vehicle scenario on pedestrian speed and the second enables the analysis of the number of people who are stationary in waiting zones across all intersections over time giving insight into the number of people per square meter in each of the waiting zones.

In Figure 12 the simulated pedestrian and vehicle average speed with and without RL are displayed using SUMO simulator.

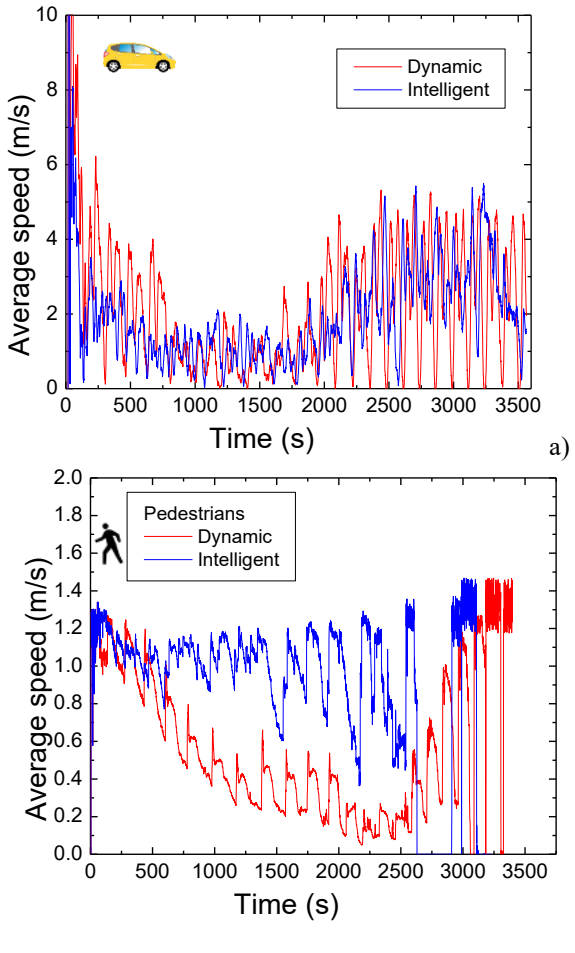


Figure 12. Comparative trends of the vehicles (a) and pedestrians (b) average speed over time for both dynamic and intelligent traffic scenarios.

In Figure13 the simulated halting for both pedestrians and vehicles is displayed.

The size of these peaks in the halting sessions (Figure 11) indicates the stress level and demonstrates pedestrian's reaction to connected vehicles cars. Aperiodic peaks in halting sessions are linked to crossing moments. Comparatively, smaller peaks are observed in the intelligent halting sessions, while higher, more cyclical peaks are observed in dynamic halting sessions.

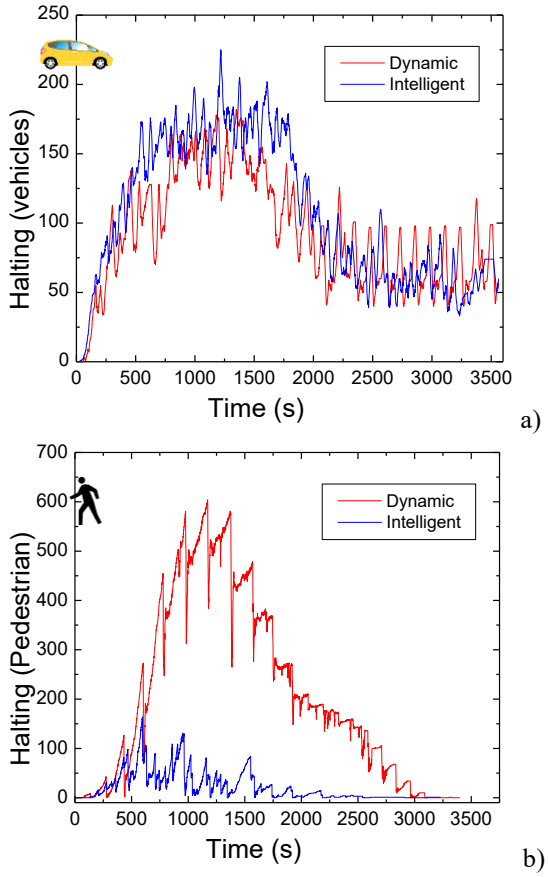


Figure 13. Comparative trends of the vehicles (a) and pedestians (b) halting over time for both dynamic and intelligent traffic scenarios.

The dynamic system strictly adheres to the phase cycle, activating the pedestrian phase with every cycle length. This behavior somewhat benefits cars, as the pedestrian phase is only active every cycle length, naturally impeding pedestrians. In both systems, during the second half hour of requests, when pedestrian and vehicle traffic are light, the system reduces the high values of waiting cars and pedestrians in queues and adjusts its behavior by analyzing the impact of its previous actions on the environment.

This study shows promising results in determining the stress level of pedestrians in the presence of connected vehicles and demonstrates the capabilities of VLC for future research. In Figure 14, the different phases along almost one hour simulation are displayed for both intersections: C1 (a) and C2 (b), respectively. On the top the nine possible phases (agent actions) are draft.

The aperiodic peaks in the halting intelligent pedestrian session (Fig. 11b) are linked to the overlap of crossing moments in C1 and C2, that corresponds to phase number 9 in both.

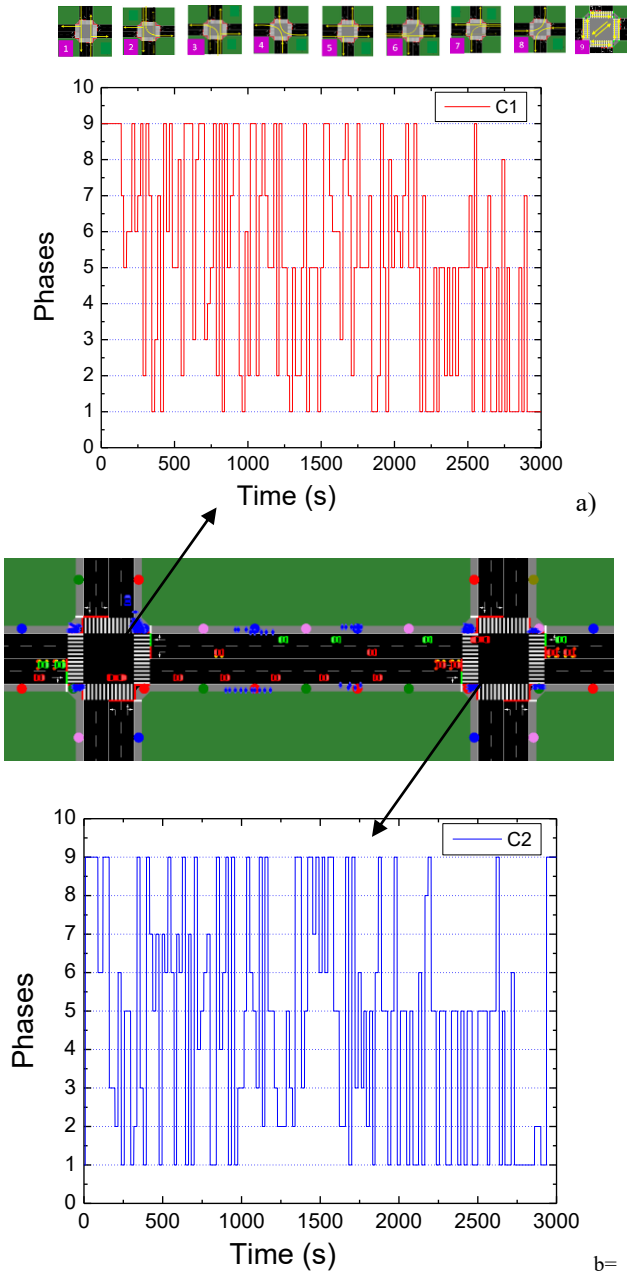


Figure 14. Comparative trends of the selected phases over time for both intersections: C1 (a), C2 (b). ON the top the nine possible phases are pointed out.

So, by integrating VLC technology among pedestrians, vehicles, and surrounding infrastructure has emerged as a pivotal advancement in optimizing traffic signals and vehicle trajectories, allowing for direct monitoring of critical factors such as queue formation, dissipation, relative speed thresholds, inter-vehicle spacing, and pedestrian corner density, ultimately contributing to enhance road safety.

VII. CONCLUSIONS

This paper sets the groundwork for advancing intelligent traffic management by highlighting the potential of VLC technology to enhance safety and efficiency at urban intersections. Our focus was on optimizing both vehicular and pedestrian traffic, addressing the previously overlooked aspect of pedestrian phases. By analyzing agents' behavior and decision-making, particularly concerning pedestrian safety, we aimed to refine the timing of pedestrian phases.

In the domain of traffic optimization, our state representation incorporates environmental information, vehicle and pedestrian distribution data from V-VLC messages, and a proposed phasing diagram guiding agent actions. We developed dynamic and intelligent control system models to securely manage traffic at two connected intersections. Through Reinforcement Learning and the SUMO simulator, we conducted a thorough analysis. With an agent at each intersection, the system optimizes traffic lights based on communication from VLC-ready vehicles, devising strategies to enhance flow and coordinate with other agents for overall traffic optimization.

Overall, the intelligent system demonstrates superior adaptability and efficiency. It manages to reduce pedestrian waiting times while still maintaining a reasonable level of vehicle flow. In comparison, the dynamic system's fixed cycle often leads to longer pedestrian wait times, which can cause significant congestion. Therefore, the intelligent system proves to be more effective in handling the traffic scenarios studied, providing a better balance between vehicle and pedestrian needs.

