AI for real-world network operation

WP2– Fundamental AI building blocks

D2.3– Formulation of transparent and trustworthy knowledge-assisted, hierarchical and distributed AI for network infrastructures

AI4REALNET has received funding from European Union's Horizon Europe Research and Innovation programme under the Grant Agreement No 101119527.

DOCUMENT INFORMATION

DOCUMENT	D2.3– Formulation of transparent and trustworthy knowledge-assisted, hierarchical and distributed AI for network infrastructures
TYPE	Report
DISTRIBUTION LEVEL	Public
DUE DELIVERY DATE	31/03/2026
DATE OF DELIVERY	23/03/2026
VERSION	V2.1
DELIVERABLE RESPONSIBLE	University of Amsterdam
AUTHOR (S)	Herke van Hoof (UvA)
OFFICIAL REVIEWER/s	Mohamed, Hassouna (Fraunhofer/UKASSEL)
	Kurt Brendlinger (UKASSEL)
	Ricardo Chavarriaga (ZHAW)

DOCUMENT HISTORY

VERSION	AUTHORS	DATE	CONTENT AND CHANGES
Version 0.1	Herke van Hoof	24/09/2025	Set up template
Version 1.0	Alberto Castagna	05/11/2025	Maze-Flatland section
Version 1.1	Milad Leyli-abadi	10/11/2025	PINNs Graph solver section
Version 1.2	Milad Leyli-abadi	12/11/2025	Expert Agent
Version 1.3	Ricardo Bessa, Ferinar Moaidi	30/12/2025	Evolving Operator Rules
Version 1.4	Mohamed Hassouna, Duarte Filipe Dias, Toni Wäfler, Anna Fedorova, Clark Borst, Patrick Zinsli, Gianvito Lopasio, Sebastiaan de Peuter, Marius Captari	16/02/2026	Contributions to first full draft
Version 1.5	Herke van Hoof	16/02/2026	Draft introduction and conclusion
Version 1.6	Herke van Hoof	02/03/2026	Finalization full draft
Version 2.0	Herke van Hoof	18/03/2026	Second draft based on internal review
Version 2.1	Herke van Hoof	23/03/2026	Revision based on feedback steering committee

ACKNOWLEDGEMENTS

NAME	PARTNER
Alberto Castagna	enliteAI
Anna Fedorova	ZHAW
Anton Fuxjäger	enliteAI
Clark Borst	TU Delft
Duarte Filipe Dias	INESC TEC
Ferinar Moaidi	INESC TEC
Gianvito Lopasio	Politecnico di Milano
Marius Captari	UvA
Milad Leyli-abadi	IRTSX
Mohamed Hassouna	Fraunhofer/UKASSEL
Patrick Zinsli	FHNW
Ricardo Bessa	INESC TEC
Sebastiaan de Peuter	UvA
Toni Wäfler	FHNW

DISCLAIMER

EXECUTIVE SUMMARY

This deliverable documents the effort of work package 2 of the AI4REALNET project to formulate transparent and trustworthy knowledge-assisted, hierarchical, and distributed AI for network infrastructures.

In this deliverable, we argue that being knowledge-assisted, hierarchical, decentralized, safe, explainable, and transparent is critical for AI building blocks that will eventually interface with human operators for critical network infrastructure.

In particular, knowledge-assisted AI aids scaling up in the presence of limited data through including inductive bias, while decentralized and hierarchical methods similarly assist scaling up by factorizing a complex environment.

On the other hand, safety is of course an essential property for a critical system, as it enforces the agent respect important safety constraints. Transparency and explainability help human operators or auditors understand the recommendations or decisions made by the agent, and thus is critical for the deployment of an AI system. These aspects also have important consequences for the ability of human operators to do their tasks in an AI-assisted or collaborative scenario, for example where it concerns cognitive of motivational aspects.

These six important properties are targeted by 18 contributions described in chapter 3 of this deliverable as follows:

- (3.1) **Physics informed graph neural network solver**, which combines data-driven and physics-based updates in a *knowledge-assisted* approach for the computation of power flows in large networks.

- (3.2) **Two adapted deepQ agents**, which aim to improve decision making in large-scale networks by reducing the action space using either the influence graph to identify promising candidates for exploration actions; or uses expert-defined filters to reduce the action space. This *knowledge-assisted* approach is accompanied by an *explainability dashboard* visualizing the environment as well as the agent's action and its impact.

- (3.3) **Adaptive genetic network programming approach,** that allows the inclusion of adaptive nodes that modify their behavior in response to the network state. This results in a *knowledge-assisted* and high-capacity architecture that is updated through a combination of reinforcement-learning and genetic strategies.

- (3.4) **Planner enhanced AI**, which aims to exploit the implicit knowledge in fast planning heuris-

tic within a data-driven AI approach. The AI component of the *knowledge-assisted* and partially *decentralized* method modifies the input representation to stimulate the planner heuristic to reach better solution, by e.g. avoiding bottlenecks where possible.

- (3.5) **Maze-flatland** introduced a *decentralized* and *hierarchical* approach to vehicle routing and scheduling by de-coupling dispatching and routing decision. Thereby, it improves the scalability of a reinforcement-learning driven approach that avoids the bottlenecks of centralized traditional methods.

- (3.6) **State-action factorization**, which provides a *decentralized* approach for complex decision-making problems by breaking up such a complex problem in smaller problems. These are identified by finding clusters of states and action dimensions that primarily influence state dimensions within the same cluster.

- (3.7) **Network-distributed Q-learning**, which learns a *decentralized* strategy by considering distinct agents for the nodes in a graph. During learning, these agents minimally communicate with each other to propagate the effect of an agent's decision on the wider context to its local update function.

- (3.8) **Communication networks in multi-agent reinforcement learning** propose a framework that allows experimentation on communicating, *decentralized* systems, and evaluates the integrated CommNet approach on this benchmark.

- (3.9) **Multiclass failure prediction** provides a framework for failure analysis and -prediction. Failures are clustered in several types, and the occurrence of such failures is predicted for several possible horizons. These predictions help to improve the system's *safety*, while being *transparent* thanks to the interpretable failure types.

- (3.10) **Soft-target imitation learning agent** provides a *knowledge-assisted* and *safe* approach by training an agent to imitate a physics-based teacher. By doing this using a soft probability distribution, the multi-modality of the learning problem is respected.

- (3.11) **Gridexplainer** provides a framework for providing *transparency* and *explanations* through a visualization of the features that influence an agent's actions. The framework is modular, providing access to various post-hoc feature attribution methods and several evaluation metrics.

- (3.12) **TraceRL** provides an interactive visualization and analysis interface that lets users explore the decision-making of the agent. Through trajectory visualization and counterfactual simulation, the agent's decision making becomes more *transparent* and *explainable*.

- (3.13) **Action alternative explainer** aims to provide decision support by *explaining* the future effects of possible actions under consideration. To make such future predictions, the operator's decision-making and how it is influenced by the predictions themselves need to be anticipated.
- (3.14) **C-PG framework for safe reinforcement learning** provides new methods for *safe* reinforcement learning with policy-gradient based methods. It advances the state of the art by providing global last-iterate convergence guarantees, and allowing both action-based and parameter-based exploration.
- (3.15) **Integration of ethical metrics** provides a *transparent* analysis of the trade-offs between several ethical criteria, such as fairness, robustness, and *safety*. An analysis of two different approaches highlights where such criteria might come in conflict.
- (3.16) **Human assessment model** provides a *transparent* and personalized assessment of the cognitive state of an operator. By monitoring the cognitive performance and stress level of the operator, critical situations can be flagged and system *safety* can be improved.
- (3.17) **Domain transparency in airspace sectorization** proposes a *transparent* AI method to aid human operators in airspace sectorization. The approach based on a Voronoi tesselation is easy to understand and modify by such operators.
- (3.18) **Design requirements for AI transparency** provides a theoretical analysis of *transparency* requirement beyond explainability. It studies in a structured manner human needs in both a co-learning and a fully autonomous scenario, providing guidelines for AI design based on task- and knowledge characteristics.

These contributions thus formulate advances to transparent and trustworthy knowledge-assisted, hierarchical, and distributed AI for network infrastructures. These approaches are typically conceived for and/or evaluated on at least one of the three project domains of power grid control, train scheduling, and air traffic management. However, as we further analyze, almost all of these contributions are also applicable (with or without adaptations) to other domains, within or outside of the project, ranging from water distribution systems to telecommunication networks or traffic flow management.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Work package 2 of the AI4REALNET project has the purpose of developing fundamental AI 'building blocks' for augmented decision-making in large-scale sequential problems. Specifically the work focuses on three aspects: (1) developing AI methods that combine data-driven components with domain specific knowledge; (2) developing scalable AI solutions; and (3) developing trustable transparent and safe systems.

Work in this work package, of course, takes place in the context of the larger project. It builds on the conceptual framework developed in work package 1 [17], and it provides input for the development of interactive AI approaches in work package 3. Furthermore, the contributions described here will be further evaluated in work package 4 [73]. Earlier development in the current work package included a position paper on AI for the operation of critical energy and mobility network infrastructures [17], that sketches challenges in this domain and suggests a general approach and specific research directions to address these, which to a large extend informs the research described in the present document.

The purpose of this report is to document the *formulation of transparent and trustworthy knowledge-assisted, hierarchical and distributed AI for network infrastructures*. To that end, the report presents how these properties are operationalized in the AI4REALNET project, and describe in detail the specific contributions or 'building blocks' aimed at incorporating and improving these properties.

In more detail, this report is organized as follows. After this brief introduction, Section 2 will describe the formulation of the properties 'knowledge-assisted', 'hierarchical', 'decentralized', 'explainable', 'transparent', and 'safe' AI within the AI4REALNET project, and explain their relation to the goals of developing trustworthy and scalable AI solutions.

After that, Section 3 will describe the concrete approaches developed in the AI4REALNET project aimed at developing AI systems that exhibit or improve these properties. Finally, Section 4 will put these contributions in the broader context of the goals of the work packages and the applicability to alternative domains inside or outside of the project.

2. FORMULATION OF DESIRABLE AI PROPERTIES FOR NETWORK INFRASTRUCTURES

As mentioned earlier, the purpose of the current work package is to develop AI building blocks that (1) combine data-driven components with domain specific knowledge (2) are scalable, and (3) are trustable transparent and safe systems.

Scalability can be achieved in several ways. As instances scale, training data and training time requirements can increase exponentially. The specific approach of combining data-driven and knowledge-driven components ('*knowledge-assisted AI*') provides strong inductive bias. It can thus contribute to scalability, by reducing the dependence on from-scratch learning and thus training data and training time requirements. Other approaches to increase scalability include *decentralized* and *hierarchical AI*. These approaches tend to split up a problem, either in parallel or higher- and lower level components. Each of these components is likely to be easier to learn and computationally cheaper to execute, and thus more scalable. By simplifying the problem to be learned (through external knowledge or decomposition), these methods might also increase the robustness of an AI system.

Turning now to trustable transparent and safe systems, we will first discuss the notion of trustworthiness. Trustworthiness is a complex concept with many different attributes. Following the EU guidelines, we adopted the Assessment list for trustworthy Artificial Intelligence (ALTAI, [55]), which was extended for critical network infrastructure within the AI4REALNET project [17]. The ALTAI distinguishes the following dimensions of trustworthy artificial intelligence: human agency and oversight; technical robustness and safety; privacy and data governance; transparency; diversity, non-discrimination, and fairness; environmental and societal well-being; and accountability.

Within work package 2, which focuses on technical building blocks, attention is focused in particular on the dimensions of transparency, and of technical robustness and safety. In particular, in this chapter we will discuss *transparency* in general as well as *explainability* specifically, and *safety*. Other trustworthiness aspects receive a larger focus in work packages 3 (on AI augmented decision making) and 4 (validation and impact assessment).

Emphasizing safety directly increases humans' trust in AI systems. Users and stakeholders are more likely to rely on AI agents when they consistently demonstrate predictable and safe behavior. Furthermore, trustworthiness arises not only from technical reliability but also from the transparency of how safety is managed and measured. When it is possible to explain actions of an AI system, quantify its risks, and prove compliance with safety constraints, it becomes a trustworthy partner in decision-

making.

In the rest of this section, we will recap the formulation of these important properties of AI systems (knowledge-assisted, hierarchical, decentralized, explainable, transparent, and safe AI). For a more comprehensive discussion, we refer the reader to the earlier deliverables discussing these topics [17, 93].

2.1. KNOWLEDGE-ASSISTED AI

Knowledge-assisted AI refers to AI systems that make use of pre-existing knowledge, typically in addition to data-driven elements. Since knowledge-assisted AI was covered in detail in earlier deliverables [17, 93] and a corresponding paper [94], here, we will give a short synopsis.

Within the context of AI4REALNET, a working definition of knowledge-assisted AI is *a hybrid approach that combines learning elements with existing implicit or explicit knowledge*. In this definition, explicit knowledge refers to knowledge that is formally represented as, e.g. mathematical equation, invariance, rule, knowledge graph, or probabilistic relation. Implicit knowledge is the knowledge implicitly contained in existing (non-AI) approaches such as planning- or optimization heuristics.

The notion of knowledge-assisted AI has a strong conceptual overlap with terms such as 'informed machine learning', 'neuro-symbolic', or 'hybrid AI'. 'Informed machine learning', as described by Von Rueden et al. [137], is a more restrictive concept that considers methods that learn from both data and formally represented and explicitly represented prior knowledge from an independent source. 'Neuro-symbolic AI', as surveyed by e.g. Yu et al. [148], Hitzler et al. [54], Sarker et al. [108], Garcez et al. [44], is an overlapping concept that describes the combination of neural and symbolic systems into a unified framework [148]. This definition includes knowledge-assisted systems where parts of the system that learn take symbolic prior information as additional input. Lastly, 'hybrid AI' is a more loose term typically describing combinations of different artificial intelligence methods, often combining data-driven machine learning with logic-based symbolic AI and thus including neuro-symbolic methods.

In the context of the operation of network infrastructures, there are a number of benefits we expect from knowledge-assisted AI. The integration of prior knowledge constitutes an inductive bias, which is widely understood to increase learning efficiency. This is particularly useful in network infrastructures, where the size of available datasets is typically of a lower order of magnitude than the internet-scale corpora used to train large language or computer vision models. Inductive biases such as those in neuro-symbolic systems are also widely understood to improve generalization [148], which helps when encountering anomalous situations due to calamities or distribution shift that would not be present in the training data. Lastly, models that conform to prior knowledge might be more easily understandable to operators working with such models, for example, aiding in explanations of neuro-symbolic systems [54].

2.2. HIERARCHICAL DECISION MAKING

Hierarchical decision making approaches divide the overall decision making problem into several levels. Within the AI4REALNET project, the most relevant approaches are those concerning hierarchical reinforcement learning. The discussion here recapitulates and extends the formulation provided in an earlier deliverable [93].

Hierarchical reinforcement learning approaches decompose a problem into a hierarchy of sub-problems or subtasks [100]. Typically, at a high level a sub-task or sub-problem is chosen, which is then executed at a lower level (perhaps by recursively setting a sub-task or sub-problem for an even lower level). Typically, control is handed over to the lower level for some time, e.g., until the sub-task is solved or a problem state is reached. From the perspective of the higher level, this induces temporal abstraction. Temporal abstraction shortens the effective planning horizon to the number of times a sub-task at this level is chosen (rather than the number of primitive actions executed at the lowest level) [93]. As a consequence, the *curse of horizon* can be alleviated [93].

Two common types of approaches for hierarchical reinforcement learning are the 'feudal hierarchy' and 'policy-tree' approaches [100]. In the feudal hierarchy type of approach, there is one policy at each level. For each pair of adjacent levels, we can distinguish a high-level 'manager' and a lower level 'worker'. In this type of approach, each 'manager' sets a goal for its worker. This worker is tasked to achieve their goal by executing primitive actions, or by in turn setting subgoals for an even lower level. A classical (and eponymous) example of such an approach is 'Feudal reinforcement learning' [27].

In the policy-tree type of approach, there are multiple policies at each level, arranged in a tree structure. Each of these policies specializes in a certain type of behavior (e.g., achieving a certain type of goal). The goal for each policy above the lowest layer is then to choose between the policies below it. The chosen policy will then be executed until some termination event is reached. The well-known options framework is an example approach of this type [128].

Hierarchical reinforcement learning has several potential advantages for the operation of network infrastructures. The temporal abstraction described above is likely beneficial in long-horizon tasks, such as the control of a power network over many decision-making intervals [93]. The decomposition over multiple levels can also allow 'information hiding', i.e., providing only coarse-grained data to the higher levels and local detail to the lower levels, thereby helping to scale to large state spaces [27]. Decomposition also results in a policy consisting of individual 'building blocks' rather than a single monolithic and complex functions. This structure can help transfer and allow for easier task division in a multi-agent setup [100].

2.3. DECENTRALIZED AI

Decentralized AI refers to control systems powered by AI in which the learning process is distributed among multiple agents. An example of decentralized AI is distributed reinforcement learning (DRL), an important challenge in multi-agent approaches. In DRL, the learning process is decentralized and the agents cooperate to achieve the same objective. Here, we provide an overview of DRL and how it is connected to the context of AI4REALNET.

In DRL, different agents act simultaneously in the environment and have to learn how to act in order to maximize a reward signal. The underlying problem structure is mathematically represented by Markov Games [114], Partially Observable Markov Games [51] or Decentralized Partially Observable Markov Decision Processes [16], depending on its characteristics.

Solutions found by DRL algorithms typically correspond to Nash equilibria (solution from which none of the agents has any incentive to deviate). Ideally, the agent find an optimal Nash equilibrium where the team reward is highest - the possible existence of sub-optimal Nash equilbria needs to be considered in approaches and their convergence guarantees.

In DRL, agents must communicate various types of information—including sampled data (observations/actions), predicted data, or knowledge (model parameters)—to efficiently achieve their shared objective. This necessity defines two main approaches in DRL: decentralized training and execution, where agents operate and learn locally using single-agent RL; and centralized training and decentralized execution, where centralized information is shared among agents and is used for learning but not for action in evaluation. Centralization, even if only during training, is favored by almost all state-of-the-art algorithms because it facilitates more efficient coordination and offers stronger theoretical convergence guarantees [4].

The core motivation for DRL is to manage problem complexity by splitting the learning task into simpler subtasks. However, this distribution often introduces some bias. Specifically, DRL aims to combat the curse of dimensionality, i.e. the challenge posed by large state and action spaces, which severely increases the sample complexity of algorithms. In simpler terms, real-world problems require an unmanageable number of samples (data points) for agents to learn optimal behavior with acceptable accuracy.

In the context of AI4REALNET, all the use-cases of the project are affected by the curse of dimensionality. Furthermore, state-of-the-art algorithms with centralized training are unfeasible in most cases. For this reason, original algorithms are required that decompose overall decisions into sub-decisions by various agents that make use only of local information local to themselves.

2.4. EXPLAINABLE AI

Integrating AI into real-world systems necessitates a deep understanding of AI choices to ensure human trust. Without such trust, AI decisions may be ignored by operators, rendering the system ineffective [2]. This challenge is particularly pronounced in safety-critical domains such as power grids, railway networks, and air traffic management, where high-performing models are often complex and opaque. Explainable AI (XAI) therefore plays a central role in supporting trustworthiness and effective Human–AI interaction.

As described in recent surveys [15, 26, 89], XAI concerns methods for explaining models learned using AI techniques and clarifying the decisions of those models. Existing approaches are commonly divided into *intrinsically* explainable models which are transparent by design, and *post-hoc* explanation techniques, which are applied to otherwise opaque models. In many critical infrastructure applications, performance requirements motivate the use of complex models, making the need for explanations of complex models particularly relevant. Within explainable AI, several types of methods can be distinguished, which will be discussed in the context of critical infrastructure below.

2.4.1. COUNTERFACTUAL AND CONTRASTIVE EXPLANATIONS

Counterfactual and contrastive explanations focus on the comparison between actual and hypothetical situations. Contrastive explanations address why a specific decision was made rather than another, while counterfactual explanations identify which input changes would alter the predicted outcome [124]. These methods are often preferred by users over case-based reasoning and are considered “actionable” [133].

While existing literature focuses heavily on supervised learning, there is a gap in applying these methods to decision-making and planning problems. Previous model-specific works include explaining plan traces using temporal logic [65] or addressing operator misconceptions [23]. Extending these ideas toward model-agnostic approaches for sequential decision-making and control environments represents an important research direction, particularly in domains where operators must understand how alternative actions could have changed system outcomes.

2.4.2. FEATURE IMPORTANCE METHODS

Feature attribution methods aim to unmask the “black box” by quantifying the influence of individual input features on a model’s prediction [107]. These are broadly categorized into perturbation-based methods and gradient based methods. *Perturbation-based methods* such as LIME [104] and SHAP [113] analyze changes in output when input features are masked or altered. *Gradient-based*

methods such as Integrated Gradients [127] and Saliency methods [120] on the other hand, compute gradients of the output with respect to the input to identify salient features.

Although feature attribution techniques are widely used in classification tasks, their application to deep reinforcement learning agents remains limited. Open challenges include defining appropriate attribution targets, handling temporal dependencies, and evaluating explanation quality in terms of faithfulness, robustness, and complexity. Addressing these challenges is particularly relevant in safety-critical control applications, where misleading explanations may undermine operator trust.

2.4.3. PROTOTYPE LEARNING

While post-hoc methods aim to approximate the behavior of a complex model, they often suffer from a lack of faithfulness to the model's internal reasoning process. As an alternative, *intrinsically explainable* models [26], including Prototype Learning approaches, have gained increasing attention [24, 106]. These models fall within the class of *transparent* models [15], in which the decision-making structure itself is interpretable without requiring a separate explanation algorithm.

In prototype learning, the model learns a set of prototypical examples (prototypes) from the training data. During inference, the model classifies a new input based on its similarity to these learned prototypes in a latent feature space. For instance, in a power grid context, a critical state might be classified as "dangerous" not because of an obscure non-linear combination of features, but because it is structurally similar to a known historical blackout scenario stored as a prototype.

This methodology offers distinct advantages for trustworthiness [2]. By surfacing the specific examples that drove the decision (e.g., "This situation is similar to another congestion event"), the system provides explanations that are inherently actionable and verifiable by human operators. Such *inherently interpretable* approaches [89] are particularly valuable for safety-critical tasks where verifying the "why" behind an AI decision is as important as the decision itself.

2.4.4. VISUALIZATION-BASED EXPLANATIONS

Visualizations convey technical explanations in an intuitive manner. In classification, saliency maps are standard for highlighting influential regions [120]. In reinforcement learning, however, visualizations may fail to convey causal relationships between state features and agent decisions [12].

To address this limitation, recent work emphasizes the importance of domain-aware visualization techniques. These include visualizing counterfactual state representations [96] and highlighting elements of the environment that require immediate attention or intervention [43]. Developing visualization methods that abstract low-level model details while preserving domain semantics is a key challenge, particularly when aiming for consistency and interpretability across heterogeneous application domains.

2.5. TRANSPARENT AI

Explainable AI (XAI) and Transparent AI are closely related but differ in scope and emphasis. Explainable AI (XAI) primarily focuses on the AI model itself. Its goal is to make the internal logic, reasoning processes, and decision mechanisms of AI systems understandable to humans. XAI techniques—such as feature attribution, rule extraction, or surrogate models—aim to answer questions like *how* and *why* a model produced a particular output. The emphasis is largely technical, centering on the interpretability of algorithms and representations.

Transparent AI, by contrast, adopts a *broader socio-technical perspective*. While it may incorporate XAI methods, transparency extends beyond model internals to include:

- **User aspects**: how information is presented, adapted to users' goals, expertise, preferences, cognitive limits, motivation and learning;
- **Domain aspects**: how system behavior and constraints are grounded in the underlying task environment, physical laws, or operational rules;
- **Technology (e.g. model) aspects**: how data sources, automation boundaries, uncertainties, and system limitations are communicated.

Although distinct, these aspects are interconnected within a triadic semiotic framework [40], as illustrated in Figure 1. As argued by Zou and Borst [155], domain transparency—rooted in Ecological Interface Design and Cognitive Work Analysis [135, 134]—provides the foundation for both user- and model-centered perspectives. This is because human and machine actions are equally constrained by the lawful structure of the control problem. In other words, regardless of whether actions are performed by humans or automation, they must comply with the same domain constraints. This so-called ecological perspective has been successfully applied in various domains and fields, such as healthcare, process control, power plants, and aviation (see [19] for an overview). By representing solution spaces (i.e., operational envelopes) that constrain both human and automated behavior, ecological information systems support safe performance monitoring, evaluation of machine actions, and effective human intervention.

Additionally, several key user aspects from a psychological angle need to be considered when designing for transparency to achieve high user acceptance and trust:

1. *User's needs in decision making and learning*. Human decision making involves the ability to understand situations, recognize critical developments, and cope with them appropriately [66]. This requires transparency to provide situation awareness which Endsley [35] conceptualizes in three levels: Level 1 is perception, which requires transparency regarding relevant situational

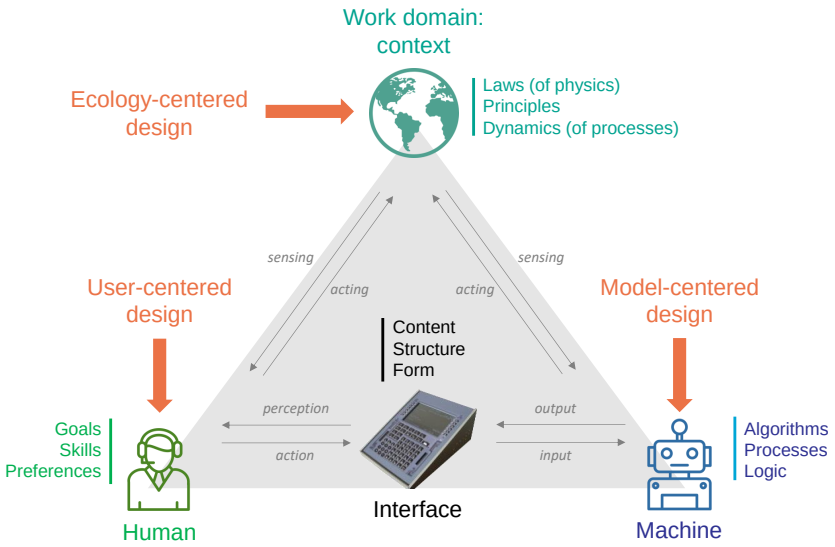

FIGURE 1 - TRIADIC SEMIOTIC PERSPECTIVE ON TRANSPARENCY, CAPTURING USER-, MODEL-, AND ECOLOGY-CENTERED PERSPECTIVES.

cues. Level 2 is comprehension, which requires transparency for an appropriate understanding of the current situation. Level 3 is projection, which requires transparency regarding further developments in the situation. Finally, decision-making includes transparency regarding leverage points and coping strategies.

2. *User's needs with respect to motivation.* Working condition and especially task content influence human motivation [99]. The most important aspects include perceived meaningfulness, an appropriate degree of autonomy, and feedback, with AI ensuring transparency in these areas.

3. *Need for human supervisory control.* Where people are assigned a passive role of supervising AI agents, they are faced with the impossible task of evaluating AI-generated recommendatiomns that they can no longer understand and taking responsibility for them [35]. 'Primitives' [140], AI agents with reduced scope that are easy to understand, can provide intuitive transparency and hence avoid the black box problem.

2.6. SAFE AI

As AI systems become increasingly powerful and pervasive, ensuring their safety has emerged as a fundamental priority. Safe AI refers to the design, development, and deployment of artificial intelligence systems that are reliable, allowing their use in safety-critical scenarios. The safety of AI encompasses several dimensions, beginning with technical robustness. This involves building models that perform reliably under different conditions and provides reliable performances in the presence of noise and adversarial manipulation. Safe AI systems should be thoroughly designed and tested to prevent unexpected or harmful behavior.

In Reinforcement Learning, safety plays a crucial role as we have to decide how to control systems and our action choices should take into account risk and ensure safety. Traditional RL agents are designed to maximize cumulative rewards, often without explicit regard for safety constraints or the potential consequences of risky actions. This limitation has led to the emergence of Safe Reinforcement Learning (Safe RL), a subfield focused on optimizing performance while ensuring that agents behave within acceptable risk boundaries. There are different aspects of safety in RL that can be addressed. Fundamentally, one can investigate safety aspects of the final policy learned, or the safety of the exploration process [45].

When looking at the optimization criteria of the final policy, Safe RL frameworks go beyond optimizing only the expected reward. Instead, they can optimize worst-case performance, a risk-sensitive criterion, or a constrained objective. Examples of risk-sensitive criteria include variance, Value at Risk (VaR), or Conditional Value at Risk (CVaR). These quantify how much risk an agent is exposed to during training and execution. Constraint objectives can be modelled in Constrained Markov Decision Processes (CMDPs) [5] and solved with Lagrangian-based methods that dynamically balance reward maximization with constraint satisfaction. When instead considering the safety of the exploration process, it is important to consider that exploration can lead to unsafe actions. Safe RL methods mitigate this risk through constrained optimization techniques such as shielding, where the agent's policy is optimized subject to safety constraints.

In the applicative scenarios of the AI4REALNET project, such as railway network and power grid control, unsafe actions may lead to cascading failures, service disruptions, or physical damage, making safety constraints non-negotiable. Safe AI techniques enable the learning of control and decision-making policies that explicitly respect operational limits, such as voltage and frequency constraints in power grids or scheduling, signaling, and capacity constraints in railway systems. By formulating these problems within constrained or risk-sensitive RL frameworks, we can leverage CMDPs, CVaR-based objectives, or worst-case optimization to ensure that learned policies remain robust under uncertainty, demand fluctuations, and rare but high-impact events.

3. FORMULATION OF AI APPROACHES FOR NETWORK INFRASTRUCTURES

This section presents the algorithms developed that formulate approaches in transparent and trustworthy knowledge-assisted, hierarchical and distributed AI for network infrastructures. The algorithms are roughly ordered by task: starting with task 2.1 (knowledge-assisted AI), continuing with task 2.2 (decentralized and hierarchical AI), and concluding with task 2.3 (safe, explainable, and transparent AI). However, some contributions go beyond a single task: these are denoted as such.

Each of the described contributions will start with a sketch of the scientific and practical context of the contribution. After that, the method formulation is explained. Each section ends with a section of applicability beyond the studied domain, within or outside of the AI4REALNET project. Where available, references to manuscripts with more detail are provided. Each contribution is accompanied by a source code release described in Deliverable 2.4. The sole exception to this is Section 3.18, which is a purely theoretical analysis.

3.1. PHYSICS INFORMED GRAPH NEURAL NETWORK SOLVER

The critical infrastructures considered in the project are represented as grids or networks. The natural way to consider this structure in AI-based models is through graphs. It allows to take into account the various potential configurations of the grids (e.g., topology changes in power grids) and benefits from graph properties such as permutation invariance. The unique consideration of the data structure may not be enough in certain domains to obtain reliable predictions. In addition, the compliance to underlying constraints should be maintained. In this section, we introduce Physics-Informed Graph Neural Networks enabling the computation of power flows while considering domain specific knowledge i.e., power grid equations (Kirchhoff's law). This work is already published and for more detailed information, the reader could refer to [74]. The Github repository of the package is accessible via `https://github.com/AI4REALNET/T2.1_graph_neural_solver`.

3.1.1. CONTEXT

The reliable operation of modern power grids depends on the continuous monitoring of infrastructures and accurate prediction of the impact of remedial actions before they are applied. Power flow simulations are central to this decision-making process, allowing operators to test different scenarios in a digital environment and ensure system stability. However, conventional simulation methods such as Newton-Raphson, which solve nonlinear physical equations iteratively, are computationally expensive and struggle to meet near real-time and large-scale industrial requirements. This challenge has become more pressing with the integration of renewable energy sources, which introduce uncertainty and variability into the grid. To address these limitations, research has increasingly explored machine learning-based solutions, including graph neural networks, due to their fast inference and ability to model grid topologies efficiently. Yet, despite their speed advantages, purely data-driven methods often face issues related to generalization, physics noncompliance, and dependence on data quality and quantity, making them less reliable for operational use in safety-critical environments like power systems.

This work presents a *knowledge-assisted AI* approach that builds on the recent advances in physics-informed neural networks (PINNs), which combine data-driven learning with physical knowledge to enhance model robustness and compliance with real-world physics. The study aims to systematically analyze the impact of integrating physical constraints into neural network models for power flow simulations, either as regularization terms in standard architectures or through hybrid two-stage approaches where machine learning outputs serve as warm-starts for physics-based solvers. The experimental framework relies on the Learning Industrial Physical Simulation (LIPS) benchmark to evaluate models not only on predictive accuracy but also on physics compliance, generalization to unseen data,

and industrial readiness. The datasets are generated using Direct Current (DC) power flow solvers—a simplified but efficient linearization of Alternating Current (AC) systems—under various grid configurations to mimic operator actions and real-world scenarios. These configurations ensure that the datasets capture realistic variations in power grid behavior while remaining computationally tractable. All experiments are reproducible, providing a standardized foundation for future research on integrating physics-informed machine learning into scalable, real-time power grid management.

3.1.2. METHOD FORMULATION

In this specific study, we consider the message-passing mechanism for both updating the phase angle values and to compute the local conservation error. Phase angle measurement refers to the determination of the phase difference between two electrical quantities, typically voltage and current and is measured in degrees or radians. The other electrical measures like active power can be deduced from phase angles using physics equations. As can be seen in Figure 2, the phase angles are initialized with zeros (flat initalization). Next, the two message-passing layers to update the phase angle and to compute local conservation are interleaved. In the following, we explain the theoretical details of the update step.

