CEAS Aeronautical Journal
https://doi.org/10.1007/s13272-026-01018-1

ORIGINAL PAPER

Constrained scenario adaptation in flight simulator training using conceptualized state spaces

H. Lilla[1] · **O. Niggemann**[2] · **T. Netzel**[1]

Received: 31 May 2026 / Revised: 31 May 2026 / Accepted: 22 September 2026
© The Author(s) 2026

Abstract
The EASA AI Roadmap 2.0 establishes trustworthiness as the backbone of AI readiness in aviation, requiring that machine learning is not only accurate, but also deterministic, traceable to formally defined concepts, bounded by explicit safety constraints, and reconstructable post-hoc. Existing approaches treat machine learning as an unconstrained predictor over a latent feature space and apply constraints only as post-hoc filters. The central methodological problem remains open: a formal mapping from heterogeneous subsymbolic training evidence to admissible scenario adaptations that preserves these properties by construction. This paper formulates and realizes such a mapping, decomposable into formally defined substeps that satisfy four structural properties: determinism, semantic grounding, constraint compliance, and reconstructability. A realization for cyber-physical scenario-based flight training is presented. Its core is a deterministic conceptualized state space based on Deep-Linking Keywords that links theory-based assessment and simulator-based performance. A deterministic finite automaton governs instructional progression. Scenario adaptation is approximated by a constrained machine learning policy within an admissible action space enforced by monotonicity constraints, phase gating, and a projection operator that clips outputs to the admissible action space. The architecture is analytically verified by construction against the four properties and demonstrated empirically on N=160 policy cases using k=5 group-wise cross-validation. The constrained machine learning component approximates the deterministic baseline policy with a profile Mean Absolute Error (MAE) of 0.126 ± 0.024 and a $+0.344$ improvement in scenario-focus classification accuracy over the majority-class baseline. Statistical generalization beyond this proof-of-concept scope is identified as the next step.

Keywords Conceptualized state spaces · Constrained machine learning · Explainable AI · Cyber-physical systems · Flight simulator training

1 Introduction

The EASA AI Roadmap 2.0 identifies trustworthiness as the central framework for AI readiness in aviation, requiring that machine learning (ML) is not only accurate, but also deterministic, interpretable, and auditable [1–3]. The EASA trustworthiness building blocks, which comprise trustworthiness analysis, AI assurance, human factors for AI, and AI safety risk mitigation, therefore make clear that AI-based applications must provide transparent, verifiable, and operationally meaningful evidence of trustworthy behavior [1–3]. This introduces a fundamental challenge: ML must be embedded into system architectures that comply with strict safety constraints and support transparent decision-making.

In pilot training, frameworks such as Competency-Based Training and Assessment (CBTA) [4] and Evidence-Based Training (EBT) [5–7] emphasize the development of holistic competencies rather than isolated task performance, which is increasingly discussed in recent aviation training research [8]. This paradigm is reflected in regulatory and curricular documents such as EASA Part-ORA [9], Part-FCL [10], the Flight Examiners' Manual [11], and the ATP(A) Integrated Course Manual [12], and is operationalized in structured training phases such as those illustrated in Fig. 1. However, the technical realization of competence-oriented training in

✉ T. Netzel
thomas.netzel@haw-hamburg.de

[1] Faculty of Aviation and Automotive Systems, Hamburg University of Applied Sciences, Berliner Tor 9, 20099 Hamburg, Germany

[2] Institute of Automation Technology, Helmut-Schmidt-University, Holstenhofweg 85, 22043 Hamburg, Germany