ACKNOWLEDGMENT

This work was sponsored by FCT – Fundação para a Ciência e a Tecnologia, within the Research Unit CTS – Center of Technology and Systems, reference UIDB/00066 and by IPL project reference IPL/IDI&CA2024/INUTRAM-ISEL

REFERENCES

[1] M. A. Vieira, M. Vieira, P. Louro, G. Galvão, and M.Vestias "Enhancing Urban Intersection Efficiency: Leveraging Visible Light Communication for Traffic Optimization," SENSORDEVICES 2024, The Fifteenth International Conference on Sensor Device Technologies and Applications, Copyright (c) IARIA, 2024. ISBN: 978-1-68558-208-1, pp. 24-29, 2024.

[2] H. Ge, Y. Song, C. Wu, J. Ren and G. Tan, "Cooperative Deep Q-Learning With Q-Value Transfer for Multi-Intersection Signal Control," in IEEE Access, vol. 7, pp. 40797-40809, 2019, doi: 10.1109/ACCESS.2019.2907618.

[3] A. Vidali, L. Crociani, G. Vizzari, and S. Bandini, "A Deep Reinforcement Learning Approach to Adaptive Traffic Lights Management." Workshop "From Objects to Agents" (WOA 2019), pp. 42-50, 2019.

[4] A. Oroojlooy, D, Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," Appl Intell 53, pp.13677–13722 (2023). https://doi.org/10.1007/s10489-022-04105-y

[5] D. O'Brien, et al. "Indoor Visible Light Communications: challenges and prospects," Proc. SPIE 7091, 709106, pp. 60-68, 2008.

[6] H. Parth., X. Pathak, H. Pengfei, and M. Prasant, "Visible Light Communication, Networking and Sensing: Potential and Challenges," September 2015, IEEE Communications Surveys & Tutorials 17(4): Fourthquarter 2015, pp. 2047 – 2077, 2015.

[7] X. Liang, X. Du, G. Wang, and Z. Han, "A Deep Reinforcement Learning Network for Traffic Light Cycle Control," in IEEE Transactions on Vehicular Technology, vol. 68, no. 2, pp. 1243-1253, Feb. 2019, doi: 10.1109/TVT.2018.2890726.

[8] I. Sousa, P. Queluz, A. Rodrigues, and P. Vieira, "Realistic mobility modeling of pedestrian traffic in wireless networks". In 2011 IEEE EUROCON-International Conference on Computer as a Tool, pp. 1-4,. IEEE , 2011.

[9] T. Ding, S. Wang, J. Xi, L. Zheng, and Q. Wang, "Psychology-Based Research on Unsafe Behavior by Pedestrians When Crossing the Street". Adv. Mech. Eng., 7, 203867, pp. 1-6, 2015.

[10] J. Oskarbski, L. Guminska, M. Miszewski, and I. Oskarbska,. "Analysis of Signalized Intersections" in the Context of Pedestrian Traffic. Transportation Research Procedia, 14, pp. 2138-2147, 2016, doi:10.1016/j.trpro.2016.05.229.

[11] Y. Elbaum, A. Novoselsky, and E. A. Kagan, "Queueing Model for Traffic Flow Control in the Road Intersection," Mathematics, 10, 3997, 2022. doi: 10.3390/math10213997.

[12] S. Yousefi, E. Altman, R. El-Azouzi, and M. Fathy, "Analytical Model for Connectivity in Vehicular Ad Hoc Networks," IEEE Transactions on Vehicular Technology, 57, pp. 3341-3356, 2008.

[13] L. Alvarez et al., "Microscopic Traffic Simulation using SUMO," In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 2575-2582. IEEE. The 21st IEEE International Conference on Intelligent Transportation Systems, Maui, USA, pp. 4.-7, Nov. 2018.

[14] A. Yousefpour et al., "All one needs to know about fog computing and related edge computing paradigms: A complete survey", Journal of Systems Architecture, Volume 98, pp. 289-330, 2019.

[15] M. A. Vieira, M. Vieira, P. Louro, and P. Vieira, "Cooperative vehicular visible light communication in smarter split intersections," Proc. SPIE 12139, Optical Sensing and Detection VII, 1213905, 17 May 2022; doi: 10.1117/12.2621069.

[16] M. A. Vieira, G. Galvão, M. Vieira, M. Véstias, P. Vieira, and P. Louro, "Traffic Signaling and Cooperative Trajectories based on Visible Light Communication," Sensors and Electronic Instrumentation Advances, pp. 23, 2023.

[17] M. A. Vieira, G. Galvão, M. Vieira, P. Louro, M. Vestias, and P. Vieira, "Enhancing Urban Intersection Efficiency: Visible Light Communication and Learning-Based Control for Traffic Signal Optimization and Vehicle Management," Symmetry 2024, 16, 240. https://doi.org/10.3390/ sym16020240.

[18] G. Galvão et al., "Traffic signals and cooperative trajectories at urban intersections: leveraging visible light communication

for implementation," Proc. SPIE 12906, Light-Emitting Devices, Materials, and Applications XXVIII, 129060O (13 March 2024); doi : 10.1117/12.3000529.

[19] Y. Elbaum, A. Novoselsky, E. Kagan, " A Queueing Model for Traffic Flow Control in the Road Intersection." Mathematics 2022, 10, 3997. doi: 10.3390/math10213997.

[20] G. P. Antonio, C. Maria-Dolores, C. AIM5LA: "A Latency-Aware Deep Reinforcement Learning-Based Autonomous Intersection Management System for 5G Communication Networks." Sensors 2022, 22, 2217. doi: 10.3390/s22062217.

[21] Y. Shi, Y. Liu, Y. Qi, Q. Han, "A Control Method with Reinforcement Learning for Urban Un-Signalized Intersection in Hybrid Traffic Environment. Sensors" 2022, 22, 779. doi: 10.3390/s22030779.

[22] M. Vieira, M. A. Vieira, G. Galvão, P. Louro, M. Véstias, P. Vieira,. "Enhancing Urban Intersection Efficiency: Utilizing Visible Light Communication and Learning-Driven Control for Improved Traffic Signal Performance," Vehicles 2024, 6, 666-692. doi.org/10.3390/vehicles6020031.

Combinatory Logic as a Model for Intelligent Systems Based on Explainable AI

Thomas Fehlmann
Euro Project Office AG
8032 Zurich, Switzerland
E-mail: thomas.fehlmann@e-p-o.com

Eberhard Kranich
Euro Project Office
47051 Duisburg, Germany
E-mail: eberhard.kranich@t-online.de

***Abstract*—Directed graphs such as neural networks can be described by *Arrow Terms* that link a finite set of incoming nodes to some response node. Scott and Engeler have shown that its powerset is a model for *Combinatory Logic*. This algebra is called *Graph Model of Combinatory Logic*. Since Combinatory Logic is Turing-complete, the model explains both traditional programming logic as well as neural networks such as the brain or *Artificial Neural Networks* as used in a *Large Language Model*. The underlying graph model is a general model for all kinds of knowledge. The graph model would yield a powerful AI-tool if used as a blueprint for implementing AI. *Chain of Thoughts* would come for free, and explainability with it. However, its performance would make such a tool impractical and useless. The paper proposes a combined approach for adding explainability to AI and creating *Intelligent Systems*. It is the strategy humans use when they try to explain their ideas. First, the generative power of neural networks is used to produce an idea or solution. Next, humans create a chain of thoughts that explain such ideas to others and try to provide evidence. AI could follow the same strategy. The architecture of such intelligent systems consists of two distinct elements: a well-trained artificial neural network for observing and generating solution approaches, and a controlling engine for fact checking and reliability assessment.**

***Keywords*—Intelligent Systems; Chain-of-Thought (CoT); Explainable AI (XAI); Artificial Neural Networks (ANN); Deep Neural Network (DNN); Combinatory Logic; Quality Function Deployment (QFD).**

I. INTRODUCTION

This paper is a revised version of the author's contribution to the 1st International Conference on Systems Explainability, held in Valencia, Spain, in autumn 2024 [1].

A. Short History of AI and its Philosophical Background

In the early 20th century, there were some shocking events taking place in mathematical logic and natural science. Gödel [2], when trying to solve some of Hilbert's 23 problems, detected that predicate logic, something with a long history dating back to the ancient Greeks, is undecidable. This insight gave birth to theoretical computer science, including the theory of computation, founded by Turing [3]. For a modern compilation, see Raatikainen [4].

Schönfinkel and Curry [5] developed *Combinatory Logic* to avoid the problems introduced when using logical quantifiers, and Church invented *Lambda Calculus* as a rival formalism [6]. Scott and Engeler developed the *Graph Model* [7], based on *Arrow Terms*, and proved that this is a model of combinatory logic. This means that you can combine sets of arrow terms to get new arrow terms, and that combinators, accelerators, and constructors can be used to create new elements of algebra.

Graphs in the form of neural networks appeared already at the origins of *Artificial Intelligence* (AI). Its first instantiation in modern times was the *Perceptron*, a network of neurons postulated by Rosenblatt [8]. It later became a directed graph [9]. Rosenblatt was also the first who postulated concepts, among perception and recognition, as constituent parts of AI [8, p. 1].

Since its origins, AI has experienced difficulties; however, today there are many AI applications that provide value for the user. In some areas, training an AI model is simpler and more rewarding than finding and programming an algorithm.

For instance, AI-powered visual recognition systems excel in recognizing and classifying objects, following the ideas established by Rosenblatt [8]. However, they have difficulty recognizing temporal dependencies and are unable to combine what they have learned, although attempts have been made to develop methods using sequential data and the ability to capture temporal patterns. AI lacks what humans use in such cases: a concept.