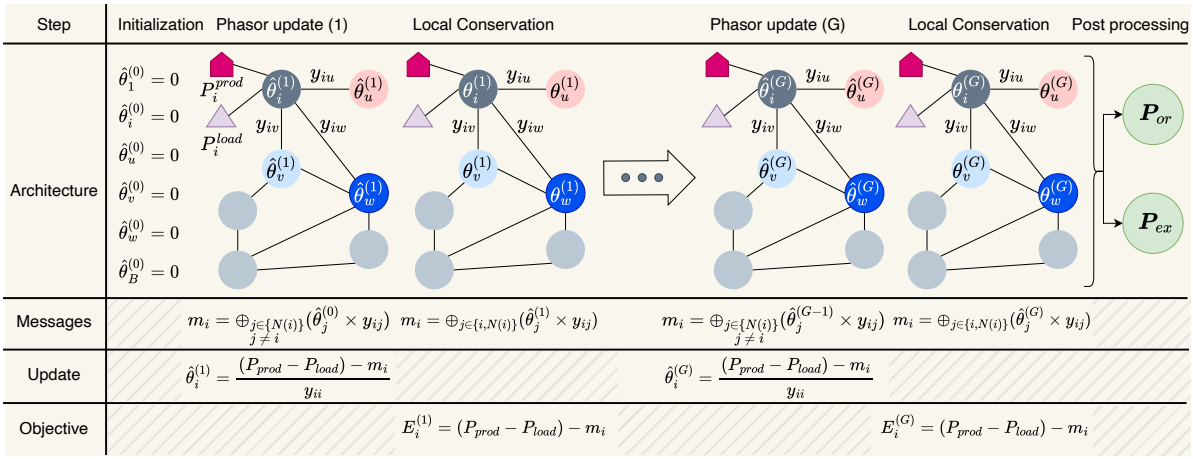

FIGURE 2 - MESSAGE-PASSING AS PHYSICS OPTIMIZATION WITH FLAT INITIALIZATION OF PHASE ANGLES. THE ARCHITECTURE CONSISTS IN INTERLEAVING TWO MESSAGE-PASSING LAYERS TO COMPUTE THE NEW θS (OPTIMIZATION) AND LOCAL CONSERVATION ERROR.

The computation of new phase angle values over message-passing layers is based on the local conservation law formulation, which for a given substation i is given by:

$$p_i^{prod} - p_i^{load} = \sum_{\ell \in N(i)} p_i^{\ell}, \tag{1}$$

where p_i^{ℓ} designates the power flow at a power line ℓ connected to a substation i. This is equivalent

to:

$$p_i^{prod} - p_i^{load} = \sum_{j \in \{i, N(i)\}} \theta_j \times y_{ij}, \tag{2}$$

where θ_j represents the phasor at a neighbor node j and y_{ij} is the admittance between two adjacent nodes i and j which is extracted from the admittance matrix Y. To compute the new phasor values at a node i, by considering $N(i) = \{u, v, w\}$ as its neighbors (see Figure 2), the Equation 2 becomes:

$$p_i^{prod} - p_i^{load} = (\theta_i \times y_{ii}) + \underbrace{(\theta_u \times y_{iu})}_{\text{message from node } u} + \underbrace{(\theta_v \times y_{iv})}_{\text{message from node } v} + \underbrace{(\theta_w \times y_{iw})}_{\text{message from node } w}, \tag{3}$$

where y_{ii} is the admittance at the node i. As can be seen, the messages computed to update the phasor include only the information contained in the neighboring nodes, and not the graph node for which the update should be computed. This could be managed using self-loops in the message-passing mechanism. Finally, the new value of our target, which is the phasor θ at node i and for a layer k, is computed as follows:

$$\theta_i^{(k)} = \frac{(p_i^{prod} - p_i^{load}) - [(\theta_u^{(k-1)} \times y_{iu}) + (\theta_v^{(k-1)} \times y_{iv}) + (\theta_w^{(k-1)} \times y_{iw})]}{y_{ii}}. \tag{4}$$

Finally, the local conservation error can easily be calculated for each layer k and at each node i from Equation (2) as:

$$E_i^{(k)} = p_i^{prod} - p_i^{load} - \sum_{j \in \{i, N(i)\}} \theta_j \times y_{ij}. \tag{5}$$

A possible extension of the above-mentioned approach is to consider a learning paradigm. As can be seen in Figure 3, instead of flat initialization, the phase angles are initialized using an MLP neural network (warm initialization). We expect that this initialization, by introducing learnable parameters, would reduce the required number of iterations (message-passing layers) for convergence. As such, back-propagation takes into account the graph operations when updating the MLP parameters, which would allow the better initialization of phase angles by considering implicitly the physical constraint. In contrast to studies that exploit the GNNs [77], our approach relies on the resolution of physics equations.

3.1.3. APPLICABILITY

The proposed approach, in its current form, is specifically tailored to address the power flow computation problem. Grid operators often require executing a large number of simulations in near real-time to assess the impact of their operational decisions on the power system. The proposed hybrid method accelerates these numerical simulations, thereby supporting fast and informed decision-making in power grid management. While the current implementation is not yet directly applicable to the

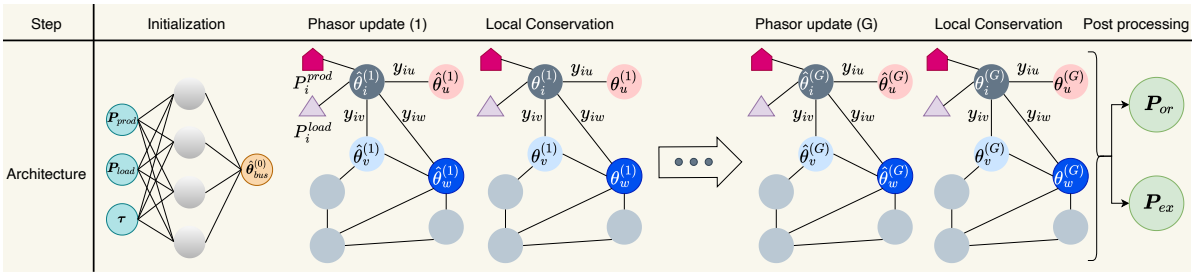

FIGURE 3 - PHYSICS-INFORMED MESSAGE-PASSING WITH WARM INITIALIZATION ($\hat{\theta}^{(0)} = MLP(P_{prod}, P_{load}, \tau; \omega)$). THE MESSAGES, UPDATES AND OBJECTIVES ARE EXACTLY THE SAME AS THE ONE SHOWN IN FIGURE 2

AI4REALNET project use cases, it can serve as a foundation or source of inspiration for developing reinforcement learning (RL) algorithms in scenarios where physical or operational constraints must be strictly satisfied. With suitable adaptations—such as incorporating domain-specific models and constraints—the method could potentially be extended to other safety-critical or physics-informed domains, including transportation networks, water distribution systems, or industrial process control.

3.2. HARNESSING EXPERT KNOWLEDGE IN DEEP RL

In line with Task 2.1 Knowledge-assisted AI, this section presents a contribution focused on augmenting traditional RL algorithms by leveraging an additional source of knowledge provided by domain experts. The specific form of expert knowledge considered in this study—originally introduced and formalized in [86]—is an independent decision-support tool for power grid operators based on a set of rules and conditions. In our approach, this expert knowledge is exploited either during the training of an RL agent or in a pre-training stage to significantly reduce the size of the action space. Moreover, an explainability dashboard is provided to support grid operators in understanding and interpreting the AI-generated recommendations, thus aligning with the objectives of Task 2.3. All contributions described here are available in the following GitHub repositories: `https://github.com/AI4REALNET/T2.1_deep_expert` and `https://github.com/AI4REALNET/T2.3_explainability_dashboard`.

3.2.1. CONTEXT

Reinforcement learning (RL) algorithms are widely used for control problems, particularly in settings where ground-truth labels—commonly required for supervised learning—are unavailable. In such scenarios, an agent interacts continuously with an environment and optimizes its future actions (policy) based on a predefined reward signal. A broad range of RL algorithms has been proposed in the literature, each relying on different strategies [11]. In many modern RL approaches, deep neural networks serve as policy learners due to their ability to handle high-dimensional and continuous observation spaces.

Although these algorithms have demonstrated promising results across various domains with relatively small action spaces, they tend to face limitations when dealing with large action sets and often exhibit instabilities during training. To address this challenge, the approach introduced in [31] proposes embedding actions into a continuous space to enable generalization, while using approximate nearest-neighbor search to achieve sub-linear (logarithmic) lookup complexity.

In this work, we propose leveraging expert knowledge as prior information to further stabilize and guide the learning process of RL algorithms. Two distinct strategies are considered to exploit existing expert knowledge in the power grid domain.

1. **Guiding the RL training using expert-defined zones:** Expert knowledge [86] is used to direct the agent's exploration toward specific zones of the grid in a *knowledge-assisted* approach. This strategy—illustrated in the top dashed rectangle of Figure 4—periodically restricts the agent's exploration to a zone selected by the expert system (highlighted in orange).

2. **Reducing the action space through expert filtering:** The same expert knowledge is used during

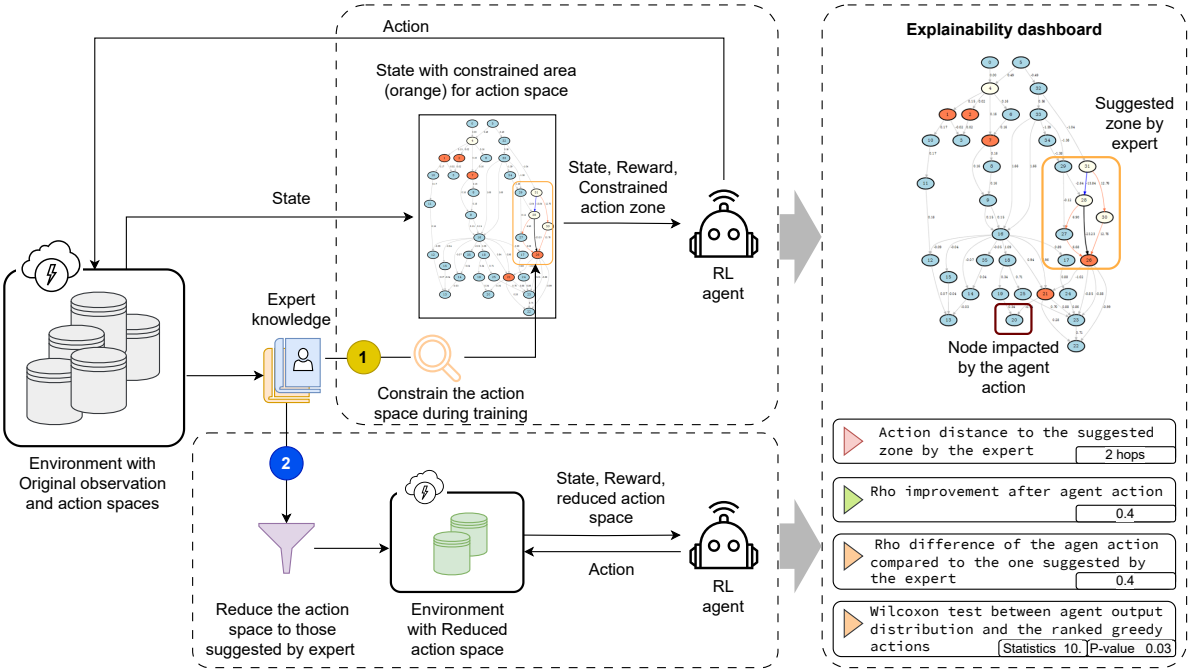

FIGURE 4 - TWO STRATEGIES TO HARNESS THE EXPERT KNOWLEDGE AND AN EXPLAINABILITY DASHBOARD. THE STRATEGIES ARE: (1) THE EXPERT KNOWLEDGE IS USED TO CONSTRAIN THE ACTION ZONE DURING THE TRAINING OF THE AGENT; (2) THE EXPERT KNOWLEDGE IS USED TO FILTER THE WHOLE ACTION SPACE SIZE TO THOSE WITH HIGHEST IMPACT. THE EXPLAINABILITY DASHBOARD GIVES A GRAPHICAL REPRESENTATION OF THE ACTION ALONGSIDE SOME STATISTICS.

preprocessing to limit the action set to those with the highest potential to resolve overload issues (~ 200 actions), instead of the full action space ($66\,811$ possible topological actions for a 36-node grid). This second strategy is illustrated in the bottom dashed rectangle of Figure 4, where the RL agent is trained in a modified environment with a reduced action space.

Finally, to improve interpretability, an *explainability* dashboard displays both the agent's selected action and the zone suggested by the expert on the same graph. Additional statistics are provided to help human operators assess the impact and relevance of recommended actions.

3.2.2. METHOD FORMULATION

This section primarily describes the formulation of both previously mentioned strategies for learning an RL agent while harnessing the expert knowledge. Afterwards, an explainability dashboard integrating a visualization tool and some statistics is proposed.

3.2.2.1 Adapted DeepQ with expert knowledge The original DeepQ algorithm strategy is based on exploration and exploitation strategies [90]. Based on a threshold, the agent may explore new actions a_t for given states s_t and adapt its strategy through time towards maximizing a pre-defined reward. When learning progress, the agent exploits the already learned policy to take best possible actions.

In the first proposed strategy, the original DeepQ algorithm is extended to harness the expert knowledge during the exploration phase of the DeepQ algorithm. The expert knowledge is provided by ExpertOp4Grid framework [86] for power grid domain. It consists in computing an influence graph (called also overload graph) around the overload of interest, and rank the substations and topologies to find a solution. It is mainly based on well-known rules and heuristics in power grid domain, proposes cheap and non-linear topological actions and does not require any training.

Hence, the adapted DeepQ algorithm revises the exploration phase by alternating between whole action space and influence zone (orange zone highlighted on the grid demonstrated in Figure 4). As such, it guides the learning procedure towards more reliable and transparent actions. It allows also to reduce the required iterations for convergence of the model. Algorithm 1 describes the different steps of the adapted strategy, and highlights the adaptation using orange color.

Algorithm 1: Adapted DeepQ algorithm harnessing expert knowledge

Initialize replay memory D to capacity N;
Initialize action-value function Q with random weights θ;
Initialize target action-value function $\hat{Q}$ with weights $\theta^{-} = \theta$;
for episode =$1, \ldots, M$ **do**
  initialize sequence $s_1 = x_1$ and preprocessed sequence $\Phi_1 = \Phi(s_1)$;
  for $t = 1, \ldots, T$ **do**
    ~~With probability ϵ select a random action a_t~~;
    Extract the observation and action space $s'_t, a'_t = ExpertKnowledge(s_t)$;
    With probability ϵ select whether original action space a_t or extracted subspace a'_t;
    With probability ϵ' select a random action in the selected action space;
    Otherwise select $a_t = \max_a Q^*(\Phi(s_t), a; \theta)$;
    Execute action a_t in emulator and observe reward r_t and next observation x_{t+1};
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\Phi_{t+1} = \Phi(s_{t+1})$;
    Store transition $\Phi_t, a_t, r_t, \Phi_{t+1}$ in D;
    Sample random minibatch of transitions $\Phi_t, a_t, r_t, \Phi_{t+1}$ from D ;
    Set $y_j = \begin{cases} r_j & \text{if episode terminates at step } j+1, \\ r_j + \gamma \max_{a'} Q(\Phi_{j+1}, a'; \theta) & \text{otherwise,} \end{cases}$;
    Perform a gradient descent step on $(y_j - Q(\Phi_j, a_j; \theta))^2$ with respect to the parameters θ;
    Every C steps reset $\hat{Q} = Q$;
  end for
end for

One of the drawbacks of this approach may be its scalability in the presence of humongous action space size, as most of the RL-based algorithms. Even if the action space exploration may be constrained during the training, the overall action space remains intact and could cause some problems at inference for policy learners.

3.2.2.2 Generic framework for RL agents harnessing the expert knowledge

To cope with the scalability problem due to the large action space, the action space size may be reduced using a teacher strategy in a pre-processing step, as suggested in [71]. The aim is to extract the most important actions

with desired impacts (reducing the overload), when observing various scenarios and states of the environment. The original teacher implementation uses a greedy search over the whole action space, which could be very time-consuming with large grids. This contribution aims to accelerate the greedy search strategy by harnessing the expert knowledge. This is shown in the bottom dashed rectangle in Figure 4. The use of expert knowledge allows also to produce a set of more contextually relevant actions, and adds transparency to the policy learned by the agent.

Once an RL algorithm (e.g., a PPO) is trained on the reduced action space, a certain number of heuristics are also considered at the inference, which are: (1) reconnect the disconnected power lines whenever it is possible (after a cool-down period); (2) return to the reference topology if it does not impact the grid state. The general workflow of our approach follows the winner solution of IDF competition [101] and is summarized in Algorithm 2.

Algorithm 2: ExpertAgent for Power Grid remedial actions

$A_{red} \leftarrow$ Reduce the action space ($A_{original}$) using Expert Knowledge
PPO $\leftarrow$ Train PPO algorithm on A_{red} obtained in Step 1
$s_t \leftarrow$ Observe a state at time t
$a_t \leftarrow$ Consider DoNothing as the default action when observing s_t
if ANY(Disconnection **IN** s_t) **then**
    s'_{t+1} = Sim($a_{reconnect}$, s_t)
    if $a_{reconnect}$ is possible **and** overload(s'_{t+1}) $<$ overload(s_t) **then**
        $a_t \leftarrow a_t + a_{reconnect}$
    end if
end if
if max_overload(s_t) > $epsilon$ **then**
    $s'_{t+1} \leftarrow$ Sim($a_{recover}$, s_t)
    if overload(s'_{t+1}) $<$ overload(s_t) **then**
        $a_t \leftarrow a_t + a_{recover}$
    else
        $a_{topology} \leftarrow PPO(s_t)$
        s'_{t+1} = Sim($a_{topology}$, s_t)
        if overload(s'_{t+1}) $<$ overload(s_t) **then**
            $a_t \leftarrow a_t + a_{topology}$
        end if
    end if
end if

3.2.2.3 Explainability To understand the RL agent decisions, an explainability dashboard is designed that provides multiple useful information. The right dashed region in Figure 4 shows the scheme of this dashboard with two main sections: (1) a visualization showing an overload graph obtained using the ExpertOp4Grid package and highlighting the highest impact zone near the overload as well as the actual agent's action; (2) some statistics allowing to analyze in more detail the agent's action and its impacts. These statistics are:

- The number of hops that the agent's action is distant from the expert interest zone;

- How much the agent's action allows to reduce the maximum overload on the grid (which is expressed by ρ);
- The difference between the ρ values affected by the action of the actual agent and the best action proposed by the expert;
- The Wilcoxon test with 95% of confidence between the top-k agent's action distribution ranks and their corresponding ranks when using a greedy strategy. If the p-value is less than 0.05, we reject the null hypothesis telling that there is no significant difference between the agent's action and best actions found using the greedy strategy. Hence, the agent's action may not be reliable.

3.2.3. APPLICABILITY

The two strategies proposed in this section are conceptually generic and can be adapted to a wide range of industrial domains, provided that expert knowledge about the underlying infrastructure or operational environment is available. While the illustrative examples and algorithms have been evaluated within the power grid domain—specifically for an AI assistant supporting operators in managing grid congestion—this choice was motivated by the availability of an existing expert-knowledge-based framework for this sector. The proposed strategies, combined with the explainability dashboard, aim to support human operators by providing actionable recommendations and enhancing the decision-making process through transparent and trustworthy insights. Their application to other domains is expected to be relatively straightforward, assuming that comparable expert knowledge can be formalized. Figure 5 highlights potential extensions to other industrial contexts. For instance, if expert knowledge regarding high-impact operational zones exists for railway traffic management or for air traffic management (ATM) sectorization tasks, it could be integrated into the suggested algorithms with minimal adaptation.

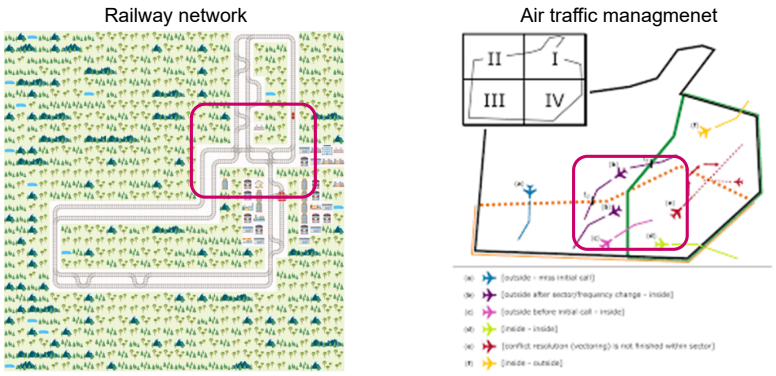

FIGURE 5 - ADAPTATION TO OTHER INDUSTRIAL DOMAINS WHERE SOME EXPERT KNOWLEDGE WOULD BE AVAILABLE (RED ZONES ON THE IMAGES).

3.3. EVOLVING POWER SYSTEM OPERATOR RULES FOR REAL-TIME CONGESTION MANAGEMENT

Within the framework of Task 2.1, *Knowledge-assisted AI*, this section presents a hybrid AI framework for power grid topology control that integrates genetic network programming (GNP), reinforcement learning, and decision trees. A novel variant of GNP is proposed, enabling the evolution of decision-making rules through learning from data within a reinforcement learning paradigm. The graph-based evolutionary representations of both GNP and decision trees support transparent and traceable decision-making, facilitating interpretability and knowledge extraction.

This work was published in the *Energy and AI* journal with the following reference: *F. Moaidi, R.J. Bessa, "Evolving power system operator rules for real-time congestion management," Energy and AI, vol. 23, pp. 100672, Jan. 2026. [https://doi.org/10.1016/j.egyai.2025.100672]*, and its code is available in the AI4REALNET GitHub: `https://github.com/AI4REALNET/GNPDT`.

3.3.1. CONTEXT

Real-time congestion management in power grids, particularly under contingency scenarios, is becoming increasingly complex due to the convergence of emerging challenges, including the variability and uncertainty of renewable energy sources (RES), the rising frequency and intensity of extreme weather events, and the increasing risk of cyberattacks on critical infrastructures. In this demanding context, control room operators must rapidly assess system conditions and define effective remedial actions to maintain grid stability, prevent overloads, and mitigate the risk of cascading failures. These remedial actions may involve generation redispatch, RES curtailment, demand response, and topological reconfiguration of the network. The large number of possible action combinations, coupled with the need for real-time decision-making, makes it extremely challenging to identify optimal solutions, which often requires the use of approximate or heuristic methods to support timely and reliable operations.

This work builds upon the longstanding concept of expert systems (ES) in power grids, traditionally rule-based approaches that integrate expert domain knowledge with physics-based models, and introduces a data-driven methodology to enhance these systems. Specifically, for the real-time congestion management problem, the formalism of Markov decision processes has been adopted. This formalism was established in the Learning to Run a Power Network (L2RPN) competition [85], to allow the augmentation of ES by using RL, enabling the ES to learn and adapt optimal policies from experience and interaction with the environment.

Compared to the state-of-the-art, this work introduces two novel contributions, which are discussed in detail below:

- GNP framework that incorporates dynamic node behavior, enabling context-aware decision-making in uncertain power system environments. Unlike conventional approaches that rely on fixed function nodes, this method employs functional nodes that dynamically adapt their behavior based on real-time system states.
- A hybrid methodology that combines Genetic Network Programming with Decision Tree (GNP-DT), and due to its graph-based structures, enhances interpretability through providing human-understandable reasoning for each control action.

In this work, a novel variant of GNP is proposed that diverges significantly from conventional formulations such as [83], particularly in its structural design and dynamic node functionality. Traditional GNP frameworks typically employ fixed-function nodes and update them in a sequential manner, resulting in rigid decision paths that struggle to adapt in high-dimensional, time-varying environments. In contrast, the proposed GNP architecture features adaptive nodes that can modify their behavior in response to evolving network states and accumulated experience. This flexibility allows the network to represent more sophisticated, state-dependent policies well-suited to the uncertain and non-linear nature of power systems, such as real-time fluctuations from RES or non-linearities inherent in AC power flow.

Crucially, this design enables context-aware decision-making. For instance, an overloaded line may not always trigger the same switching action; instead, the node selects an appropriate control based on additional context, such as nearby contingencies, mimicking expert reasoning under diverse operational scenarios. As training progresses (i.e., learning from data and experience), these evolving heuristics capture expert-like judgment, resulting in a responsive and interpretable control mechanism that adapts in real-time to changing grid conditions. Furthermore, the possibility of seeding at initialization the GNP population with expert-derived decision graphs (e.g., from a pre-existing ES) can accelerate convergence in early learning phases by reducing the need for extensive trial-and-error.

Interpretability is a central contribution of the proposed method, achieved through the integration of graph-based structures. In particular: (a) unlike conventional deep RL approaches that operate as "black-boxes", the GNP framework structures decision-making within an explicit graph-based representation. Each node in the graph corresponds to a programmed decision heuristic, allowing the entire policy to be visualized and traced. This structure enables understanding the rationale behind specific topology actions at any point during learning. In contrast, end-to-end deep RL methods, such as those in [147, 152, 71], often produce uninterpretable decisions encoded in high-dimensional neural parameters, limiting reasoning and reducing trust from grid operators; (b) a multistage decision tree (DT) was employed to extract human-readable rules from the trajectory of actions created by top-performing decision graphs in GNP, which were collected into a pool representing high-quality policy behavior. Separate DTs are subsequently trained to capture the key elements of the control logic. This *hierar-*

chical rule extraction supports a modular interpretation, in which each stage informs and constrains the subsequent one.

3.3.2. METHOD FORMULATION

The core concept of the proposed method involves evolving heuristic control policies, originally derived from operator expertise or pre-existing expert systems, encoded as a decision graph. For power grids, the congestion management task can be modeled as a Markov decision process, defined as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$.

At each timestep t, the agent observes the system state s_t and selects an action a_t according to a predefined policy $\pi(a_t \mid s_t)$. The environment then transits to a new state s_{t+1} based on the dynamics:

$$s_{t+1} \sim P(s_{t+1} \mid s_t, a_t), \quad a_t \sim \pi(a_t \mid s_t) \tag{6}$$

The RL objective is to find the optimal policy π^* that maximizes the expected cumulative reward over an episode of length T:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau\sim\pi} \left[\sum_{t=0}^{T} r(s_t, a_t) \right], \tag{7}$$

where $\tau = (s_0, a_0, s_1, \ldots, s_T)$ denotes a trajectory under policy π, and the expectation $\mathbb{E}_{\tau\sim\pi}$ is over trajectories sampled under the environment dynamics.

As noted in [147, 85], the winning deep reinforcement learning agent of the L2RPN WCCI 2020 Challenge adopted the episode-duration reward formulation, reinforcing its validity as a baseline reward signal for grid control learning tasks. While this formulation encourages robust strategies for prolonging operation, it does not directly account for operational costs, which were only evaluated post hoc. To address this limitation, we use a *cost-aware reward* that incorporates a formal operational cost model inspired by the reward proposed by Grid2Op contributors and closely aligned with the L2RPN scoring metric [156].

The baseline knowledge for this work, as well as for the GNP rule evolution, is the ES developed by RTE [84], which aimed to emulate the decision-making process of human operators. The ES is formulated as a network graph for knowledge representation.

To overcome the limitations of rigid decision-making in traditional GNP, the proposed method integrates an RL-enhanced GNP framework that supports dynamic and context-aware behavior. In this approach, each node is capable of altering its output depending on current grid conditions, such as line overloads, generation/load levels, or switching states. For example, a judgment node that initially selects the reconfiguration action at 'Bus A in zone 1' under moderate congestion may instead recommend reconfiguration at 'Bus C in zone 3' if RES fluctuations create a localized congestion between the two zones. The node's behavior is therefore not statically defined, but instead learns a mapping from

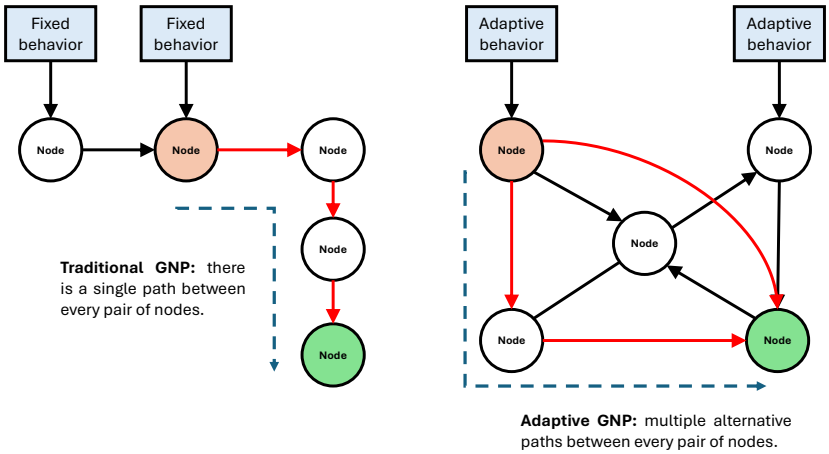

FIGURE 6 - TRADITIONAL GNP VS. ADAPTIVE GNP.

state features to decision criteria (e.g., change in priority of flexible units or threshold of activating a flexible unit), enabling adaptive control that aligns with requirements for real-time management. Furthermore, the execution sequence of the decision graph is not fixed, in contrast to traditional GNP, which follows a hardcoded node traversal. Instead, the proposed method allows the policy graph to activate different substructures conditionally, based on current state inputs. These conditions include the level of congestion, fault locations, substation configurations, and the recent history of control actions. The adaptive traversal mechanism in the proposed GNP supports conditional connections between nodes, especially under time-varying network constraints.

This structural advancement is depicted in Figure 6, which compares the traditional and proposed GNP formulations. On the left, the traditional GNP follows a linear structure in which each node is associated with a fixed function, and execution proceeds through a predefined sequential path, formalized in Equation 8.

$$n_{t+1} = \text{Next}(n_t), \quad a_t = f_{n_t}(s_t) \tag{8}$$

where n_t denotes the node visited at time t, $\text{Next}(n_t)$ is the deterministic next node pointer, s_t is the observed grid state at time t, f_{n_t} is the decision rule implemented at node n_t, and a_t is the resulting action taken by the agent. Each node processes its input and passes control unconditionally to the next node, regardless of the evolving state of the grid. The system response is therefore insensitive to diverse operational contexts, limiting its effectiveness in dynamic environments.

In contrast, the proposed GNP framework (right) organizes the policy graph into an interconnected network of nodes, each with adaptive behavior. The nodes evaluate the state vector, which can include the power flow on the lines, AC power flow constraints, grid topology, and dynamically determine their output. The graph includes multiple possible transitions from each node, with links encoding condition-based execution paths learned through RL. For example, depending on whether the over-

load is localized or widespread, a node might route control to a sub-policy targeting either demand-side response or topological reconfiguration. The arrows between nodes represent these conditional transitions, learned from high-performance decision graphs during training. Node connectivity reflects logical dependencies and execution flexibility, rather than fixed ordering, as expressed in Equation 9.

$$n_{t+1} \sim \pi_{\text{trans}}(n_{t+1} \mid n_t, s_t), \quad a_t = f_{n_t}(s_t) \tag{9}$$

where π_{trans} is a stochastic policy over node transitions.

The system thus constructs control policies by composing rule segments that are most relevant under the current grid state, forming context-specific decision pathways that align with expert behavior while remaining responsive to uncertainty. This structural difference is summarized in Table 1. Thus, in contrast to classic GNP, where node transition probabilities remain static, adaptive GNP dynamically adjusts these transitions based on RL feedback, enabling decision pathways to evolve in response to the agent's performance. In general, the proposed structure enables the GNP policy to behave as a state-driven control mechanism, capable of decomposing complex decisions into subgraphs, dynamically coordinating local actions, and maintaining interpretability for humans.

TABLE 1 - COMPARISON OF TRADITIONAL GNP VS. ADAPTIVE GNP EXECUTION

Component	Traditional GNP	Adaptive GNP
Node execution	$a_t = f_{n_t}(s_t)$	$a_t = f_{n_t}(s_t)$
Node transition	$n_{t+1} = \text{Next}(n_t)$	$n_{t+1} \sim \pi_{\text{trans}}(n_{t+1} \mid n_t, s_t)$
Policy structure	Fixed	Conditional (learned)
Graph behavior	Deterministic	State-driven

The interaction between the GNP evolutionary process and the RL-based adaptive learning is further illustrated in Figure 7, which presents a flowchart summarizing the information flow, feedback loop, and optimization stages within the proposed GNP-RL collaborative framework. Let G_i denote the i-th individual graph in the population, representing a policy μ_i that governs decision-making in a Markov decision process with state space $\mathcal{S}$, action space $\mathcal{A}$, and reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. Each graph consists of *judgment nodes* and *processing nodes*; *judgment nodes* perform condition-based branching, while *processing nodes* issue actions $a_t \in \mathcal{A}$. During interaction with the environment, a graph follows its encoded logic to generate trajectories $\pi_i = (s_0, a_0, r_0, s_1, \dots)$. During execution, each node n maintains heuristic parameters θ_n, which are updated based on local feedback from the environment, typically as a function of the observed state, i.e., $\theta_n \leftarrow f_n(s_t)$. This adaptive mechanism refines node behavior across generations, supporting the learning dynamics. The process continues until a failure condition or episode termination is met. The cumulative reward along π_i defines the fitness F of G_i, Equation 10, which reflects the operational lifespan of the graph under dynamic grid conditions. This formulation inherently prioritizes policies that maintain safe grid operation.

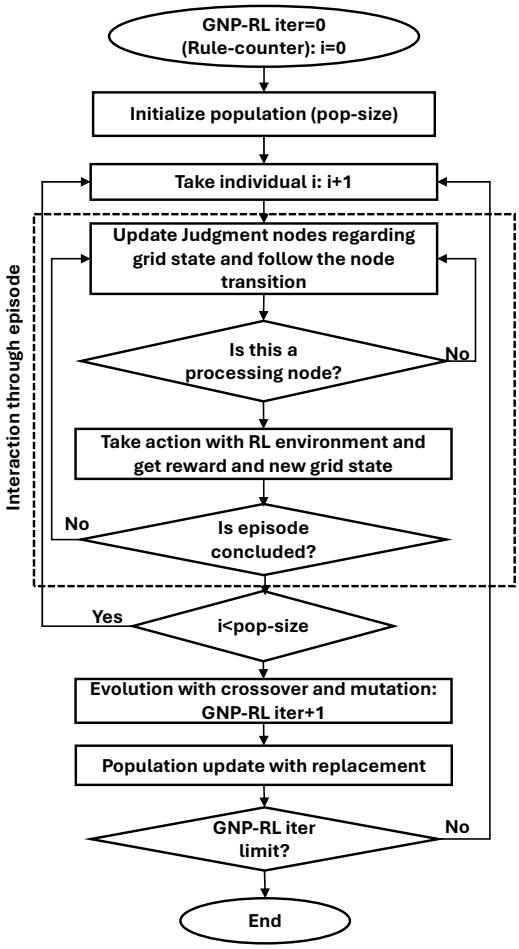

FIGURE 7 - FLOWCHART ILLUSTRATING THE FUSION BETWEEN GNP AND RL, SHOWING HOW THE RL-BASED FEEDBACK UPDATES GUIDE THE EVOLUTIONARY PROCESS TOWARD ADAPTIVE AND OPTIMAL DECISION GRAPH STRUCTURES.

$$F(G_i) = \sum_{t=0}^{T-1} r_{t+1} \tag{10}$$

A two-stage crossover mechanism is employed to balance exploitation and exploration:

1. **Exploitation-driven crossover**: selects both parents P_1, P_2 from the top-performing 50% of the population to preserve high-fitness substructures.