Published online: 16 November 2026

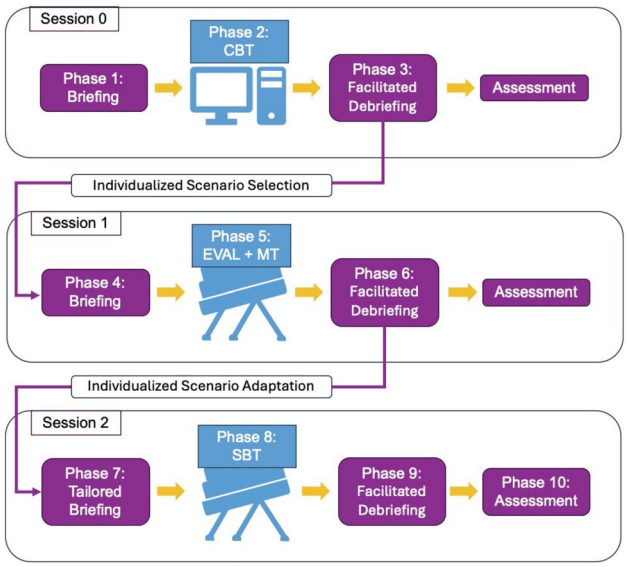


Fig. 1 Structured representation of the EBT learning phases according to the EBT Manual [5]. The proposed CPS architecture extends this framework by introducing a preceding CBT-based theory phase, enabling a consistent linkage between knowledge assessment and simulator-based performance within a unified training process

data-driven systems remains insufficiently formalized. In particular, there is still no consistent method to transform heterogeneous data sources, such as theory-based assessments and simulator telemetry, into a unified, deterministic, and semantically consistent system representation.

Conceptualization approaches for cyber-physical systems (CPS) [13] provide a principled foundation for transforming subsymbolic data into symbolic state representations. In the present context, this idea is extended through the use of Deep-Linking Keywords [14], which create a shared semantic reference between theory-based knowledge work and simulator-based performance. However, existing conceptualization approaches do not explicitly address how such a semantically grounded state space can serve as the basis for integrating ML under safety-critical constraints.

This leads to a central problem: there is no formally defined transformation from heterogeneous training evidence to admissible scenario adaptation that simultaneously ensures determinism, semantic grounding, constraint compliance, and reconstructability. Formally, this work seeks a mapping $\Phi : \mathcal{D}_{\text{raw}} \to \Delta_{\text{scenario}}$ that transforms heterogeneous raw measurements, including theory-based assessment scores and simulator telemetry, into admissible adaptations of the flight simulator scenario, decomposable into formally defined substeps that jointly satisfy four structural properties: (P1) determinism, (P2) semantic grounding to a finite concept set $\mathcal{K}$, (P3) constraint compliance to an admissible policy space Π_c, and (P4) reconstructability from a persistent decision trace τ. The constructive realization of Φ is developed in Sect. 3.

Against this background, this work addresses the following research questions:

RQ1: Can heterogeneous theory-based and simulator-based training evidence be transformed via Deep-Linking Keywords into a deterministic, semantically grounded, and compact ML feature space?

RQ2: Can an ML policy be trained to approximate the decision structure of a rule-based baseline policy while constraint compliance is guaranteed by construction?

RQ3: Do semantic interpretability and auditability follow structurally from the conceptualized state-space representation, through deterministic state construction and persistent decision traces respectively?

This work builds directly on the CPS prototype introduced in [14]. There, Deep-Linking Keywords served as a semantic interface between theory-based assessment and simulator-based performance, and scenario adaptation operated on heuristic KSA aggregation through rule-based scoring. The keyword-based representation was not yet formalized as a deterministic state space, and the adaptation logic was not embedded in a structurally constrained policy framework.

The present work closes these methodological gaps. The transition from heterogeneous raw evidence to admissible scenario adaptation is formulated as the deterministic mapping Φ with explicit structural properties. The state space derived from this formulation is then coupled with a constrained policy approximation. The flight simulator setup introduced in our preceding work [14] serves as the running instance; the contribution targets a general methodological pattern for cyber-physical training systems under safety-critical AI constraints.

This paper makes four contributions.

First, the transition from heterogeneous training evidence to admissible scenario adaptation is formulated as a deterministic mapping $\Phi : \mathcal{D}_{\text{raw}} \to \Delta_{\text{scenario}}$. The mapping is decomposed into formally defined substeps and characterized by four structural properties: determinism, semantic grounding, constraint compliance, and reconstructability.

Second, the cyber-physical conceptualization approach of Niggemann et al. [13] is extended by a deterministic three-stage state-space construction consisting of discretization, semantic assignment, and aggregation. This construction lifts the heuristic Deep-Linking Keyword aggregation from our preceding work [14] into a formal symbolic learner representation suitable for diagnosis, learning, and audit.

Third, ML is embedded into this representation as a constrained policy approximation $\pi_{\text{ML}} \in \Pi_c$. Admissibility is enforced by construction through monotonicity constraints, deterministic instructional control, and a projection operator onto an admissible action space.