Logical skills such as inference and deduction provide quite a challenge, as exemplified by the ARC Price challenge, a sort of intelligence test for AI models, proposed by Chollet [10]. A *Large Language Model* (LLM) easily summarizes texts or books but it still does not understand what is written in it, in the sense that the US National Council of English Teachers calls *Literacy*, see [11], [12].

Artificial Neural Networks (ANN) can be divided into four types: Recurrent Neural Network (RNN), Fuzzy Neural Network (FNN), Convoluted Neural Network (CNN) and Deep Neural Network (DNN). DNN are the most successfully used for LLM and thus the most important type of ANN regarding explainability because they rely on many hidden layers. Among the rapidly developing literature, Gerven & Bothe's classification are a good start [13]. *Natural Neural Networks*, in analogy to ANNs, are abbreviated by NNN.

IBM defines *Explainable Artificial Intelligence* (XAI) as a set of processes and methods that allow human users to comprehend and trust the results created by machine learning

algorithms [14]. Current approaches to XAI attempt to use statistical correlations as a basis for reasoning. From a theoretical perspective, this is unlikely to work, because of Gödel [2]. However, engineers try to use statistical methods to circumvent Gödel's undecidability. Sometimes the results look convincing. Dallanoce [15] compiled a list of available processes and methods for XAI.

Artificial General Intelligence (AGI) is a type of artificial intelligence (AI) that falls within the lower and upper limits of human cognitive capabilities across a wide range of cognitive tasks. The creation of AGI is a primary goal of AI research and companies such as OpenAI and Meta, but what exactly AGI refers to is controversial [16].

Current attempts towards AGI focus on the investigation of *Chain of Thoughts* (CoT). Using an LLM based on DeepSeek with CoT enabled yields the feeling that the LLM does some "reasoning" because it displays the intermediate results obtained with the processing of some query [17]. The real innovation behind DeepSeek is using a hash function to avoid processing useless branches in an LLM. This is not what reasoning really is, namely the use of logic based on factual knowledge to find previously unknown answers. The hash function is still based on statistics from a suitable training set [18].

B. Research Questions

The aim of this paper is to recall prior work in logic and AI to understand how neural networks work. To do this, we investigate the following three research questions:

RQ 1: How are neural networks and especially DNNs linked to the graph model?
RQ 2: Does CoT relate to a sequence of arrow terms?
RQ 3: Can the graph model explain AI?

The motivation for this is that we are experiencing the fourth AI hype in sixty years and that its acceptance in society is currently transitioning from admiration to rejection. Because the nature of AI is poorly understood not only by society but also by the AI research community. We believe that the graph model is an excellent way to understand what intelligence is, both natural and artificial. However, it is not an answer to how to construct XAI.

C. Paper Structure

We first explain combinatory logic (Section II) and the motivation for building a model (Section III). Then we compare DNNs with graphs and explain how arrow schemes represent what a DNN does and have an outlook on the architecture of intelligent systems (Section IV). Finally, we present the method for designing intelligent systems (Section V) and explain how we want to go ahead (Section VI).

II. COMBINATORY LOGIC

In the past decades, there has been a lack of attention and consequently of publications on *Combinatory Logic.* Nevertheless, it explains quite a bit what artificial intelligence can do and what not.

A. Combinatory Logic and Axiom of Choice

Combinatorial Logic is a notation that eliminates the need for quantified variables in mathematical logic, and thus the need to explain what the meaning of existential quantifiers $\exists x \in M$ is, see Curry [5] and [19]. Eliminating quantifiers is an elegant way to avoid the *Axiom of Choice* [20] in its traditional form. Combinatory Logic can be used as a theoretical model for computation and as design for functional languages (Engeler [21]); however, the original motivation for combinatory logic was to better understand the role of quantifiers in mathematical logic.

Combinatory logic is based on *Combinators* which were introduced by Schönfinkel in 1920. A combinator is a higher-order function that uses only functional applications, and earlier defined combinators, to define a result from its arguments.

The combination operation is denoted as $M \bullet N$ for all combinatory terms M, N. To make sure there are at least two combinatory terms, we postulate the existence of two special combinators **S** and **K**.

They are characterized by the following two properties (1) and (2):

$$\mathbf{K} \bullet P \bullet Q = P \tag{1}$$

$$\mathbf{S} \bullet P \bullet Q \bullet R = P \bullet Q \bullet (P \bullet R) \tag{2}$$

P, Q, R are terms in combinatory logic. The combinator **K** acts as projection, and **S** is a substitution operator for combinatory terms. Equations (1) and (2) act like axioms in traditional mathematical logic.

Like an assembly language for computers, or a Turing machine, the **S**-**K** terms become quite lengthy and are barely readable by humans, but they work fine as a foundation for computer science. The power of these two operators is best understood when we use them to define other, handier, and more understandable combinators.

The identity combinator for instance is defined as

$$\mathbf{I} := \mathbf{S} \bullet \mathbf{K} \bullet \mathbf{K} \tag{3}$$

Indeed, $\mathbf{I} \bullet M = \mathbf{S} \bullet \mathbf{K} \bullet \mathbf{K} \bullet M = \mathbf{K} \bullet M \bullet (\mathbf{K} \bullet M) = M$. Association is to the left. Moreover, **S** and **K** are sufficient to build a Turing machine. Thus, combinatory logic is Turing-complete. For proof, consult Barendregt [22, pp. 17-22].

B. Functionality by the Lambda Combinator

Curry's *Lambda Calculus* [23] is a formal language that can be understood as a prototype programming language. The **S**-**K** terms implement the lambda calculus by recursively defining the *Lambda Combinator* $\mathbf{L_x}$ for a variable x as follows:

$$\begin{gathered} \mathbf{L_x} \bullet x = \mathbf{I} \\ \mathbf{L_x} \bullet Y = \mathbf{K} \bullet Y \text{ if } Y \text{ different from } x \\ \mathbf{L_x} \bullet M \bullet N = \mathbf{S} \bullet \mathbf{L_x} \bullet M \bullet \mathbf{L_x} \bullet N \end{gathered} \tag{4}$$

The definition holds for any term **x** of combinatory logic. Usually, on writes suggestively $\lambda x.M$ instead of $\mathbf{L_x} \bullet M$, for any combinatory term M. *Lambda Terms* $\lambda x.M$ offer the

possibility of programmatic parametrization. Note that $\lambda x.M$ is a combinatory term, as proofed by (4), and that this introduces a kind of variable in combinatory logic with precisely defined binding behavior.

The Lambda combinator allows writing programs in combinatory logic using a higher-level language. When a lambda term is compiled, the resulting combinatorial term looks like machine code in traditional programming languages.

C. The Fixpoint Combinator

Given any combinatory term Z, the *Fixpoint Combinator* $\mathbf{Y}$ generates a combinatory term $\mathbf{Y} \bullet Z$, called *Fixpoint of* Z, that fulfills $\mathbf{Y} \bullet Z = Z \bullet (\mathbf{Y} \bullet Z)$. This means that Z can be applied to its fixpoint as many times as wanted and still yields back the same combinatory term.

In linear algebra, such fixpoint combinators yield an eigenvector solution $\mathbf{Y} \bullet Z$ to some problem Z.

According to Barendregt in his textbook about Lambda calculus [22, p. 12], the fixpoint combinator can be written as

$$\mathbf{Y} := \lambda f.\left(\lambda x.f \bullet (x \bullet x)\right) \bullet \left(\lambda x.f \bullet (x \bullet x)\right) \qquad (5)$$

Translating (5) into an $\mathbf{S}$– $\mathbf{K}$ term demonstrates how combinatory logic works, see [24].

When translated into arrow terms, the fixpoint combinator contains loops. Fixpoint operations are related to infinite loops, thus, to programming constructions that never end and have no normal form. Applying $\mathbf{Y}$, or any equivalent fixpoint combinator to a combinatory term Z, usually does not terminate. An infinite loop can occur, and must sometimes occur, otherwise Turing would be wrong and all finite state machines would reach a finishing state [3].

D. A few More Sample Combinators

The following samples are taken from Zachos 1978 [25], where all proofs are given:

- Composition:
  $\mathbf{B} \bullet P \bullet Q \bullet R = P \bullet Q \bullet R$ by
  $\mathbf{B} := \mathbf{S} \bullet (\mathbf{K} \bullet \mathbf{S}) \bullet \mathbf{K}$
- Exchange of arguments:
  $\mathbf{C} \bullet P \bullet Q \bullet R = P \bullet R \bullet Q$ by
  $\mathbf{C} := \mathbf{S} \bullet (\mathbf{B} \bullet \mathbf{B} \bullet \mathbf{S}) \bullet (\mathbf{K} \bullet \mathbf{K})$
- Argument identification:
  $\mathbf{W} \bullet P \bullet Q = P \bullet Q \bullet Q$ by
  $\mathbf{W} := \mathbf{S} \bullet \mathbf{S} \bullet (\mathbf{K} \bullet \mathbf{I})$
- Composition:
  $\mathbf{\Phi} \bullet O \bullet P \bullet Q \bullet R = O \bullet (P \bullet R) \bullet (Q \bullet R)$ by
  $\mathbf{\Phi} := \mathbf{B} \bullet (\mathbf{B} \bullet \mathbf{S}) \bullet \mathbf{B})$
- Composition:
  $\mathbf{\Psi} \bullet O \bullet P \bullet Q \bullet R = O \bullet (P \bullet Q) \bullet (P \bullet R)$ by
  $\mathbf{\Psi} := \mathbf{B} \bullet (\mathbf{B} \bullet \mathbf{W} \bullet (\mathbf{B} \bullet \mathbf{C})) \bullet \mathbf{B} \bullet \mathbf{B} \bullet (\mathbf{B} \bullet \mathbf{B})$
- Fixpoint Combinator:
  $\mathbf{Y} \bullet R = R \bullet (\mathbf{Y} \bullet R)$ by
  $\mathbf{Y} := \mathbf{W} \bullet \mathbf{S} \bullet (\mathbf{B} \bullet \mathbf{W} \bullet \mathbf{B})$

There is no negation combinator, because with a negation $\boldsymbol{N}$ we would have $\mathbf{Y} \bullet \boldsymbol{N} = \boldsymbol{N} \bullet (\boldsymbol{Y} \bullet \boldsymbol{N})$. This contra-intuitive example explains why so few people dare to work with combinatory logic. However, it also strengthens our point that it is highly rewarding to try it.