2. **Exploration-driven crossover**: pairs elite individuals (top e_i%) with non-elite ones to introduce novel combinations and maintain diversity.

It has been recognized that a single decision graph may not generalize optimally across diverse grid conditions due to varying fault locations, network topologies, and intertemporal dependencies. Therefore, the elite policies of the GNP algorithm are used to build a rule pool, consisting of the best-performing decision graph for each episode in the last generation. In particular, the elite graph rep-

resents an action selection trajectory that achieves the highest possible performance in an episodic evaluation; this is quite important, as it has optimized the impact of an action at the current grid condition (current timestep) and its subsequent impact in the next time steps. Therefore, elite-derived actions per timestep will be used to generate labeled datasets (i.e., consisting of grid state features and the corresponding actions), in which these datasets are then used to train a multistage DT to capture different components of control logic.

The sequence of DT models is defined as follows:

(i) The first DT model (DT1) to classify the flexibility type (e.g., line reconnection vs. bus reconfiguration);

(ii) The second DT model (DT2) to estimate the required number of flexibility actions;

(iii) The third DT model (DT3) to estimate the specific line(s)/bus(es) to operate.

(iv) The fourth DT model (DT4) will be used only if the detected flexibility type involves bus reconfiguration (bus splitting) regarding DT1. The set of top-k most probable feasible topology reconfiguration actions for the bus identified by DT3 is extracted from DT4, based on the predicted class probabilities at the leaf nodes.

This module is a hierarchical approach in which each stage constrains and informs the subsequent one. For instance, the flexibility type determined in the first stage conditions the search space for the next model, which predicts how many units must be activated, and in turn, this output influences the final model that selects which specific assets to operate. Formally, each stage is modeled as a conditional expectation:

$$\hat{y}_r = \mathbb{E}\left[y_r \mid X, \hat{y}_{r-1}, \hat{y}_{r-2}, \ldots, \hat{y}_{r-k}, \theta\right] + \epsilon_r \tag{11}$$

where $\hat{y}_r$ denotes the predicted decision component at rule stage r, conditioned on the input features X, preceding predicted components $\hat{y}_{r-1}, \hat{y}_{r-2}, \ldots, \hat{y}_{r-k}$, and model parameters θ. The parameter set θ represents the learnable weights of the conditional model (i.e., a DT model), including decision thresholds, node weights, and any coefficients used to combine input features and prior stage predictions. The DT models are constructed for minimizing a suitable loss function (e.g., mean squared error or cross-entropy) over the training episodes, using the observed target outputs y_r. The residual term ϵ_r captures unmodeled variability at stage r. This layered rule extraction allows the DT ensemble to capture interdependencies among rule components, while enabling both interpretability and domain-relevant fidelity.

3.3.3. APPLICABILITY

The methodology described in this section is conceptually domain-agnostic, as it relies on a general framework for evolving, refining, and extracting decision-making logic from expert knowledge through RL and graph-based representations. However, its practical instantiation necessarily builds on domain-specific expert knowledge, typically encoded as an expert system or rule-based controller, which is assumed to be representable as a directed decision graph.

In this work, the algorithms were described and validated in the context of power grid operation, leveraging an existing expert system for real-time congestion management. Nevertheless, the proposed framework can be transferred to other application domains provided that a small set of structural conditions is satisfied.

Firstly, an expert system or expert-derived policy representation must be available, either as an explicit rule base, a decision tree, a flowchart, or a procedural decision logic. Crucially, this expert knowledge must be representable as a graph composed of conditional (judgment) nodes and action (processing) nodes. This requirement is not restrictive, as many operational domains—such as transportation systems, industrial process control, telecommunications, water networks, manufacturing systems, or medical decision support—already rely on structured operational guidelines, standard operating procedures, or heuristic decision trees that can be naturally mapped to such graph-based representations.

Secondly, the control problem must admit a formulation as a Markov decision process. While the specific reward design is inherently domain-dependent, both sparse (survival-based) and dense (cost-aware) reward formulations can be accommodated within the same learning framework.

Thirdly, the decision recommendations produced by the evolved policy must be codifiable into interpretable rule structures, such as decision trees. The final decision tree extraction stage is a key element of the methodology, as it decouples learning from deployment: the reinforcement learning and evolutionary processes are used offline to improve and optimize expert knowledge, while the resulting distilled rules provide a lightweight, transparent, and auditable decision support mechanism suitable for real-time operation.

Importantly, the methodology does not assume that the expert system is optimal or complete. On the contrary, it explicitly treats expert knowledge as a structured prior that can be refined, adapted, and extended through interaction with data and simulated environments. The genetic network programming component enables the exploration of alternative decision paths, the reinforcement learning loop provides performance-driven feedback, and the decision tree extraction consolidates high-performing behaviors into human-readable rules.

3.4. PLANNER ENHANCED AI

In planner-enhanced AI, we leverage the *implicit knowledge* contained within established heuristic planning algorithms (such as collision avoidance rules and priority logic), resulting in a *knowledge-assisted* approach. However, by themselves, scalable planning heuristics can be far from optimal. We aim to improve these methods using a learning component to modify the problem representation. Further technical details, training procedures, and results are available in our workshop paper accepted at the *ICML 2025 Workshop on Programmatic Representations for Agent Learning*.
Source code corresponding to the contribution described here is available in the following repository: `https://github.com/AI4REALNET/flatland-blackbox`.

3.4.1. CONTEXT

Critical infrastructure domains, such as railway networks and automated logistics, rely heavily on the efficient coordination of multiple agents sharing limited resources. This problem is formalized as Multi-Agent Path Finding (MAPF) [125], where the core challenge lies in the trade-off between solution quality (optimality) and computational efficiency (scalability). Classical planners generally fall into two broad categories. First, optimal solvers, such as Conflict-Based Search (CBS) [115], utilize explicit knowledge and rigorous constraints to guarantee minimal flow-time, yet they are NP-hard and fail to scale to the real-time demands of large infrastructure networks. Second, sub-optimal heuristic solvers, like Prioritized Planning (PP) [118], utilize implicit knowledge embedded in greedy heuristics to solve problems quickly in a largely *decentralized* manner. While scalable, these heuristic methods often produce inefficient solutions, such as unnecessary delays, due to a lack of global coordination.
Our method addresses this dichotomy by integrating recent advancements in black-box differentiation and neuro-symbolic AI. Unlike standard “Predict-then-Optimize” frameworks [33], which typically predict real-world parameters to feed into a solver based on accuracy, our method adopts a decision-focused learning approach. We learn to *modify* the problem representation explicitly to guide a sub-optimal solver toward solutions that are closer to those a optimal solver. This contrasts with previous learning-based MAPF approaches that predominantly modify local planner decisions, such as agent prioritization [150] or conflict resolution strategies [59]. By targeting the global graph representation, our method aims to influence the solver’s behavior holistically. This global graph serves as a centralized element, after which the individual agents can make individual plans, which are *decentralized* in as much as they order the agents, which then do not take into account ‘later’ agents, although they do take the paths of ‘earlier’ agents as constraints.
This approach specifically targets the properties of safety, trustworthiness, and scalability. By maintaining the heuristic solver at the core of the decision-making loop, we ensure that all hard constraints,

such as collision avoidance and track validity are met by construction, a guarantee often lacking in purely learning-based methods. The learned component strictly assists in optimizing efficiency without compromising the safety guarantees inherent to the solver's logic.

3.4.2. METHOD FORMULATION

Our method builds on Prioritized Planning [118], a highly scalable, but regularly very sub-optimal planning heuristics for MAPF. Our goal is to induce prioritized planning to present it a modified representation of the original problem instances, where edge traversal costs can be modified to encourage the agent to take or avoid that edge. This modification is to be learned by a neural network. We can thus see this method as a global, differentiable representation learning method.

We define the MAPF instance on an undirected graph $G = (V, E)$ with a set of agents $\mathcal{A}$. A joint plan is considered feasible if it avoids any vertex conflicts or edge conflicts. We define the edge-usage vector $y \in \mathbb{N}^M$, where y_e counts how many times edge e is traversed by any agent. Given a vector of non-negative edge costs w, the total plan cost is defined as $c(w, y) = \sum_{e \in E} w_e y_e$. The objective is to solve the discrete optimization problem $y^\star(w) = \arg\min_{y \in \mathcal{Y}} c(w, y)$, where $\mathcal{Y}$ is the set of all feasible collision-free plans. Prioritized Planning solves this heuristically by assigning a fixed priority order to agents and planning their paths sequentially. Each agent computes a shortest path using A*, treating the paths of higher-priority agents as time-dependent obstacles. This defines a deterministic mapping from edge costs w to a feasible joint plan $y(w)$.

We now aim to train the neural network to output a modified cost vector $\tilde{w}$ for which the resulting plan under PP $y(\tilde{w})$ is closer to an optimal solution. To train a neural network to optimize the cost vector $\tilde{w}$, we must propagate learning signals through the PP algorithm. However, the mapping $\tilde{w} \mapsto y(\tilde{w})$ is piecewise constant, yielding zero gradients almost everywhere. To address this, we employ the black-box differentiation technique proposed by Pogančić et al. [102]. We define the task loss $L(\hat{y}, y^\star)$ as the mean squared error between the predicted plan usage $\hat{y}$ produced by PP and the optimal plan $y^\star$ generated by Explicit Estimation Constrained Based Search (EECBS). We then construct a perturbed cost vector $w' = \tilde{w} + \lambda \frac{\partial L}{\partial \tilde{y}}$ using a scalar $\lambda > 0$. Evaluating PP with this perturbed cost vector yields a perturbed solution $y_\lambda = y(w')$. The surrogate gradient can then be computed as follows:

$$\nabla_{\tilde{w}} f_\lambda(\tilde{w}) = -\frac{1}{\lambda}(\hat{y} - y_\lambda), \tag{12}$$

where $f_\lambda(\tilde{w})$ is a continuous interpolation of the linearized cost function. This gradient effectively signals how the edge weights should be adjusted, by inflating costs on edges that are overused and discounting costs on edges that are underused, in an attempt to encourage improved global flow and guide the heuristic planner to more closely mimic the global patterns of the optimal plan. A neural network $\mathcal{N}_\theta$ (parameterized by θ) is introduced to map instance features x, which include the map

topology, static obstacles, and agent start/goal pairs, into the vector of edge costs $w = \mathcal{N}_\theta(x)$. The gradients obtained via Equation 12 are back-propagated through $\mathcal{N}_\theta$ to update the parameters θ. An overview of the method training diagram can be found in Figure 8.

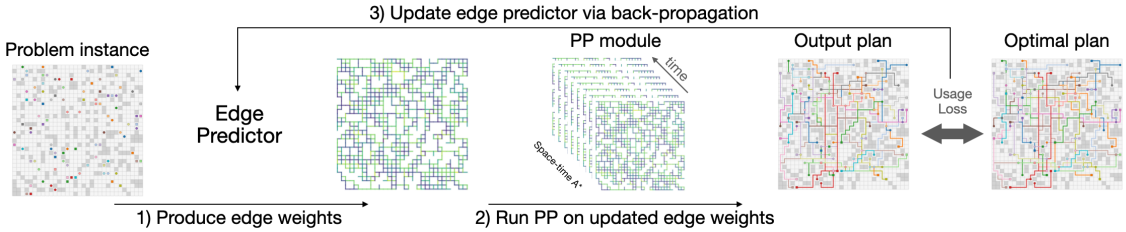

FIGURE 8 - DIFFERENTIABLE MAPF TRAINING FRAMEWORK THAT LEARNS COST ADJUSTMENTS VIA BLACK-BOX GRADIENTS FROM EXPERT PLAN COMPARISONS.

3.4.3. APPLICABILITY

This method is designed for domains where safety constraints are non-negotiable, but operational efficiency is paramount. Within the AI4REALNET project, the primary application is the train domain, through the Flatland environment. The method is suitable for infrastructure capacity planning, where it can optimize the routing of trains through complex station throats or switching networks. By learning from an optimal solver that typically requires significant computation time, the knowledge-assisted system can provide improved scheduling suggestions to operators in real-time.

Beyond the specific scope of this project, the method is applicable to any scenario requiring the simultaneous planning of paths of many agents. As an example, consider the movement planning of fleets of robots in large warehouses. In these scenarios, collision avoidance is often handled by simple local rules (implicit knowledge) due to decentralized execution, but global throughput requires learned coordination to prevent congestion.

Applying this method to real-world critical infrastructure requires specific adaptations. First, the graph topology must be adapted from the grid-based worlds common in academic MAPF benchmarks [125] to the general graph structures of railway networks, which include switches, signals, and block sections. The GNN architecture proposed in our work facilitates this transition, as GNNs are naturally invariant to the underlying graph structure. Second, real-world trains operate in continuous time rather than discrete timesteps. The underlying PP solver would have to therefore be adapted to continuous time MAPF. Finally, while the current method relies on supervised learning from an optimal solver, expert-free learning would be a possible avenue for future work. For extremely large railway networks where generating optimal ground-truth solutions is intractable, the framework could be adapted to use Reinforcement Learning to learn modified representations of the problem, where an objective function such as negative flow-time could serve as the reward signal.

3.5. MAZE-FLATLAND

Efficient railway operations require real-time adaptability to disruptions such as delays or infrastructure failures. At the core of this challenge lies the Vehicle Routing and Scheduling Problem (VRSP), which determines how trains should be dispatched and routed to minimise conflicts, delays, and deadlocks. In practice, this often involves abstracting the problem space, simplifying schedules, routes, or constraints, to enable real-time decision-making within computational limits.

Traditional Operational Research (OR) methods remain standard in real-world applications, however OR are limited by their centralised and iterative nature. In contrast, Reinforcement Learning (RL) offers *decentralised* adaptability through learning-based control. However, existing RL approaches in rail environments often underperform due to poor scalability, unbalanced training, and limited exploration of dispatching behavior.

To address these limitations, we introduce *Maze-Flatland*, a *semi-hierarchical* Multi-Agent Reinforcement Learning (MARL) framework that separates dispatching and routing to improve scalability and coordination in railway operations.

The framework and all associated materials are publicly released to support future research and adaptation under real-world constraints (`https://github.com/enlite-ai/maze-flatland`).

3.5.1. CONTEXT

The VRSP has been widely studied, notably through the *Flatland-RL* challenges. The *Flatland-RL* challenge addresses the VRSP by providing a grid-world simulation of a railway network, where multiple train agents must be coordinated under operational constraints, failures, and dynamic interactions.

In its first edition [69] at NeurIPS 2020, the winning team, *An_Old_Driver*, employed an OR-based approach combining Prioritized Planning (PP), Large Neighbourhood Search (LNS), and Minimum Communication Policy (MCP), integrating replanning and deadlock prevention. On the RL track, the *JBR_HSE* team achieved strong performance by combining tree-based observations with neighbour communication and by training a supervised-learning *scheduler* to dispatch trains based on their estimated success probabilities.

Following these works, [61] introduced a TreeLSTM architecture to encode future paths, allowing agents to anticipate deadlocks and reduce congestion. With extended observation depth, their method achieved performance comparable to OR baselines but severely underutilised available resources in low-density scenarios. More recently, [151] introduced Local Critic Proximal Policy Optimization (LCPPO), extending PPO [111] with local critics under the Centralized-Training-Decentralized-Execution (CTDE) paradigm. Although it improved success rates over previous RL baselines, its scalability remains limited.

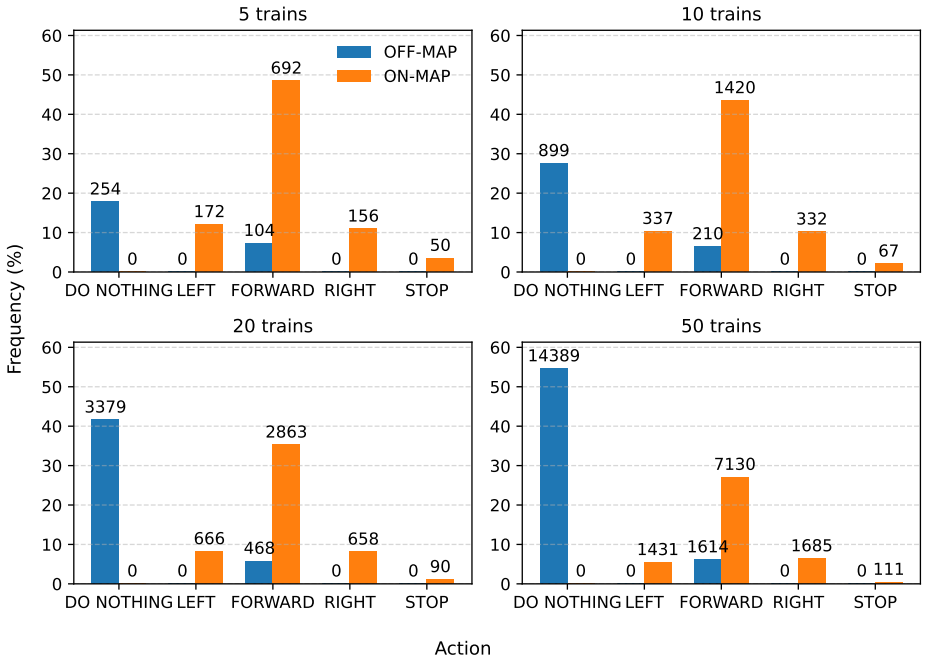

FIGURE 9 - ACTION DISTRIBUTION ACROSS 20 EPISODES FOR A TRAINED AGENT ON RANDOM MAPS WHERE AT LEAST 90% OF TRAINS ARRIVE TO THEIR TARGET.

Despite these advances, challenges persist in coordination, observation design, and dispatch control. Nevertheless, recent MARL frameworks [110] demonstrate that multi agent learning can capture domain specific interactions and provide a promising foundation for realistic and adaptive railway scheduling.

These limitations motivated a closer examination of agent behavior within the *Flatland-RL* environment, a minimal simulation framework for studying train coordination under realistic operational constraints, while abstracting away communication issues.

At each timestep in *Flatland-RL*, each train selects one of five actions: *DO NOTHING* (maintain current state), *STOP* (pause for one timestep), or a movement action: *FW* (forward), *LEFT*, or *RIGHT* (turn). We analysed agent behaviour across multiple scenarios and categorised experience into two classes: *ON-MAP*, when the train is navigating toward its destination, and *OFF-MAP*, when the train remains idle at its origin awaiting dispatch.

Our analysis revealed a strong imbalance between *OFF-MAP* and *ON-MAP* experiences (Figure 9). Since dispatching is implicitly tied to the *FORWARD* action, fewer than 15% of these actions correspond to actual departures, while at higher densities agents increasingly select *DO NOTHING* to delay dispatch under congestion.

This behavioral imbalance highlights a broader limitation of monolithic RL systems in structured environments: agents tend to overfit to frequently encountered sub-tasks while overlooking rare but crit-

ical decisions. In railway scheduling, dispatch timing is exactly such a decision, infrequent yet highly consequential. Thus, exploration of off-map behaviour remains limited, causing the learning process to focus disproportionately on routing. Because most successful RL approaches [69, 61] rely on curriculum learning over increasingly complex instances, this bias compounds across training stages, further suppressing exploration of dispatch behaviour. Overcoming this issue calls for an architectural separation of decision domains. The *semi-hierarchical* framework proposed in the next section introduces distinct state and action spaces for dispatching and routing, isolating their learning signals and enabling focused, task-specific policies.

Maze-Flatland advances learning-based railway scheduling by introducing a two-level policies to learn dispatching and routing independently. Following the intuition from the *JBR_HSE* team, which introduced a dispatcher, we propose a semi-hierarchical architecture that decouples the dispatching decision from the routing. This separation enables each policy to specialise in their sub-task, thereby improving scalability, coordination, and learning stability under dynamic railway conditions.

3.5.2. METHOD FORMULATION

We address the VRSP in railway operations through a semi-hierarchical decentralised MARL formulation that separates decision making into two sublevels: dispatching and routing. The routing policy becomes active only after a train has been dispatched, differing from fully hierarchical RL approaches [30].

The problem is modeled as a Partially Observable Markov Decision Process (POMDP) [62] to overcome the scalability limits of full-state observation, which would otherwise require complete knowledge of the network and all train states. The environment is divided into dispatching and routing subspaces (S_d, S_r), each with its corresponding action set (A_d, A_r). Transitions T and rewards R follow standard Markov Decision Process (MDP) definitions. Agents begin in (S_d, A_d) and, upon a successful dispatch action, transition into (S_r, A_r) for on-map routing. Partial observability restricts each agent to local information, including top-k paths and potential conflicts with nearby trains.

To align with the formal definition and clarify the algorithmic process, each agent has two policies: Multi-Agent Departure Scheduling (**MADS**) for dispatching and Multi-Agent Path Finding (**MAPF**) for routing. The semi-hierarchical MARL framework is outlined in Alg. 3. The environment and the observation spaces for dispatching (Obs_{disp}) and routing (Obs_{rout}) are initialised in lines 1–2, followed by policy initialisation at line 3. For each timestep t_i in an episode, the joint action vector $\mathcal{A}_i$ is computed from the two policies. If an agent has not yet departed, its observation from Obs_{disp} is passed to **MADS** to select an action (lines 7–9); otherwise, Obs_{rout} and **MAPF** are used (lines 10–12). Finally, the environment steps with $\mathcal{A}_i$ (line 15).

Algorithm 3: Maze-Flatland: High-level Semi-Hierarchical interaction for VRSP in rail operations

initialise environment E
initialise observation builders for dispatching (Obs_{disp}) and routing (Obs_{rout})
initialise policies **MADS** (π_{mads}) and **MAPF** (π_{mapf})
for each timestep t_j **do**
    Compute $\mathcal{A}_t \leftarrow \langle a_t^0, a_t^1, \ldots, a_t^n \rangle$
    for each agent α^i **do**
        if α^i is outside of the map **then**
            $s_t^i \leftarrow Obs_{disp}(E)$
            $a^i \leftarrow \pi_{mads}(s_t^i)$
        else
            $s_t^i \leftarrow Obs_{rout}(E)$
            $a^i \leftarrow \pi_{mapf}(s_t^i)$
        end if
    end for
    step E with $\mathcal{A}_i$
end for

ACTION AND OBSERVATION SPACES

The proposed framework defines two custom action–observation spaces for dispatching and routing, as shown in Figure 10. At each timestep, the environment provides local observations to agents, processed by **MADS** for dispatching or **MAPF** for routing. A decision controller triggers actions only when necessary, such as at switches, after which the agent receives a reward.

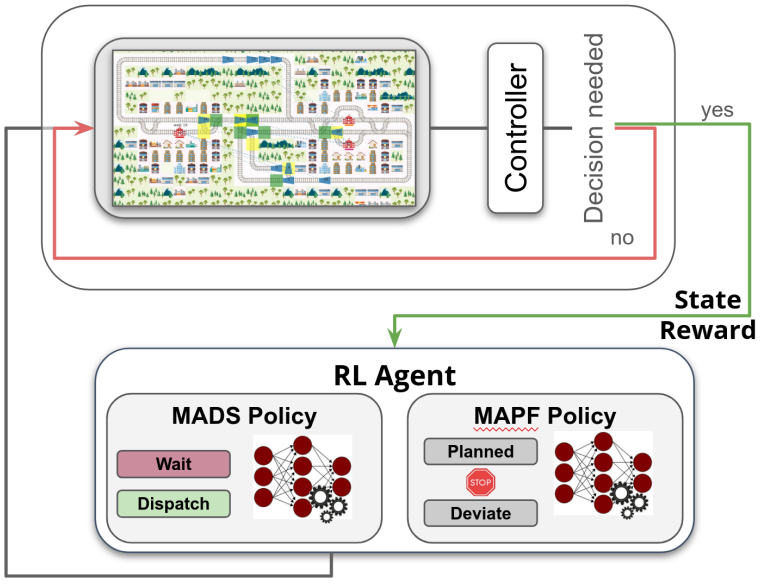

FIGURE 10 - SEMI-HIERARCHICAL CONTROL LOOP WITH DECISION CONTROLLER FOR SKIPPING.

MADS governs train release decisions, choosing between *Dispatch* and *Wait*. Its observations combine global traffic indicators (e.g., map occupancy, remaining time) with conflict estimates derived from the overlap between the top-k paths of active and waiting trains. This enables adaptive scheduling that balances throughput and congestion.

MAPF manages routing once trains are on the map. At switches, agents may *follow*, *deviate*, or *stop*, with deviations triggering replanning along the shortest path. Its observation compresses the original tree-based representation [41] into binary branches (*shortest/alternative*) while retaining minimal features such as distance, cost, and deadlock presence (Figure 11). Agents also perceive the next encountered train along their projected path to support decentralised coordination.

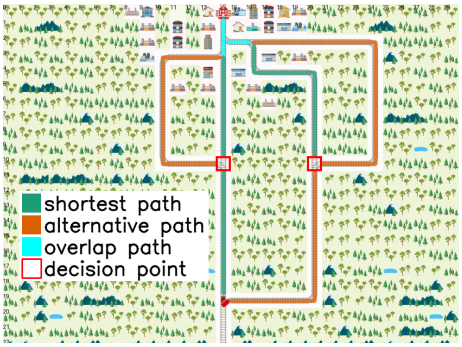

FIGURE 11 - MAPF — REPRESENTATION OF AN OBSERVATION.

3.5.3. APPLICABILITY

Maze-Flatland was designed with the specific use case of *automated re-scheduling in railway operations* [3] in mind. The framework captures the essential features of railway operations, dispatching, routing, and congestion management, making it directly applicable to timetable optimisation, real-time rescheduling, and conflict resolution in simulated or digital-twin railway systems. Beyond rail transportation, the semi-hierarchical formulation and decentralised multi-agent principles are transferable to other networked domains where agents share limited-capacity infrastructure and decision-making alternates between *planning* and *re-planning* phases. Such systems include air-traffic coordination, drone logistics, and urban mobility networks, where traditional RL methods often under-explore infrequent yet critical decisions.

While *Maze-Flatland* illustrates this decomposition within the railway domain, extending it to other domains requires domain knowledge to identify imbalanced decision processes. By isolating such processes and partitioning the decision space accordingly, the same semi-hierarchical principle can enable specialised policies and more efficient learning across diverse systems.

3.6. STATE AND ACTION FACTORIZATION

Addressing control problems with reinforcement learning has proven successful [119, 82], however state-of-the-art algorithms do not scale to realistic environments with a large number of states and actions. In such cases, algorithms are affected by the so-called *curse of dimensionality*, i.e., the amount of data/computation required to achieve a good solution may be out of reach.

Distributed Reinforcement Learning (DRL) can be considered to mitigate this problem by distributing the learning process among multiple agents [149]. In this framework, each agent can observe just a limited part of the state space and take only a small number of actions, but all the agents cooperate to achieve a common goal. The main idea of DRL is thus breaking the complexity of the original RL problem by creating smaller and simpler subproblems.

Source code corresponding to the contribution described here is available in the following repository: `https://github.com/AI4REALNET/dynamic-state-action-factorization`.

3.6.1. CONTEXT

Designing subproblems for DRL is a crucial task that may critically affect the performance of the *decentralized* learning algorithms. Therefore, we propose an original algorithm to obtain a factorization of the state and the action space, presented in more details in [80].

The algorithm is *domain-agnostic*, making it suitable for any complex decision-making problem without requiring prior domain knowledge. It computes correlations between state and action components using mutual information, an information-theoretic measure that quantifies the predictive power of input variables (state or action) regarding future state variable evolution (target variables). By grouping highly correlated state-action pairs, the algorithm creates simpler, potentially independent subproblems that can facilitate distinct learning processes.

Related works include distributed control theory [9] in which, however, the dynamical model of the system is supposed to be known. In our case, no prior knowledge about the system is required and the correlations between variables is entirely based on sampled data. Other related works may include domain-specific analysis, such as segmentation of power grids [87], in which the factorization is based on a specific human intervention and configuration of the grid. On the contrary, our method is meant to be domain-agnostic and widely applicable to any complex decision problem.

3.6.2. METHOD FORMULATION

We consider a *factored* structure for a Markov Decision Process $\mathcal{M}$ (MDP) [103] in which the state and action vectors have components $\mathbf{s} = (s_1, s_2, \ldots, s_n) \in \mathcal{S}$, $\mathbf{a} = (a_1, a_2, \ldots, a_m) \in \mathcal{A}$. Moreover, we

suppose that our MDP $\mathcal{M}$ can be seen as composed of K independent MDPs $\mathcal{M} = (\mathcal{M}_k)_{k=1}^K$, with each one being defined as $\mathcal{M}_k = \left(\mathcal{S}_k, \mathcal{A}_k, P_k, R_k, H\right)$.
In this formulation, each state space $\mathcal{S}_k$ and action space $\mathcal{A}_k$ of the MDP $\mathcal{M}_k$ are taken as a subset of the space $\mathcal{S}$ or $\mathcal{A}$ by combining the domains of only some components of the vector **s** or **a**. The underlying MDPs presents transition probabilities $P_k : \mathcal{S}_k \times \mathcal{A}_k \times \mathcal{S}_k \to [0, 1]$ and the reward functions $R_k : \mathcal{S}_k \times \mathcal{A}_k \to [0, 1]$. The global transition probability and reward function of $\mathcal{M}$ are defined as

$$P(\mathbf{s}'|\mathbf{s}, \mathbf{a}) = \prod_{k=1}^{K} P_k(\mathbf{s}_k{}' \,|\, \mathbf{s}_k, \mathbf{a}_k), \quad R(\mathbf{s}, \mathbf{a}) = f\left(R_1(\mathbf{s}_1, \mathbf{a}_1), R_2(\mathbf{s}_2, \mathbf{a}_2), \ldots, R_K(\mathbf{s}_K, \mathbf{a}_K)\right),$$

where the variables in bold $\mathbf{s}_k \in \mathcal{S}_k, \mathbf{a}_k \in \mathcal{A}_k$ refer to the state/action vectors of the MDP $\mathcal{M}_k$ (not to be confused with the scalar components of the vectors **s**, **a** of the original MDP $\mathcal{M}$). The term $f(\cdot)$ refers to a fixed monotonous function that combines the rewards of all the MDPs. It is easy to show that with this formulation we can consider – without loss of generality – independent policies on each MDP $\mathcal{M}_k$ to optimize the global MDP $\mathcal{M}$.
In order to represent correlations between variables, we can start by defining a fully connected graph $\mathcal{G} = (V, E)$ in which

- V is the set of nodes containing all the state and action components from **s**, **a** and all the next state components from **s**′
- E is the set of edges representing the interactions among components

$$E = \{(x_i, s'_j) \,|\, x_i, s'_j \in V \text{ and } c(x_i, s'_j) \geq \delta\}, \tag{13}$$

  where $c(x_i, s'_j)$ is a metric that measures how much a variable x_i (state or action component) is important to predict the variable s'_j (next state component), with δ being a suitable threshold.

With the above definition, the presence of an edge (x_i, s'_j) in the graph $\mathcal{G}$ means that the component of the next state s'_j can be predicted by the component x_i of the state or action vector, thus x_i and s'_j belong to the same MDP. In this formulation, we can discard the weak connections by means of the threshold δ, and we can obtain either an undirected or a directed graph if the metric is symmetric or asymmetric, respectively. On this graph, we can then run a clustering algorithm to divide the variables into communities $\left(\widehat{\mathcal{S}}_k, \widehat{\mathcal{A}}_k\right)_{k=1}^{\widehat{K}}$.
Our algorithm uses an information-theoretic measure to build the adjacency matrix $\widehat{I}_{\mathcal{G}}$ that describes the correlation among state and action variables, and finds a factorization of the original MDP $\mathcal{M}$.
The first step is to collect a dataset $\mathcal{D}$ from $\mathcal{M}$ with a sufficiently exploratory policy π_e (see, e.g., Mutti et al. [95]). In this dataset $\mathcal{D} = \left\{(\mathbf{s}, \mathbf{a}, \mathbf{s}')_t\right\}_{t=1}^{T}$, each entry is a transition that starts from state **s**, plays an action $\mathbf{a} \sim \pi_e(\cdot \,|\, \mathbf{s})$ and reaches state **s**′.

We define a random variable S' that represents the next state vector and a random variable X that is the concatenation of the state vector and the action vector:
$S' = (S'_1, S'_2, \ldots, S'_n), X = (S_1, S_2, \ldots, S_n, A_1, A_2, \ldots, A_m)$.
Each entry of the dataset $\mathcal{D}$ is a joint realization of X and S'. We can now define the connectivity matrix $\widehat{I}_{\mathcal{G}}$ as a matrix of size $(n \times (n+m))$ having a row for each component of S' and a column for each component of X. Each entry $\widehat{I}_{\mathcal{G}}[i,j]$ is computed based on the Mutual Information (MI) between the component i of the next state vector, S'_i, and the component j of state-action vector, X_j,

$$\widehat{I}_{\mathcal{G}}[i,j] = \begin{cases} 1 & \text{if } \mathrm{MI}(S'_i, X_j) \geq \delta \\ 0 & \text{otherwise} \end{cases}, \quad \text{where } \mathrm{MI}(S'_i, X_j) := \mathbb{E}\left[\log\left(\frac{p(s'_i, x_j)}{p(s'_i)\,p(x_j)}\right)\right]$$

quantifies the amount of information (or, equivalently, reduction in uncertainty) that knowing either variable provides about the other, δ is a suitable threshold. The quantity $\mathrm{MI}(S'_i, X_j)$ cannot be computed exactly as, in practice, we do not have access to those probability distributions, but it is approximated using the dataset $\mathcal{D}$.
At this point, the binary matrix $\widehat{I}_{\mathcal{G}}$ can be transformed into a pseudo-block diagonal matrix, arranging the columns so that variables X_j that have an impact on the same components S'_i are close together. Based on such diagonal blocks, we can define the set of clusters $\left(\widehat{\mathcal{S}}_k, \widehat{\mathcal{A}}_k\right)_{k=1}^{\widehat{K}}$ and run a DRL algorithm on each corresponding MDP.

3.6.3. APPLICABILITY

Our algorithm is domain-agnostic, i.e., it can be applied "as is" to any complex decision making problems. Consequently, it can be applied to all the use cases of the AI4REALNET project. In our work [80], we tested our algorithm on the power grid use case, showing that it is in line with domain-expert analysis (see Figure 12).