Fourth, interpretability and auditability are integrated as architectural properties rather than post-hoc add-ons: semantic grounding (P2) follows from the structure of the conceptualized state space, and reconstructability (P4) follows from persistent decision traces over deterministically constructed intermediate representations.

The framework is instantiated for stall recognition and recovery training within the EBT/CBTA paradigm and empirically validated as a proof-of-concept (Sect. 4).

2 State of the art

The state of the art relevant to this work can be structured into four research strands: regulatory and pedagogical frameworks for adaptive pilot training (see Fig. 1), data-driven approaches for scenario adaptation, explainability-oriented approaches for safety-critical AI, and formal and conceptual modeling approaches. Taken together, these strands show that important partial solutions exist, but that the research questions formulated in Sect. 1 are not yet finally answered.

2.1 Regulatory and pedagogical frameworks for adaptive pilot training

Competency-Based Training and Assessment (CBTA) [4] and Evidence-Based Training (EBT) [5–7] define the current paradigm of pilot training by shifting the focus from fixed hour-based qualification models toward competence-oriented training and assessment [11].

The increasing relevance of competence-oriented pilot training is also reflected in recent aviation training research [8, 15–17]. This paradigm is formalized in regulatory and curricular documents such as EASA Part-FCL [10], Part-ORA [9], the ATP(A) Integrated Course Manual [12], and associated examiner guidance such as the Flight Examiners' Manual [11]. These frameworks specify structured training phases such as Evaluation and Maneuvers Training (EVAL + MT) and Scenario-Based Training (SBT), summarized in Fig. 1, but they do not provide a formal computational model for representing competence states, integrating heterogeneous learner data, or embedding ML into deterministic adaptation processes.

In parallel, computer-based training has become an established component of theoretical pilot training and enables the systematic acquisition of declarative, procedural, and reflective knowledge, encompassing strategic and adaptive knowledge components [6, 18]. This creates the didactic basis for linking theory and simulator training more tightly in adaptive competence-oriented systems.

2.2 Data-driven approaches for adaptive scenario generation

A second research strand addresses data-driven adaptation of training scenarios. Existing work includes classical ML approaches [19] and tree-based boosting methods [20–22], deep learning [23, 24], and reinforcement-learning-based approaches [25]. These approaches show that training scenarios can in principle be adapted on the basis of learner behavior and performance data. Other systems use AI-based methods to support pilot training workflows more broadly, for example in wind-related training support, adaptive trainer systems, intervention-oriented learning environments, or automated scenario-generation pipelines [19, 25–29].

However, many existing adaptive learning architectures implicitly treat ML as a direct mapping from telemetry to adaptation decisions. As a consequence, the learned representation often remains latent, weakly interpretable, and insufficiently connected to instructional semantics. Existing approaches therefore primarily demonstrate predictive or adaptive capability rather than formally constrained integration into semantically grounded and safety-critical system architectures.

In particular, most approaches presuppose an already available feature space and rarely address how this feature space should be constructed in a semantically grounded, deterministic, and audit-ready manner.

2.3 Explainability-oriented approaches in safety-critical AI

A third research strand concerns explainability and trustworthy AI. In aviation, explainability is increasingly treated as a regulatory requirement rather than an optional model property [1–3]. In ML research, concept-based methods such as Concept Bottleneck Models [30] and TCAV [31] show that explanations become more meaningful when they are linked to human-understandable concepts. In parallel, post-hoc feature-attribution methods such as SHAP [32, 33] have been used to increase transparency in aviation-related adaptive systems [19, 34].

Nevertheless, many current approaches still treat explainability primarily as a post-hoc analytical layer rather than as an integral part of the underlying system architecture and decision logic. This is increasingly viewed as insufficient in safety-critical aviation contexts [3, 35]. The relevant trustworthiness building blocks defined by EASA are summarized in Fig. 2.