It is a specific human behavior to identify complicated behavior with simple explanations, such as "Exchange of arguments." If you expand that combinator, it would be near to unreadable; same with the fixpoint operator $\mathbf{Y}$, as shown in [24].

III. The Graph Model of Combinatory Logic

The graph model is a versatile model for knowledge in all its different forms. It is highly recursive and Turing-complete, which means it can also be used to describe conventional algorithmic programming. The LISP language was once created to allow programming in a framework close to the graph model [26].

A. A Logic Needs a Model

A *Model* for a logical structure is a set-theoretic construction that has the properties postulated for the logic and can be proved to be non-empty. Then it means that such logic makes sense as far as it describes an existing structure and can be used to prove something about the model.

Let $\mathcal{L}$ be a non-empty set. Engeler [7] defined a *Graph* as the set of ordered pairs:

$$\langle \{a_1, a_2, \dots, a_m\}, b \rangle \qquad (6)$$

with $a_1, a_2, \dots, a_m, b \in \mathcal{L}$. We write $\{a_1, \dots, a_m\} \to b$ for the ordered pair to make notation mnemonic, i.e., referring to directed graphs, and call them *Arrow Terms*. These terms describe the constituent elements of directed graphs with multiple origins and a single node. We refer to $\mathcal{L}$ as *Observations,* and to terms $\{a_1, \dots, a_m\} \to b$ as *Concepts,* i.e., a non-empty finite set of arrow terms with level 1 or higher.

We extend the definition of arrow terms to a powerset by including all formal set-theoretic objects recursively defined as follows:

$$\begin{gathered}\text{Every element of } \mathcal{L} \text{ is an arrow term.}\\ \text{Let } a_1, \dots, a_m, b \text{ be arrow terms.}\\ \text{Then } \{a_1, \dots, a_m\} \to b \text{ is also an arrow term.}\end{gathered} \qquad (7)$$

The left-hand side of an arrow term is a finite set of arrow terms, and the right-hand side is a single arrow term. This definition is recursive. Elements of $\mathcal{L}$ are also arrow terms. The arrow, where present, should suggest the ordering in a graph, not logical imply.

B. Einstein-Notation for Arrow Terms

To avoid the many set-theoretical parenthesis, the following notation, called *Arrow Schemes*, is applied, in analogy to the Einstein notation [27, p. 6]:

- a_i for a finite set of arrow terms, i denoting some Choice Function selecting finitely many specific terms out of a set of arrow terms a.
- a_1 for a singleton set of arrow terms; i.e., $a_1 = \{a\}$ where a is an arrow term. (8)
- $\emptyset$ for the empty set, such as in the arrow term $\emptyset \to a$.
- $a_i + b_j$ for the union of two observation sets a_i ,b_j.

The application rule for M and N now reads:

$$M \bullet N = (a_i \to b) \bullet N = \{b | \exists a_i \to b \in M;\ a_i \subset N\} \qquad (9)$$

Arrow schemes always represent sets of arrow terms. $(a_i \to b) \subset M$ is the subset of level 1 arrow terms in M, provided $a_i \in M$ and $b \in M$. Thus, $(a_i \to b)_j$ denotes a concept, together with two choice functions i, j. Each set element has at least one arrow.

The choice function i chooses specific observations a_i out of a (larger) set of observations a. This is what Zhong describes as *Grounding* when linking observations to real-world objects [28]. In AI, grounding is crucial for linking AI engines to the real world. If a denotes knowledge, i.e., an infinite set of arrow terms of any level, a_i can become part of a concept consisting of specific arrow terms referring to some specific object, specified by the choice function i. Choice functions therefore have the power of focusing knowledge on specific objects in specific areas. That makes choice functions interesting for intelligent systems and AI.

There is a conjunction of choice functions, thus $a_{i,j}$ denotes the union of a finite number of grounded arrow schemes:

$$a_{i,j} = a_{i,1} \cup a_{i,2} \cup \ldots \cup a_{i,m} = \bigcup_{k=1}^{m} a_{i,k} \qquad (10)$$

There is also cascading of choice functions. Assume $N = (a_j \to b)_k$, then:

$$M = \left(\left((a_j \to b)_k \to b_i\right)_l \to c\right) \text{ and } M \bullet N = (b_{i_l} \to c) \qquad (11)$$

The choice function might be used for grounding an arrow scheme to observations.

An arrow scheme without outer indices represents a potentially infinite set of arrow terms. Thus, writing a, we mean knowledge about an observed object. Adding an index, a_j, indicates such a grounded object together with a choice function j that chooses finitely many specific observations or knowledge.

While on the first glimpse, the Einstein notation seems like just another way of denoting arrow terms, for representing such data in computers it means that the simple enumeration of finite data sets is replaced by an intelligent choice function providing grounding that must be computed and can be either programmed or guessed by an intelligent system.

For practical applications, the choice function is an important part of deep learning. It means learning by generalization. The more choices you get on the left-hand side, the more knowledge you acquire. The ARC price competition for instance is easily solvable if we can generalize our choice functions good enough, drawing conclusions from the samples into general rules. However, generalization is not easily available with current AI technology. *Controlling Combinators*, see Section IV.C, are a workaround.

C. The Graph Model of Combinatory Logic

The algebra of observations represented as arrow terms is a combinatory algebra and thus a model of combinatory logic. The following definitions demonstrate how the graph model implements Curry's combinators **S** and **K** fulfilling equations (1) and (2), following [5].

- $\mathbf{I} = a_1 \to a$ is the Identification, i.e., $(a_1 \to a) \bullet b = b$
- $\mathbf{K} = a_1 \to \emptyset \to a$ selects the 1st argument:
  $\mathbf{K} \bullet b \bullet \mathrm{c} = (b_1 \to \emptyset \to b) \bullet b \bullet \mathrm{c} = (\emptyset \to b) \bullet \mathrm{c} = \mathrm{b}$
- $\mathbf{KI} = \emptyset \to a_1 \to a$ selects the 2nd argument:
  $\mathbf{KI} \bullet b \bullet \mathrm{c} = (\emptyset \to c_1 \to c) \bullet b \bullet \mathrm{c} = (c_1 \to c) \bullet c = \mathrm{c}$
- $\mathbf{S} = \left(a_i \to \left(b_j \to c\right)\right)_1 \to (d_k \to b)_i \to \left(a_i + b_{j,i} \to c\right)$

(12)

Therefore, the algebra of observations is a model of combinatory logic. The interested reader can find complete proofs in Engeler [7, p. 389].

The *Lambda Theorem* from Barendregt [23] says that with **S** and **K,** an abstraction operator can be constructed that adds algorithmic skills to knowledge represented as arrow schemes, following equation (4).

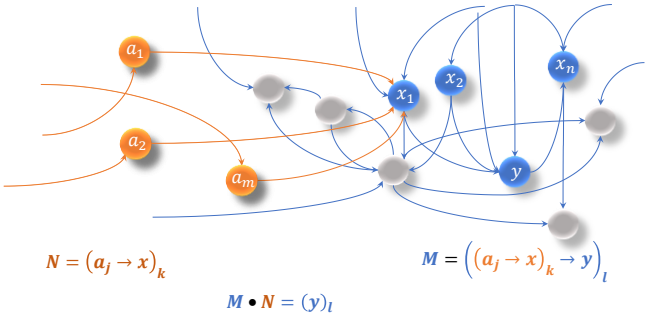


Figure 1. A Neural Network becomes a Combinatorial Algebra

As the name "graph model" suggests, arrow terms are an algebraic way of describing neural networks. Thus, something that nature uses to acquire and work with knowledge.

Figure 1 illustrates the effect of the combination according to equation (9). It becomes apparent that the graph model describes graphs indeed, with loops. Repeatedly applying equation (9) leads to what we perceive as the "response of a neural network". The combination of knowledge and combinators thus plays a significant role in AI.

However, Figure 1 is not only a picture of an abstract graph. It can also be understood as a part of a *Deep Neural Network* (DNN) – or of a *Natural Neural Network* (NNN). Engeler [29] associated neuroscience with the graph model in 2019, by explaining how a brain works. He used the graph model as an algebraic representation of NNN.

IV. TOWARDS INTELLIGENT SYSTEMS

Barceló et al. has shown in 2019 that modern neural network architectures are Turing-complete [30]. This is also a property of the graph model but not of every DNN. An architecture for intelligent systems should be suitable for using conventional algorithmic programming instead of complex arrow notations.

A. Solving the World Formula

Artificial neural networks learn continuously by using corrective feedback loops to improve their predictive analytics. Neural networks perform supervised learning tasks, building knowledge from training data where the right answer is provided in advance. In contrast, in unsupervised learning, algorithms learn patterns exclusively from unlabeled data [31]. There exist mixed forms; a famous example of semi-supervised learning has led to the creation of ChatGPT [32].