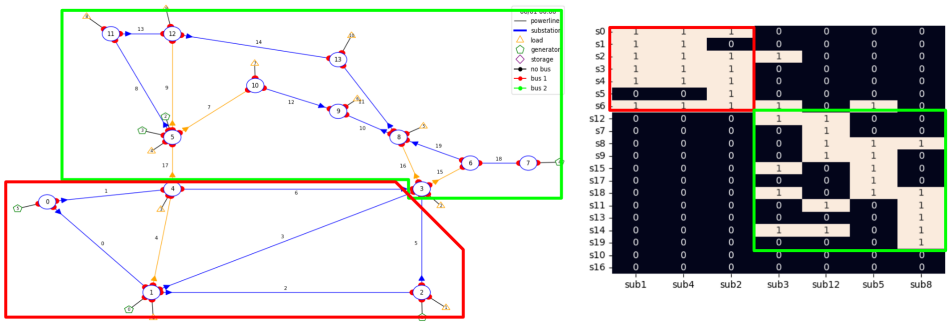

FIGURE 12 - EXAMPLE APPLICATION IN THE POWER GRID DOMAIN.

3.7. NETWORK-DISTRIBUTED Q-LEARNING

We consider the class of control problems where the system can be modeled as a directed graph. In particular, we are interested in problems where the nodes of the graph can be seen as decision points, connected to other decision points by directed edges, along which the consequences of the decisions propagate. In such cases, we want to adapt a state-of-the-art reinforcement learning algorithm, namely Q-learning [141], to make it *decentralized* and exploit the graph structure of the problem.

Source code corresponding to the contribution described here is available in the following repository: `https://github.com/AI4REALNET/network-distributed-q-learning`.

3.7.1. CONTEXT

A wide variety of problems can be modeled as mentioned above, such as the Train Dispatching Problem [68], a special case of the more general Vehicle Rescheduling Problem [76], the real-time control of power grids [63], or the job-shop scheduling problem [143]. Usually, this class of problems presents a high degree of stochasticity, meaning that the environment is affected by random events that are not easily predictable, making planning ahead difficult. To face this challenge, solutions are required to adapt to rapid changes in the environment in real time, while also being able to scale to very large networks.

Classical optimization techniques have been successful because they can provide optimal solutions by planning ahead, but they also present some limitations. In particular, to successfully build the optimization problem one needs to have a full model of the environment which is not always available, especially when dealing with very complex systems with high stochasticity. Moreover, the optimal solution is usually computed before the system starts to operate, and any unpredictable event that disrupts the planned schedule will require a complete replanning, which can quickly become unpractical to do real-time when dealing with very large instances of the problem.

On the other hand, Reinforcement Learning (RL) techniques, and in particular Multi-Agent Reinforcement Learning (MARL), are specifically designed to deal with real-time decision-making in stochastic environments, which makes them promising candidates in this scenario. Although less mature than classical optimization approaches and notoriously difficult to train, RL agents can observe changes or departures from the expected environment behavior, and they can learn to act optimally even in unforeseen circumstances.

Most RL techniques are model-free, meaning that they do not require a model of the environment to work properly, and they can adapt in real time to changes without needing to be retrained from scratch. On the other hand, most of the time, especially in MARL, we are not able to provide theo-

retical guaranties on the optimality of the solution found by the agents as we can do with classical optimization techniques, but the agents may be able to learn empirically optimal policies even when there are stochastic events involved.

3.7.2. METHOD FORMULATION

We first discuss how we represent decision problems on graphs, and then describe our approach for Q-learning on these graphs.

Decision problems on graphs Let's represent our network as a directed graph $G = (N, E)$, where N is the set of decision points (nodes) of the network, and E is the set of connections between the decision points (edges).

Each node $n \in N$ in the network has visibility of the state of the environment up to a certain depth d in the graph, which we will call the *observation depth*. Observation depth d poses a limit on the information that a node can obtain from the environment, and it is a parameter that has to be carefully tuned as it creates a trade-off between the amount of available information and the complexity of the learning process.

As we will see later, it is important that nodes are able to observe the state of their successors in the graph ($d \geq 1$), since they will be affected by the immediate consequences of the decision taken by the node. For instance, if a node is controlling a switch in a railway network, it is important that it is able to observe the state of the tracks that are connected to the switch, in order to avoid collisions between trains, or to avoid sending a train to a track that is already crowded. A bad decision taken from the current node would increase the delay of the train, affecting the performance of the next nodes that will be responsible of routing that train in the future. We will call the set of all the local observations of depth d that a node n can make the *local state space* of the node, and we will denote it as $S_{n,d}$.

The resulting state space of the global Markov Decision Process [MDP,] representing the whole network is the combination of all possible local states of the environment observed by the nodes in the network. Given observation depth d, the generic state s of the network can be represented by the following matrix:

$$s = \begin{bmatrix} \boldsymbol{s}_{d,1} \\ \boldsymbol{s}_{d,2} \\ \vdots \\ \boldsymbol{s}_{d,|N|} \end{bmatrix} \tag{14}$$

where each row $\boldsymbol{s}_{d,j}$ is the local observation of depth d made from node j.

As we mentioned before, each node $n \in N$ in the network is a decision point, that is, the node is required to make a local decision at each time step. Basing on the specific problem we are trying to solve, the set of actions that are available at each decision point may vary. To maintain a general formulation, we will consider that each node $n \in N$ has a set of actions A_n available at each time step. The action space A of the MDP representing the network is once again the combination of the action spaces of all the nodes in the network, such that:

$$a = \begin{bmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \vdots \\ \boldsymbol{a}_{|N|}, \end{bmatrix} \tag{15}$$

with each row $\boldsymbol{a}_j \in A_j$ being the action taken by node j.

Now we can decompose the global MDP into a set of *reduced* instances of that MDP, one for each agent in the network. A *partition* of a set is a grouping of its elements into non-empty subsets, such that every element is included in exactly one subset. After partitioning the network into $|N|$ partitions, one for each node, we can define a notion of *reduced MDP* with respect to each partition. For a generic node $n_j \in N$, we define a new MDP with a smaller state space, which is equal to the space of local node observations of depth d, $S_{j,d}$, and a smaller action space, which is equal to the space of local actions, A_j.

In the case of the global MDP it is often very easy and natural to formulate a reward function in terms of the global objective we are trying to optimize. Furthermore, as all the information of the environment is available at each time step, it is also easy to guide an agent towards the optimal solution.

In the case of a reduced MDP, an agent operating in the reduced MDP has only a partial view of the environment, as it is not able to observe what is happening in the entire network, but only up to a certain depth away from the node making the observation; we can say that the environment is *partially observable* from the point of view of each agent. This means that each agent will never see the effects of its actions on the portion of the network that it cannot observe directly. This makes the formulation of the reward function crucial in guiding the agents towards the optimal solution, as it is the only source of information that potentially gives them a hint about the global effects of their actions.

Q-learning on graphs We propose a scalable, distributed version of the Q-Learning algorithm [141], specifically adapted to this class of problems, which allows agents to learn independently with a minimal communication framework between neighbor nodes.

Let us now consider a generic partition j, and denote with n_j the node associated to that partition. Agent j will be responsible for making decisions in the reduced MDP associated to partition j, given by the state space $S_{j,d}$ and the action space A_j.

At each time step t, agent j will observe the state $s_t \in S_{j,d}$ of the environment, take action $a_t \in A_j$, and observe the reward r_t. From our previous assumptions, the effect of action a_t will be immediately observed by the nodes that are directly connected to node n_j through its outgoing edges. We will denote with $F_j = \{n_1, n_2, \dots, n_{|F_j|}\}$ the set of nodes that are directly connected to node n_j through its outgoing edges. The effect of action a_t will be observed by the nodes in F_j, and they will communicate an estimate of their value function to agent j calculated from their q-table:

$$\hat{v}_i(s_{t+1}) := \max_{a'} q_i(s_{t+1}, a') \quad i = 1, 2, \dots, |F_j|. \tag{16}$$

Agent j receives the estimates of the value functions of the nodes in F_j, and uses them to update its own q-table (instead of the regular update [141]):

$$q_j(s_t, a_t) \leftarrow (1-\alpha) q_j(s_t, a_t) + \alpha \left(r_t + \gamma \sum_{i=1}^{|F_j|} w_i(a_t) \cdot \hat{v}_i(s_{t+1}) \right) \tag{17}$$

$$= (1-\alpha) q_j(s_t, a_t) + \alpha \left(r_t + \gamma \sum_{i=1}^{|F_j|} w_i(a_t) \cdot \max_{a'} q_i(s_{t+1}, a') \right) \tag{18}$$

The weights w_i are normalized and action-dependent, as they are used to give more or less importance to different value function estimates, depending on how much action a_t has affected each node in F_j. The values of the weights w_i are determined by the specific problem we are trying to solve, and they are not hyperparameters of the algorithm.

For instance, if we were to solve the train dispatching problem, we would associate each node to a switch in the railway network, and each edge as a track in the railway connecting two consecutive switches. In this specific case, when a node decides to send a train into a certain track, the effect of that decision will only affect the node that is directly connected to the specific edge associated to that track, therefore, the weight w_i associated to the value function estimate of the node receiving the train will be 1, and all the other weights will be 0, since the decision taken will not affect the other nodes connected to the switch.

If we were to operate a power grid, we could associate each node to an active element of the grid, such as a power plant, and each edge as a power line. Let us assume that the only two available actions in this case are to increase or decrease the power generation of the power plant. If a node decides to

increase its power generation, the effect of that decision will likely equally affect all the nodes that are directly connected to the power plant through the power lines, therefore the weights will be equal to $1/|F_j|$ for all the nodes in F_j.

3.7.3. APPLICABILITY

As mentioned before, the class of problems targeted by the algorithm presented so far is a wide class of decision making problems, in particular any problem that can be modeled as decision making on nodes of a graph. For instance, the train dispatching problem, the vehicle rescheduling problem, the job shop scheduling problem, the traffic flow management problem. The method is applicable "as is", provided that a proper partitioning of the global MDP is selected and an informative reward function is chosen in each reduced MDP.

Concerning the use-cases of AI4REALNET, we tested this method on the railway use case using the Flatland simulator. We observed that agents can empirically achieve the optimal solution in case of deterministic scheduling and in case of limited stochastic malfunctions. An immediate extension of this method can be done for the power grid and the air traffic management use cases.

3.8. COMMUNICATION NETWORK IN MULTI-AGENT RL

3.8.1. CONTEXT

The deployment of autonomous learning systems in complex, safety-critical domains raises fundamental challenges related to coordination, robustness, and trustworthiness. In many real-world applications, decision-making is inherently **multi-agent**, *decentralized*, and subject to strong interdependencies between agents. Classical single-agent reinforcement learning and fully centralized control approaches are ill-suited for such settings, as they either ignore coordination effects or introduce scalability and reliability limitations. Multi-agent reinforcement learning (MARL) provides a principled framework for addressing these challenges, but naive decentralization often leads to unstable learning dynamics, inefficient coordination, and unsafe system behavior.

In our work, the central challenge addressed is how to enable effective and safe coordination among *decentralized* agents, while maintaining scalability, interpretability, and alignment with *trustworthiness requirements*. In particular, the work targets environments where agents operate under partial observability and must resolve conflicts without centralized control, making purely reward-driven learning insufficient. This motivates the integration of structured coordination mechanisms and domain knowledge into the learning process.

The objective of this contribution is to implement and evaluate explicit inter-agent communication within a MARL framework for railway traffic control in the Flatland3 environment. In Flatland, each agent has a discrete set of actions: it may move forward, remain idle, or turn left or right if permitted by the current track configuration. Since all agents operate in a shared space and can block one another, coordination becomes essential for overall system performance. Furthermore, the environment enforces strict movement constraints, ensuring that agents can only take valid actions at any given time and track their position. These features make Flatland particularly suited for evaluating coordination and planning in constrained, partially observable settings.

To evaluate the performance of different tested algorithms we introduce context-dependent metrics: deadlocks, completion rate, score. The design and integration of measurable system-level metrics aligns with the guideline requirement for trustworthy AI, that ethical principles be translated into *verifiable and testable criteria*. By comparing communication-enabled and baseline agents under controlled conditions, this approach contributes to accountability and transparency: it makes trade-offs between performance, safety, and coordination explicit rather than implicit.

Our contribution is the following:

- **Framework establishment.** Implementation of a robust, modular and reproducible baseline that can serve as a foundation for further experimentation, including deep dive analysis of mes-

sages and communication between agents.

- **CommNet integration on PPO and DDDQN.** Implementation of a communication network (CommNet) module for two base policies, PPO and DDDQN.
- **Performance comparison.** Quantitative comparison of each policy with and without CommNet using self-developed context-adjusted metrics, such as deadlock number and arrival rate.

The contribution described here is available in the following GitHub repository:
https://github.com/AI4REALNET/flatland-commnet

3.8.2. METHOD FORMULATION

Baseline models We defined a Flatland3 environment that enables the integration and evaluation of different reinforcement learning algorithms. The focus lies on Dueling Double Deep Q-Network (DDDQN), and Proximal Policy Optimisation (PPO) and their respective integration with a communication network, namely CommNet.

DDDQN combines two foundational ideas from deep reinforcement learning: Double Q-Learning and Dueling Network Architectures. The combined method benefits of both approaches: it uses the dueling architecture to learn better value estimations, and applies double Q-learning updates to reduce overestimation bias. The result is a more stable and accurate value-based RL algorithm.

PPO belongs to the class of actor-critic methods and is widely favored for its simplicity and strong empirical performance. PPO maintains two neural networks: the actor, which parameterizes the policy, and the critic, which estimates the value function. The agent interacts with the environment to collect trajectories, and then uses these trajectories to compute advantage estimates, which are used to update the policy in a direction that increases expected return. PPO alternates between sampling new trajectories from the environment and performing multiple epochs of mini-batch stochastic gradient ascent on the collected data.

The CommNet architecture addresses a key limitation in MARL: the inability of agents to learn differentiable communication protocols end-to-end during training. We integrate the CommNet architecture [126] into both our DDDQN and PPO baselines to facilitate explicit message passing among agents. CommNet enables end-to-end differentiable communication by having each agent broadcast a continuous latent vector that is mean-pooled and re-injected into policy networks. Despite producing non-symbolic messages, this approach has been empirically shown to improve coordination under partial observability: Sukhbaatar et al. report significant gains on cooperative navigation and traffic junction tasks [126]. CommNet's minimal architectural footprint, requiring only a small change in the network, allows its adoption for systematically evaluating communication dynamics in the Flatland3 environment.

Model	Score	Arrival rate	Deadlock
DDDQN	-0.207	0.053	0.002
CommNet-DDDQN	-0.215	0.232	0.011
PPO	-0.117	0.293	0.038
CommNet-PPO	-0.273	0.039	0.008

TABLE 2 - AGGREGATED EVALUATION RESULTS

To evaluate the performance of the considered MARL algorithm and assess whether communication leads to measurable improvements of the models in coordination and task performance, we defined the following KPIs:

- **Deadlock Rate**: The percentage of agents failing to reach their targets getting stuck in deadlocks, facing each other. A lower % rate is better.
- **Arrival Rate**: The percentage of agents successfully reaching their destinations in time. A higher % rate is better.
- **Score**: The cumulative reward per episode, reflecting task completion efficiency, deadlock avoidance and penalty reduction. A higher value is better.

Evaluation results Table 2 summarizes the results across the metrics for all tested algorithms. The comparison between DDDQN and its communication-augmented variant, CommNet-DDDQN, highlights the tangible benefits of inter-agent communication in value-based methods. Although the performance difference in score is marginal and likely within variance, we observed a major improvement in the Completion rate, which shows a more than fourfold increase. This suggests that communication enables agents to better coordinate and avoid conflicts, leading to more successful deliveries. The Deadlock rate increases slightly, which is an acceptable trade-off considering the gain in task completion. This trade-off implies that while communication improves collaboration, it may introduce some complexity that occasionally results in conflict. Nonetheless, CommNet proves especially effective in enhancing coordination in environments where DDDQN alone struggles to scale.
In the PPO-based comparison, the results show a more nuanced picture. Standard PPO achieves the best Score and the highest Completion rate among all tested models, clearly demonstrating its effectiveness in high-level planning and execution. However, the Deadlock rate of PPO is also the highest, indicating a tendency for agents to collide or enter unsolvable configurations during complex episodes. CommNet-PPO, on the other hand, underperforms on both Score and Completion, with only a modest improvement in Deadlock rate. This underwhelming performance could stem from architectural in-

compatibilities between CommNet and PPO, where the added communication layers might disrupt the stable training dynamics of policy-gradient methods. The lower deadlock rate suggests some improvement in safety or coordination, but it comes at a considerable cost to overall task performance. These results highlight that while CommNet enhances coordination in value-based approaches like DDDQN, its integration into actor-critic methods such as PPO may require more careful tuning or structural adaptation to yield benefits.

3.8.3. APPLICABILITY

The Flatland environment is specific for the train scheduling problems and was developed for this particular domain. The combination of CommNet and PPO or DDDQN can be used, however, for different types of applications. Integration of communication in multi-agent reinforcement learning has shown performance improvements in different scenarios and for multiple base models. Specifically CommNet framework was shown to perform well in different applications, from steel manufacturing complex video games [153, 149].

CommNet is a modular and interpretable model and can build a foundation for future oversight, auditing, and explainability mechanisms — the core of trustworthy AI requirements. To adapt it to other scenarios, developers would need to choose the metrics appropriate for their use cases and define the benchmarks to adequately estimate the algorithm's performance.

3.9. MULTICLASS FAILURE PREDICTION

This section presents a comprehensive failure analysis and prediction framework for Reinforcement Learning (RL) agents in power grid topology optimization. This work has been published in [72], where a more detailed description is available. The code is available in the AI4REALNET GitHub: `https://github.com/AI4REALNET/failure_prediction`.

3.9.1. CONTEXT

Optimizing the topology of transmission networks using Deep Reinforcement Learning (DRL) is a promising avenue for autonomous grid management. However, the interpretation of agent behavior—specifically, understanding why an agent survives or fails in specific scenarios—remains a significant challenge. The "black box" nature of DRL agents often obscures the root causes of failure, making it difficult to differentiate between failures caused by stochastic environment conditions (e.g., adversarial attacks, extreme loads) and those caused by poor agent policy.
To address this challenge, this method targets the properties of *Safe AI*, *Trustworthy AI*, *and Explainable AI*. By identifying common failure patterns and predicting them in advance, the system enhances:

- **Safety:** By providing early warnings of imminent grid collapse, allowing for remedial actions (either by the agent or a human operator).
- **Trustworthiness:** By moving beyond simple binary success/fail metrics to a granular understanding of failure modes.
- **Transparency:** By employing clustering to categorize failures into interpretable types (e.g., "Disconnected Power Lines" vs. "Decreased Load Consumption") and using feature importance analysis to highlight critical grid regions.

Theoretical foundations are drawn from unsupervised learning (clustering) for pattern recognition and supervised learning (gradient boosting) for time-series forecasting. While previous work has explored fault prediction in grids using physics-based models or LSTMs for specific failure types, this approach is novel in its specific application to the outcome of topology-based DRL agents, creating a bridge between agent performance and grid physics.

3.9.2. METHOD FORMULATION

The formulation of the algorithm is two-fold, consisting of a descriptive analysis to categorize failure types and a predictive framework to forecast them.

3.9.2.1 Problem Formulation The problem is modeled within the context of the L2RPN (Learning to Run a Power Network) framework. We consider an environment where an agent interacts with a power grid. A failure occurs when the grid constraints (e.g., thermal limits) are violated, leading to a game-over state. The objective is to:

1. **Cluster Failure Types:** Given a dataset of failed episodes, group them into k distinct clusters $C = \{c_1, ..., c_k\}$ based on the grid state at the moment of failure compared to the initial state.

2. **Predict Failure Horizon:** Given an observation obs_t at time t, predict the discrete time-to-failure class $y \in \{\text{Safe}, \text{Fail}_5, \text{Fail}_3, \text{Fail}_1\}$, corresponding to survival or failure in 5, 3, or 1 time-steps respectively.

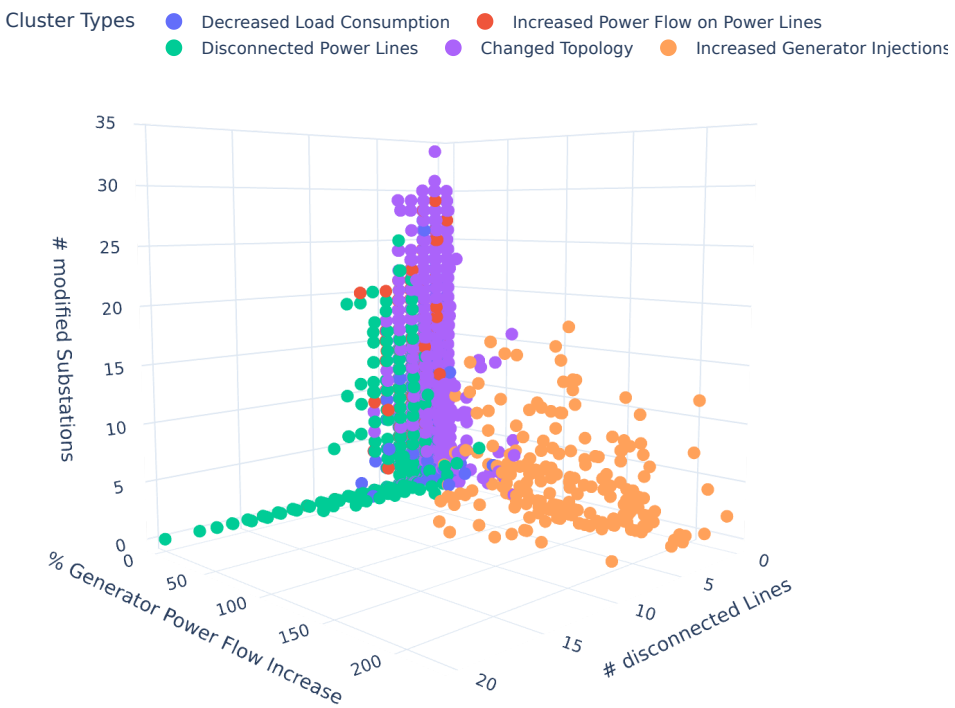

FIGURE 13 - 3D VISUALIZATION OF THE FIVE CLUSTERS, WHERE EACH POINT REPRESENTS A FAILURE OF THE AGENTS.

3.9.2.2 Algorithm Description The method proceeds in two stages:

1. **Descriptive Clustering:** To understand failure modes, data is collected from agent interactions (e.g., observation of line capacity ρ, generator injections, and topology changes). Dimensionality reduction is performed using Principal Component Analysis (PCA) to retain 85% of the variance. Subsequently, k-means clustering is applied. The analysis identifies five distinct failure clusters:

Changed Topology, *Decreased Load Consumption*, *Disconnected Power Lines*, *Increased Generator Injections*, and *Increased Power Flow*, as illustrated in Figure 13. This categorization provides the necessary labels to understand the ”nature” of the threat facing the agent.

2. **Multiclass Forecasting:** For real-time prediction, the problem is framed as a multi-class classification task. The input features include line loadings (ts_{overflow}), topological changes, and temporal data. Several models were evaluated, including Random Forest, XGBoost, and neural networks (CEM, GANDALF). The **Light Gradient-Boosting Machine (LightGBM)** was identified as the optimal model, achieving an accuracy of 82% and a binary accuracy (Safe vs. Failure) of 87%. The model outputs a probability distribution over the defined time horizons, effectively alerting the system to instability up to 25 minutes (5 time-steps) in advance.

   To ensure explainability, we further analyzed the decision-making process of the best forecasting model using feature importance. We utilized the **gain metric** from the LightGBM model, which quantifies the improvement in accuracy contributed by a specific feature (e.g., the loading of a specific line or a generator’s output) to the decision trees. By aggregating these scores, we identified the grid components most critical for predicting failures. Figure 14 visualizes the top 10 most important lines, generators, and loads, highlighting specific sub-regions (A, B, C) that are structurally significant for grid stability.

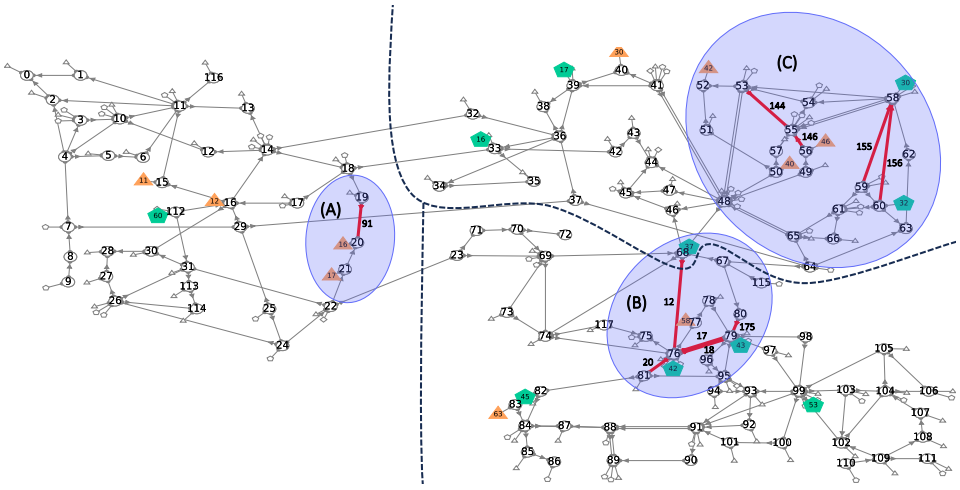

FIGURE 14 - TOP 10 IMPORTANT LINES (RED), GENERATORS (GREEN), AND LOADS (YELLOW) FOR FAILURE PREDICTION. SUB-GRIDS ARE SEPARATED WITH DOTTED LINES. THREE SIGNIFICANT REGIONS (A,B,C) OF IMPORTANT GRID FEATURES ARE HIGHLIGHTED.

3.9.3. APPLICABILITY

This method is primarily applicable to the **Power Grid** domain within the AI4REALNET project, specifically for transmission system operations involving topology optimization (e.g., L2RPN/Grid2Op envi-

ronments). It is designed to act as a supervisory layer for DRL agents.

3.9.3.1 Use Cases:

- **Operator Support:** The prediction module can serve as an early warning system for human operators in control rooms, displaying the probability of failure and the likely time horizon.
- **Agent Training:** The identified clusters can inform "Curriculum Learning," where agents are specifically trained on scenarios known to cause specific types of failures (e.g., focusing training on "Disconnected Power Lines" scenarios).
- **Safe RL Shielding:** The prediction can trigger a fallback policy or a safe-mode controller if the probability of imminent failure crosses a safety threshold.

3.9.3.2 Adaptability to Other Domains: The framework is generic enough to be adapted to other critical infrastructure domains (e.g., **Railway Networks** or **Air Traffic Management**) where "failure" can be defined as a specific state (e.g., deadlock, separation violation) and simulation data is available.

- **As-Is:** The LightGBM and clustering pipeline can be applied directly to tabular data extracted from other environments, provided the features (state variables) are standardized.
- **Adaptations Needed:** The specific feature engineering (e.g., ts_{overflow} for power lines) is domain-specific. For a railway use case, features would need to represent train delays, block occupancies, and switch states. Furthermore, the "survival time" definitions would need to be adjusted to the relevant operational timescales of the new domain.

3.10. SOFT-TARGET IMITATION LEARNING AGENT

3.10.1. CONTEXT

Deep Reinforcement Learning (DRL) has demonstrated significant potential for controlling critical infrastructures, particularly in power grid topology optimization (L2RPN) [85]. However, deploying autonomous agents in safety-critical domains requires high robustness and reliability, properties that standard DRL algorithms (such as PPO or DQN) often struggle to guarantee due to training instabilities and the challenge of exploring vast combinatorial action spaces.

This contribution describes a *knowledge-assisted* approach [52] that addresses the challenge of **robustness** and **sample efficiency**. Standard imitation learning approaches typically utilize "hard" targets (one-hot encoded vectors), where the student agent is trained to mimic the single best action chosen by an expert (teacher). However, in complex network control problems, there is often *multimodality*—meaning multiple distinct actions may lead to similarly good outcomes. Forcing a neural network to commit to a single "correct" action when alternatives are valid can lead to training instability and poor generalization.

The Soft-Target Graph Neural Network (GNN) agent targets the properties of **Knowledge-Assisted AI** and **Safe AI**. By distilling the knowledge of a physics-based greedy search (the teacher) into a Graph Neural Network (student) via a soft probability distribution, the method preserves information about the relative quality of alternative actions. This results in an agent that is not only more robust to unseen grid topologies but also more transparent in its preference ranking compared to black-box RL baselines. Extensive evaluations on the **IEEE 118-bus benchmark (WCCI 2022)** demonstrate that this approach outperforms both the expert teacher and specialized State-of-the-Art (SOTA) DRL agents designed for this challenge [52]. This work has been published at the ECML PKDD 2025 ADS Track [52], where a more detailed description is available. The code is available in the AI4REALNET GitHub: `https://github.com/AI4REALNET/soft_label_gnn`.

3.10.2. METHOD FORMULATION

The core of the method is a Teacher-Student distillation framework designed for discrete action spaces, specifically topology control (line switching) in power grids.

Teacher Agent (Knowledge Source). Instead of a human expert, the method employs a "Greedy" teacher agent. This teacher utilizes an internal physics simulator (e.g., Grid2Op) to simulate all available actions at a given timestep. It evaluates the quality of resulting states based on a utility function, such as the remaining transmission margin or the prevention of cascading failures. While this search is highly effective, it is computationally expensive and unsuitable for real-time inference.

Soft-Target Distillation. Unlike standard behavior cloning, which treats the action with the highest utility as the sole ground truth (Hard Target), this method converts the teacher's utility values into a probability distribution (Soft Target). Let $Q(s, a)$ be the utility of action a in state s calculated by the teacher. The target distribution y is computed using a softmax function with a temperature parameter τ:

$$y(a|s) = \frac{\exp(Q(s,a)/\tau)}{\sum_{a' \in \mathcal{A}} \exp(Q(s,a')/\tau)} \tag{19}$$

This distribution reflects the *uncertainty* and *nuance* of the decision: if multiple actions are effective, the probability mass is distributed among them rather than concentrated on a single arbitrary winner.

Graph-Based State Representation. To effectively process the complex and dynamic topology of power grids, the agent discards fixed-vector representations in favor of a graph-based approach. We model the power grid as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where the construction is component-centric rather than bus-centric:

- **Nodes ($\mathcal{V}$):** We treat every individual grid component—loads, generators, storage units, and crucially, *each end of a transmission line*—as a distinct node. The feature matrix is constructed such that each row corresponds to a grid asset and each column to a specific attribute (e.g., power injections, voltage measurements, cooldown timers, and maintenance status). Features not applicable to a specific component type are zero-padded.

- **Edges ($\mathcal{E}$):** Edges are determined by the electrical connectivity of the grid. This includes connections between components within a substation, if they are connected via a busbar as well as transmission lines linking different substations.

- **Node-Centric Feature Encoding:** A key distinction of our approach is that transmission line features—such as power flow, thermal loading, and voltage limits—are encoded directly into the nodes representing the respective line ends, rather than being treated as edge attributes. This results in a uniform node-based feature representation that captures all relevant grid states while maintaining a simple and efficient graph structure.

The agent utilizes the architecture to perform message passing on this structure. By aggregating information from these component-level nodes, the GNN generates an embedding that is permutation invariant, allowing the agent to generalize to grid topologies not seen during training.

Student Architecture and Training. The student model is a Graph Neural Network (GNN), specifically utilizing a Graph Attention Network (GAT) architecture. The student outputs a policy distribution $\pi_\theta(a|s)$, illustrated in Figure 16. The network is trained to minimize the Cross-Entropy loss (equivalent

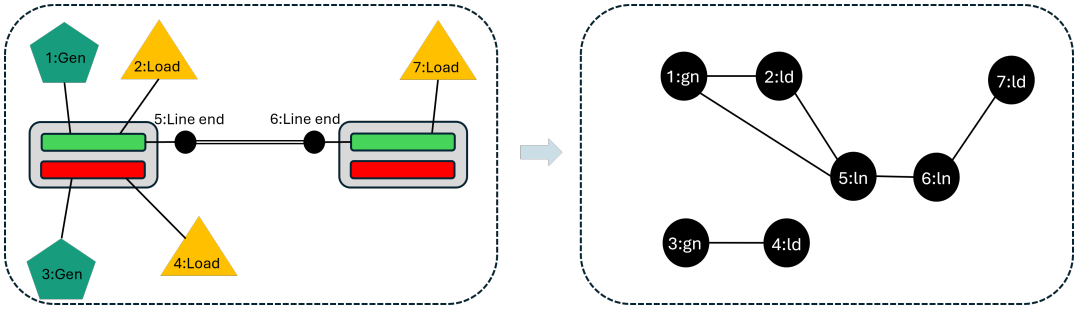

FIGURE 15 - THE RAW GRID OBSERVATION IS TRANSFORMED INTO A COMPONENT-BASED GRAPH WHERE NODES REPRESENT ELECTRICAL ELEMENTS (E.G., LOADS, GENERATORS, LINE ENDS).

to Kullback-Leibler divergence) between the teacher's soft distribution y and the student's predicted distribution π_θ:

$$\mathcal{L}(\theta) = -\sum_{a \in \mathcal{A}} y(a|s) \log \pi_\theta(a|s) \tag{20}$$

By matching the distribution, the agent learns not just "what to do", but the relative safety of all available options, leading to significantly higher survival rates and robustness compared to PPO and standard imitation learning baselines [52].

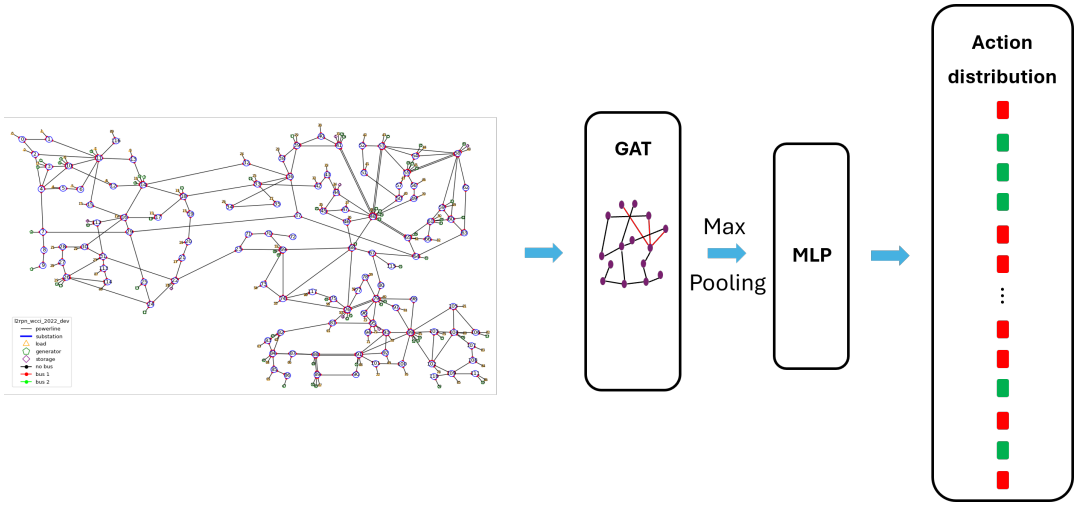

FIGURE 16 - OVERVIEW OF THE GRAPH-BASED SOFT-TARGET AGENT. A GAT MODEL AGGREGATES THE GRAPH STRUCTURE OF THE STATE TO COMPUTE EMBEDDINGS, WHICH ARE USED TO PREDICT THE SOFT PROBABILITY DISTRIBUTION OVER TOPOLOGICAL ACTIONS.