At the same time, research in explainable aviation systems increasingly stresses that explanations must be embedded into operational contexts and user communication

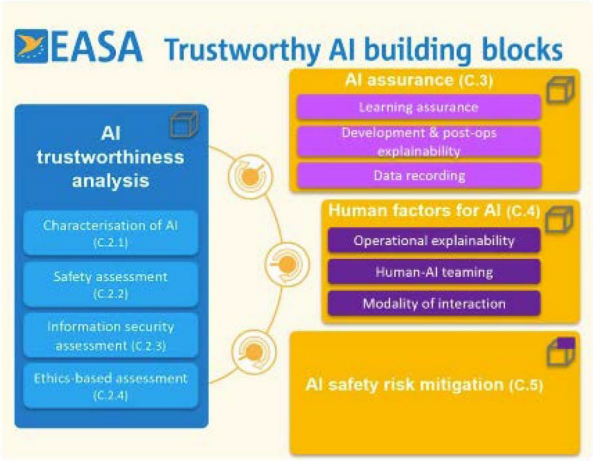


Fig. 2 Overview of the building blocks for trustworthy AI in aviation defined by EASA [3]. The figure provides the regulatory frame for formal models, explainable policies, and auditability

rather than treated as purely technical visualizations [36]. Related work also points to technical, ethical, and human-factors-related barriers, including reliability, bias, privacy, and user trust [37]. Degas et al. [35] further emphasize that existing XAI approaches are often primarily developer-oriented and insufficiently aligned with the operational needs of human decision-makers in aviation. However, most current approaches still do not structurally embed explanation into the state space, transition logic, or adaptation mechanism itself.

Beyond post-hoc explainability, a parallel line of research investigates how constraint compliance can be enforced within the ML architecture itself. In the aviation domain, Fenn et al. [38] have explicitly formulated safety as an architectural property and proposed property-driven design patterns for AI-enabled autonomous systems. At the architecture level, constraint-enforcing network designs guarantee output admissibility by construction rather than by regularization [39, 40]. At the inference level, predictive safety filters replace inadmissible actions with safe alternatives at runtime, ensuring constraint compliance for learning-based control of nonlinear dynamical systems [41]. However, these approaches typically presuppose an already given feature space and do not address how such a feature space should be constructed in a semantically grounded and deterministic manner from heterogeneous subsymbolic evidence.

2.4 Formal and conceptual modeling approaches

A fourth research strand addresses formal and conceptual system modeling. Conceptualization approaches for CPS describe how subsymbolic data can be transformed into symbolic representations suitable for structured system analysis and control [13]. Related ontology- and semantics-oriented work further emphasizes the importance of conceptually grounded representations for interoperable, explainable, and reusable CPS architectures [42, 43]. In educational and adaptive systems, deterministic finite automata and related formal control models have been used to represent learning progressions and state-dependent adaptation behavior [44–47].

However, AI-based flight training systems currently do not provide a formal DFA-based model that is simultaneously didactically interpretable, compatible with ML, and aligned with regulatory AI assurance requirements.

More generally, rule extraction and automata-oriented representations have been studied as a means to translate subsymbolic models into formally analyzable control structures [48–50].

In a complementary direction, validation frameworks for approximated propositional logic system descriptions have been developed that introduce formal metrics for the diagnosability, co-similarity, and expressivity of learned symbolic models in model-based fault diagnosis [51].

Recent work in adaptive flight simulator training likewise emphasizes the need for semantically grounded and interoperable scenario representations [52]. However, these approaches focus primarily on ontology-oriented scenario description and interoperability rather than on deterministic learner-state construction or constrained ML integration.

For the present work, the key point is that symbolic state modeling and data-driven adaptation are still rarely integrated. Niggemann et al. [13] explicitly identify the transition from subsymbolic information to symbolic and semantically interpretable concepts as a central unresolved problem, especially the discretization of continuous signals into states and the semantic assignment of these states to concepts.

A first domain-specific step toward such an integration is represented by a CPS-based framework for AI-driven scenario adaptation in flight simulator training [14]. There, a CPS prototype linked theory-based assessment and simulator-based evidence through Deep-Linking Keywords and aggregated them into competence-related Knowledge, Skill, and Attitude (KSA) representations using weighted heuristic scoring functions.

The prototype demonstrated the feasibility of semantically grounded keyword linking and cross-domain evidence aggregation between theory and simulator training. However, the resulting representation remained heuristic and was not yet formalized as a deterministic state space with explicit transition logic, constrained action spaces, or architecturally embedded interpretability and auditability.

2.5 Research gap

Existing work provides important partial solutions, but no unified approach currently realizes the transition from heterogeneous subsymbolic training evidence to a semantically grounded and deterministic symbolic state space for adaptive flight training. Consequently, the literature still lacks a formal architecture that combines deterministic state-space construction, constrained ML, and interpretable and auditable adaptation decisions. This gap motivates the solution proposed in this paper.