In both cases, the principle is the same as with *Six Sigma Transfer Functions* (SSTF) [33]: One has to solve an equation (13), where the expected response $\boldsymbol{y}$ is known but neither the required controls $\boldsymbol{x}$ nor the transfer function $\boldsymbol{A}$ itself, which cause this response, are known. Transfer functions are abundant in technology and science – just to mention the *Fast Fourier Transform* (FFT) of audio and video signals from analog to digital [34] – and AI-enabled applications belong also to that category. In either case, the problem to be solved is:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \tag{13}$$

The equation (13) is often called the "World Formula" [35]. In the case of AI, the world formula describes *Deep Learning* (DL), i.e., the process of parametrizing the model so that it provides the expected answers.

In AI, the transfer function $\boldsymbol{A}$ is usually represented as a large sparse matrix. For small dimensions, the easiest way to solve equation (13) is the Eigenvector method used by Saaty for the *Analytic Hierarchy Process* (AHP), for decision making method [36]. The method also works for *Quality Function Deployment* (QFD) [33, p. 34].

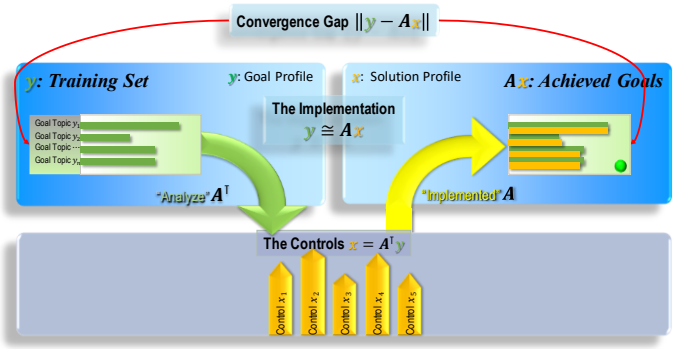


Figure 2. Solving the World Formula

The idea of the Eigenvector solution method is to calculate the *Principal Eigenvector* $\boldsymbol{y_E}$ with the property:

$$\boldsymbol{y_E} = \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y_E} \tag{14}$$

The principal eigenvector exists due to the *Perron-Frobenius* theorem [33, p. 365]. Setting $\boldsymbol{x_E} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y_E}$ yields an approximate solution to equation (13), using equation (14), provided that $\boldsymbol{y} \cong \boldsymbol{A}\boldsymbol{x_E}$ is close enough. The Euclidean distance (15) is called the *Convergence Gap*:

$$\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x_E}\| \tag{15}$$

The eigenvector method is not applicable for large AI matrices, because the solution is numerical and not algebraic. New methods suitable for large sparse matrices representing neural networks had to be invented, see for example Hinton [31].

The breakthrough for solving such matrices, and thus enabling machines for deep learning, happened in 2012 and involves breaking up those matrices into smaller pieces that can be managed in parallel. Its impact on humanity and society might become comparable with the FFT transform of 1977 that made the analog/digital conversion of audio and video in real-time possible and thus stood at the origins of the Internet of today.

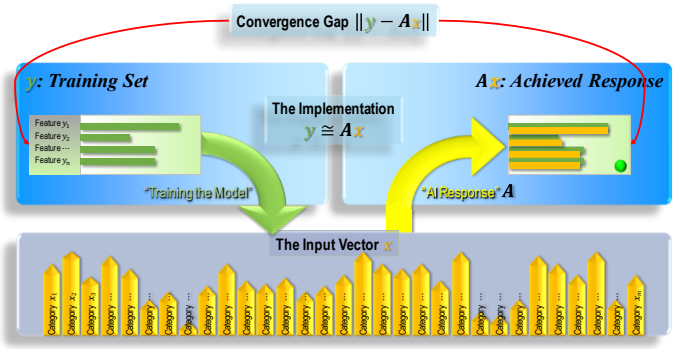


Figure 3. Deep Learning as a Transfer Function

Once the neural net has been sufficiently trained, the model can be used to predict responses not just for the training set but for any query submitted to the AI. The convergence gap in Figure 3, the vector distance between the true response and the response received, is what we consider the inaccuracy, or uncertainty, of the trained model.

B. How Arrow Schemes describe DNNs

While it is obvious how an NNN is represented by arrow schemes, this is not equally clear for ANNs. The reason is that directed graphs contain loops while looping in ANNs is very restricted. There exist certain architectures for ANNs that allow for loops, within narrow limits; however, a *Multi-Layered Perceptron* (MLP) as used for LLMs does not [13].

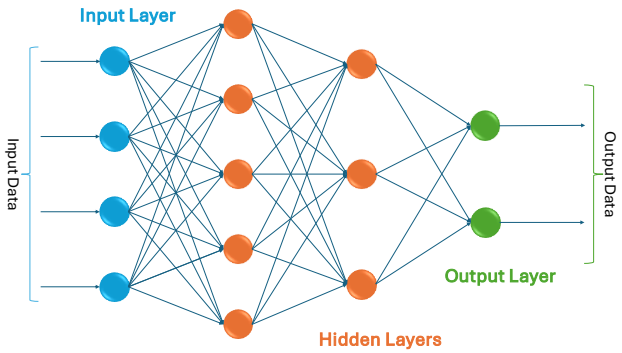


Figure 4. Multi-Layered Perceptron as an DNN

Consequently, a DNN has only a limited ability to emulate an NNN. In principle, every arrow scheme $a_i \to b$ describes one node in a directed but not loop-free graph. Some arrow schemes describe algorithmic concepts such as in equation (12) or as explained in equation (5). Other arrow schemes simply connect observations a_i to some response b. General knowledge has many facets.

It would be wonderful if we had the ability to look at an LLM and identify arrow schemes for each node. This would add full explainability to AI, but unfortunately, this has no practical value. Neither combinatory terms nor arrow schemes have normal forms. Very often there is a wide variety of solutions that are equivalent but widely different; not only formally but also in effectiveness.

This makes explainability of AI difficult. The lack of normal forms blocks all attempts to find the one sequence of arrow schemes that explains what AI is doing. AI engineers have no other choice than trying to train their DNNs such that the response meets expectations but without exactly knowing what happens. It is comforting, however, that they share the same sad fate with neuroscientists. It is astonishing how long-forgotten theoretical results such as the lack of a normal form in combinatory logic yields economically relevant results, nowadays, in the evolving AI ecosystem. Consult Lachowski [37] for a survey of the performance challenges that occur around combinatory logic.

However, there is a famous saying that nothing is too difficult for the engineer ("Inventor of Anything"). Recent findings suggest that AI is capable of recognizing chains of thought that lead to the observation of a specific response [38]. This complements earlier findings that describe CoT as a prompting technique [17]. Thus, there exist AI architectures that allow us to identify at least some arrow schemes that describe what AI does. It is not necessarily the whole truth, just as it is not when people explain their thoughts to colleagues. But it should be enough to convince them.

Having a complete sequence of arrow schemes describing approximatively some DNN would lead to explainable AI that even is able to get certified for safety-critical applications. However, the problem with hidden layers remains. While the QFD method uses identifiable topics for each layer [39], an DNN has none; they remain hidden and unknown. Thus, much of the intermediate reasoning also remains hidden. RQ 2 remains at least partially unanswered. If the input data and response can only be captured by arrow schemes, the intermediate steps must be guessed based on domain knowledge, but it is not known exactly what the AI engine did consider. AI might change behavior and create hazardous changes to the hidden layers. Low-rank adaptation (LoRA) of LLMs is an attempt to limit such change [40]. In QFD, on the contrary, intermediate stages are identifiable based on their topic; for example, when deploying customer needs, we first go to user stories and then to testable features.

Another approach to better explainable AI is already well established: *Retrieval-Augmented Generation* (RAG) might avoid hallucinations for LLMs [41] by referencing knowledge databases and including them into the generation of responses. RAG impacts the architecture of intelligent systems by connecting neural networks to knowledge databases [42]. RAG corresponds to grounding arrow schemes using the choice function; RAG is indispensable for explainable AI.

This is the motivation for looking at AI architecture. In some way, it must be complemented by functionality that controls the behavior of AI. With such controls an AI-engine can perform safety-critical tasks. When certifying an AI-engine for safety, it is not necessary to convert all nodes of an DNN into arrow schemes, but we can focus on the overall result, because these results are not presented plainly but reviewed by a controlling combinator first. If an AI fails on such tasks, we do not have a white-box trace of all nodes including their arrow schemes that have contributed to this failure, but we are at least as good as with traditional safety-preserving methods and techniques.

C. The Architecture of Intelligent Systems

Intelligent systems using AI are based upon *Controlling Combinators*. Controlling combinators are derived from the idea behind fixpoint combinators, see equation (5) but refer to effective factual knowledge or to skills. Examples from Engeler include controlling combinators for learning mathematics, or for playing violin [29].

A *Controlling Operator* $\mathbf{C}$ acts on a controlled object X by its application $\mathbf{C} \bullet X$. Control means that knowledge needed to execute a task that is represented by arrow schemes in X is sufficiently well-known and described. This implies the need for a metric that measures the convergence gap. Note that $\mathbf{C}$ itself a term of the graph model of combinatory logic and thus a combinatory algebra term. Then, accomplishing control can be formulated by (16):

$$\mathbf{C} \bullet X = X \tag{16}$$

The equation (16) is a theoretical statement, referring to a potentially infinite loop. For solving practical problems, X must be approximated by finite subterms.