3.10.3. APPLICABILITY

While the Soft-Target GNN agent is primarily designed for and evaluated on the **Power Grid** use case (specifically the IEEE 118 L2RPN WCCI 2022 benchmark). the contributions of this work are twofold and extend to wider domains. It is crucial to distinguish between the generalizability of the *Soft-Target Distillation* framework and the specific utility of the *Graph-Based Architecture*. The core principle of

distilling a teacher's utility function into a soft probability distribution is applicable to any domain characterized by discrete action spaces and the availability of a simulator (Oracle). Unlike the specific GNN architecture, this learning paradigm is not limited to graph-structured data. It is valuable in any control setting where *multimodality* exists—i.e., where multiple distinct actions yield similarly effective outcomes. It is applicable "as is" to domains characterized by:

- **Discrete Action Spaces:** The softmax formulation naturally handles discrete choices (e.g., switching lines on/off, changing integer tap positions).
- **Availability of a Simulator/Oracle:** The method requires a teacher capable of evaluating actions during training (knowledge extraction).

In such cases, enforcing a hard target (one-hot label) introduces arbitrary bias, whereas soft labels preserve the "ranking" of viable options. This approach is particularly suitable for safety-critical control systems, where a "safe" action is preferable to a "perfect but risky" action, or complex logistics, where multiple routes may result in the same delivery time.

Adaptation to other domains. Within AI4REALNET, this method could be adapted for the **Railway Use Case**. If a simulation-based dispatcher (Oracle) exists that can rank potential dispatching commands based on delay reduction, the Soft-Target framework could be used to train a fast neural dispatcher. This would be preferable to hard classification if multiple dispatching schedules yield similar delays. However, adaptations would be needed if the action space becomes continuous or if the oracle is too computationally expensive to query for every training step.
We identify two specific domains outside of power systems where the combination of Soft-Target Imitation and Graph Representations would be immediately valuable. The first is **Urban Traffic Signal Control**. Similar to power grids, traffic networks function as flow-based graphs where intersections (nodes) regulate the flow of vehicles (edges). A simulator can act as the teacher, calculating the delay reduction for all possible phase switches. Soft labels would allow a student agent to learn that multiple phase configurations might clear congestion equally well. The second domain is **Software-Defined Networking (SDN) Routing**. In telecommunication networks, data packets must be routed through specific nodes to avoid bottlenecks. A Soft-Target agent could learn to mimic a centralized controller's optimal paths, treating the network topology as a graph. The soft labels would teach the agent to identify *all* non-congested paths, providing robustness if the primary path fails.

3.11. GRIDEXPLAINER: EXPLAINABLE RL FOR POWER GRIDS

This section describes *GridExplainer*, a comprehensive framework designed to enhance the *transparency and explainability* of Deep Reinforcement Learning (DRL) agents in power grid operations. It addresses the critical need for interpretability in safety-critical infrastructure by providing a modular toolbox for feature attribution, rigorous evaluation, and interactive visualization. The code is available in the AI4REALNET GitHub: `https://github.com/AI4REALNET/GridExplainer`.

3.11.1. CONTEXT

Deep Reinforcement Learning (DRL) has emerged as a promising approach for optimizing topology control and power flow increasingly complex power grids. However, its deployment is hindered by the "black-box" nature of deep neural networks. In critical infrastructure domains like power systems, the lack of transparency raises significant concerns regarding reliability, accountability, and trustworthiness.

To address these challenges, this method targets the properties of **Explainable AI (XAI)**, **Transparency**, and **Trustworthy AI**. The primary motivation is to bridge the gap between the high performance of DRL agents and the operational requirement for interpretable decision-making. By elucidating the specific features (e.g., line loads, generation values) that influence an agent's topological actions, GridExplainer aims to:

- **Facilitate Trust:** Enabling human operators to understand the rationale behind automated interventions and fostering trust among stakeholders and regulatory bodies.
- **Ensure Safety:** Allowing for the verification of agent behavior against physical grid constraints, ensuring that the agent is not exploiting simulation artifacts.
- **Support Debugging:** helping developers identify identifying reasoning flaws or biases in the agent's policy.

The framework is grounded in the theory of post-hoc feature attribution, utilizing established methods such as Shapley values and gradient-based approximations to assign importance scores to input features. It specifically addresses the gap in existing literature where XAI methods are rarely applied or rigorously evaluated within the specific context of power grid topology optimization.

3.11.2. METHOD FORMULATION

The GridExplainer framework is formulated as a modular toolbox tailored for the *Grid2Op* environment. It consists of two main pillars: a suite of feature attribution algorithms and an evaluation protocol.

Feature Attribution Algorithms The core of the method is the generation of attribution maps that quantify the contribution of each input feature (e.g., the loading of line i or the state of substation j) to the agent's chosen action. As evidenced by the implementation, the framework integrates the *Captum* library to support a diverse set of attribution methods, ensuring a comprehensive analysis:

- **Perturbation-Based Methods:**
  - *Feature Permutation [39]:* This method measures feature importance by randomly shuffling individual features within a batch and observing the degradation in the model's output score. A significant drop indicates high importance.
  - *KernelSHAP [81]* A model-agnostic method based on game theory that approximates Shapley values via weighted linear regression, treating features as players in a coalition.
- **Surrogate-Based Methods:**
  - *LIME (Local Interpretable Model-agnostic Explanations) [104]:* LIME explains individual predictions by training an interpretable surrogate model (e.g., a sparse linear regressor) on perturbed samples in the local neighborhood of the input. The coefficients of this surrogate model serve as the feature attributions.
- **Gradient-Based Methods:**
  - *Saliency [121]:* The baseline gradient approach that computes the gradient of the output with respect to the input features. It identifies features that, if changed slightly, would most affect the model's prediction.
  - *Input $\times$ Gradient [117]:* An extension of Saliency that mitigates the gradient saturation problem by multiplying the gradient by the input feature value itself.
  - *Integrated Gradients (IG) [127]:* IG provides an axiomatic attribution by integrating the gradients along a linear path from a baseline input (e.g., a zero vector) to the actual input. This satisfies the axiom of *Completeness*, ensuring the attributions sum up to the difference between the model's output at the input and the baseline.

Evaluation Metrics To ensure the explanations themselves are reliable, the framework formulates a quantitative evaluation based on specific XAI metrics :

- **Faithfulness:** Assesses whether the feature importance scores provided by an explanation method accurately reflect the model's decision-making process. It operates on the assumption that if a feature is truly important, removing or altering it should significantly affect the model's output.

For example, the *Faithfulness Estimate* (FE) metric systematically masks or perturb features identified as important and measures the impact on the prediction [6]:

$$FE = \sum_{i=1}^{N} |f(x_i) - f(x_i^{\setminus j})| \tag{21}$$

Where $f(x_i)$ is the model prediction for the original input x_i, and $f(x_i^{\setminus j})$ is the prediction when feature j is removed or masked. This metric ensures that AI-driven explanations remain reliable and aligned with the actual decision-making process.

- **Robustness:** This evaluates the stability of explanations against small input perturbations, under the assumption that the model's output remains relatively unchanged. A robust explanation should not fluctuate significantly when the input is slightly modified. An example metric is the *Average Sensitivity* (S), estimated via Monte Carlo sampling [146]:

$$S = \mathbb{E}_{\delta \sim D}[||\Phi(f, x) - \Phi(f, x + \delta)||^p]^{1/p} \tag{22}$$

Where $\Phi(f, x)$ represents the explanation for the original input, $\Phi(f, x + \delta)$ is the explanation for the perturbed input (with noise δ sampled from distribution D), and $|| \cdot ||^p$ denotes the p-norm distance (e.g., L2). This ensures the interpretability method is not sensitive to insignificant noise, a critical requirement for safety-critical applications.

- **Complexity:** Indicates how concise the explanations are, i.e. how few features are used to explain a model prediction. This is based on the assumption that it is too difficult for users to understand an explanation if the number of features used in the explanation is too large. One way of measuring the conciseness is through the *Entropy* of the attribution distribution [18]. Lower entropy indicates a sparser, less complex explanation that focuses on fewer features, which is generally preferred for human interpretation.

- **Randomization:** Assesses the validity of the method by comparing explanations of the trained model against those of an initialized (randomized) model. One way of measuring the effect is through *Model Parameter Randomization Test* [1, 53] which randomises the parameters of single model layers and measures the distance of the explanation (or their respective entropies [53]) to the original explanation.

Visualization The formulation includes an extension to the *grid2viz* tool. This integrates the calculated attribution values directly into the graphical representation of the power grid, allowing operators to visualize "hotspots" or critical grid elements that drove a specific topological action. These hotspots can be for instance identified by a feature attribution method or any other explainability methods but

are not limited to that. An example plot is shown in Fig. 17.

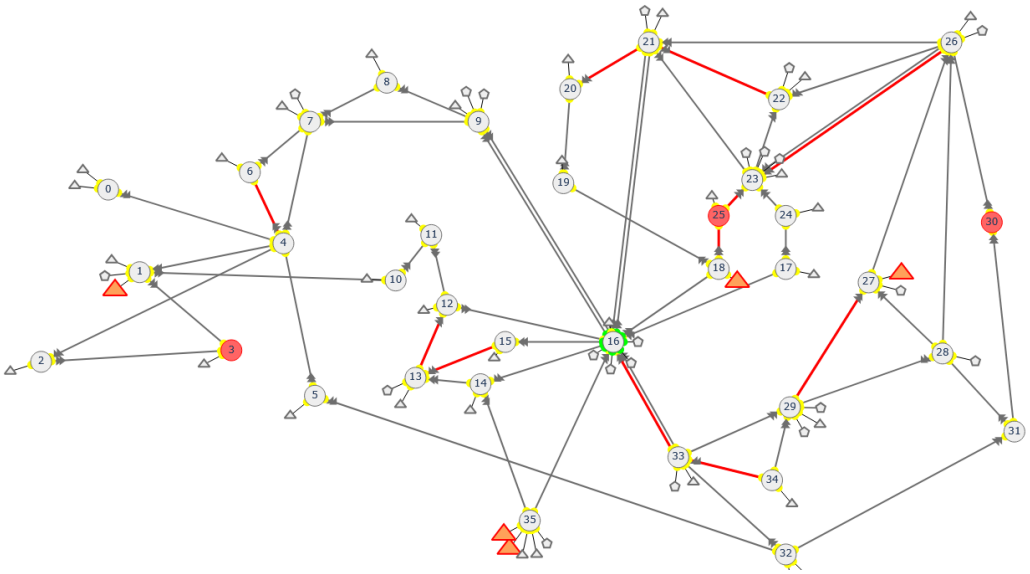

FIGURE 17 - EXAMPLE PLOT THAT HIGHLIGHTS CRITICAL AREAS SUCH AS LINES AND LOADS IN THE POWER GRID.

3.11.3. APPLICABILITY

The GridExplainer framework is explicitly designed for the **Power Systems** domain, specifically for Transmission System Operators (TSOs) utilizing the L2RPN (Learning to Run a Power Network) / *Grid2Op* ecosystem.

Use Cases within AI4REALNET:

- **Post-Hoc Analysis:** Analyzing critical failures or unexpected maneuvers during agent training to refine reward functions or observation spaces.
- **Real-Time Decision Support:** Providing "sanity checks// for human operators when an RL agent proposes a complex topology change (e.g., bus splitting) by highlighting the lines causing the congestion that necessitated the action.

Adaptability to Other Domains: While the implementation is tightly coupled with *Grid2Op* data structures (observation vectors), the underlying methodology is domain-agnostic.

- **Adaptations Needed:** The attribution engine (wrapping *Captum*) can be applied to any domain using PyTorch-based DRL agents (e.g., Railway or Air Traffic Management) provided the input features are tabular or grid-like. However, the visualization layer is specific to power grids

and would require complete redevelopment for other infrastructures (e.g., visualizing track segments for railways).

- **As-Is:** The evaluation metrics (Faithfulness, Robustness) are mathematical properties of the explanations and can be applied "as-is" to benchmark XAI methods in any other domain, including all AI4REALNET use cases.

3.12. TRACERL FOR INTERPRETABLE & INTERACTIVE ANALYSIS

As Artificial Intelligence (AI) systems increasingly operate in complex and safety-critical environments, the need for interpretable and human-centered decision-making systems has become central. While Reinforcement Learning (RL) offers powerful autonomous capabilities, its decision processes often remain opaque and difficult for humans to audit or understand. On one hand, traditional RL approaches tend to focus primarily on performance, often disregarding the need for transparent and explainable decision-making. On the other hand, research in Explainable Artificial Intelligence (XAI) typically emphasises interpretability and transparency, sometimes at the expense of performance and scalability. To bridge the gap between agent usability and explainability, this section introduces *TraceRL*, an interactive visualisation and analysis interface that operates on top of an *Agent-as-a-Service* (*A3S*) backend. Through trajectory visualisation and counterfactual simulation, it allows users to explore how an agent acts, test alternative decisions at selected timesteps, and observe the resulting effects on environment evolution and KPIs. In the current implementation, the backend exposes a single next-action recommendation per step; alternative choices are explored through explicit human overrides followed by simulation.

Source code corresponding to the contribution described here is available in the following repository: `https://github.com/AI4REALNET/agent-as-a-service-trace-rl`.

3.12.1. CONTEXT

Integrating AI into real-world systems necessitates understanding AI choices to ensure human trust. Without it, AI decisions may be disregarded [2]. Additionally, attention to AI ethics and regulation have grown exponentially in recent years, making XAI a core component of AI solutions [8]. Consequently, the XAI field is gaining significant traction. XAI focuses on explaining the behaviour of AI models. For instance, XAI methods can be used to detect bugs in trained agents and to identify hidden biases.

Explainability can take different forms. Models can be intrinsically explainable (e.g. decision trees), transparent (e.g. linear regression), or inherently interpretable (e.g. rule-based systems) [10, 26, 89]. By contrast, models such as deep neural networks are not intrinsically explainable and require post-hoc methods to ensure transparency and interpretability [10, 26].

Generally, explainability mechanisms need to be integrated from the beginning of system design to enable continuous observation, interpretation, and human feedback throughout development. When treated as an add-on rather than a core design principle, such mechanisms often become non-reusable and difficult to maintain. Moreover, when integrated too late in the development process, it becomes challenging to ensure transparency, traceability, and user trust, as the system's internal logic may already be opaque or tightly coupled to non-interpretable components. To mitigate these challenges,

we introduce *TraceRL*, a framework for explainability and human-AI interaction that operates through trajectory exploration, "what-if" scenario analysis, and interactive simulation via the *A3S* backend. This approach goes beyond the single static explanations that conventional models provide, enabling humans to explore multiple solutions and better understand the agent's decision-making process.

TraceRL serves as a bridge between AI agents and human users, enabling *transparent* inspection, interactive evaluation, and safe experimentation. The following sections detail its architecture, core components, and how it operationalises *explainability* within real and simulated environments.

3.12.2. METHOD FORMULATION

TraceRL is both model- and environment-agnostic, designed to operate between the human user and the agent–environment system. *TraceRL* enables users to unroll and visualise the agent's decision-making process step by step, providing an interpretable view of how actions evolve over time and how they contribute to the goal achievement.

TraceRL consists of two main components: (i) a visualisation platform that loads pre-computed trajectories and allows single- or multi-agent rollouts to be explored, and (ii) an *A3S* backend hosting a live instance of the agent, a digital environment, and any additional services required for monitoring, simulation, or analysis. The *A3S* backend returns a single recommended action for the current state; TraceRL enables users to evaluate alternatives by overriding that action and requesting new simulated rollouts.

The visualisation platform enables users to load and inspect previously stored trajectories. Users can navigate a dynamic graph that represents the sequential interaction between an AI agent and its environment. Alongside the rendered state, the platform shows the actions that transition the agent between states, together with Key Performance Indicators (KPIs) and logged events generated during past rollouts.

An example of the visual interface is shown in Figure 18, where a trajectory with five trains in Flatland-RL is rendered. The left side of the interface hosts the dynamic, navigable graph, while the right side displays statistics and the corresponding train actions for the selected timestep.

Trajectories are recorded using a dedicated wrapper, based on the OpenAI gym [20] and the maze-rl [37] interfaces, that ensures a standardised format for storing agent–environment interactions. This wrapper, in combination with the visualisation platform, is designed to support single-agent and multi-agent decision-making processes, and can accommodate search-based algorithms (e.g., Monte Carlo Tree Search or beam search) provided they expose trajectories in the same logging format.

While the visualisation platform enables the rendering and exploration of recorded trajectories, *TraceRL* goes beyond static analysis by supporting live communication between humans and the agent–environment system, providing a deeper level of explainability through alternative scenarios. This

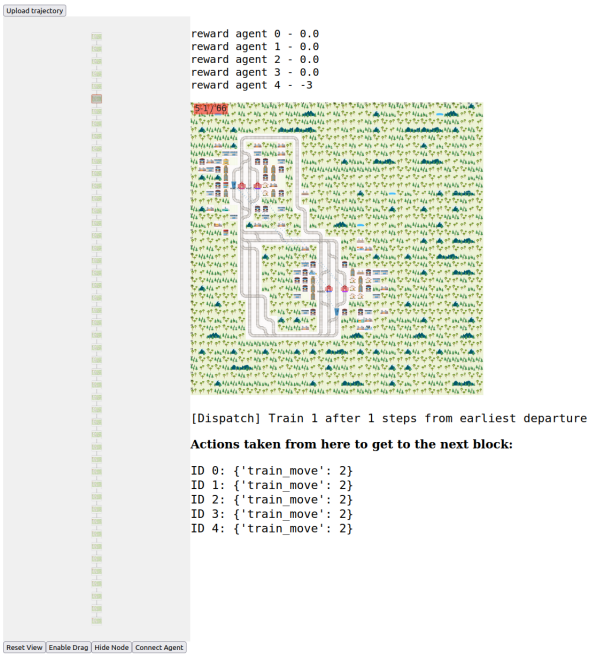

FIGURE 18 - *TRACERL* - SCREENSHOT OF THE USER INTERFACE.

bidirectional communication is central to the framework's design, allowing users not only to observe but also to interact with the agent to gain deeper insight into its behaviour.

The human interacts with the framework by focusing on specific decision steps and providing alternative choices to the original agent's action. In response, the *Agent-as-a-Service* backend parses the human input through its endpoints and leverages the agent–environment loop to return the resulting state and metrics to the user. Finally, the visualisation platform updates dynamically to reflect the new information. This exchange connects human judgement with autonomous reasoning, enabling interpretability and repeatable "what-if" exploration through simulation.

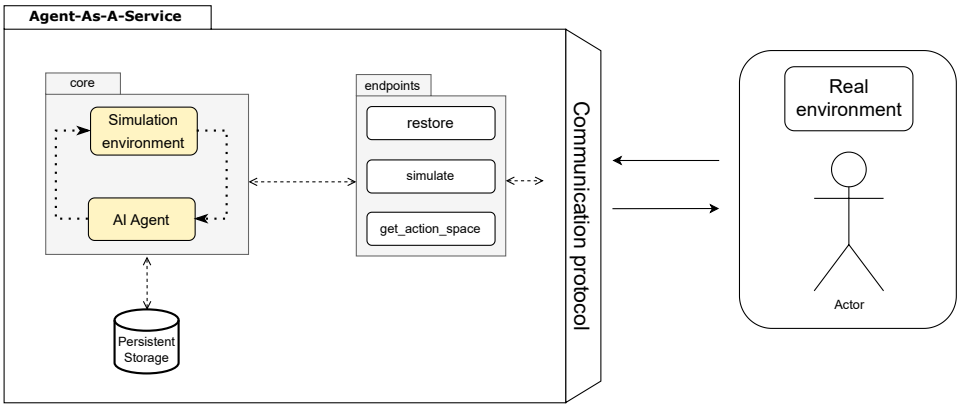

FIGURE 19 - *TRACERL* - CONCEPTUAL ARCHITECTURE.

Figure 19 illustrates the conceptual architecture of *TraceRL*, showing the *Agent-as-a-Service* layer linking human users, the agent–environment loop, and persistent storage through defined communication endpoints. The three endpoints enable users to query the action space (*get_action_space*), run simulations from a given state with or without human input (*simulate*), and restore specific states (*restore*).

(a) INITIAL STATE OF THE TRAJECTORY. **(b) ACTION CONTROL PANEL FOR HUMAN INPUT.** **(c) UPDATED GRAPH.**

FIGURE 20 - *TRACERL* - EXAMPLE OF HUMAN–AGENT INTERACTION IN *TRACERL*.

Figure 20 illustrates an example of human input in *TraceRL*. First, a trajectory is selected and visualised. The user then selects a decision block to override one or more actions (block highlighted in red at Fig. 20a). Subsequently, the system queries the agent for the available actions and presents the user with a control panel (Fig. 20b). After the user selects an alternative action to explore, the *Agent-as-a-Service* backend processes the input, runs a simulation and returns the new trajectory. Finally, *TraceRL* updates the live graph accordingly (Fig. 20c).

The human-influenced cells are shown with a light-blue background, whereas the AI-generated actions produced in response to human input are highlighted in light yellow. The decision blocks from the original trajectory retain a neutral background.

The modular design makes *TraceRL* a flexible foundation for testing, auditing, and explaining RL systems in both simulated and real-world settings. The following section outlines its applicability across different domains and scenarios where interpretability, safety, and human–AI interaction are essential.

3.12.3. APPLICABILITY

TraceRL is designed to be both agent- and domain-independent; however, some requirements must be met. While it is not built for a specific environment, *TraceRL* currently supports environments defined through the Gym [20] or maze-rl [37] interface. This ensures compatibility with a wide range of reinforcement learning frameworks that follow the standard agent–environment interaction pattern, where the agent receives an observation, selects an action, and receives a reward and new state from the environment.

By relying on the Gym API, *TraceRL* can directly interface with most commonly used research environments that follow the standard agent–environment interaction protocol. Although designed to be general-purpose, *TraceRL* currently supports only discrete human input. This means that user interventions, such as providing overriding agent actions or selecting alternatives, must be represented as discrete choices rather than continuous values. The extension to continuous action spaces will be addressed in the future.

The Gym interface also allows developers to integrate custom environments with minimal adaptation, provided they implement the standard methods `reset()`, `step()`, and `render()`. For environments that do not conform to the Gym standard, integration remains possible through an adapter layer that translates between the native API and the expected agent–environment communication protocol. This design maintains flexibility while ensuring interoperability across a wide range of reinforcement learning frameworks.

Beyond domain applicability, *TraceRL* serves as a practical tool for analysing, debugging, and improving agent behaviour. It supports behaviour inspection and replay to understand why agents act as they do, as well as safety validation through simulation of high-risk scenarios before deployment. The framework also facilitates policy comparison and counterfactual reasoning, enabling users to test alternative strategies or interventions and observe their causal impact. Moreover, *TraceRL* could be extended to log human feedback for preference learning and calibration, helping align agent policies with human expectations. These capabilities make it suitable not only for research and development but also for operator training, audit support, and safe experimentation in complex real-world systems.

3.13. EXPLAINER FOR ACTION ALTERNATIVES

When making decisions, operators must consider what long-term outcomes different candidate actions lead to, and how desirable these outcomes are. In a large networked system, this reasoning is especially complicated as the effect of actions propagates across the system and through time. To address this, we introduce an AI that explains what the expected outcomes of actions will be, simplifying the reasoning process, and allowing operators to focus on evaluating which actions have the best outcomes. In contrast to many systems for *explainable AI*, this leaves the initiative with the operator to pick an action, rather than presenting a single recommendation. The challenge here is that when the AI has incomplete knowledge of the control goals or constraints of the operator, this creates uncertainty about the outcomes that are likely to be achieved; and for explanations to be faithful, they must correspond to outcomes under all goals and constraints that the operator is likely to follow. We tackle here how to generate such explanations.

Source code corresponding to the contribution described here is available in the following repository: `https://github.com/AI4REALNET/T2.3_explaining_action_alternatives`.

3.13.1. CONTEXT

The AI4REALNET project looks at how AI can support human operators of critical networked systems. One of the many challenges of operating these systems is their size and complexity. For any each intervention in the system, the operator must choose the optimal course of action from among a set of candidate actions. This first requires a type of counterfactual ("what if") reasoning to determine what effects each action will have on the system and how this affects long-term outcomes. For example, when evaluating a change in the bus connections at a substation in a power network, an operator must consider how this will redistribute the load across the power network, and whether this increases the likelihood of overloads in the future. Next, the outcomes of the various actions can be evaluated for desirability, weighing up with action is best. Note that reasoning through these outcomes becomes exponentially more difficult for large networked systems as we must consider how the effects of actions propagate through the system (reconfiguring a substation affects power loads across the network) and through time (the current configuration can make it harder to redirect power in the future).

We propose to explain to operators what outcomes they can expect to achieve under the different actions they have available to them. The aim here is to largely remove the need for operators to predict future outcomes and instead allow them to focus on *evaluating* actions based on their explained outcomes instead. To do this we show operators the successor features

$$\lambda^{\pi}(s_t, a_t) = \mathbb{E}_{T,\pi}\left[\sum_{k=0}^{\infty} \gamma^k \phi(s_{t+k}, a_{t+k})\right]$$

of each available action. These successor features capture the expected state (and action) features we will encounter after taking action a_t in the current state s_t under the system dynamics and the policy π. The state-action features $\phi(s, a)$ can be arbitrarily chosen to capture any state or action information relevant to the control problem. For example, they can capture whether a power line is at capacity, or how many bus connections an action reconfigures.

Prior work has considered successor feature as explanations found them to be an effective way to show outcomes of actions but differed significantly from our setting in what the explanations tried to achieve. Khan et al. [64] and Yau et al. [145] used successor features to show which states are likely to be visited after an action is taken, helping to explain that an action enables reaching desirable states. Yau et al. [145] additionally showed specific outcomes, further helping to explain why the action is desirable. Lin et al. [78] uses successor features to contrast between two given actions, allowing users to understand the trade-off in outcomes between the two. The difference between these works and ours is that they focus on helping operators understand a given AI policy for auditing purposes. Explanations only serve to further this understanding and therefore explain outcomes exclusively under this AI policy.

We aim to *explain* outcomes that the human operator is likely to achieve themselves, rather than what an independently running AI will achieve. The challenge here is that we must consider that future actions will be taken under the operator's own policy (rather than the AI agent's policy), and under the influence of future explanations from the AI. This challenge is exacerbated when the AI has only partial knowledge of the goals and constraints under which the operator is working, as is usually the case in real world applications. When this happens, the AI is uncertain about the true policy of the operator. For explanations to be truthful, the AI must therefore show in its explanation the expected outcomes under all policies it thinks the operator could be following, thus reflecting the full range of potential outcomes that can be achieved. This is not only a methodological issue, but also a representational one. When different policies can achieve different outcomes, their mixture can attain any intermediate outcome, but only a fraction of the time. This would cause actions that allow us to attain many outcomes to appear less interesting than actions that are more specific to a small set of outcomes.

3.13.2. METHOD FORMULATION

We formulate the control problem as an MDP $\langle S, A, T, R_\omega, \gamma \rangle$ where S is the state space, A the action space, $T : S \times A \to \Delta(S)$ is a stochastic transition function and γ is the discounting rate. The reward function that the operator is trying to maximize R is parameterized by parameter $\omega \in \Omega$. Let there be a feature function $\phi(s, a)$ mapping state-action pairs to their features. We will assume that the AI has full knowledge of this MDP, but that it does not know what the correct value for ω is, i.e. it does

not know the exact reward function the operator is optimizing for. For any state s, let us denote by e an explanation for all actions available in that state. We will slightly abuse notation and denote by e_a the explanation for action a in e. Finally, assume that the operator follows policy $\pi(a|s, e, \omega)$ which depends on the explanation e. We will assume that – apart from the reward parameter ω – this policy is also known to the AI.

To recover the successor features, we must represent the problem from both the point of view of the AI and the operator. To do this we augment the original MDP definition of the control problem into a belief MDP by incorporating the belief of the assistant about the operator's reward parameters ω into the state. The state of this belief MDP is a tuple (s, B) where s is the state of the control problem and $B \in \Delta(\Omega)$ is the AI's current posterior belief over ω. In each time step the AI will generate an explanation e and the operator will choose an action according to $\pi(a|s, e, \omega)$. The original state s transitions according to the normal transition function T of the original control problem. The belief updates according to a Bayes' rule according to the likelihood of the user's action under various reward parameters. The new belief B' is therefore $B'(\omega) \propto \pi(a|s, e, \omega)B(\omega)$.

We now calculate the successor features in this augmented decision problem to recover the correct outcome expectation, which the AI will use as the explanation e. For any state s of the original control problem and any action a, and for a current belief B of the AI about the operator's reward function, we can use the standard Bellman equation for successor features to find the successor features of this augmented problem:

$$\lambda(s, B, a) = \phi(s, a) + \gamma \mathbb{E}_{s' \sim T(s'|s,a); a' \sim \hat{\pi}(a'|s')} \left[\lambda(s', B', a')\right]$$

where $\phi(s, a)$ is the state-action feature function of the original control problem and B' is the updated belief after the AI has observed the operator taking action a. The policy $\hat{\pi}(a|s)$ represents the mixture of potential operator policies under the AI's belief B': $\hat{\pi}(a|s) = \int \pi(a|s, \omega)B'(\omega)d\omega$. Note that for notational brevity we have dropped the dependency here on the explanation e; by definition $e := \lambda(s', B', a')$.

The above definition allows us to model and explain outcomes for a wide range of control problems and operators. Through the operator's policy $\pi(a|s, e, \omega)$ we can make various assumptions about how the operator reacts to and relies on the AI's explanations. For example, when the reward function is linear ($R_\omega(s, a) = \phi(s, a) \cdot \omega$), we can consider an operator who relies exclusively on the explanations by defining $\pi(a|s, e, \omega) \propto \exp(\beta e_a \cdot \omega)$, where β is a temperature parameter. Under this policy, the operator uses the explanation for each action (the successor features) to calculate the Q-value of that action and then selects an action with probability proportional to these Q-values.

3.13.3. APPLICABILITY

The method is applicable in any domain that can be represented as an MDP. This includes all usecases of the AI4REALENT project. The method works best in domains where state-action features $\phi(s, a)$ can be defined that are easily interpretable by human operators.

3.14. POLICY GRADIENT METHOD FOR SAFE REINFORCEMENT LEARNING

3.14.1. CONTEXT

The development of *safe reinforcement learning* arises from the need to deploy RL agents in real-world, safety-critical environments where constraint violations are unacceptable. Traditional RL focuses on maximizing the expected cumulative reward, but this is often insufficient in applications such as energy systems, where safety, stability, or fairness constraints must be guaranteed. This motivates the study of *Constrained Reinforcement Learning* (CRL), formalized as *Constrained Markov Decision Processes* (CMDPs), which incorporate cost functions and associated thresholds representing safety or operational limits.

Among the algorithmic families for CMDPs, *policy gradient* (PG) methods have been particularly successful in continuous-control tasks because of their robustness to noise, ability to handle continuous state–action spaces, and natural incorporation of prior knowledge. PG methods learn directly in the policy space by adjusting the parameters of a stochastic policy, rather than approximating value functions. However, guaranteeing convergence of PG methods in constrained settings remains challenging. Standard policy gradient primal–dual algorithms provide only asymptotic or averaged convergence guarantees, and their rates often depend on the dimension of the state and action spaces. Moreover, most existing work considers only *action-based exploration*, neglecting *parameter-based exploration*, where a stochastic hyperpolicy samples policy parameters directly.

Source code corresponding to the contribution described here is available in the following repository: `https://github.com/AI4REALNET/safe-constrained-policy-gradient`.

3.14.2. METHOD FORMULATION

These limitations are addressed in our contribution [91] through the *C-PG framework*, a general policy-gradient-based primal–dual method for constrained and risk-constrained reinforcement learning. The framework introduces *global last-iterate convergence guarantees*—a property rarely achieved in RL—under weak gradient domination assumptions, while being *dimension-free*, i.e., independent of the cardinality of the state and action spaces. This makes the algorithm highly scalable to continuous domains. Furthermore, C-PG unifies action-based and parameter-based approaches within a single theoretical formulation and extends naturally to risk-sensitive constraints (e.g., CVaR or mean-variance), thus directly targeting properties of *safety, robustness, and trustworthiness*—key desiderata in the AI4REALNET project's safe RL objectives.

The method is founded on a *gradient-based primal–dual approach* to constrained optimization in the

policy space. The core problem is formulated as the minimization of an expected cost function (or risk measure) subject to inequality constraints on expected costs. To solve this constrained problem, a *regularized Lagrangian function* is introduced. The *C-PG algorithm* alternates between a gradient descent step on the primal variable and a gradient ascent step on the dual variable.
Two concrete implementations of C-PG are derived:

- C-PGAE (Action-based Exploration): learns the parameters of a stochastic policy using score-function estimators like to REINFORCE or GPOMDP.
- C-PGPE (Parameter-based Exploration): learns a stochastic hyperpolicy that samples policy parameters directly, thus decoupling environment stochasticity from policy variability and reducing gradient variance.

Both variants extend to *risk-constrained optimization problems* (RCOP) by adopting a *unified risk measure* framework that can be used to represent different risk metrics, such as Expected Cost, Mean-Variance, CVaR, or Chance Constraints. This extension allows C-PG to capture *risk-aware constraints* while preserving theoretical guarantees in the risk-neutral case. The main theoretical result is that C-PG achieves *dimension-free last-iterate global convergence* under standard regularity and gradient domination assumptions.

3.14.3. APPLICABILITY

The C-PG framework and its variants are applicable to a wide range of *continuous and constrained control problems*, where both performance and safety must be jointly optimized.
In the context of the AI4REALNET project, the method is particularly suitable for:

- Power grid management: learning safe control policies for voltage regulation, frequency stability, and optimal power dispatch under physical and safety constraints.
- Railway network operations: optimizing train scheduling, control, and energy use while respecting safety margins, delay limits, and resource constraints.