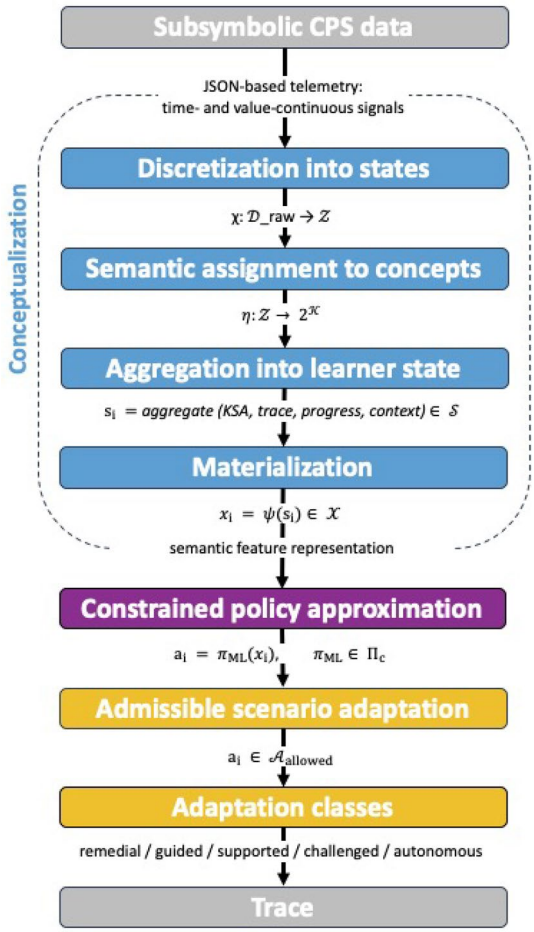


Fig. 3 Conceptualized CPS data-to-policy pipeline. Subsymbolic CPS data, represented as time- and value-continuous telemetry, are transformed through a three-step conceptualization process consisting of discretization, semantic assignment, and state construction. This results in a semantically grounded state representation $x_i \in \mathcal{X}$, which serves as the input to a constrained ML-based policy approximation π_{ML}. The policy operates within a formally restricted policy space and produces admissible scenario adaptations $a_i \in \mathcal{A}_{\mathrm{allowed}}$, which are mapped to ordered adaptation classes ranging from remedial to autonomous. All decisions are recorded in a traceable and auditable manner, ensuring full reconstructability of the decision process. *Color coding:* Blue indicates the state representation layer (RQ1), in which heterogeneous CPS data are transformed into a semantically grounded feature space. Purple denotes the decision model (RQ2), representing the constrained policy approximation. Orange represents the admissible action space and scenario adaptation (RQ2). Grey indicates the audit layer (RQ3), capturing traceability and reconstructability of all adaptation decisions

3 Solution

The realization of the formal mapping from heterogeneous raw measurements to admissible scenario adaptations, Φ , introduced in Sect. 1, is organized as a data-to-decision pipeline in which the conceptualized learner state space $\mathcal{S}$ serves as the central architectural element. The state space provides the symbolic learner representation, the semantic foundation of the operational ML feature space, and the reference structure for interpretable and auditable adaptation decisions. At system level, the architecture is described by the pipeline

$$\mathcal{D}_{\mathrm{raw}} \xrightarrow{\chi} \mathcal{Z} \to \mathcal{S} \to (Q, \delta) \to \mathcal{A} \to \pi \to \Delta_{\mathrm{scenario}} \tag{1}$$

where $\mathcal{Z}$ denotes the intermediate space of discrete diagnostic states produced by the discretization mapping χ, $\mathcal{S}$ the conceptualized learner state space, (Q, δ) the deterministic instructional control logic, $\mathcal{A}$ the admissible action space, and π the adaptation policy. The transition from $\mathcal{Z}$ to $\mathcal{S}$ comprises semantic assignment and aggregation, detailed in Sect. 3.1. Each step of Φ is formally defined: subsymbolic raw evidence is first discretized into diagnostic states, then mapped to a semantically grounded learner state space, then processed by a deterministically controlled and constraint-bounded policy layer, and finally translated into admissible symbolic scenario outputs. Figure 3 summarizes this transformation and highlights the role of the conceptualized state space as the central interface between subsymbolic data processing and constrained policy-based decision making.