Thus, the control problem is solved by a *Control Sequence* $X_0 \subseteq X_1 \subseteq X_2 \subseteq \cdots$, a series of finite subterms and the controlling operator $\mathbf{C}$, starting with an initial X_0 and determined by (17):

$$X_{i+1} = \mathbf{C} \bullet X_i, i \in \mathbb{N} \tag{17}$$

This is called *Focusing*. The details can be found in Engeler [29, p. 299]. The controlling operator $\mathbf{C}$ gathers all faculties that may help in the solution. The inclusion operator in equation (17) is explained by the graph model. The control problem is a repeated process involving substitution, like finding the fixpoint of a combinator, and thus increasing the number of arrow schemes, and especially of choice functions, in the resulting focusing process.

Controlling combinators both collect and use empirical data for continuous training. Such an intelligent system incorporates the necessary functional processes for fine-tuning based on feedback received. For further details, please refer to the authors' paper on solving the control problem [43].

If the "Skills Definition" in Figure 5 is a training set, the program scheme represents *Deep Learning* [44]. In case the definition is linked to some feedback or hash function as in DeepSeek, it is *Reinforcement Learning* [45]. In all these

cases, the controlling combinators use the convergence gap; the measurable variation between actual behavior and expectations and requirements. In AI, the convergence gap is the same as in equation (15), but it is called "Loss Function". This term originates from *Signal Theory* and originally describes the loss of fidelity in analog sound transmission. Since the discovery of the *Fast-Fourier Transform* (FFT) [34], one understands that A/D-convergence is not a loss, but an acquisition of enough knowledge to reach a specific threshold for high-fidelity rendering of music. Deep learning uses the same principles.

Both come as (large) vectors and thus the Euclidean distance is easily computable. If learning is continuous, e.g., by experiences, by external feedback from a tutor, or by physical sensors, it is called an *Intelligent System*. Expectations and correct answers might also come from an external knowledge database, allowing the intelligent system to learn autonomously.

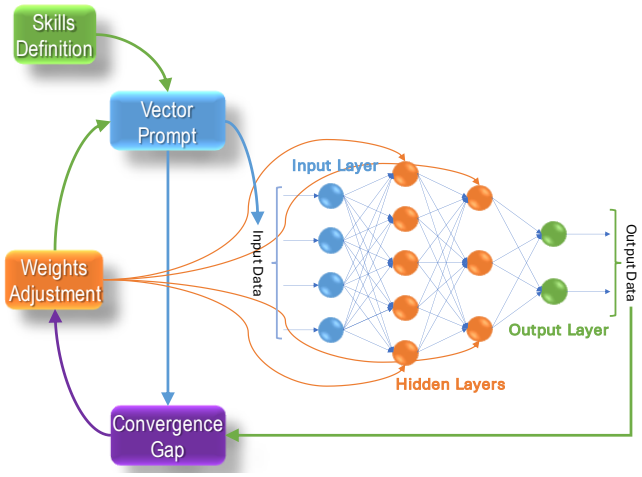


Figure 5. Controlling Combinators for self-learning intelligent systems

The architecture for RAG now extends. Instead of embedding the reference into response generation [42], and hoping it works, we set up functional processes for comparing LLM results with evidence from the knowledge database and calculating the convergence gap.

The convergence gap of such a system fully explains its behavior. Under well-defined conditions, such a system can be certified, even for safety critical tasks.

It is also possible to add more than one AI engine to an intelligent system, compare results and go forward with the most reliable one. Insufficient training, biases, and hallucinations therefore would become detectable.

Figure 6 shows an example of an intelligent system design that relies on two separate visual recognition engines analyzing the same scenario, one through a camera and the other through a Lidar. Such architecture requires that the reliability of each AI engine be known, under certain conditions, such as weather. In this way, the intelligent system can explain why it selected one or the other response.

If both AI engines provide an identical answer, this increases the overall reliability of the intelligent system's response significantly.

The graph model delivers the metrics for defining controlling combinators by inclusion, and it also allows us to combine knowledge and thus reliability correctly, by equation (9). This is discussed in the following section.

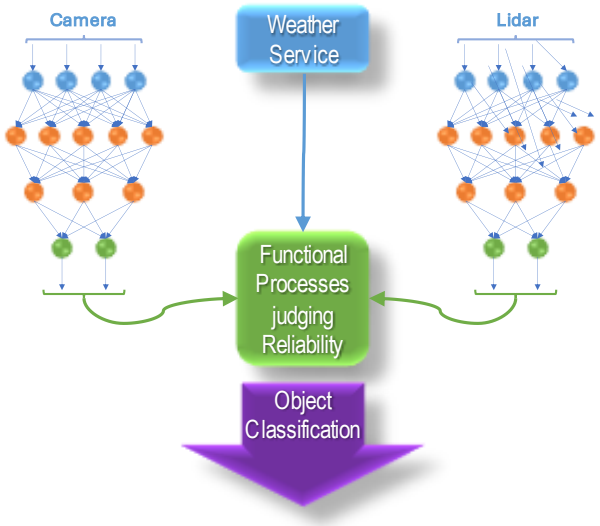


Figure 6. An Intelligent System that selects the most reliable AI response

Figure 6 shows a simple example of an intelligent system that combines two different AI engines that are used to visually recognize identical objects. Reliability might depend upon illumination and weather. Figure 5 and Figure 6 both show the importance of calculating reliability when operating intelligent systems.

V. Designing Intelligent Systems

Intelligent systems rely on the capability of measuring the quality of knowledge [46]. To this purpose, it is necessary to consider how software can be measured. This is not so easy, as software is not a tangible entity. The standard solution to this measurement challenge is to measure the functionality of software. As always in measurement theory, this is best achieved by constructing a model for the functionality and measuring the relevant model elements.

A. The COSMIC Model for Functionality

The COSMIC standard identifies layers. The layers' boundaries detect the flow of data moving from one object into another. Every *Data Movement* transports a *Data Group*, identifying the data moved from one object to another.

Data groups hold the information needed to assess privacy protection needs, or safety risk exposure, of data. They also transport knowledge from one *Object of Interest* into another. In certain cases, the data groups contain enough information to allow for generating code out of a COSMIC model [47].

The constituent element of the COSMIC model is a *Functional Process*. A functional process is an object together with a set of data movements, classified as either **R**ead or **W**rite, or **E**ntry or e**X**it. These data movements connect the functional process with *Persistent Data Stores*, or *Devices* respectively *Other Applications*, e.g., an AI engine. They

represent the *Functional User Requirements* (FUR) for the software being measured [48, p. 42].

Figure 7 is a sample *Data Movement Maps* according to ISO/IEC 19761 COSMIC. The connectors represent *Data Movements*. The data groups that they convey can be viewed as a single data set. Each data group should occur in a model only once. Its uniqueness is indicated by color-filled trapezes at their origin. Another move of the same data group between the same objects within a COSMIC functional process lets the trapeze blank.

Objects of Interest: For data movement maps, we draw four types of lifelines:

- **Functional Processes**: Objects that perform one or several functional processes in the COSMIC sense.
- **Persistent Data Store**: Objects that persistently hold data.
- **Devices**: A device can be a system user or anything providing or consuming data. They send (Entry) or receive (eXit) data groups from or to functional processes.
- **Other Applications**: Other applications use functional processes the same way as devices do; however, they typically represent other software or systems that can be modeled the same way using data movement maps.

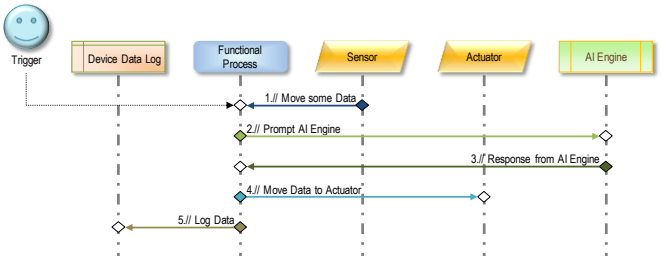


Figure 7. Sample Data Movement Map

The lifeline for functional processes represents, for example, a *Virtual Machine* (VM) or an *Electronic Control Unit* (ECU) that performs various calculations and implements several functional processes as defined in the COSMIC manual [48, p. 42]. *Triggers* usually indicate the starting data movement of one COSMIC functional process. Thus, one functional process lifeline can have several triggers. Lifelines representing persistent data stores can provide data services for more than one functional process. LLMs, or other AI engines appear as *Another Application* in the data movement maps.

B. Assigning Reliability Scores to Objects of Interest

A typical data movement map for an intelligent system consists of devices such as sensors, actuators, and persistent data sources, collecting data and delivering them to a functional process that prepares them as input to a suitable AI engine such as a neural network, a specific knowledge base, or some search engine. Another functional process will then work with its response and execute recommendations using a *Generative Pretrained Translator* (GPT) to communicate the response to a user through an output device. This is necessary because all knowledge in intelligent systems is represented by token vectors.

However, neither sensors, actuators nor an AI tool can be trusted 100%. Thus, data groups need to have special attribute for this: *Reliability*.

If data is persistently stored, it retains its reliability. All other data has some degree of reliability that is either known from physical devices, by assessing an AI tool with a test set, or by the functional processes the data groups go through. Reliability originates from devices or other applications. Although reliability is measured commonly by percentage numbers, it is a standard deviation, not a linear reliability average [49].

C. Combining Reliabilities with Regard to FUR

The data movement map describes how data groups move through the software. The functional processes combine these reliabilities by combining uncertainties associated with knowledge-based actions. In our example, the reliability of the data groups originating from the functional process in Figure 7 is the combination of the reliability of all incoming data. The expected overall reliability of the intelligent system is calculated from the uncertainty expectations of the output of the functional process.