Beyond these, the method applies directly to autonomous driving and robotics, ensuring safety constraints such as collision avoidance or energy limits, as well as to industrial process control in general. In domains with discrete or low-dimensional CMDPs, C-PG can be used *as is*. For highly stochastic or partially observable environments, adaptations may include recurrent or belief-state policy representations and variance reduction for gradient estimation. In large-scale distributed control problems (e.g., multi-agent energy systems), the primal–dual updates can be decentralized with consensus terms, preserving the safe and trustworthy learning properties central to the C-PG design.

3.15. INTEGRATION OF METRICS FOR ETHICAL DIMENSIONS IN MULTI-OBJECTIVE RL

3.15.1. CONTEXT

Designing *trustworthy-by-design* AI systems remains challenging because high-level ethical principles must be translated into concrete and measurable criteria. While the seven dimensions proposed in the EU Framework for Trustworthy AI offer a useful conceptual framework, their practical integration into AI system design and evaluation remains fragmented [75]. Current evaluation practices suffer from limited standardization of criteria, metrics, and benchmarks, as well as uneven research maturity across ethical dimensions. For instance, Fairness is supported by a relatively rich set of metrics, whereas dimensions such as human agency and oversight, and societal and environmental well-being lack established and widely applicable evaluation methods, making metric selection highly use-case dependent [88].

In addition, ethical dimensions are typically assessed in isolation despite being inherently interdependent, leaving trade-offs between objectives implicit and difficult to analyze. This limits transparency and reduces the practical value of trustworthiness assessments, particularly in complex and safety-critical applications. Multi-objective optimization, and in particular multi-objective reinforcement learning (MORL), provides a methodological basis for integrating multiple ethical metrics *transparently* by explicitly modelling competing objectives. By enabling systematic exploration and comparison of trade-offs, such approaches support a shift from isolated indicators toward holistic, use-case-aware optimization frameworks supporting trustworthiness-by-design.

Power grid operation provides a concrete and highly relevant use case for studying the integration of ethical metrics using MORL [132]. Decisions in this domain directly affect system *safety*, economic efficiency, environmental sustainability, and fairness in service provision. At the same time, the operational complexity of power systems makes it impractical to reduce these concerns to a single performance indicator. This creates a natural setting in which ethical dimensions must be evaluated jointly rather than independently. By examining how multiple metrics can be structured, combined, and interpreted within a unified framework, power grid control serves as a representative example of the broader challenges faced by Trustworthy AI evaluation in regulated, high-impact domains.

This contribution implements and evaluates MORL extensions, including gated tiered scalarisation and preference-conditioned learning and analyses interactions and conflicts between operational objectives via multi-objective reward structures reflecting robustness, fairness, sustainability, and structural system properties.

The codebase for this contribution is publicly available in the Github repository: `https://github.`

com/AI4REALNET/Grid2Op_MORL

3.15.2. METHOD FORMULATION

We consider the problem of **sequential decision-making for power grid operation** under uncertainty. At each discrete time step t, the system is described by a high-dimensional state $s_t \in \mathcal{S}$, representing grid topology, line loadings, generator states, and demand. The agent selects an action at $a_t \in \mathcal{A}$, corresponding to topology modifications or control interventions, and the environment transitions according to unknown dynamics.

Unlike standard formulations that optimize a single scalar objective, grid operation inherently involves **multiple competing objectives**. These include, but are not limited to: **Robustness and safety** (avoiding overloads and blackouts), **Fairness** (balanced service provision across regions or loads), **Sustainability** (reducing reliance on carbon-intensive generation), and **Structural efficiency** (limiting excessive or risky grid reconfigurations).

This motivates a **multi-objective reinforcement learning (MORL)** formulation, where the reward at each time step is a vector

$$\mathbf{r}_t = \left(r_t^{(1)}, r_t^{(2)}, \ldots, r_t^{(K)}\right) \in \mathbb{R}^K \tag{23}$$

with each component corresponding to a distinct evaluation dimension. The learning objective is not to maximize a single expected return, but to identify policies that represent meaningful trade-offs among these objectives.

We applied two different approaches:

AI4realnet-morl [136]. The approach is focused on a scalarised MORL pipeline based on a multi-objective extension of Proximal Policy Optimization (MOPPO). Policy learning was embedded in an outer-loop weight selection procedure using Optimistic Linear Support (OLS), which iteratively proposes linear weight vectors to approximate the convex coverage set (CCS) of the Pareto front. Within this formulation, each policy was trained under a fixed linear combination of objectives, while OLS ensured systematic exploration of trade-offs rather than reliance on manually chosen or heuristic weight settings. From an action-design perspective, this approach relies on a discrete, pre-filtered topology action space derived from domain knowledge. By restricting the action set to a curated subset of meaningful switching operations, the combinatorial complexity of Grid2Op's native action space was substantially reduced.

Staged pipeline [58]. The algorithm implemented a staged Teacher–Tutor–Junior Student–Senior Student architecture, combining expert action generation, imitation learning, and reinforcement learning. Learning is structured as a staged process that progressively embeds expert knowledge, reduces combinatorial complexity, and stabilises exploration before full reinforcement learning is applied. The pipeline addresses the challenge of the combinatorial explosion of admissible topological actions by

constructing a reduced action set using expert heuristics.

The transition to the *staged* pipeline was motivated by two of the most critical challenges encountered using the AI4Realnet-morl method: the combinatorial explosion of the action space and the fragility of early-stage random exploration. By leveraging expert-generated reduced action spaces and supervised pre-training, the pipeline achieved reliable initial performance before transitioning to reinforcement learning. The staged architecture allows to introduce domain knowledge early and gradually, allowing the agent to transition from imitation to autonomous control while remaining within a meaningful region of the policy space. The stages build up on each other, contributing to the stepwise development of trustworthiness.

Stage 1 – Safety and Robustness Stage. Agent used: Teacher. Designed to ensure stable and secure grid operation by prioritizing objectives related to system survival, overload prevention, and risk minimization. This stage dominates in critical or near-failure states and prevents the agent from trading safety for secondary gains.

Stage 2 – Performance and Structural Efficiency Stage. Agent used: Tutor. Designed to optimize operational efficiency and limit unnecessary or risky control actions once the system operates within safe limits. This stage encourages effective grid management without compromising stability.

Stage 3 – Trustworthiness Objectives Stage. Agents used: Junior Student, Senior Student. Designed to promote longer-term ethical considerations such as fairness and sustainability, including balanced service provision and reduced reliance on carbon-intensive generation. This stage becomes active only when safety and operational constraints are satisfied.

Used environments [47] We tested the AI4realnet-morl algorithm on the l2rpncase14sandbox environment, a widely used benchmark scenario in the Grid2Op ecosystem. This environment offers a moderate system size with sufficient topological flexibility to study sequential decision-making under contingencies, while computationally tractable for extensive MORL experimentation [136]. The environment consists of a small transmission grid with limited topological depth. It contains 14 substations, 20 lines, 6 generators, and 11 loads; no storage units are present in this environment.

The staged pipeline was used on a substantially larger benchmark, l2rpnneurips2020track1. It counts 36 substations, 59 power lines, 22 generators and 37 loads (including loads that represent interconnection points to an external grid); no storage units are present in this environment.

Metrics The evaluation of agent behaviour is based on a set of explicitly defined metrics that reflect different operational objectives in power-grid control. The metrics defined in this section form the basis of the reward signals used during training and are used for post-hoc analysis across large sets of runs.

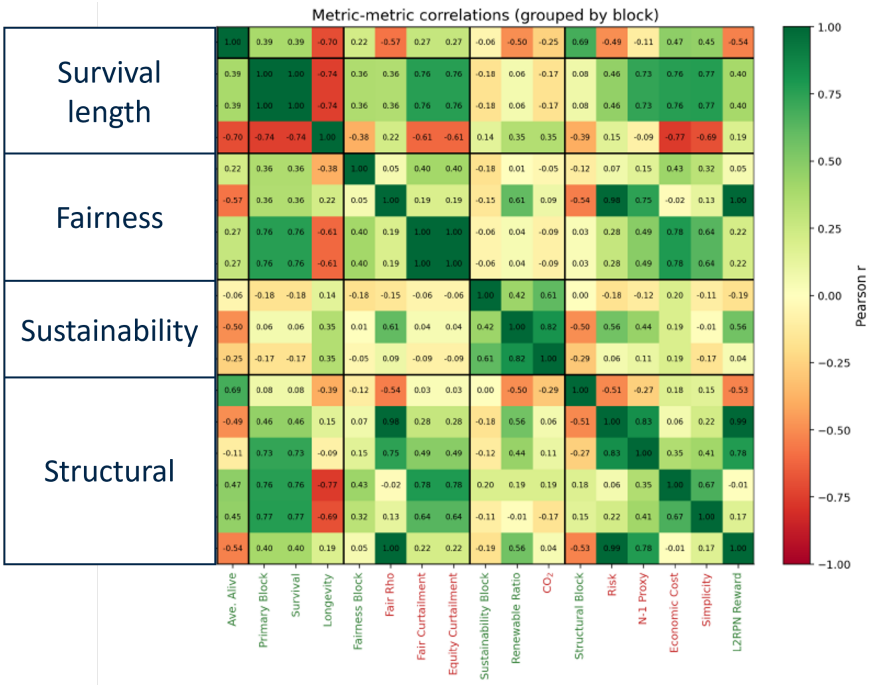

FIGURE 21 - PAIRWISE PEARSON CORRELATIONS BETWEEN TRANSFORMED BASE METRICS, GROUPED BY OBJECTIVE BLOCK.

The ordering and grouping of metrics follows the block structure, so metrics are grouped by their associated block and by their role. The following blocks were used:

1. **Survival**: the number of environment steps before termination and proxy based on episode length.
2. **Fairness**: the regional variance of average transmission line loading and the variance of curtailment magnitude.
3. **Sustainability**: a relative emissions proxy and maximum observed line loading.
4. **Structural**: economic cost, complexity proxy and line loading.

Interplay of metrics for technical and non-technical dimensions for the staged pipeline Pairwise correlations provide insights into local relationships and potential conflicts between metrics. The metrics were calculated for the staged baseline pipeline in the medium environment. Figure 21 shows the correlation matrix, with metrics grouped according to their associated objective blocks.

Strong positive correlations can be observed within blocks, particularly for fairness- and structural-related metrics, confirming that these groups capture coherent operational concepts. For example, curtailment-based fairness measures and economic cost exhibit substantial positive correlation, indicating that improvements in equity often coincide with decreased operational expense.

More interestingly, pronounced negative correlations emerge across blocks. The L2RPN Reward metric shows a strong negative correlation with both the primary block and the average episode survival time (Ave. Alive). This reflects a fundamental trade-off: policies that aggressively intervene to extend short-term robustness may incur long-term costs or increased operational stress, reducing the Ave. Alive metric. Similarly, sustainability-related metrics, such as renewable ratio and COx_2 reduction, display weak or negative correlations with robustness-oriented measures, reinforcing the notion that environmental objectives are not naturally aligned with survival-focused control.

These interactions help explain why scalarised reward functions tend to favor specific objectives implicitly and why improvements along one dimension often degrade performance along another. The observed correlation structure, therefore, motivates the use of grouped metrics and multi-objective learning, as no single metric, or linear combination thereof, adequately captures the full performance landscape.

Four representative configurations were selected for a more detailed analysis: a single-objective PPO baseline and three configurations with a distinct preference profile within the multi-objective space. They are referred to by their main (not exclusive) optimisation intent: Survival, Fairness, and Sustainability. The **Survival** configuration prioritises robustness-related objectives, with increased emphasis on episode longevity and system stability proxies. This configuration represents conservative, resilience-oriented grid operation. The **Fairness** configuration emphasises equitable system behaviour by assigning greater weight to fairness-related objectives. This configuration trades operational margins for more balanced resource allocation across regions. The **Sustainability** configuration explicitly prioritises environmental objectives, focusing on the renewable ratio and CO_2 reduction. This configuration represents environmentally driven operation and accepts potential degradation in robustness and economic performance as a consequence of aggressive sustainability optimisation.

Table 3 compares the four agents across a set of base metrics. The **PPO baseline** achieves strong performance in robustness-related metrics, including survival, but exhibits comparatively weaker outcomes in multiple fairness-related metrics. The **Survival** configuration achieves the highest average episode survival among the actively controlling agents and maintains competitive performance across most other metrics. The **Fairness** configuration substantially improves curtailment equity and fairness-related measures, achieving the strongest performance along these dimensions. These gains are accompanied by slightly increased economic cost and reduced robustness, illustrating the inherent trade-off between equitable operation and conservative control. The **Sustainability** configuration dominates renewable ratio and CO_2 reduction metrics, outperforming all other agents along these dimensions. However, this dominance is associated with significantly reduced survival and robustness-related performance, highlighting that environmental objectives impose a disproportionate cost on system stability.

TABLE 3 - COMPARISON OF BASELINE AGENT AND SELECTED MORL CONFIGURATIONS TRAINED WITH THE STAGED PIPELINE USING AVERAGED BASE METRICS (GREEN - BEST AND RED - WORST ACHIEVED VALUES FOR THE METRIC).

Metric	Target	PPO baseline	Survival	Sustainability	Fairness
Ave. alive	High	378.84	393.32	142.97	309.62
Survival	High	1.241	1.25	1.217	1.257
Fairness ρ	Low	23.028	21.457	17.573	17.309
Fair curtail	Low	11.432	8.821	12.066	7.556
Equity curtail	Low	1.78E-05	1.37E-05	1.88E-05	1.85E-05
Renewable ratio	High	0.609	0.62	0.645	0.637
CO_2	Low	73.6	71.78	68.01	71.63
Risk	Low	0.068	0.064	0.066	0.054
N-1 proxy	Low	0.721	0.716	0.716	0.701
Economic cost	Low	4.31	4.08	4.68	4.31
Simplicity	Low	0.03	0.03	0.039	0.021
L2RPN Reward	High	5.083E-03	5.12E-03	5.198E-03	5.211E-03

Overall, the comparison demonstrates that MORL enables controlled navigation of the multi-objective trade-off space rather than merely interpolating between baseline behaviors. Distinct objective preferences give rise to qualitatively different performance profiles, which cannot be captured by a single scalar ranking.

3.15.3. APPLICABILITY

The code base of this contribution was developed for the Grid2OP environment and hence is not directly transferable for any other use cases. However, it can be extended with the metrics for other trustworthiness dimensions. The transition to a larger environment would make it possible to include features such as storage units or more flexible generator control. Such extensions could meaningfully alter the sustainability–survival trade-off observed in this work by introducing new degrees of freedom for balancing long-term reliability and emissions related objectives.

The idea of using MORL to achieve simultaneously multiple technical and non-technical objectives can be applied to many different scenarios. The main challenge for the method is an informed and context-based choice of the metrics to include in the reward. Our work shows that when the set of metrics was used to describe the same dimensions, their progress during the training process was mostly aligned, but not equal. It means that the choice of the metrics for the same trustworthiness dimension can

influence the result and final reward.

As mentioned in the context definition, multiple trustworthiness dimensions are so far under-researched in terms of operational metrics to describe their performance. Our work gives an example of sustainability metrics, which is one of the underrepresented dimensions.

Application to other domains can follow the same approach by adopting metrics selected for the KPIs identified for each use case. For further analysis and choice of further metrics, functional and non-functional requirements build a solid foundation. They were developed based on the trustworthiness dimensions and are aligned with the specific objectives of the use cases, hence they can secure the alignment with the domain-specific context.

3.16. HUMAN ASSESSMENT MODEL

3.16.1. CONTEXT

The AI4REALNET project aims to have a human-centered AI approach to take advantage as much as possible of the knowledge, experience, and decision making capabilities of the human. As a key aspect to bring the human even closer to the AI and design a full human-in-the-loop concept, we believe that the human needs to interact directly with the AI. Our concept leverages this interaction and connect directly the human mental status to the AI, in order to use it to better adapt its response/reaction/suggestions to the human. In this sense, the AI tries to understand the human's psychological status in a way that the human does not perceive directly, using data from a wearable device (implicit interaction). We believe that by providing operator context to the AI4REALNET AI agents, it will be possible to achieve a higher level of empathy and trust in the AI system, leading to increased acceptability.

The Human Assessment Module (HAM) was developed to test this concept and provide real-time quantification of the cognitive and stress levels of the operator, based on the Electrocardiogram (ECG) acquired by wearable devices during operational scenarios. These levels aim to represent the mental state of the operator, and jointly with an explainable component, they aim to provide operator context to the AI agents. In this project the objective is that the AI agents combine both operational awareness context and this new operator context to support better management of the autonomy level of the system in a seamless and implicit manner, such as reducing the number of notifications or suggestions if the operator is more stressed or with a lower cognitive capability.

The concept of the HAM system also aims to address an important aspect: personalization of the operator mental state assessment. Based on previous studies by INESC TEC, cognitive and stress level behavior differs from person to person, so it is of great importance to understand how to personalize these algorithms for each subject. For example, Figure 22 represents data from a dataset from INESC TEC, which illustrates how 2 subjects (top: subject 1; bottom: subject 2) behave in the react-time task pre- and post-stress situation - it is clear that the behavior is different and a personalization methodology is needed to train algorithms specifically based on each subject's reaction and behavior.

To achieve the HAM personalized human-in-the-loop concept, a methodology was developed to address both cognitive and stress levels. This methodology focuses on the 3 main steps illustrated in Figure 23 :

1. Individual protocol: combination of reaction time tasks and Trier Social Stress Test (TSST) to collect data (Electrocardiogram and Visual Analog Scales – VAS) of each subject and understand the behavior in different scenarios: (a) baseline- to understand the normal/reference status of

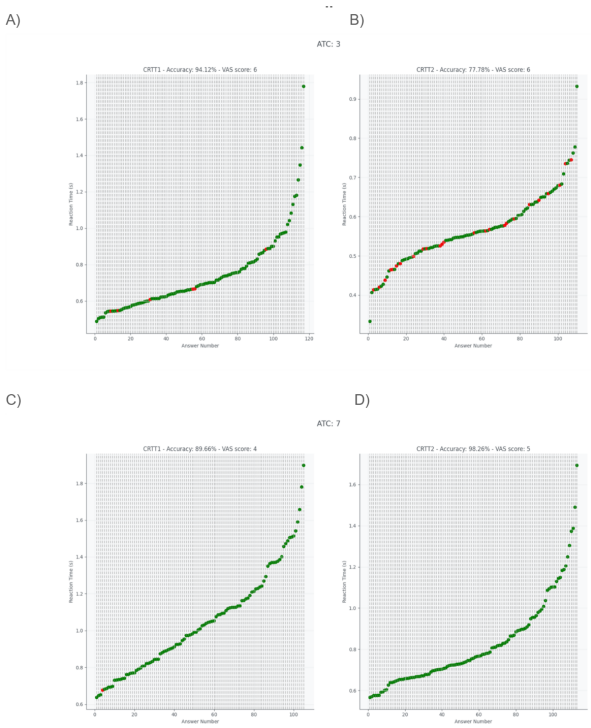

FIGURE 22 - SUBJECT 1 (TOP) AND 2 (BOTTOM) REACTION TIME TO A PRE-DEFINED TASK BEFORE (LEFT) AND AFTER (RIGHT) STRESS INDUCTION BASED ON TRIER SOCIAL STRESS TEST (TSST). GRAPHS SHOW REACTION TIME (S) ON THE VERTICAL AXIS AGAINST ANSWER NUMBER ON THE HORIZONTAL AXIS.

the human in rest; (b) pre-stress: after resting how does the human reacts to a reaction time task; (c) TSST- stress induction protocol to understand how the physiology of the changes when under stress; (d) post-stress: how the human react to a reaction time task after stress induction. This protocol was already defined previously [105].

2. Algorithm personalisation: this is the core of the work performed within Task 2.3, where two workflows were developed to train regression models and other machine learning models according to the data acquired in the individual protocol. These workflows will be further explained below. The output of these workflows is a personalised model that will then be used in the real-time assessment of stress and cognitive levels.

3. Human-in-the-loop real-time assessment: during AI4REALNET data collection campaigns, the objective is that the operator uses a small wearable device that streams ECG data in real-time for the HAM server, making use of the personalised model to infer both stress and cognitive level of that specific subject. This information will then be integrated into the Interactive AI system to provide operator awareness to the AI agents.

Taking all this into account, we believe that HAM contributes to explainable, safe, and trustworthy AI by providing *transparent* and personalized implicit assessments of human cognitive state that are

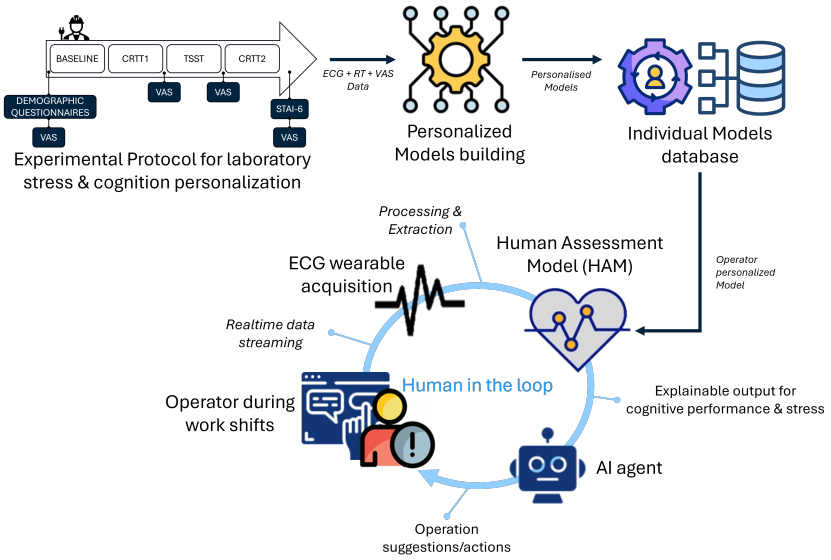

FIGURE 23 - HAM CONCEPT FOR THE DEVELOPMENT AND DEPLOYMENT OF PERSONALIZED MODELS

explicitly integrated into a hierarchical human–AI decision architecture, linking physiological evidence to interpretable human-state indicators and system-level actions. This layered representation enables traceable, *safety-oriented AI* adaptation, supporting transparency, accountability, and trust in human–AI collaboration.

Source code corresponding to the contribution described here is available in the following repository: `https://github.com/AI4REALNET/Human-Assessment-Module`.

3.16.2. METHOD FORMULATION

The experimental paradigm used to elicit stress in participants and to assess the alteration in cognitive skill was replicated from [38]. It is composed of a baseline condition, a first 2-choice reaction time task (CRTT1), the Trier Social Stress Test (TSST), and a second reaction time task (CRTT2), lasting approximately 40min in total. During its deployment, the ECG was acquired through a medical-grade wearable device in form of a t-shirt (Vital Jacket®, Biodevices S.A, Matosinhos, Portugal) with a sampling frequency of 500Hz. Moreover, demographic and psychological questionnaires were administered along the pipeline, as the top left part of Figure 23 illustrates, namely the Visual Analog Scale (VAS) and the STAI-6 questionnaire.

The HAM consists of two main components: a regressor to predict the cognitive performance of human operators using ECG data extracted during the 2 choice reaction time tasks and a classifier to distinguish stress from no stress conditions using ECG data extracted from the initial baseline and the TSST. The HAM was firstly tested on an already existing dataset of INESC TEC concerning air traffic controllers (ATCs). The developed pipeline is schematically depicted in 24.

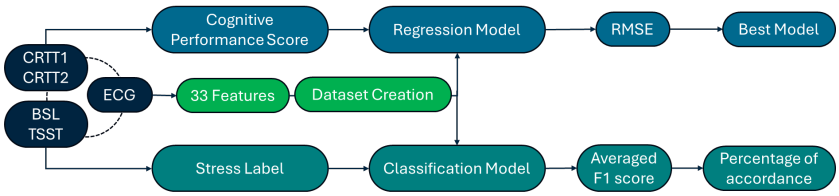

FIGURE 24 - THE DEVELOPED WORKFLOW TO PROCESS THE DATA AND TO TRAIN THE PERSONALISATION MODELS.

Cognitive performance score development: The cognitive performance (CP) score was designed to quantify an individual's cognitive state using the behavioural data collected from the CRTTs trials. Three versions were defined: (a) accuracy; (b) CP1, which divides the overall accuracy over the mean reaction time; (c) CP2, which extends CP1 by also incorporating reaction time standard deviation at the denominator. This last strategy was implemented to capture potential reaction' stability patterns, hypothesising that a better performance is characterised also by a lower variability throughout the trial. The three formulations of the CP score were tested statistically between CRTT1 and CRTT2 in order to discover which held the stronger discriminative power. Moreover, regression analysis was performed to understand which one was better predicted with ECG data. After preliminary analyses, CP2 was selected as the best target variable for regression analysis, because it also accounts for mental skills nuances, even if CP1 demonstrated a slightly higher outcome.

ECG pre-processing and feature extraction: ECG recordings were pre-processed analogously with the hrv-analysis Python library to filter out noise, detect R peaks and correct for ectopic beats using physiologically-validated constraints. The resulting tachogram was then used to derive the heart rate variability (HRV) series and calculate a set of 23 features:

- From the time domain, mean NN, median NN, range NN, SDNN, NN50, pNN50, NN20, pNN20, RMSSD, CVSD and CVNN, mean HR, std HR, min HR and max HR were extracted [38, 112].
- From the frequency domain, power in the very low, low and high frequency bands (VLF, LF and HF, respectively), the ratio between LF and HF, as well as the LF and HF computed in normalised units were included [112].

Notably, HRV features are known to depend on age and gender [138]. Therefore, the possibility of using population-validated HRV limit values, as presented in [138], was investigated to normalise the individual physiological datasets. Such an approach, that would preserve their proportionality and simply change scale, aimed at assessing whether it could enhance the separability between the stress and no stress classes.

Moreover, the identification of the R peaks served as a first step for the computation of an additional pool of 10 metrics that characterise the ECG morphology temporal properties [97, 25]. Specifically,

these parameters measures the time distance (in ms) between the P, Q, R, S, and T fiducial points of the heartwave. P, Q, S and T locations were determined after applying a second order Butterworth low-pass filter with a cut off frequency of 10 Hz to the raw signal and considering previously established physiological time intervals [13].
Since ECG intervals are strongly dependent on the heart rate (HR), a further normalisation was applied on ST and QT intervals to allow for comparability between experimental sections. Therefore, the well-known Bazett formula (Equations 24 and 25) was implemented to correct for ECG morphology HR dependency.

$$QT_c = \sqrt{QT/RR}, \tag{24}$$

$$ST_c = \sqrt{ST/RR}. \tag{25}$$

Notably, several strategies were considered for feature extraction: preliminary tests were conducted with 5min, 2.5min and 1min long windows, with and without overlap. Overall models performance was evaluated to select the best strategy in order to optimise both prediction and classification tasks. Ultimately, the best results were obtained with the shortest window, applying a 50% overlap. Interestingly, this approach presented several advantages, among which the possibility to increase the dataset size for model training and ensure a quick physiological update of the models' outcome in the real-time condition. The re-scaling of HRV metrics worsened the results, therefore this augmentation was discarded and raw features were used to build the individual dataset.

Feature selection: Due to limited sample size, it was decided to implement a feature selection strategy to avoid the curse of dimensionality. To preserve the personalisation framework, it was conducted separately for each participant. An initial analysis was carried out to understand which would be the best feature set size to enhance models performance. Sets of 20, 10, 5 parameters were considered. Several techniques were evaluated: Pearson or Spearman correlation, ANOVA or Kruskal-Wallis tests (depending on the data normality, assessed with a Shapiro-Wilk test with $\alpha = 0.05$), mutual information and logistic regression. It was finally decided that regression analysis used mutual information selected parameters, whereas classifiers used the logistic regression ones.

Regression analysis: Several algorithms were implemented, namely: Bayesian, linear, polynomial, k nearest neighbors (KNN), support vector machine (SVM), ridge, Huber, and RANSAC regressors. Data were split following a 80-20 ratio between train and test sets. Each model underwent a hyper-parameter tuning procedure with a cross-validated grid search strategy to optimise the root mean square error (RMSE). In each iteration, to avoid data leakage, each validation set was scaled according to the mean and standard deviation of current train set. As models will work with novel, real-time

data, the best one was chosen considering the best generalisation capabilities, assessed through the lowest RMSE achieved on the test set. The R2 was also computed.
Overall, personalised models displayed promising yet highly variable results on test data, considering a mean R2 value of 0.41 ± 0.37, which indicates a moderate generalisability. This outcome suggests to increase dataset size, possibly by replicating the experimental paradigm or adding another CRTT section. Possibly, it could be also linked to an incapability of ECG-related features only to capture cognitive mechanisms alterations as measured with the designed score.

Classification: This AI experiment comprised the KNN, SVM, decision tree, gradient boosting, naive bayes and linear discriminant analysis models. Hyperparameters tuning followed the above mentioned procedure that optimised, in this case, the weighted F1 score, due to an imbalance between the stress/no stress labels. Interestingly, as all classifiers presented a similar outcome, it was decided to adopt the percentage of confidence, i.e., the mean F1 score across models, as an evaluation metric. Balanced accuracy and the true positive rate (TPR) for the stress condition were also extracted.
This problem achieved far better results than the regression task, even with unseen observations. Classifiers were trained with a larger (but still relatively small) dataset, but the almost perfect distinction between stress and no stress conditions may indicate overfitting. However, the high TPR for recognising the critical condition, e.g., stress, seems promising for the real-time implementation of the proposed framework as the most critical condition at which the AI agents must intervene is correctly identified with high confidence.

Real-time deployment: A feasibility study was carried out with another database of pre-acquired ECG recordings to test the real-time implementation of the personalised models.
After identifying the ATC's ID, their associated personalised models for cognitive performance and stress were loaded through a custom made RESTful API connected to a database. Each 30s, a request was sent using the API to collect the ECG signal to update a buffer which size matched the window length used during models building. The procedure was developed to check for data continuity, to make sure that a new segment is always appended,to clean and locate fiducial points from the 1 minute long segment to then extract morphology and HRV features. From the entire set of features, a selection is made to select the features that were used to train the specific model only. The real-time dataset is then used to make predictions using the HAM model and print two strings that were conceived to provide interpretability to the AI output. Two custom thresholds were set concerning the CP score and the stress agreement percentage. Notably, both values range between 0 (= severe impairment of cognitive performance; = no stress) and 1. Such a textual information will be used not only to monitor the well-being of the workers, but also to frequently update and make AI agents aware about their psychophysiological status, so that it can adapt the workload and assist individuals without an explicit

prompt.

Notably, even with a sampling frequency of 500Hz, ECG pulling and processing, and model inference were executed without introducing observable delays in the generation of the interpretative outputs. This suggests that the proposed architecture is technically compatible with online monitoring scenarios, supporting its potential integration into adaptive human-AI interaction during work shifts.

3.16.3. APPLICABILITY

The Human Assessment Module (HAM) is conceived as a transversal capability for safety-critical and cognitively demanding operational environments, where human performance, stress regulation, and decision-making reliability directly influence system safety, efficiency, and resilience. By continuously estimating operators' cognitive performance and stress levels through unobtrusive ECG monitoring and personalised AI models, HAM enables AI systems to adapt their behaviour in response to the human's internal state, supporting more balanced workload distribution, safer automation, and more trustworthy human–AI collaboration.

The method is particularly well suited for domains involving critical infrastructure operations, where operators must maintain sustained attention, manage complex and dynamic system states, and respond to time-critical events under high responsibility. In such contexts, HAM can serve as a real-time human-state sensing layer that informs adaptive decision-support systems, alert management strategies, or autonomy modulation mechanisms.

HAM provides a domain-agnostic and modular framework for personalised psychophysiological monitoring and adaptive human–AI interaction. While the current implementation demonstrates feasibility within critical infrastructure operations, its architecture supports scalable extension to a wide range of socio-technical systems in which human cognitive state represents a limiting factor for safety, reliability, and operational resilience. With appropriate domain-specific calibration and sensor integration, HAM has the potential to become a foundational component of next-generation human-centred AI systems across multiple critical sectors.

3.17. DOMAIN TRANSPARENCY IN AIRSPACE SECTORIZATION

3.17.1. CONTEXT

Dynamic Airspace Sectorization (DAS) addresses the need to continuously adapt airspace sector boundaries in response to evolving traffic demand, weather conditions, and operational constraints [46]. Rather than relying on static sector configurations, DAS seeks to redistribute airspace in a way that balances controller workload, maintains safety, and preserves operational efficiency across the network. This problem is inherently complex, as it requires simultaneous consideration of multiple, often competing, constraints that must remain satisfied under all circumstances.

Among the principal constraints are workload balancing across sectors, geometric and topological requirements for sector shapes, traffic flow continuity, coordination requirements between adjacent sectors, and compliance with safety regulations and operational procedures [28, 46, 144]. Additionally, sectorizations must remain interpretable and manageable for human operators, ensuring that transitions between configurations do not introduce unnecessary complexity or risk. These constraints define the feasible solution space within which any sector design must lie, and they are equally binding for both human designers and automated decision-support systems.

To support human operators in designing appropriate and operationally viable sectorizations, a dedicated human–machine interface (HMI) has been developed. The interface is designed to make domain constraints explicit and *transparent*, enabling users to explore, adjust, and evaluate sector configurations while remaining continuously aware of their operational implications. By presenting relevant metrics, visual feedback, and constraint-related information in an integrated environment, the HMI facilitates informed decision-making and iterative refinement of sector designs. This approach ensures that human expertise remains central to the sectorization process while being augmented by computational support, thereby fostering effective collaboration between human operators and future automated tools in dynamic airspace management.

Software repository: `https://github.com/AI4REALNET/ATMSectorization`.

3.17.2. METHOD FORMULATION

Modeling airspace sectorization using Voronoi tessellation is particularly advantageous because it naturally produces convex polygonal regions, which align with common operational and geometric requirements for airspace sectors [144]. Convex sectors simplify traffic flow management, coordination between adjacent sectors, and compliance with safety and workload constraints. Moreover, Voronoi-based representations enable intuitive human interaction: by manually adjusting the positions of Voronoi generator points (sector centers), operators can directly and predictably reshape sec-

tor boundaries while preserving convexity and full airspace coverage. This provides a transparent and computationally efficient mechanism for iterative sector design, allowing human planners to explore alternative configurations through simple spatial manipulations of control points while maintaining structurally valid sector geometries.

Airspace. Let $\Omega \subset \mathbb{R}^2$ be a bounded polygonal region describing the airspace of interest (see Figure 25). Let $\mathcal{L}$ denote a set of airway polylines (or flows), and let $\mathcal{A}(t)$ denote the set of aircraft positions at time $t \in [t_0, t_1]$, provided by a traffic simulation or prediction model. Aircraft i follows a polyline (airway) with position $x_i(t) \in \Omega$ for $t \in [t_0, t_1]$. A sector entry/exit event occurs when $x_i(t)$ crosses $\partial V_k(S)$.