To make the realization concrete, the following subsections use stall training as a running example, building on the CPS-based stall adaptation framework presented in our preceding work. This example combines theory-based assessment, event-based simulator evidence, and adaptive scenario progression within a single competence-relevant training context. Within the EASA-oriented competence structure, the subsequent formalization focuses primarily on Flight Path Management, Manual Control (FPM), since stall recognition and recovery are most directly observable through manual flight path control. The stall case is therefore not introduced as a special-purpose exception, but as an illustrative instantiation of the general system logic developed in this work.

3.1 Conceptualization of training evidence

To answer RQ1, the proposed system extends the conceptualization approach for CPS discussed in Sect. 2 by introducing a semantically grounded state construction for training evidence. The core idea is that heterogeneous and partly continuous evidence from assessment and simulator operation must first be transformed into symbolic and interpretable concepts before it can serve as a reliable basis for adaptation decisions.

The transformation proceeds in three steps, as illustrated in Fig. 3. The first step turns continuous and event-like raw signals into discrete diagnostic states. The second step assigns each diagnostic state to the competence concepts it provides evidence for. The third step aggregates the assigned evidence into a unified learner representation. The remainder of this subsection introduces each step formally and illustrates it on the stall case.

Step 1: Discretization. This step turns continuous, time-stamped raw measurements into discrete, diagnostically meaningful events that the rest of the architecture can reason about symbolically. Each raw evidence record is a typed tuple

$$d \in \mathcal{D}_{\text{raw}} \subseteq \mathcal{D}_{\text{theory}} \times \mathcal{D}_{\text{eval}} \times \mathcal{D}_{\text{trace}} \times \mathcal{D}_{\text{prog}} \times \mathcal{D}_{\text{ctx}}. \tag{2}$$

This tuple combines per-keyword theory scores $\mathcal{D}_{\text{theory}} = [0,1]^{|\mathcal{K}|}$, per-keyword simulator-evaluation scores $\mathcal{D}_{\text{eval}} = ([0,1] \cup \{\perp\})^{|\mathcal{K}|\times 2}$ for skill and attitude (with $\perp$ marking non-assessed keywords), telemetry-derived trace features $\mathcal{D}_{\text{trace}} \subset \mathbb{R}^9$ comprising reaction delay d_{reaction}, AoA reduction $\Delta\alpha$, pitch unloading $\Delta\theta$, velocity recovery Δv, altitude loss h_{loss}, maximum bank $\phi_{\max}$, yaw instability i_y, secondary stall count n_{sec}, and outcome code o_{stall}, instructional progress descriptors $\mathcal{D}_{\text{prog}} = Q \times [0,1]^4$ comprising the current phase $q \in Q$ and cumulative alpha aggregates, and scenario context $\mathcal{D}_{\text{ctx}}$ (scenario type, active keyword set, base reference values, tolerance thresholds). A deterministic discretization mapping

$$\chi : \mathcal{D}_{\text{raw}} \to \mathcal{Z} \tag{3}$$

transforms raw CPS evidence into discrete diagnostic states and event abstractions. In the present implementation, this includes event detection, temporal aggregation, maneuver-level feature extraction, and threshold-based identification of relevant behavioral patterns, yielding for each defined metric a discrete severity label in $\{0, 0.5, 1.0\}$ via metric-specific minor/major thresholds. In the stall case, this step yields symbolic states such as stall-warning phases, confirmed stall states, recovery phases, secondary stall events, and tolerance violations. Maneuver categories such as approach-to-stall, accelerated stall, secondary stall, or skidding stall are therefore not interpreted as raw telemetry patterns. They are semantically operationalized flight states derived from symbolic state construction. The architecture thereby treats stall training as a conceptual state-transition problem rather than as a signal-driven prediction problem.

Step 2: Semantic assignment. This step links each detected event to the competence concepts it carries information about. Because a single event often informs multiple competencies, the assignment is set-valued:

$$\eta : \mathcal{Z} \to 2^{\mathcal{K}} \tag{4}$$

so that each detected state can contribute evidence to one or more competence-related keywords. In the stall case, recovery-related evidence may simultaneously inform Flight Path Management (Manual) and stall-recovery keywords.