The way data is combined in functional processes depends on the FUR that implements them. This requires an understanding of the discipline of requirements engineering for knowledge-processing systems [50]. As explained before, requirements address knowledge with distinct levels of certainty: While observations are usually not completely certain and learning concepts will never be fully reliable, there are rules, the so-called *Lambda Concepts*, that work in a mechanical way, preserving reliability. It is not a promising idea to implement Lambda concepts by arrow schemes. There exist much more efficient tools: conventional programming, that in turn also have a model expressible as arrow schemes. Typically, in intelligent systems Lambda concepts are implemented as programs according to rules, mostly in Python [51].

Depending on the FUR, functional processes can also reduce uncertainty, for instance when it selects the most reliable response between various kinds of AI engines as shown in Figure 6. The diverse ways of combining reliability are discussed in more detail in [46].

D. Propagating Reliability through Functional Processes

Lambda concepts might extract one source and discard the second or do substitution. Such operations might keep uncertainty unchanged by a functional process. Functional processes that implement Lambda concepts combine according to the Max principle, which preserves the maximum reliability of the different input data groups and propagates the maximum degree of knowledge reliability, while the normal combination of different uncertainties usually increases uncertainty and is called the $Comb$ principle. Since subsequent uncertainties can correct the initial uncertainty of data groups from the first source, the reliability is expected to decrease only according to the expected value obtained with the statistical sensitivity analysis, which means that the reliability decreases at a lower rate than with naive multiplication.

Let u_i be the uncertainty of the i^{th} input in a functional process with n input data groups. The uncertainty of the output of this functional process is:

$$u_{Comb} = \sqrt{\sum_{i=1}^{n} u_i^2} \quad (18)$$

This is the Euclidean distance between the uncertainties of the input data groups. Thus, the reliability propagation follows the same statistical rules as the profiles in Six Sigma transfer functions [33, p. 34].

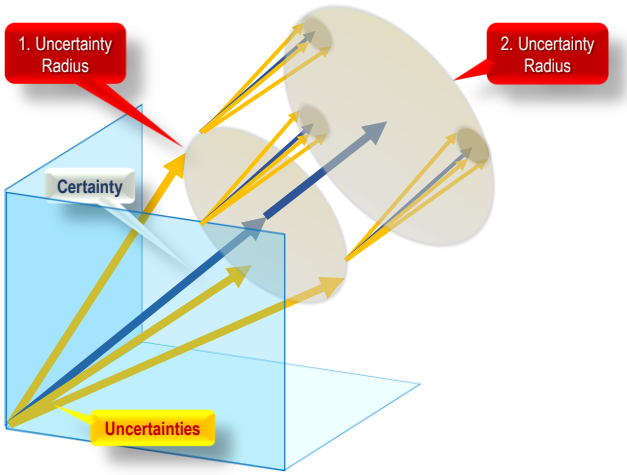


Figure 8. Combination of Uncertainties originating from AI input

The *Reliability* of a functional process with n input data movements carrying data groups with uncertainties $u_1, u_2, \dots, u_n$ is defined as:

$$Reliability(u_1, u_2, \dots, u_n) = 1 - u_{Comb} \quad (19)$$

With graphical visualization (Figure 8) we aim to explain the statistical methods used without going into formalisms. The figure proofs equation (19) in the special case of three dimensions ($n = 3$).

Assume there are some uncertainties originating from the first input source. The response range in the n-dimensional space of all responses is produced by some functional process. By combining this with the uncertainty originating from the second input source, it is unknown where the second uncertainty builds over the previous partial response.

Thus, the bundle of outcomes with an encompassing second uncertainty radius representing the expected uncertainty of a functional process with two inputs combined can correct part of the first. Expected uncertainties thus must be combined by using equation (19).

Reliability of a data group might change when originating from different functional processes. Also, a functional process might produce more than one data group as an eXit or Write data movement with different reliabilities, dependent from the data group.

E. Making an LLM Reliable

Not if we combine with a source of information with known reliability. We should set up a functional process connecting The LLM with some factual repository such as Wolfram|Alpha, or whatever is suitable for its topics. Best we feed the facts to the LLM and let the LLM apply its pretrained conversational capabilities as a transformer to transfer factual knowledge into arguments and explanations. Then the reliability combines from the reliability of the facts with the reliability of the LLM as a transformer and we can chain it with a Lambda concept that checks whether the LLM has kept well to the original facts. Thus, the FUR we have against the LLM is that we expect it to reproduce available facts with a known reliability. Here we assume 95%. As a further assumption, Wolfram|Alpha has been measured to be 98% reliable in the chosen context.

In Figure 9, there are two functional processes, one feeding the facts to the LLM and one comparing results. The persistent store serves for logging results, learning, and for communication between the two functional processes. The data movement map explains how the data groups are moved from one object to another.

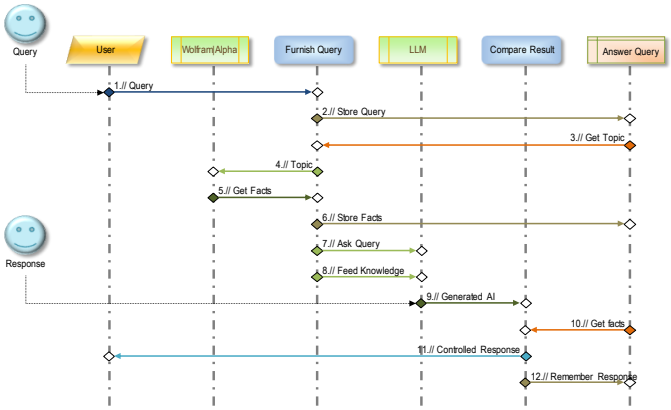


Figure 9. Data Movement Map for Combining LLM with Wolfram|Alpha

The *Compare Result* functional process in Figure 8 combines uncertainties according to the statistical methods explained in equations (18) and (19) as follows:

$$\sqrt{5\%^2 + 2\%^2} = 5.4\% \quad (20)$$

With regard to equation (19), this yields a reliability of 94.6% for the query functionality represented in Figure 9. Figure 9 and the equation (20) have been created and computed using an Excel-based tool from the authors, which is available to interested reader [33].

F. The Future of AI: Intelligent Systems

The current hype with AI is suffering from the same problem earlier attempts had: QFD, Expert Systems, and many other machine-based reasoning and decision tools could not explain how reliable they are. You could believe them or not; and sometimes, the non-believers proved to be true.

The new ability to build LLMs with recognizable CoT still does not answer the question, how reliable they are. Taking the graph model as an explanation of reliability, recognizing that knowledge and algorithmic processing are from the same source, namely the graph model, shows a way how to build intelligent systems with known reliability. It makes AI-based decisions and process control suitable for legal assessment and technical certification.

It is obvious that the reliability of a data group might change during operations of an intelligent system. Functional processes can calculate reliability, keep a log trace of it, and use it to guide processing through all programming steps in the data movement map.

In theory, intelligent systems consist of a controlling combinator, in most cases realized by implementing some functional processes, and an ANN part, typically implementing an LLM that is trained on the specific knowledge domain. This reflects equations (16) and (17). Obviously, both parts, the controlling combinator and the knowledge acquisition combinator. Both can be described as arrow schemes in the graph model, but they are implemented differently, in the most effective way. It is indeed not necessary to train an LLM to do reasoning, because a controlling combinator implementing functional processes in Python are much more effective. Explaining, maintaining, and improving such a controlling combinator is much easier in Python (or any other suitable programming environment) than training an LLM. However, it is possible to train an LLM in logical reasoning, but this is not needed. It is much more straightforward, and way more effective, to use Engeler's controlling combinator as a design paradigm for intelligent systems.

VI. Conclusion and Future Work

Like humans, the result of any ANN is as unreliable as any result from an uneducated NNN. Without logical foundations, logical reasoning, and feedback from the environment, humans are also just hallucinating.

Intelligent systems can do better, except getting feedback and learning from it. The key is to combine combinators representing neural networks with combinators doing logical derivations. This is primarily a design principle, but secondly also an operating paradigm.

In this paper, we have provided evidence for:

- RQ 1: DNNs can be represented in the graph model of combinatory logic as well as any other neural network, including the brain or QFD;
- RQ 2: CoT does not relate to a defined and unique sequence of arrow schemes because of missing normal form, but can be explained using other arrow schemes, such as QFD;
- RQ 3: Intelligent systems explain how AI behavior can be controlled.

The graph model of combinatorial logic does not provide an alternative for implementing AI, but it is an excellent guide and theoretical foundation for what can be done with AI, for explaining AI, but also for learning where AI meets its limits. The current step forward is collecting several designs of intelligent systems with controlling combinators, finding methods for measuring reliability and defining suitable convergence gaps. This work in progress of the authors is shared with interested parties; the authors have no institution or sponsor to help with this [52].

It is possible that ANNs can learn logical reasoning based on facts. However, combining AI engines with feedback loops originating from reality requires much less effort. Testing AI results for feasibility and physical soundness using traditional programming methods creates trustworthiness and adds credibility and explainability to AI results.

It remains the idea that AI could be explained by searching for arrow schemes that provide the same responses. Since combinatory logic does not have normal forms, this seems feasible. It could be used as a validation process for AI. However, for now, this is a future research project.

Acknowledgment

The authors would like to thank Lab42 in Davos for asking excellent questions and promoting the ARC Challenge [10], now ARC Prize [53], and to the anonymous reviewers who helped to improve this paper. Special thanks to Erwin Engeler for his suggestions and availability for valuable discussions with his former student.

References

[1] T. M. Fehlmann and E. Kranich, "The Graph Model of Combinatory Logic as a Model for Explainability," in *ThinkMind Digital Library https://www.thinkmind.org*, Valencia, Spain, Proc. Int. Conf. Systems Explainability (EXPLAINABILITY 2024), pp. 40-46, 2024..