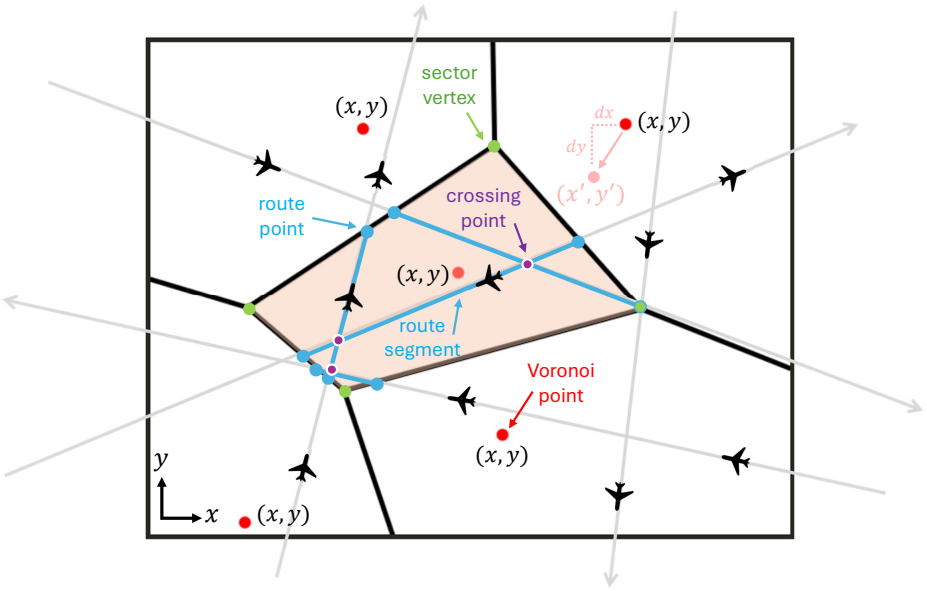

FIGURE 25 - AIRSPACE VORONOI TESSELLATION.

Voronoi Tessellation and Fortune's Algorithm. A Voronoi tessellation partitions the plane based on proximity to a set of generator points (sites). Given a set of sites

$$S = \{p_1, p_2, \ldots, p_n\}, \quad p_i \in \mathbb{R}^2,$$

the Voronoi cell associated with site p_i is defined as

$$V_i = \{x \in \mathbb{R}^2 \mid \|x - p_i\| \leq \|x - p_j\|, \ \forall j \neq i\}.$$

Each cell therefore contains all points closer to its site than to any other. Boundaries between cells lie on perpendicular bisectors between pairs of sites, while Voronoi vertices occur at points equidistant from three or more sites. The resulting diagram forms a partition of the plane into convex polygonal

regions and is dual to the Delaunay triangulation.

Fortune's algorithm [42] provides an efficient method for constructing the Voronoi diagram with time complexity $O(n \log n)$. It is based on a sweep-line approach in which a horizontal line moves across the plane, processing sites in descending order. As the sweep progresses, a dynamic curve known as the beach line represents the current boundary of influence of processed sites and is composed of parabolic arcs defined by distance to the sweep line and to individual sites.

Taskload prediction per sector and time interval. Fix a time step Δt and let $\mathcal{T} = \{t_0, t_0+\Delta t, \ldots, t_1\}$. For sector k at time t, define:

$$N_k(t) := \#\{i \mid x_i(t) \in V_k(S)\} \quad \text{(aircraft inside)}, \tag{26}$$

$$E_k(t) := \#\{\text{entries/exits of any } i \text{ across } \partial V_k \text{ in } [t-\Delta t, t]\}. \tag{27}$$

A basic *taskload* model is a weighted sum

$$\tau_k(t) \;=\; \alpha\, N_k(t) \;+\; \beta\, E_k(t), \tag{28}$$

with weights $\alpha, \beta \geq 0$. Optionally extend τ_k with additive penalties:

- Conflict proximity inside V_k: $C_k(t)$ (e.g., number of pairs closer than a threshold).
- Crossing/intersection density (airways vs. boundaries): $X_k(t)$.
- Geometric irregularity of V_k (e.g., perimeter/area ratio): G_k (time-independent).

Then

$$\tau_k(t) \;=\; \alpha N_k(t) + \beta E_k(t) + \gamma C_k(t) + \delta X_k(t) + \eta G_k. \tag{29}$$

When explicit trajectory simulation is expensive, one can approximate the instantaneous sector workload by aggregating H over the cell:

$$\widehat{\tau}_k(t) \;\approx \sum_{c:\, g_c \in V_k(S)} w_c\, H(g_c, t), \tag{30}$$

where w_c is a cell area (or integration) weight and $H(\cdot, t)$ is a time-indexed forecast of density.

Taskload balancing objective. We quantify cross-sector imbalance at time t by standard deviation

$$\sigma(t; S) \;=\; \sqrt{\frac{1}{K}\sum_{k=1}^{K} \big(\tau_k(t) - \bar{\tau}(t)\big)^2}, \qquad \bar{\tau}(t) = \frac{1}{K}\sum_{k=1}^{K} \tau_k(t). \tag{31}$$

Over the horizon $\mathcal{T}$ we can minimize, for example, the time-averaged imbalance

$$J_{\text{bal}}(S) \;=\; \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \sigma(t;S). \tag{32}$$

To account for constraint violation penalties, let $M(t) := \max_k \tau_k(t)$ be the peak workload. Given a safety limit M_{max}, introduce an overload penalty

$$P_{\text{over}}(S) \;=\; \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \phi(M(t) - M_{\text{max}})\,, \tag{33}$$

where $\phi(u) = 0$ if $u \le 0$, and grows linearly/quadratically/exponentially if $u > 0$. Additional penalties can be added for short dwell times ("sector hopping"), excessive boundary crossings, or pathological shapes.

Human-Machine Interface and usage. The airspace sectorization interface in Figure 26 enables interactive design and assessment of dynamic sector configurations through a Voronoi-based representation. Operators can directly manipulate the sector layout by dragging Voronoi centers to reshape boundaries, removing centers to merge regions, or adding new centers to introduce additional sectors. This center-based interaction provides a simple and transparent mechanism for iteratively refining sector geometry while maintaining valid polygonal partitions of the airspace.

To support operational feasibility, the interface provides continuous feedback on constraint compliance. Users can inspect detected constraint violations, including route segments that become too short and route crossing points that lie too close to sector boundaries. These indicators help operators identify configurations that may increase coordination complexity, reduce predictability, or violate geometric and procedural requirements, and they guide targeted adjustments to the sector design.

In addition to geometric and route-based constraints, the interface integrates workload-oriented decision support. For any selected sector, users can inspect the predicted taskload profile over time and immediately identify exceedances of a specified maximum taskload threshold. This temporal view enables assessment of whether a candidate sectorization remains robust under demand fluctuations, and it supports proactive redesign before overload conditions occur.

Finally, the interface supports network-level balancing by providing cross-sector taskload balance indicators. In particular, it summarizes the distribution of taskload across sectors using the standard deviation of taskload values, enabling operators to quickly assess whether workload is evenly shared or concentrated in a subset of sectors. By combining interactive geometry manipulation with constraint- and workload-centered feedback, the interface facilitates efficient exploration of sectorizations that are both operationally compliant and balanced.

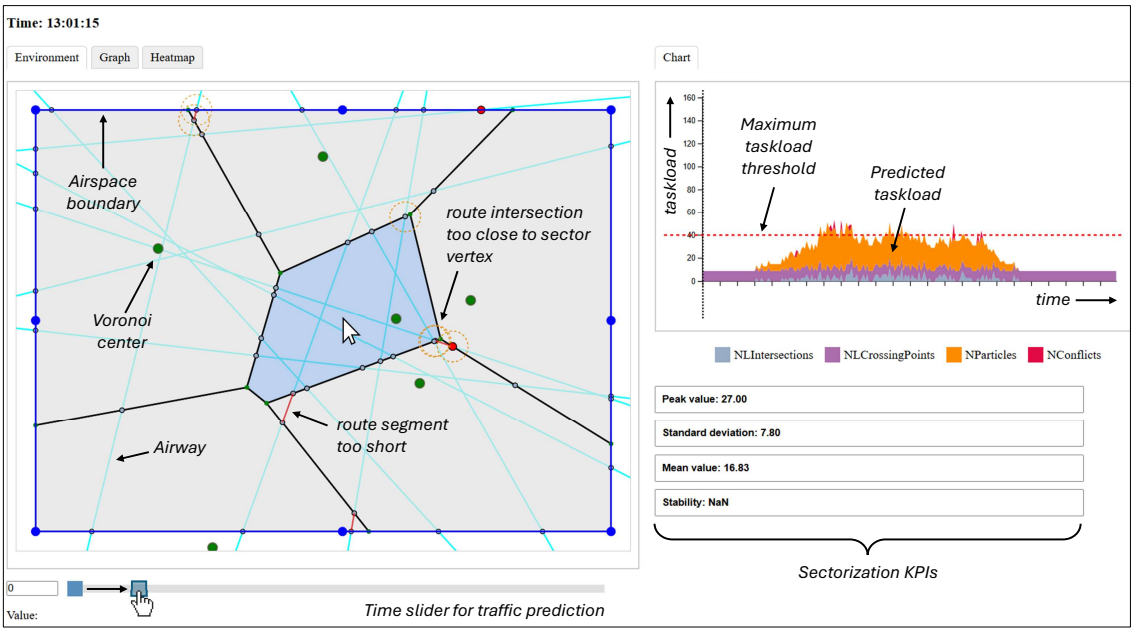

FIGURE 26 - HMI FOR DOMAIN TRANSPARENCY IN AIRSPACE SECTORIZATION.

3.17.3. APPLICABILITY

Voronoi-based sectorization can be applied to a wide range of other domains that require spatial partitioning under workload, coverage, or coordination constraints. In **urban service and emergency response zoning**, Voronoi regions can define responsibility areas for police, fire, or medical units. Adaptations would include replacing Euclidean distance with travel-time metrics over road networks, incorporating population or incident density into workload estimates, and enforcing administrative or geographic boundaries. In **telecommunications planning**, Voronoi diagrams can approximate coverage regions of base stations. This requires weighted or anisotropic distance metrics reflecting signal propagation, terrain, and interference, as well as constraints related to capacity balancing and coverage overlap for seamless connectivity. In **logistics and distribution**, Voronoi partitioning can support territory design for warehouses or delivery depots. Domain-specific adaptations include demand-weighted workload models, network-based travel distances, and constraints on delivery time and route continuity. In **environmental monitoring and multi-robot systems**, Voronoi regions can allocate monitoring or coverage responsibility among sensors or agents. These applications require dynamic updates, obstacle-aware distance measures, and workload definitions based on sensing requirements or energy constraints.

Across domains, the main adaptations involve redefining distance metrics, workload models, and operational constraints. The underlying Voronoi framework remains valuable due to its convex partitions, transparency, and ease of manual adjustment through generator-point manipulation.

3.18. DESIGN REQUIREMENTS FOR AI TRANSPARENCY

3.18.1. MOTIVATION FOR AUTOMATION TRANSPARENCY BEYOND EXPLAINABILITY

Effective use of AI recommendations requires sustained cognitive effort – monitoring the situation, interpreting outputs, and judging applicability. Many systems, however, do not elicit the motivation needed for such deeper engagement [21]. Research and practice have rather focused on transparency and trust, particularly through explainable AI [32]. However, providing more information does not necessarily increase meaningful engagement, nor does it guarantee appropriate reliance [32, 70]. This motivates a broader view of automation transparency that goes beyond explainability alone.

Automation transparency, in this broader sense, concerns whether humans can develop an appropriate understanding of what the automation does, why it behaves as it does, how it interacts with operational constraints and other actors, and when it is likely to be unreliable. In complex work environments, transparency depends heavily on motivation: if humans do not perceive engaging with the automation as worthwhile, feasible under time pressure, and aligned with their responsibility, they will rationally minimize effort, often resulting in superficial use of AI support. This challenge is consistent with evidence from a meta-analysis showing that human-AI combinations, on average, performed worse than the better of humans or AI alone, with particularly pronounced performance losses in decision-making tasks [130].

Explainable AI can contribute to transparency, but it is not sufficient for it [32]. In T2.3.2 we therefore worked conceptually (see Hamouche et al. [49], Waefler et al. [139]) emphasizing that recommendations (even with explanations) often fail to trigger meaningful engagement: humans may not develop the understanding needed to integrate AI advice appropriately, and explanations may not be used in ways that improve judgment. Consequently, transparency should be treated as a socio-technical characteristic of work, not merely as an explanation feature. This directly points to work design: what matters is how AI is embedded into the work and tasks, shaping roles and responsibilities and thus human engagement. This also clarifies why usability is necessary but insufficient: usability can optimize how users interact with an interface, but it cannot guarantee motivated engagement or appropriate reliance. The primary lever for these outcomes is work design, i.e., whether the AI configuration preserves human autonomy, maintains coherent human task involvement as well as an active human role, and provides feedback that supports human skill and knowledge development over time.

Similarly, if we aim to develop a co-learning AI (T3.3) or move towards a fully autonomous AI with human supervision (T3.4), we must design the technical systems in a way that humans are not left with residual “monitor/approve” role. Regardless of the AI’s level of autonomy, task and function allocation – and thus human roles – should be human-centered and therefore aligned with work design criteria.

However, our literature research suggests that such work-design-grounded approaches remain comparatively rare in current AI design guidance. We therefore developed design and evaluation criteria based on work design theories, presented in the following chapter, to inform the T3.3 and T3.4 AI designs. [1]

3.18.2. ANALYSIS: HUMAN NEEDS REGARDING AUTOMATION TRANSPARENCY IN THE CO-LEARNING SCENARIO

Automation transparency needs regarding human learning. In complex, high-skill operational domains such as railway operations, AI support is often introduced to improve prediction accuracy and decision performance. However, systems that primarily "solve the problem" (i.e., through recommendations) rarely create systematic learning opportunities for human operators [131] and may even contribute to deskilling over time [36]. As an analogy, "students cannot learn to solve complex math problems if a teacher simply tells them the answer" [49]. Learning requires support that helps people understand the underlying problem and the reasoning path – not only the final output.

This is particularly critical in safety- and resilience-relevant systems: long-term performance depends on sustained human expertise. Human operators must remain capable of monitoring dynamic conditions, detecting critical developments, understanding system interdependencies, and acting effectively under uncertainty. Our approach therefore follows a human-capability-centric objective: AI should not only deliver correct outputs, but also help humans develop, maintain, and refine skills that make hybrid human-AI work effective in the long run. As argued above, recommendations and explanations alone are not a reliable standalone mechanism for learning in complex tasks [21, 32]. Deliberate learning requires structured support.

To structure such deliberate learning support, we ground our approach in Kolb's experiential learning cycle [67], which describes learning across four phases. First, humans have a concrete experience, which they reflect on regarding what happened, what worked, and what not. Based on this reflection, they develop abstract concepts and generalizable principles. These principles are then tested through active experimentation in a new but comparable situation. The cycle then begins again with experiences and updated knowledge about what worked and what did not. Each phase implies different opportunities for AI to support operators' active learning.

However, before learning can be supported systematically, the relevant learning targets must be defined. For this, we drew on the Supportive AI Framework (see Waefler et al. [139]; developed in WP1, see D1.1), which conceptualizes network resilience as depending on human macrocognitive performance, such as situation monitoring, detection and comprehension, and sound decision-making grounded in explicit and tacit contextual knowledge. The framework further distinguishes four sup-

[1] Since this is literature-based analysis, there is no corresponding software deliverable.

port modes through which AI can support humans: transparency, exploration, animation, and mirroring – modes that go beyond recommendations and explanations. As a theoretical foundation to specify learning targets for monitoring, detection, comprehension, and projection [139], we drew on Endsley's model of situation awareness (SA) and Decision-Making [34]. It assumes that achieving situation awareness requires progressing through three distinct cognitive stages: 1) perception, referring to knowledge of relevant situational cues, 2) comprehension, referring to the ability to interpret information correctly, and 3) projecting, referring to the ability to anticipate how a situation may develop. Knowledge about leverage points and coping strategies is captured by decision-making [34]. While the Supportive AI Framework developed in WP1 helps define the broader learning fields (e.g., environmental knowledge) and distinguishes four different support modes, the situation awareness levels provide a structured way to define specific learning targets within those fields.

Building on these learning targets and support modes, we structured AI-supported learning along Kolb's experiential learning cycle and derived concrete learning-support functionalities for each phase. The approach links a) learning targets (situation awareness levels 1-3 and decision-making), b) Kolb's four phases (experience, reflection, conceptualization, and experimentation), and c) support Human-AI interaction modes – transparency (TP), exploration (EX), animation (AN), and mirroring (MR]) to specify actionable AI functions. The example in (table 1) 1 focuses on decision-making. Hamouche et al. [49] covers all levels of situation awareness as well as decision-making. The full paper will be presented at the "16th International Conference on Human Interaction & Emerging Technologies: Artificial Intelligence & Future Applications".

Automation transparency needs regarding human motivation. Co-learning not only requires a specific design focus to deliberately support learning, it also requires humans to stay cognitively engaged: they must monitor evolving conditions, interpret cues, and judge when AI advice applies. Whether they do so depend on motivation shaped by task and role design. Without motivation-supportive design, engagement drops and humans are more likely to either overrely on the system or ignore it. To specify what "motivation-supportive design" means, we adopt a work design lens.

Established work design theories translate motivation into actionable characteristics of work, such as autonomy, feedback, or task identity (i.e., being involved in coherent task units) [48]. Extensions of the model further highlight knowledge characteristics (including e.g., information processing, problem solving, job complexity) as key conditions for sustained cognitive engagement [92] for knowledge-intensive work. Complementary perspectives such as SMART work design emphasize the need to preserve agency and mastery while keeping demands tolerable in complex socio-technical systems [99].

Our literature research indicates that much of the human-AI design literature emphasizes trust, trans-

	Generic AI-functions	Examples for AI-functions
	Learning target: Improved decision-making	
Concrete experience	How might AI help people experience alternative decisions and their impact on outcomes?	
	• Real-time effects of decisions on the networks KPIs [TP; EX] • Impact transparency [TP]	• The AI visualizes in real time how different decision options affect key network indicators and KPIs • The AI makes visible which network areas are most affected by the human's current decision
Reflection	How might AI help humans reflect on their decisions they made?	
	• Comparison of decision-levers [TP; MR] • Decision rationale replay [MR]	• The AI shows which network indicators, metrics, and recommendations the human considered when deciding and which were ignored, helping them reflect on how specific inputs influenced the chosen action. Decision rationale replay [MR] • The AI replays the decision paths, high-lighting moments of trade-offs (e.g. efficiency vs. resilience) and prompting reflection on the reasoning behind each choice
Abstract Conceptualization	How might AI help humans abstract their decisions into rules or models?	
	• Include decision principles from past cases [TP; EX] • Mirrors operators tendencies/biases in decision making [MR; AM]	• The AI auto-drafts an IF-THEN-UNLESS rule from similar successful network cases helping operators formalize effective decision principles • Highlights systematic priority for one or few service classes
Active Experimentation	How might AI help humans test their decision strategies?	
	• Impact feedback on KPIs and on sectors around [TP] • Sandbox mode "what-if" [EX] • Peer strategy suggestion [MR]	• The AI allows operators to test decision A versus B in a simulated network environment and immediately see the impact on key performance indicators • The AI visualizes how each decision affects key network KPIs and shows the effects on surrounding sectors in real time • The AI shows successful decision strategies from other operators in comparable network situations, encouraging experimentation with alternative approaches to improve outcomes

TABLE 4 - CO-LEARNING AI FUNCTIONALITIES FOR DECISION-MAKING.

parency, and explainability, while relatively few contributions explicitly derive guidelines for AI design based on established motivation theories or work design theories. Where work design theories are referenced, they are often used to interpret or evaluate effects rather than to guide AI design. A detailed synthesis is currently being prepared for scientific publication.

However, motivation needs to be treated as a design objective and anchored in work-design-grounded criteria. Building on this, we developed evaluation anchors that can inform both design and evaluation of AI designs. The final model we draw on includes five task characteristics (task variety, task significance, task identity, autonomy and feedback) and five knowledge characteristics (skill variety, job complexity, information processing, problem solving, and specialization). While the task characteristics capture how work is structured, the knowledge characteristics reflect the cognitive demands and cognitive engagement that the work requires. These knowledge characteristics are particularly relevant for preserving expertise and preventing deskilling over time [21, 79, 98]. The model assumes that fulfilling these characteristics fosters three critical psychological states: experienced meaningfulness, experienced responsibility, and knowledge of own results, which in turn promote intrinsic work motivation.

To turn these criteria into a practical tool for AI design and evaluation, we translated the criteria into qualitative anchors through a theory-driven synthesis: building on core job design literature, we conducted a targeted review of how AI may affect each characteristic and synthesized the resulting insights, criterion by criterion. The corresponding anchors inform AI design for the co-learning scenario (T3.3). Table 5 presents an excerpt from this operationalization for the criterion task identity. Hamouche et al. [50] covers all criteria. The full paper was presented at the “15th International Conference on Human Interaction & Emerging Technologies”.

We first defined the criterion task identity as “the extent to which a job requires completing a ‘whole’ and identifiable piece of work” [48, p. 257]. A key challenge posed by AI for task identity is that human work can be reduced to isolated steps (e.g., monitoring, and approving outputs), which can weaken people’s understanding of how their actions contribute to the overall outcome [79, 98]. Based on this, we derived the design goal that human-AI workflows should enable humans to complete coherent, end-to-end task units and to clearly recognize the meaningfulness of their contribution to the whole.

To facilitate the application of such criteria for human-AI function allocation and hence for task design, we have developed a method in T2.3.3 [154] (paper to be presented at the “16th International Conference on Human Interaction & Emerging Technologies: Artificial Intelligence & Future Applications”). The method is based on the Supportive AI Framework [139] and was developed in the tradition of sociotechnical system design [129], of complementary function allocation [142], and of joint cognitive system [57]. For the design process, it also draws on instruments from the user experience field such as user story maps and user story boards. The method proposes a three-step approach:

High	Medium	Low
Humans are assigned a complete task (including planning, preparation, execution and fol-low-up tasks) and see how their work contributes to the whole. AI supports without fragmenting the process and by making contributions transparent.	Humans are involved in selected task steps. AI handles key steps and makes contribution partially transparent.	Humans only perform isolated actions without seeing how their input fits into a complete task. AI executes most of the process.

TABLE 5 - EVALUATION ANCHORS FOR TASK IDENTITY.

- Step 1, as-is analysis: Understanding the task within the sociotechnical system
- Step 2, role design: Development of psychologically coherent roles for human and AI
- Step 3, collaboration design: Design of the interaction between human and AI

This sequence ensures psychologically sound design: Understanding current work (Step 1), establishing coherent roles (Step 2), interface design (Step 3). This sequential grounding in actual work practices enables the joint cognitive system to function effectively. Both design steps (Steps 2 and 3) comprise two distinct phases: a descriptive phase and a normative phase. The descriptive phase focuses on developing potential role and collaboration designs and analyzing their impact on humans. The normative phase evaluates these designs against psychological criteria. This separation transforms subjective impressions into measurable factors, creating an objective, standardized foundation for design decisions.

Step 1: As-is analysis The as-is analysis is based on two established approaches: the Sociotechnical System Analysis (STS) and the Hierarchical Task Analysis (HTA). STS focuses on human work activities and provides insights into working conditions, task design, and their implications for overall system performance [129]. HTA structures tasks hierarchically into goals, subgoals, operations, and plans, thereby making task components and their interrelations explicit [122]. Where necessary, complementary approaches such as Activity-Centred Task Analysis (ACTA) or Goals, Operators, Methods, and Selection Rules (GOMS) analysis can be incorporated (see Stanton et al. [122], for an overview).
The results of the as-is analysis are summarized in a table providing a comprehensive overview of all tasks and subtasks. It also describes the task within the sociotechnical system and the selected evaluation criteria. This step deliberately focuses on how the work is actually carried out (work-as-done), as only an understanding of real work practices provides a basis for the subsequent design steps.

Step 2: Role design The objective of Step 2 is to design a psychologically realistic and meaningful human role. This is achieved through an iterative procedure: potential AI functions are identified for each subtask, the resulting changes to human roles are described, and the revised roles are evaluated against psychology-based criteria. This cycle repeats until the human role is deemed psychologically acceptable.

Step 2 employs user story maps, which provide a process-oriented visualisation of system requirements and support a shared understanding among stakeholders [123]. The user story map extends the as-is table from Step 1 by adding two rows that distinguish between supportive AI functions (which assist humans) and automated AI functions (which fully automate tasks, replacing humans). This distinction explicitly encourages consideration of AI as a means of human support rather than automation alone. As potential AI functions are specified, the user story map transitions from an "as-is" to a "to-be" representation, defining both the future human role ("User Task") and the corresponding AI functions.

The impact of proposed AI functions on the human role is evaluated iteratively using selected criteria.

Step 3: Collaboration Design as Interface Concretisation Whilst AI functions and psychologically sound human roles have been developed in the previous step, this step focuses on designing human–AI collaboration. A user story board is used to establish a shared understanding among stakeholders and to identify the interactions required for effective collaboration. Based on the role design from Step 2, the initial to-be story board is iteratively refined, supported by design templates such as predefined interface elements.

The resulting collaboration design is evaluated using established human–AI collaboration frameworks (e.g., Amershi et al. [7], Hoffman et al. [56], Huchler, N., et al. [60], Schmidt and Herrmann [109], Shneiderman et al. [116]). As in Step 2, the process follows an iterative cycle of design, evaluation, and refinement until a satisfactory solution is achieved. All artifacts generated serve as inputs for subsequent AI development.

3.18.3. ANALYSIS: HUMAN NEEDS REGARDING AUTOMATION TRANSPARENCY IN THE FULL AUTONOMOUS SCENARIO

From a human factors' perspective, the combination of humans and autonomous AI presents a number of challenges. These involve well-known problems of supervisory control, which were already described by Bainbridge [14] as "ironies of automation." Endsley [36] discussed these challenges and concerns regarding the use of AI. A major problem is the black box nature of AI. Humans are faced with the impossible task of evaluating AI-generated suggestions that they can no longer understand and taking responsibility for them. This effect even appears when AI provides explanations [22]. Further

challenges include difficulties in developing adequate situation awareness, deskilling, de-motivation, or automation complacency.

These negative effects on humans are exacerbated by the black box nature of autonomous AI in conjunction with the passive role assigned to humans in terms of supervisory control. To address these two problems while still leveraging the benefits of autonomous AI, we turn to Wahde and Virgolin [140] concept of interpretable primitives. A primitive is an autonomous AI agent with reduced scope, so that its purpose and functioning are easy for humans to understand (e.g. prioritizing train A over train B). To avoid the black box problem, many primitives that are understandable to humans are used instead of a comprehensive but incomprehensible AI. The human's role is to orchestrate the primitives by defining strategies, setting priorities, or directing their deployment. In this way, humans are assigned an active role that includes task characteristics that are considered prerequisites for human engagement and upskilling (e.g. Parker and Grote [98]).

The following sections describe what we developed in T2.3.2 regarding identifying suitable primitives in critical network control. The criteria and the method for applying these criteria are described in Dettling et al. [29]. This paper will be presented in July 2026 at "The Applied Human Factors and Ergonomics International Conference" (AHFE).

Criteria for deriving primitives. As a first step, we derived evaluation criteria that define what makes a good candidate for an interpretable primitive. The criteria were derived from the definition by Wahde and Virgolin [140] of interpretable primitives as high-level components whose purpose and mode of operation are easily understandable for humans and that can be combined to realize complex operations while still remaining human-readable. Based on this definition, we translated interpretability into three assessment criteria with indicators and qualitative rating scales: clarity of purpose, process transparency, and granularity.

- Clarity of purpose captures whether the purpose of a primitive candidate can be stated clearly in its context. A candidate scores high when it supports a clearly identifiable goal, refers to a defined object, and can be named so that its purpose is recognizable once the context is known. Candidates score lower when the purpose remains broad, mixes several aims, or depends heavily on interpretation.

- Process transparency captures whether it is realistic to specify how the primitive candidate works in a short and checkable way. A candidate scores high when it corresponds to a clear action function (e.g., identify, filter, check, prioritize) and when the required information basis (inputs) can be stated. Candidates score lower when the function bundles multiple actions or when inputs remain unclear or implicit.

- Granularity captures whether the scope of a candidate is appropriate as a building block. A candidate scores high when it forms a closed unit with clear boundaries and remains manageable and combinable as part of a set. Candidates score lower when they are too broad or too fine-grained (micro-steps that would create too many small primitives and increase orchestration effort), or when it cannot be combined cleanly with other units.

Primitive derivation process. The following steps describes how the hierarchical task analysis (HTA; Stanton et al. [122]) is used to derive and select primitive candidates in a structured way.

- Step 1: Collect task and goal material for disruption management.: Data (e.g. by interviews, observations, etc.) needs to be collected in order to describe tasks in normal operations as well as during disruptions. The data needs to be coded so that tasks can be linked to the goals they serve.

- Step 2: Construct the hierarchical task analysis (HTA) as the derivation basis: The coded tasks and goals are consolidated into a hierarchical task analysis (HTA). The HTA structures the work into goals, sub-goals, and the tasks/subtasks that contribute to them, which provides a systematic map of possible candidate primitives.

- Step 3: Identify suitable primitive candidates using the criteria: The three criteria as described above are applied to sub-tasks at the lower levels of the HTA as initial candidate functions. If subtasks are rated weakly based on these criteria we move to the next level of sub-tasks. This is repeated in a bottom-up approach until sub-tasks receive a strong rating across clarity of purpose, process transparency, and granularity. Sub-tasks at this point are considered interpretable primitives.

3.18.4. APPLICABILITY

The design requirements described in this chapter are generic in nature and based on psychological findings. Therefore, these requirements are not domain-specific. However, the requirements refer to the combination of humans and AI in complex, knowledge-intensive tasks in which humans are experts and should remain experts even after the integration of AI, so that humans and AI complement each other. Consequently, the requirements do not apply to non-experts or consumer behavior. Nor do they apply to AI that aims to replace humans

4. CONCLUSION

This deliverable has focused on formulating AI approaches for transparent and trustworthy knowledge-assisted, hierarchical and distributed AI for network infrastructure. Out of the targeted properties, we can largely see two groups: First, there is a group that is aimed at performance, and specifically allowing AI methods to be applied to large-scale infrastructures. This group contains the properties of knowledge-assisted, hierarchical, and distributed AI. Secondly, there is a group that is aimed at improving the trustworthiness of AI methods, in particular the safety, transparency, and explainability. This report has first formulated these six properties. Where the scaling-related properties are concerned, knowledge-assisted AI helps by reducing data requirements through stronger inductive biases; hierarchical AI helps by splitting complex problems in smaller (higher- and lower-level) sub-problems; and de-centralized AI helps by also splitting complex problems, but now in parallel components.
Where the trustworthiness aspects are concerned, explainable and transparent AI both support the ability of human supervisors to understand the AI recommendations better, and improve the auditability of such methods. AI safety, on the other hand, ensures important constraints are respected.
In this report, subsequently, 18 different contributions towards realizing these six properties were described. Table 6 lists the contributions that target each of these properties. It is clear that many contributions target more than one property. These co-occurences are visualized in Figure 27. One can see that knowledge-assisted AI co-occurs with most properties. Besides that, the scalability-related properties and the trustworthiness-related properties tend to co-occur within these two groups.
The contributions outlined in this report were to large extend conceived for and/or evaluated on at least one of the three project domains: power grid control, train scheduling, and air traffic management. However, almost all of these contributions are also applicable (with or without adaptations) to other domains, within or outside of the project. Table 7 describes applicability and transferability of these methods, with examples of alternative application domains.

Knowledge-assisted AI	Sections 3.1, 3.2, 3.3, 3.4, 3.10
Hierarchical AI	Sections 3.3, 3.5
Decentralized AI	Sections 3.4, 3.5, 3.6, 3.7, 3.8
Explainable AI	Sections 3.2, 3.8, 3.9, 3.11, 3.12, 3.13
Transparent AI	Sections 3.11, 3.12, 3.15, 3.16, 3.17, 3.18
Safe AI	Sections 3.8, 3.9, 3.10, 3.14, 3.15, 3.16

TABLE 6 - CONTRIBUTIONS TARGETING THE SIX DESCRIBED PROPERTIES.

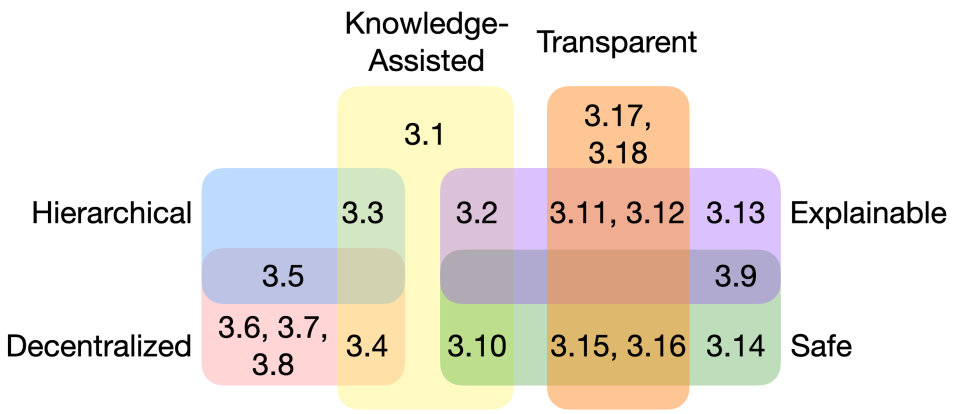

FIGURE 27 - PROPERTIES TARGETED BY THE CONTRIBUTIONS DESCRIBED IN THIS DOCUMENT.

TABLE 7 - APPLICABILITY AND TRANSFERABILITY OF LISTED CONTRIBUTIONS

Results	Knowledge representation type	Potential to transfer knowledge to new domains	Examples
Physics informed solver (3.1)	Enhanced neural network	Transfers to other power grid instances. With adaptations, similar approach could be used in other safety-critical and physics-informed domains.	Transportation networks, water distribution systems, industrial process control.
Expert DeepQ (3.2)	Neural network	Transfers to other power system instances with the same state and action space. Adaptable to domain where similar insights are available.	Railway scheduling and air traffic management.
Evolving operator rules (3.3)	Expert system (in graph form) & decision rules	Transfers to other instances with the same state and action space. It can be applied with adaptations to other domains as long as expert systems can be represented as graphs	Rule-based industrial process control, expert systems to manage pumps in water networks, reconfiguration of telecommunications networks.
Planner enhanced AI (3.4)	Enhanced neural network	Transfers to other planning instances in the same domain. Method applicable to multi-agent path planning problems.	Warehouse robot fleet planning.
Maze-flatland (3.5)	Semi-hierarchical framework	Transfers to random maps in the same domain. Method can be applied with adaptation to domains which require planning and re-planning phases.	Air-traffic coordination, drone logistics, urban mobility networks.
State and action factorization (3.6)	Factorization	Transfers to other instances in same state-action space. Method can be used in any high-dimensional decision making problem.	Air-traffic management, railway scheduling.