Step 3: Aggregation into a learner state. This step combines the semantically assigned evidence into a single, structured learner representation that downstream components can use directly. The resulting conceptualized learner state is defined as

$$s_i \in \mathcal{S} = \mathcal{S}_{\text{KSA}} \times \mathcal{S}_{\text{trace}} \times \mathcal{S}_{\text{prog}} \times \mathcal{S}_{\text{ctx}} \tag{5}$$

with per-keyword KSA aggregates $\mathcal{S}_{\text{KSA}} \subset ([0,1] \cup \{\perp\})^{|\mathcal{K}|\times 4}$ (Knowledge, Skill, Attitude, and total scores per keyword), maneuver-specific trace features $\mathcal{S}_{\text{trace}} \subset \mathbb{R}^9$, instructional progress $\mathcal{S}_{\text{prog}} \subset Q \times [0,1]^4$, and scenario context $\mathcal{S}_{\text{ctx}}$ (scenario type, active keyword set, and base reference values). No latent representation is introduced at any stage; every coordinate of s_i is defined as a function of $\mathcal{K}$, scenario context, or telemetry-derived maneuver descriptors, satisfying (P2). Conceptualization is therefore not a preprocessing step. It is the epistemic boundary of the learning system that determines what the ML component can know, learn, and explain. The model is not permitted to infer directly from raw telemetry or unconstrained latent embeddings. It operates exclusively on semantically assigned and deterministically constructed learner-state representations. The earlier keyword-based KSA aggregation introduced in the predecessor CPS prototype is in this way lifted into a formal state-space representation.

To make the transformation operational, conceptualization is implemented as a deterministic algorithmic procedure. Algorithm 1 presents the pseudocode of the conceptualization process.

Algorithm 1 ConceptState: Conceptualized learner state construction

Require: $\mathcal{D}_{\text{raw}}$
Ensure: s_i, x_i
1: $\mathcal{Z} \leftarrow \chi(\mathcal{D}_{\text{raw}})$
2: **for all** $z \in \mathcal{Z}$ **do**
3: $\quad \mathcal{K}_z \leftarrow \eta(z)$
4: $\quad$ **for all** $k \in \mathcal{K}_z$ **do**
5: $\quad\quad$ update $\alpha_{\text{K}}^{(k)}$,
6: $\quad\quad\quad\quad \alpha_{\text{S}}^{(k)}, \alpha_{\text{A}}^{(k)}$
7: $\quad$ **end for**
8: **end for**
9: $s_i \leftarrow \text{aggregate}(\text{KSA}, \text{trace},$
10: $\quad\quad\quad \text{progress}, \text{context})$
11: $x_i \leftarrow \psi(s_i)$
12: **return** (s_i, x_i)

Lines 1: Raw CPS evidence is transformed into discrete diagnostic states and event abstractions by deterministic event detection and temporal aggregation.

Lines 2–8: Each detected state is assigned to one or more Deep-Linking Keywords. The corresponding keyword-level competence values are updated according to the associated evidence.

Lines 9–10: The keyword-level competence values are aggregated together with trace-derived, progress-related, and contextual variables into a unified learner representation

Line 11: The conceptualized learner state is materialized as an operational feature snapshot for downstream policy execution.

Line 12: The conceptualized learner state and its operational feature snapshot are returned.

For policy execution, the abstract conceptualized learner state is materialized as an operational feature snapshot

$$x_i = \psi(s_i) \in \mathcal{X} \tag{6}$$

where $\mathcal{X}$ is directly derived from $\mathcal{S}$ and preserves its semantic structure. The snapshot includes keyword-level features, state-level aggregate features, trace-derived maneuver features, progress-level variables, and context features. The ML feature space is therefore not introduced as an independent or latent representation. It is an operational realization of the conceptualized learner state.

The conceptualization procedure is deterministic by construction, since identical input evidence yields identical event abstractions, keyword assignments, and aggregated learner states, satisfying (P1). The result is a deterministic and semantically grounded representation of training evidence that jointly satisfies (P1) and (P2). This demonstrates the operationalizability claim of RQ1: a formal state construction that consistently transforms heterogeneous theory-based and simulator-based evidence into a symbolic learner representation suitable for diagnosis, policy learning, and audit.

3.2 Constrained adaptation model

Scenario adaptation in this architecture serves a directed instructional purpose. The parameters of the next training scenario are adjusted in response to the diagnosed competence state of the learner, so that subsequent practice targets identified weaknesses while remaining compatible with the regulatory and instructional structure of the training process. Adaptation is therefore not an end in itself but an instructional mechanism whose behavior must satisfy three properties. First, the adaptation must be derived from the conceptualized learner state, so that decisions are semantically traceable in line with (P2). Second, the adaptation must be bounded by safety, regulatory, and instructional constraints, in line with (P3). Third, identical learner states under identical instructional phases must yield identical scenario updates, in line with (P1).