[2] K. Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I," *Monatshefte für Mathematik und Physik,* vol. 38, no. 1, pp. 173-198, 1931.

[3] A. Turing, "On computable numbers, with an application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society,* vol. 42, no. 2, pp. 230-265, 1937.

[4] P. Raatikainen, "Gödel's Incompleteness Theorems," in *The Stanford Encyclopedia of Philosophy*, E. N. Zalta, Ed., 2020.

[5] H. Curry and R. Feys, Combinatory Logic, Vol. I, Amsterdam: North-Holland, 1958.

[6] A. Church, "The Calculi Of Lambda Conversion," *Annals Of Mathematical Studies 6,* 1941.

[7] E. Engeler, "Algebras and Combinators," *Algebra Universalis,* vol. 13, pp. 389-392, 1981.

[8] F. Rosenblatt, "The Perceptron: A Perceiving and Recognizing Automaton (Project PARA)," Cornell Aeronautical Laboratory, Inc., Buffalo, 1957.

[9] M. Minsky and S. Papert, Perceptrons: An Introduction to Computational Geometry, 2nd edition with corrections ed., Cambridge, MA: The MIT Press, 1972.

[10] F. Chollet, "On the Measure of Intelligence," arXiv:1911.01547 [cs.AI], Cornell University, Ithaca, NY, 2019.

[11] K. Wijekumar, B. J. Meyer, and P. Lei, "High-fidelity implementation of web-based intelligent tutoring system improves fourth and fifth graders content area reading comprehension," *Computers & Education,* vol. 68, pp. 366-379, 2013.

[12] A. Peterson, "Literacy is More than Just Reading and Writing," National Council of Teachers of English, 23 March 2020. [Online]. Available: https://ncte.org/blog/2020/03/literacy-just-reading-writing/. [Accessed 14 May 2025].

[13] M. v. Gerven and S. Bohte, "Artificial Neural Networks as Models of Neural Information Processing," in *Frontiers in Computational Neuroscience*, Lausanne, 2017.

[14] IBM TechXchange Community, "What is explainable AI?," IBM Reserach, Available at: https://www.ibm.com/topics/explainable-ai. [Accessed 15 May 2025].

[15] F. Dallanoce, "Explainable AI: A Comprehensive Review of the Main Methods," Medium.com, San Francisco, California, 2022.

[16] M. R. Morris, J. Sohl-Dickstein, N. Fiedel, T. Warkentin, and A. Dafoe, "Levels of AGI for Operationalizing Progress on the Path to AGI," arXiv:2311.02462v4 [cs.AI], Cornell University, 2024.

[17] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," arXiv:2201.11903v6 [cs.CL], Cornell University, 2023.

[18] S. Xu, W. Xie, L. Zhao, and P. He, "Chain of Draft: Thinking Faster by Writing Less," arXiv:2502.18600v2, Cornell University, Ithaca, NY, 2025.

[19] H. Curry, J. Hindley, and J. Seldin, Combinatory Logic, Vol. II, Amsterdam: North-Holland, 1972.

[20] T. M. Fehlmann and E. Kranich, "Intuitionism and Computer Science – Why Computer Scientists do not Like the Axiom of Choice," *Athens Journal of Sciences,* vol. 7, no. 3, pp. 143-158, 2020.

[21] T. M. Fehlmann and E. Kranich, "A General Model for Representing Knowledge - Intelligent Systems Using Concepts," *Athens Journal of Sciences,* vol. 11, pp. 1-18, 2024.

[22] H. Barendregt and E. Barendsen, Introduction to Lambda Calculus, Nijmegen: University Nijmegen, 2000.

[23] H. P. Barendregt, "The Type-Free Lambda-Calculus," in *Handbook of Math. Logic*, vol. 90, J. Barwise, Ed., Amsterdam, North Holland, 1977, pp. 1091 -1132.

[24] T. M. Fehlmann and E. Kranich, "The Fixpoint Combinator in Combinatory Logic - A Step towards Autonomous Real-time Testing of Software?," *Athens Journal of Sciences,* vol. 9, no. 1, pp. 47-64, 2022.

[25] E. Zachos, "Kombinatorische Logik und S-Terme," ETH Dissertation 6214, Zurich, 1978.

[26] Lisp-Community, "Common Lisp," 2015. [Online]. Available: https://lisp-lang.org/. [Accessed 15 May 2025].

[27] T. M. Fehlmann, Autonomous Real-time Testing – Testing Artificial Intelligence and Other Complex Systems, Berlin, Germany: Logos Press, 2020.

[28] V. Zhong, J. Mu, L. Zettlemoyer, E. Grefenstette, and T. Rocktäschel, "Improving Policy Learning via Language Dynamics Distillation," arXiv:2210.00066v1, Cornell University, 2022.

[29] E. Engeler, "Neural algebra on "how does the brain think?"," *Theoretical Computer Science,* vol. 777, pp. 296-307, 2019.

[30] P. Barceló, J. Pérez, and J. Marinković, "On the Turing Completeness of Modern Neural Network Architectures," arXiv: 1901.03429v1 [cs.LG], Cornell University, Ithaca, NY, 2019.

[31] G. Hinton, "A Practical Guide to Training Restricted Boltzmann Machines," in *Neural Networks: Tricks of the Trade*, Springer Lecture Notes in Computer Science, vol 7700. Berlin, Heidelberg, 2012, pp. 599-619.

[32] S. Wolfram, What is ChatGPT doing ... and Why Does it Work?, Champaign, IL: Wolfram Media, Inc., 2023.

[33] T. M. Fehlmann, Managing Complexity – Uncover the Mysteries with Six Sigma Transfer Functions, Berlin, Germany: Logos Press, 2016.

[34] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation,* vol. 19, pp. 297-301, 17 August 1964.

[35] T. M. Fehlmann and E. Kranich, "The World Formula and the Theorem of Perron-Frobenius: How to Solve (Almost All) Problems of the World," *Athens Journal of Sciences,* vol. 10, no. 2, pp. 95-110, 2023.

[36] T. L. Saaty and J. M. Alexander, Conflict Resolution: The Analytic Hierarchy Process, New York, NY: Praeger, Santa Barbara, CA, 1989.

[37] Ł. Lachowski, "On the Complexity of the Standard Translation of Lambda Calculus into Combinatory Logic," *Reports on Mathematical Logic,* vol. 53, pp. 19-42, 2018.

[38] Z. Jin and W. Lu, "Self-Harmonized Chain of Thought," arXiv:2409.04057v1 [cs.CL], Cornell University, 2024.

[39] T. M. Fehlmann and E. Kranich, "How to Explain Artificial Intelligence to Humans - Learning from Quality Function Deployment," in *EuroSPI*, Munich, 2024.

[40] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," arXiv:2106.09685v2 [cs.CL], Cornell University, Ithaca, NY, 2021.

[41] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv:2312.10997v5 [cs.CL] , Cornell University, 2024.

[42] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," arXiv:2005.11401v4 [cs.CL], Cornell University, 2021.

[43] T. M. Fehlmann and E. Kranich, "Making Artificial Intelligence Intelligent - Solving the Control Problem for Artificial Neural Networks by Empirical Methods," in *Human Systems Engineering and Design (IHSED2024): Future Trends and Applications. AHFE (2024) International Conference*, Split, 2024.

[44] M. Ganaie, M. Hu, A. Malik, M. Tanveer and P. Suganthan, "Ensemble deep learning: A review," arXiv:2104.02395 [cs.LG], Cornell University, Ithaca, NY, 2022.

[45] DeepSeek-AI, "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning," arXiv:2501.12948v1, Cornell University, Ithaca, NY, 2025.

[46] T. M. Fehlmann and E. Kranich, "Measuring the Quality of Intelligent Systems," in *Intelligent Systems and Applications*, vol. 1066, K. Arai, Ed., IntelliSys 2024, Amsterdam, Lecture Notes in Networks and Systems, Springer, Cham, 2024, pp. 438-455.

[47] A. Oriou, E. Bronca, B. Bouzid, O. Guetta, and K. Guillard, "Manage the automotive embedded software development with automation of COSMIC," in *IWSM Mensura 2014*, Rotterdam, 2014.

[48] COSMIC Measurement Practices Committee, "The COSMIC Functional Size Measurement Method – Version 4.0.2 – Measurement Manual," The COSMIC Consortium, Montréal, 2017.

[49] L. S. Jutte, K. L. Knight, B. C. Long, J. R. Hawkins, S. S. Schulthies, and E. B. Dalley, "The Uncertainty (Validity and Reliability) of Three Electrothermometers in Therapeutic Modality Research," *Journal of Athletic Training,* vol. 40, no. 3, pp. 207-210, 2005.

[50] T. M. Fehlmann and E. Kranich, Requirements Engineering for Cyber-Physical Products, Systems, Software and Services Process Improvement. EuroSPI 2023 ed., Vols. Systems, Software and Services Process Improvement. EuroSPI 2023, M. Yilmaz, C. P. A. Riel and R. Messnarz, Eds., Grenoble: Communications in Computer and Information Science, Springer, Cham, 2023.

[51] S. F. Python, "Python Setup and Usage," 2001-2024. [Online]. Available: https://docs.python.org/3/using/index.html. [Accessed 27 April 2024].

[52] T. M. Fehlmann, "Intelligent Systems - A Recollection," Euro Project Office AG, 9 May 2024. [Online]. Available: https://web.tresorit.com/l/AXX78#FaBkGqfY2cF_JsVmX70_ng.

[53] Lab 42, "ARC Price," Mike Knoop, Davos and San Francisco, 2024.