TABLE 7 - (CONTINUED)

Results	Knowledge representation type	Potential to transfer knowledge to new domains	Examples
Network-distributed Q-learning (3.7)	Set of Q-value tables	Limited direct transfer. Applicable to other domains requiring decision making on nodes in a graph.	Train dispatching, vehicle rescheduling, job-shop scheduling, traffic flow management.
CommNet in multi-agent RL (3.8)	Communicatic network	Direct transfer to other train scheduling instances. Can be applied with adaptation to other domains requiring communication.	Steel manufacturing, air traffic control.
Multiclass failure prediction (3.9)	Failure clusters, light gradient-boosting machine	Direct transfer to other instances in the domain. Can be applied with adaptations to other infrastructure domains.	Railway networks, air traffic management.
Soft-target imitation learner (3.10)	Graph neural network	Direct transfer to other power grid instances. Method applicable with adaptation.	Railway domain, urban traffic signal control, software-defined networking routing.
Grid-explainer (3.11)	Explainability framework.	Applicable to power system instances. Underlying methodology domain agnostic (with visualization redevelopment).	Railway, air traffic management domains.
TraceRL (3.12)	Interpretation framework	Method applicable to methods with Gym or Maze-RL interface with discrete human input.	Power grid control, railway operations, air traffic management.

TABLE 7 - (CONTINUED)

Results	Knowledge representation type	Potential to transfer knowledge to new domains	Examples
Action alternative explainer (3.13)	Successor features	Limited direct transfer, method is domain agnostic so can in principle be applied to any domains with interpretable features.	Power grid control, railway operations, air traffic management.
Safe policy gradient (3.14)	Policy parameters	Transferable to other instances in training domain with same state and action space. Method is applicable to continuous and constrained control problems.	Power grid management, railway network operations.
MORL Ethical metric integration (3.15)	Metric assessment methodology	Applicable with modifications to other domains with trustworthiness metrics.	Power grid control, railway operations, air traffic management.
Human assessment model (3.16)	Human assessment methodology	Models are task and individual specific. Methodology is (with calibration and sensor integration) applicable to many safety-critical and cognitively demanding operational environment.	Power grid control, railway operations, air traffic management.
Sectorizatior interface (3.17)	Voronoi cell representation	No direct transfer. Method is applicable to domains requiring spatial partitioning under constraints.	Urban services zoning, telecommunications planning, logistics territory design.
Transparency requirements (3.18)	Human needs analysis	Domain agnostic analysis applicable to human-AI collaboration in complex, knowledge-intensive tasks.	Power grid control, railway operations, air traffic management.

REFERENCES

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian J. Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *CoRR*, abs/1810.03292, 2018. URL `http://arxiv.org/abs/1810.03292`.

[2] Daehwan Ahn, Abdullah Almaatouq, Monisha Gulabani, and Kartik Hosanagar. Will we trust what we don't understand? impact of model interpretability and outcome feedback on trust in ai. *arXiv preprint arXiv:2111.08222*, 2021.

[3] AI4REALNET. Railway network use case 1 – automated re-scheduling in railway operations. `https://ai4realnet.eu/portfolio-item/railway-network-use-case-1/`, 2024. Accessed: 2025-11-05.

[4] Stefano V. Albrecht, Filippos Christianos, and Lukas Schäfer. *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*. MIT Press, 2024.

[5] Eitan Altman. *Constrained Markov decision processes*. Routledge, 2021.

[6] David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks, 2018. URL `https://arxiv.org/abs/1806.07538`.

[7] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. Guidelines for human-AI interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13. ACM, 2019. ISBN 978-1-4503-5970-2. doi: 10.1145/3290605.3300233. URL `https://dl.acm.org/doi/10.1145/3290605.3300233`.

[8] and European Data Protection Supervisor. *AI Act Regulation (EU) 2024/1689 – Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act) (Text with EEA relevance)*. Publications Office of the European Union, 2025. doi: doi/10.2804/4225375.

[9] Gianluca Antonelli. Interconnected dynamic systems: An overview on distributed control. *IEEE Control Systems Magazine*, 33(1):76–88, 2013.

[10] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

[11] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE signal processing magazine*, 34(6):26–38, 2017.

[12] Akanksha Atrey, Kaleigh Clary, and David Jensen. Exploratory not explanatory: Counterfactual analysis of saliency maps for deep reinforcement learning. *arXiv preprint arXiv:1912.05743*, 2019.

[13] Dustin Axman, Joana S Paiva, Fernando de La Torre, and Joao PS Cunha. Beat-to-beat ecg features for time resolution improvements in stress detection. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 1290–1294. IEEE, 2017.

[14] Lisanne Bainbridge. Ironies of automation. *Analysis, design and evaluation of man–machine systems*, pages 129–135, 1983.

[15] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

[16] Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27 (4):819–840, 2002.

[17] Ricardo Bessa, Mouadh Yagoubi, and Milad Leili-abadi. Design of a holistic framework for ai in critical network infrastructures. Technical Report D1.1, AI4REALNET project, 2024.

[18] Umang Bhatt, Adrian Weller, and José MF Moura. Evaluating and aggregating feature-based model explanations. *arXiv preprint arXiv:2005.00631*, 2020.

[19] Clark Borst, John M. Flach, and Joost Ellerbroek. Beyond ecological interface design: Lessons from concerns and misconceptions. *IEEE Transactions on Human-Machine Systems*, 45:164–175, 2015. ISSN 2168-2291. doi: 10.1109/THMS.2014.2364984. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6963425`.

[20] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[21] Zana Buçinca. Optimizing decision-maker's intrinsic motivation for effective human-AI decision-making. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–5. ACM, 2024. ISBN 979-8-4007-0331-7. doi: 10.1145/3613905.3638179. URL https://dl.acm.org/doi/10.1145/3613905.3638179.

[22] Zana Buçinca, Siddharth Swaroop, Amanda E. Paluch, Finale Doshi-Velez, and Krzysztof Z. Gajos. Contrastive explanations that anticipate human misconceptions can improve human decision-making skills, 2025. URL http://arxiv.org/abs/2410.04253.

[23] Tathagata Chakraborti, Sarath Sreedharan, Yu Zhang, and Subbarao Kambhampati. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. In *International Joint Conference on Artificial Intelligence*, pages 156–163, 2017.

[24] Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. This looks like that: deep learning for interpretable image recognition. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.

[25] João Paulo TRIGUEIROS DA SILVA CUNHA and Joana Isabel SANTOS PAIVA. Biometric method and device for identifying a person through an electrocardiogram (ecg) waveform, January 5 2021. US Patent 10,885,361.

[26] Arun Das and Paul Rad. Opportunities and challenges in explainable artificial intelligence (xai): A survey. *arXiv preprint arXiv:2006.11371*, 2020.

[27] Peter Dayan and Geoffrey E Hinton. Feudal reinforcement learning. *Advances in neural information processing systems*, 5, 1992.

[28] Daniel Delahaye, Marc Schoenauer, and Jean-Marc Alliot. Airspace sectoring by evolutionary computation. In *1998 IEEE International Conference on Evolutionary Computation Proceedings*, pages 218–223, 06 1998. ISBN 0-7803-4869-9. doi: 10.1109/ICEC.1998.699504.

[29] Nerissa Dettling, Samira Hamouche, and Toni Waefler. Avoiding black box problems by assigning an active role to humans in the control of autonomous AI: A methodological approach. In *17th International Conference on Applied Human Factors and Ergonomics (AHFE 2026)*.

[30] Thomas G Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of artificial intelligence research*, 13:227–303, 2000.

[31] Gabriel Dulac-Arnold, Richard Evans, Hado van Hasselt, Peter Sunehag, Timothy Lillicrap, Jonathan Hunt, Timothy Mann, Theophane Weber, Thomas Degris, and Ben Coppin. Deep reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679*, 2015.

[32] Simon Eisbach, Markus Langer, and Guido Hertel. Optimizing human-AI collaboration: Effects of motivation and accuracy information in AI-supported decision-making. *Computers in Human Behavior: Artificial Humans*, 1(2):1–12, 2023. ISSN 29498821. doi: 10.1016/j.chbah.2023.100015. URL https://linkinghub.elsevier.com/retrieve/pii/S2949882123000154.

[33] Adam N Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68:9–26, 2022.

[34] M. R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995. doi: 10.1518/001872095779049543.

[35] Mica R Endsley. Ironies of artificial intelligence. *Ergonomics*, 66(11):1656–1668, 2023.

[36] Mica R. Endsley. Ironies of artificial intelligence. *Ergonomics*, 66(11):1656–1668, 2023. ISSN 0014-0139, 1366-5847. doi: 10.1080/00140139.2023.2243404. URL https://www.tandfonline.com/doi/full/10.1080/00140139.2023.2243404.

[37] enliteai. Maze-rl:applied reinforcement learning with python. https://github.com/enlite-ai/maze, 2025. URL https://github.com/enlite-ai/maze. GitHub repository, accessed 2025-11-18.

[38] Rodrigues et al. A wearable system for the stress monitoring of air traffic controllers during an air traffic control refresher training and the trier social stress test: A comparative study. *The Open Bioinformatics Journal*, 11(1):106–116, 2018. doi: 10.2174/1875036201811010106.

[39] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019. URL http://jmlr.org/papers/v20/18-760.html.

[40] John Flach. Supporting productive thinking: The semiotic context for cognitive systems engineering (CSE). *Applied Ergonomics*, 59:612–624, 2017. ISSN 0003-6870. doi: https://doi.org/10.1016/j.apergo.2015.09.001. URL https://www.sciencedirect.com/science/article/pii/S0003687015300739. The Legacy of Jens Rasmussen.

[41] Flatland Association. flatland-rl: Openai gym environment for railway management. https://github.com/flatland-association/flatland-rl, 2025. GitHub repository, accessed 2025-07-22.

[42] Steven Fortune. A sweepline algorithm for voronoi diagrams. *Algorithmica*, 2(1):153–174, 1987.

[43] Anton R Fuxjäger, Kristian Kozak, Matthias Dorfer, Patrick M Blies, and Marcel Wasserer. Reinforcement learning based power grid day-ahead planning and ai-assisted control. *arXiv preprint arXiv:2302.07654*, 2023.

[44] Artur d'Avila Garcez, Sebastian Bader, Howard Bowman, Luis C Lamb, Leo de Penning, BV Illuminoo, Hoifung Poon, and COPPE Gerson Zaverucha. Neural-symbolic learning and reasoning: A survey and interpretation. *Neuro-Symbolic Artificial Intelligence: The State of the Art*, 342(1): 327, 2022.

[45] Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

[46] Ingo Gerdes, Axel Temme, and Michael Schultz. Dynamic airspace sectorisation for flight-centric operations. *Transportation Research Part C: Emerging Technologies*, 95:460–480, 2018.

[47] Grid2Op Project. Grid2op documentation (latest). `https://grid2op.readthedocs.io/en/latest/`, 2025. Accessed documentation for the Grid2Op framework for sequential decision making in power systems.

[48] Richard J. Hackman and Greg R. Oldham. Motivation through the design of work: Test of a theory. *Organizational Behavior and Human Performance*, 16:250–279, 1976.

[49] Samira Hamouche, Nerissa Dettling, Julia Usher, Manuel Renold, and Toni Waefler. A methodical approach to AI-supported human learning in complex task environments. In *16th International Conference on Human Interaction & Emerging Technologies: Artificial Intelligence & Future Applications*.

[50] Samira Hamouche, Nerissa Dettling, and Toni Wafeler. Applying job design criteria for effective human-AI collaboration. In *Human Interaction and Emerging Technologies*, 197, 2025. doi: 10.54941/ahfe1006695.

[51] Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 709–715. AAAI Press / The MIT Press, 2004.

[52] Mohamed Hassouna, Clara Holzhüter, Malte Lehna, Matthijs de Jong, Jan Viebahn, Bernhard Sick, and Christoph Scholz. Learning topology actions for power grid control: A graph-based soft-label imitation learning approach. In Inês Dutra, Mykola Pechenizkiy, Paulo Cortez, Sepideh Pashami, Arian Pasquali, Nuno Moniz, Alípio M. Jorge, Carlos Soares, Pedro H. Abreu, and João Gama, editors, *Machine Learning and Knowledge Discovery in Databases. Applied Data Science*

Track and Demo Track, pages 129–146, Cham, 2026. Springer Nature Switzerland. ISBN 978-3-032-06129-4.

[53] Anna Hedström, Leander Weber, Sebastian Lapuschkin, and Marina MC Höhne. Sanity checks revisited: An exploration to repair the model parameter randomisation test, 2024. URL https://arxiv.org/abs/2401.06465.

[54] Pascal Hitzler, Aaron Eberhart, Monireh Ebrahimi, Md Kamruzzaman Sarker, and Lu Zhou. Neuro-symbolic approaches in artificial intelligence. *National Science Review*, 9(6):nwac035, 2022.

[55] AI HLEG. Assessment list for trustworthy artificial intelligence, 2020.

[56] Robert R. Hoffman, Shane T. Mueller, Gary Klein, and Jordan Litman. Metrics for explainable AI: Challenges and prospects, 2018. URL http://arxiv.org/abs/1812.04608.

[57] Erik Hollnagel and David D. Woods. *Joint Cognitive Systems: Foundations of Cognitive Systems Engineering*. CRC Press, 2005. ISBN 978-0-429-12222-4. doi: 10.1201/9781420038194.

[58] Y. Hu, B. Chen, and K. Tang. L2rpn_nips_2020_a_ppo_solution: Learning to run a power network (l2rpn) – ppo solution, 2020. URL https://github.com/AsprinChina/L2RPN_NIPS_2020_a_PPO_Solution. GitHub repository for a PPO-based solution to the NeurIPS 2020 L□earning to Run a Power Networkc□ompetition (L2RPN) robustness track; ranked 2nd.

[59] Taoan Huang, Sven Koenig, and Bistra Dilkina. Learning to resolve conflicts for multi-agent path finding with conflict-based search. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11246–11253, 2021.

[60] Huchler, N.,, Adolph, L., André, E., Bauer, W., Bender, N., Müller, N., ..., and Suchy, O. Kriterien für die mensch-maschine-interaktion bei KI. ansätze für die menschengerechte gestaltung in der arbeitswelt, 2020. URL https://www.plattform-lernende-systeme.de/files/Downloads/Publikationen/AG2_Whitepaper2_220620.pdf.

[61] Yuhao Jiang, Kunjie Zhang, Qimai Li, Jiaxin Chen, and Xiaolong Zhu. Multi-agent path finding via tree lstm. *arXiv preprint arXiv:2210.12933*, 2022.

[62] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[63] Adrian Kelly, Aidan O'Sullivan, Patrick de Mars, and Antoine Marot. Reinforcement learning for electricity network operation. *arXiv preprint arXiv:2003.07339*, 2020.

[64] Omar Khan, Pascal Poupart, and James Black. Minimal sufficient explanations for factored markov decision processes. In *Proceedings of the international conference on automated planning and scheduling*, volume 19, pages 194–200, 2009.

[65] Joseph Kim, Christian Muise, Ankit Jayesh Shah, Shubham Agarwal, and Julie A Shah. Bayesian inference of linear temporal logic specifications for contrastive explanations. In *International Joint Conferences on Artificial Intelligence*, 2019.

[66] Gary Klein. Macrocognitive measures for evaluating cognitive work. In *Macrocognition Metrics and Scenarios*, pages 47–64. CRC Press, 2018.

[67] A. Y. Kolb and D. A. Kolb. The learning way: Meta-cognitive aspects of experiential learning. *Simulation & Gaming*, 40(3):297–327, 2009. doi: 10.1177/1046878108325713.

[68] Leonardo Lamorgese, Carlo Mannino, Dario Pacciarelli, and Johanna Törnquist Krasemann. Train dispatching. *Handbook of optimization in the railway industry*, pages 265–283, 2018.

[69] Florian Laurent, Manuel Schneider, Christian Scheller, Jeremy Watson, Jiaoyang Li, Zhe Chen, Yi Zheng, Shao-Hung Chan, Konstantin Makhnev, Oleg Svidchenko, et al. Flatland competition 2020: Mapf and marl for efficient train coordination on a grid world. In *NeurIPS 2020 Competition and Demonstration Track*, pages 275–301. PMLR, 2021.

[70] John D Lee and Katrina A See. Trust in automation: Designing for appropriate reliance. *Human Factors*, 2004.

[71] Malte Lehna, Jan Viebahn, Antoine Marot, Sven Tomforde, and Christoph Scholz. Managing power grids through topology actions: A comparative study between advanced rule-based and reinforcement learning agents. *Energy and AI*, 14:100276, 2023.

[72] Malte Lehna, Mohamed Hassouna, Dmitry Degtyar, Sven Tomforde, and Christoph Scholz. Fault detection for agents on power grid topology optimization: A comprehensive analysis, 2024. URL `https://arxiv.org/abs/2406.16426`.

[73] Bruno Lemetayer, Luca Saporetti, Manuel Schneider, Ricardo Bessa, and Roman Liessner. Evaluation and test protocols. Technical Report D4.1, 2025.

[74] Milad Leyli-Abadi, Antoine Marot, and Jérôme Picault. Study design and demystification of physics informed neural networks for power flow simulation. *arXiv preprint arXiv:2509.19233*, 2025.

[75] Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. Trustworthy ai: From principles to practices. *ACM Computing Surveys*, 55(9):1–46, aug 2022.

[76] Jing-Quan Li, Pitu B Mirchandani, and Denis Borenstein. The vehicle rescheduling problem: Model and algorithms. *Networks (N. Y.)*, 50(3):211–229, October 2007.

[77] Nan Lin, Stavros Orfanoudakis, Nathan Ordonez Cardenas, Juan S Giraldo, and Pedro P Vergara. Powerflownet: Power flow approximation using message passing graph neural networks. *International Journal of Electrical Power & Energy Systems*, 160:110112, 2024.

[78] Zhengxian Lin, Kin-Ho Lam, and Alan Fern. Contrastive explanations for reinforcement learning via embedded self predictions. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Ud3DSz72nYR.

[79] Xuan Liu and Yuxuan Li. Examining the double-edged sword effect of AI usage on work engagement: The moderating role of core task characteristics substitution. *Behavioral Sciences*, 15 (2):206, 2025. ISSN 2076-328X. doi: 10.3390/bs15020206. URL https://www.mdpi.com/2076-328X/15/2/206.

[80] Gianvito Losapio, Davide Beretta, Marco Mussi, Alberto Maria Metelli, and Marcello Restelli. State and action factorization in power grids. *arXiv preprint arXiv:2409.04467*, 2024.

[81] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions, 2017. URL https://arxiv.org/abs/1705.07874.

[82] Jerry Luo, Cosmin Paduraru, Octavian Voicu, Yuri Chervonyi, Scott Munns, Jerry Li, Crystal Qian, Praneet Dutta, Jared Quincy Davis, Ningjia Wu, et al. Controlling commercial cooling systems using reinforcement learning. *arXiv preprint arXiv:2211.07357*, 2022.

[83] Shingo Mabu, Kotaro Hirasawa, Masanao Obayashi, and Takashi Kuremoto. Enhanced decision making mechanism of rule-based genetic network programming for creating stock trading signals. *Expert Systems with Applications*, 40(16):6311–6320, 2013. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2013.05.037.

[84] A. Marot, B. Donnot, S. Tazi, and P. Panciatici. Expert system for topological remedial action discovery in smart grids. In *Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion (MEDPOWER 2018)*, pages 1–6, 2018. doi: 10.1049/cp.2018.1875.

[85] A. Marot, G. Donnot, B.and Dulac-Arnold, A. Kelly, A. O'Sullivan, J. Viebahn, M. Awad, I.M. Guyon, P. Panciatici, and C. Romero. Learning to run a power network challenge: a retrospective analysis. *Proceedings of Machine Learning Research (NeurIPS 2020 Competition and Demonstration Track)*, 133:112–132, 2021. doi: https://doi.org/10.48550/arXiv.2103.03104.

[86] Antoine Marot, Benjamin Donnot, Sami Tazi, and Patrick Panciatici. Expert system for topological remedial action discovery in smart grids. In *Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion (MEDPOWER 2018)*, pages 1–6. IET, 2018.

[87] Antoine Marot, Sami Tazi, Benjamin Donnot, and Patrick Panciatici. Guided machine learning for power grid segmentation. In *2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–6. IEEE, 2018.

[88] Louise McCormack and Malika Bendechache. A comprehensive survey and classification of evaluation criteria for trustworthy artificial intelligence. *AI and Ethics*, 5(3):1973–1994, jul 2025. doi: 10.1007/s43681-024-00590-8. URL `https://doi.org/10.1007/s43681-024-00590-8`.

[89] Dang Minh, H Xiang Wang, Y Fen Li, and Tan N Nguyen. Explainable artificial intelligence: a comprehensive review. *Artificial Intelligence Review*, 55(5):3503–3568, 2022.

[90] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[91] Alessandro Montenegro, Marco Mussi, Matteo Papini, and Alberto Maria Metelli. Last-iterate global convergence of policy gradients for constrained reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, pages 126363–126416. Curran Associates, Inc., 2024. doi: 10.52202/079017-4014.

[92] Frederick P. Morgeson and Stephen E. Humphrey. The work design questionnaire (WDQ): Developing and validating a comprehensive measure for assessing job design and the nature of work. *Journal of Applied Psychology*, 91(6):1321–1339, 2006. ISSN 1939-1854, 0021-9010. doi: 10.1037/0021-9010.91.6.1321. URL `https://doi.apa.org/doi/10.1037/0021-9010.91.6.1321`.

[93] Marco Mussi, Gianvito Losapio, Alberto Maria Metelli, and Marcello Restelli. Position paper on ai for the operation of critical energy and mobility network infrastructures. Technical Report D2.1, AI4REALNET project, 2024.

[94] Marco Mussi, Alberto Maria Metelli, Marcello Restelli, Gianvito Losapio, Ricardo J Bessa, Daniel Boos, Clark Borst, Giulia Leto, Alberto Castagna, Ricardo Chavarriaga, et al. Human-ai interaction in safety-critical network infrastructures. *IScience*, 28(9), 2025.

[95] Mirco Mutti, Lorenzo Pratissoli, and Marcello Restelli. Task-agnostic exploration via policy gradient of a non-parametric state entropy estimate. *Proceedings of the AAAI Conference on*

Artificial Intelligence, 35(10):9028–9036, May 2021. doi: 10.1609/aaai.v35i10.17091. URL https://ojs.aaai.org/index.php/AAAI/article/view/17091.

[96] Matthew L Olson, Roli Khanna, Lawrence Neal, Fuxin Li, and Weng-Keen Wong. Counterfactual state explanations for reinforcement learning agents via generative deep learning. *Artificial Intelligence*, 295:103455, 2021.

[97] Joana S Paiva, Susana Rodrigues, and Joao Paulo Silva Cunha. Changes in st, qt and rr ecg intervals during acute stress in firefighters: A pilot study. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3378–3381. IEEE, 2016.

[98] Sharon K. Parker and Gudela Grote. Automation, algorithms, and beyond: Why work design matters more than ever in a digital world. *Applied Psychology*, 71(4):1171–1204, 2022. ISSN 0269-994X, 1464-0597. doi: 10.1111/apps.12241. URL https://iaap-journals.onlinelibrary.wiley.com/doi/10.1111/apps.12241.

[99] Sharon K Parker and Caroline Knight. The smart model of work design: A higher order structure to help see the wood from the trees. *Human Resource Management*, 63(2):265–291, 2024.

[100] Shubham Pateria, Budhitama Subagdja, Ah-hwee Tan, and Chai Quek. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 54(5):1–35, 2021.

[101] Adrien Pavão, Antoine Marot, Jules Sintes, Viktor Eriksson Möllerstedt, Laure Crochepierre, Karim Chaouache, Benjamin Donnot, Van Tuan Dang, and Isabelle Guyon. Ai challenge for safe and low carbon power grid operation. *Energy and AI*, page 100564, 2025.

[102] Marin Vlastelica Pogančić, Anselm Paulus, Vit Musil, Georg Martius, and Michal Rolinek. Differentiation of blackbox combinatorial solvers. In *International Conference on Learning Representations*, 2019.

[103] Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

[104] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[105] Susana Rodrigues, Joana S. Paiva, Duarte Dias, Marta Aleixo, Rui Manuel Filipe, and João Paulo S. Cunha. Cognitive impact and psychophysiological effects of stress using a biomonitoring platform. *International Journal of Environmental Research and Public Health*, 15(6), 2018. ISSN

1660-4601. doi: 10.3390/ijerph15061080. URL https://www.mdpi.com/1660-4601/15/6/1080.

[106] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, May 2019. ISSN 2522-5839. doi: 10.1038/s42256-019-0048-x. URL https://doi.org/10.1038/s42256-019-0048-x.

[107] Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J Anders, and Klaus-Robert Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of the IEEE*, 109(3):247–278, 2021.

[108] Md Kamruzzaman Sarker, Lu Zhou, Aaron Eberhart, and Pascal Hitzler. Neuro-symbolic artificial intelligence. *AI Communications*, 34(3):197–209, 2021.

[109] Albrecht Schmidt and Thomas Herrmann. Intervention user interfaces: a new interaction paradigm for automated systems. *interactions*, 24(5):40–45, 2017. ISSN 1072-5520. doi: 10.1145/3121357. URL https://dl.acm.org/doi/10.1145/3121357.

[110] Stefan Schneider, Anirudha Ramesh, Anne Roets, Ciprian Stirbu, Farhad Safaei, Faten Ghriss, Jan Wülfing, Mehmet Güra, Nima Sibon, Rick Gentry, et al. Intelligent railway capacity and traffic management using multi-agent deep reinforcement learning. In *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1743–1748. IEEE, 2024.

[111] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[112] Fred Shaffer and Jay P Ginsberg. An overview of heart rate variability metrics and norms. *Frontiers in public health*, 5:258, 2017.

[113] L. Shapley. A value for n-person games. In *Classics in game theory*. Princeton University Press, 1953.

[114] Lloyd S Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10): 1095–1100, 1953.

[115] Guni Sharon, Roni Stern, Ariel Felner, and Nathan R Sturtevant. Conflict-based search for optimal multi-agent pathfinding. *Artificial intelligence*, 219:40–66, 2015.

[116] Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, Niklas Elmqvist, and Nicholas Diakopoulos. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Pearson, 2016. ISBN 978-0-13-438038-4.

[117] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Learning important features through propagating activation differences, 2017. URL https://arxiv.org/abs/1605.01713.

[118] David Silver. Cooperative pathfinding. In *Proceedings of the aaai conference on artificial intelligence and interactive digital entertainment*, volume 1, pages 117–122, 2005.

[119] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

[120] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[121] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[122] Neville A. Stanton, Paul M. Salmon, Laura A. Rafferty, Guy H. Walker, Chris Baber, and Daniel P. Jenkins. *Human Factors Methods: A Practical Guide for Engineering and Design*. CRC Press, London, 2 edition, September 2017. ISBN 978-1-315-58739-4. doi: 10.1201/9781315587394.

[123] Toni Steimle and Dieter Wallach. *Collaborative UX Design: Lean UX und Design Thinking: teambasierte Entwicklung menschzentrierter Produkte*. dpunkt.verlag, 2., aktualisierte und erweiterte auflage edition, 2023. ISBN 978-3-86490-881-1.

[124] Ilia Stepin, Jose M Alonso, Alejandro Catala, and Martín Pereira-Fariña. A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access*, 9:11974–12001, 2021.

[125] Roni Stern, Nathan Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, TK Kumar, et al. Multi-agent pathfinding: Definitions, variants, and benchmarks. In *Proceedings of the International Symposium on Combinatorial Search*, volume 10, pages 151–158, 2019.

[126] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2244–2252, dec 2016. URL https://proceedings.neurips.cc/paper/2016/hash/

55b1927fdafef39c48e5b73b5d61ea60-Abstract.html. Presented at the 30th International Conference on Neural Information Processing Systems (NIPS/NeurIPS 2016), Barcelona, Spain.

[127] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[128] Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

[129] Eberhard Ulich. *Arbeitspsychologie*. vdf Hochschulverlag AG, 7. auflage edition, 2011. ISBN 978-3-7281-4042-5.

[130] Michelle Vaccaro, Abdullah Almaatouq, and Thomas Malone. When combinations of humans and AI are useful: A systematic review and meta-analysis. *Nature Human Behaviour*, 8(12): 2293–2303, 2024. ISSN 2397-3374. doi: 10.1038/s41562-024-02024-1. URL https://www.nature.com/articles/s41562-024-02024-1.

[131] Karel van den Bosch, Emma M. Van Zoelen, Tjeerd A. J. Schoonderwoerd, Anthia Solaki, Birgit van der Stigchel, and Ivana Akrum. Design and effects of co-learning in human-AI teams. *Journal of Artificial Intelligence Research*, 82:1445–1493, 2025. doi: https://doi.org/10.1613/jair.1.1684.

[132] Erica van der Sar, Alessandro Zocca, and Sandjai Bhulai. Optimizing power grid topologies with reinforcement learning: A survey of methods and challenges. *Foundations and Trends in Electric Energy Systems*, 9(1):1–119, aug 2025. doi: 10.1561/3100000048. URL https://doi.org/10.1561/3100000048.

[133] Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596*, 2, 2020.

[134] Kim J. Vicente. *Cognitive Work Analysis: Towards Safe, Productive, and Healthy Computer-Based Work*. L. Erlbaum Associates Inc., USA, 1999. ISBN 0805823964.

[135] Kim J Vicente and Jens Rasmussen. Ecological interface design: Theoretical foundations. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(4):589–606, 1992.

[136] E. Vilches. Ai4realnet-morl: Multi-objective reinforcement learning for grid2op, jan 2024. URL https://gitlab.inesctec.pt/cpes/european-projects/ai4realnet/ukassel/

`ai4realnet-morl`. Research software developed within the AI4REALNET European project for multi-objective reinforcement learning applied to power grid environments using Grid2Op.

[137] Laura Von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, et al. Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):614–633, 2021.

[138] Andreas Voss, Rico Schroeder, Andreas Heitmann, Annette Peters, and Siegfried Perz. Short-term heart rate variability—influence of gender and age in healthy subjects. *PloS one*, 10(3): e0118308, 2015.

[139] Toni Waefler, Samira Hamouche, and Andrina Eisenegger. The supportive AI framework: From recommending to supporting. In Dylan D. Schmorrow and Cali M. Fidopiastis, editors, *Augmented Cognition*, pages 303–317. Springer Nature Switzerland, 2025. ISBN 978-3-031-93724-8. doi: 10.1007/978-3-031-93724-8_22.

[140] Mattias Wahde and Marco Virgolin. The five is: key principles for interpretable and safe conversational ai. In *Proceedings of the 2021 4th International Conference on Computational Intelligence and Intelligent Systems*, pages 50–54, 2021.

[141] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

[142] Toni Wäfler, Gudela Grote, Anna Windischer, and Cornelia Ryser. KOMPASS: A method for complementary system design. In *Handbook of cognitive task design*, Human factors and ergonomics, pages 477–502. Lawrence Erlbaum Associates Publishers, 2003. ISBN 978-0-8058-4003-2. doi: 10.1201/9781410607775.ch20.

[143] Hegen Xiong, Shuangyuan Shi, Danni Ren, and Jinjin Hu. A survey of job shop scheduling problem: The types and models. *Computers & Operations Research*, 142:105731, 2022.

[144] Min Xue. Airspace sector redesign based on voronoi diagrams. *Journal of Aerospace Computing, Information, and Communication*, 6(12):624–634, 2009. doi: 10.2514/1.41159.

[145] Herman Yau, Chris Russell, and Simon Hadfield. What did you think would happen? explaining agent behaviour through intended outcomes. *Advances in Neural Information Processing Systems*, 33:18375–18386, 2020.

[146] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Suggala, David I Inouye, and Pradeep K Ravikumar. On the (in) fidelity and sensitivity of explanations. *Advances in Neural Information Processing Systems*, 32, 2019.

[147] Deunsol Yoon, Sunghoon Hong, Byung-Jun Lee, and Kee-Eung Kim. Winning the l2{rpn} challenge: Power grid management via semi-markov afterstate actor-critic. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LmUJqB1Cz8.

[148] Dongran Yu, Bo Yang, Dayou Liu, Hui Wang, and Shirui Pan. A survey on neural-symbolic learning systems. *Neural Networks*, 2023.

[149] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pages 321–384, 2021.

[150] Shuyang Zhang, Jiaoyang Li, Taoan Huang, Sven Koenig, and Bistra Dilkina. Learning a priority ordering for prioritized planning in multi-agent path finding. In *Proceedings of the International Symposium on Combinatorial Search*, volume 15, pages 208–216, 2022.

[151] Yuan Zhang, Umashankar Deekshith, Jianhong Wang, and Joschka Boedecker. Improving the efficiency and efficacy of multi-agent reinforcement learning on complex railway networks with a local-critic approach. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 34, pages 698–706, 2024.

[152] Bo Zhou, Hongsheng Zeng, Yuecheng Liu, Kejiao Li, Fan Wang, and Hao Tian. Action set based policy optimization for safe power grid management. In *ECML/PKDD*, 2021. doi: https://doi.org/10.48550/arXiv.2106.15200.

[153] Changxi Zhu, Mehdi Dastani, and Shihan Wang. A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems*, 38 (1):4, jan 2024. doi: 10.1007/s10458-023-09633-6. URL https://doi.org/10.1007/s10458-023-09633-6.

[154] Patrick Zinsli, Stephanie Kalt, Nerissa Dettling, Samira Hamouche, and Toni Waefler. Augmented cognition requires a psychologically sound human role: a methodical approach. In *The Applied Human Factors and Ergonomics International Conference (AHFE)*.

[155] Yiyuan Zou and Clark Borst. Towards a unified taxonomy for algorithmic transparency: insights from uncrewed air traffic management. *Cognition, Technology & Work*, 9 2025. ISSN 1435-5558. doi: 10.1007/s10111-025-00826-5. URL https://link.springer.com/10.1007/s10111-025-00826-5.

[156] Île-de-France Region and RTE. Paris region ai challenge for energy transition: Low-carbon grid operations, 2023. URL https://www.iledefrance.fr/sites/default/files/medias/2023/05/Description_Challenge_RTE.pdf. Accessed: 2025-10-03.