To support these properties, the conceptualized learner state is embedded into a deterministic adaptation framework consisting of instructional control, an admissible action space, and a rule-based baseline policy. Adaptation is therefore not treated as a free optimization problem but as a formally restricted decision process in which all admissible decisions are structurally constrained.

Instructional control. The instructional progression of the system is modeled as a deterministic finite automaton (DFA), illustrated in Fig. 4, that captures the phase structure prescribed by EBT and CBTA. The conceptualized learner state and the current instructional phase jointly define the execution state of the system:

$$\mathcal{M} = (Q, \Sigma, \delta, q_0, F) \tag{7}$$

$$\xi_i = (s_i, q_i) \tag{8}$$

Here, Q denotes the instructional phases of the training process, Σ the admissible transition events, δ the deterministic transition function, q_0 the initial phase, and F the set of terminal phases. The execution state $\xi_i = (s_i, q_i)$ serves as the basis for all subsequent adaptation decisions.

As illustrated in Fig. 4, the automaton governs the progression of training phases independently of the ML layer. The transition function δ is not influenced by data-driven components. The sequencing of instructional phases therefore remains fixed and reproducible, consistent with (P1). The conceptualized learner state may inform adaptation decisions but cannot alter the formal control structure of the training process.

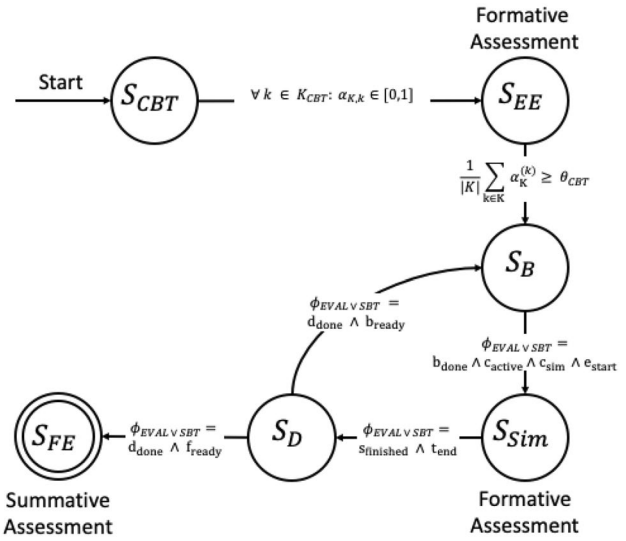


Fig. 4 DFA-based modeling of the training process. The automaton represents the deterministic control structure governing the sequence of training phases. State transitions are defined by formalized conditions based on competence-related markers and system events. ML-based adaptation is not modeled as a separate state, but is executed after the evaluation simulator phase and determines the parameterization of subsequent training phases

Action space. Adaptation actions are defined as typed delta vectors over a finite set of adaptation levers. Let $\mathcal{L}$ denote the universe of scenario parameter levers and $\mathcal{V}_\ell$ the typed admissible value range of lever ℓ, describing the permitted resulting values after additive application of the lever delta. The full action space $\mathcal{A}$ is

$$\mathcal{A} = \prod_{\ell \in \mathcal{L}} \mathcal{V}_\ell, \tag{9}$$

where $\prod_{\ell\in\mathcal{L}} \mathcal{V}_\ell$ denotes the Cartesian product of the admissible value ranges $\mathcal{V}_\ell$ over all scenario parameter levers $\ell \in \mathcal{L}$, and the admissible action space restricts $\mathcal{A}$ to the case-specific subset of permitted levers $\mathcal{L}_{\text{allowed}} \subseteq \mathcal{L}$:

$$\mathcal{A}_{\text{allowed}} = \prod_{\ell \in \mathcal{L}_{\text{allowed}}} \mathcal{V}_\ell. \tag{10}$$

In the current implementation, $\mathcal{L}$ contains ten levers organized into five parameter domains: `location` (altitude_ft, airspeed_ias), `weather` (wind_speed_kt, turbulence_level, gust_factor, wind_direction_variation), `aircraft` (yaw_instability), `engine` (power_response_delay), and `scenario` (repeat_stall, tighten_tolerances). Numeric levers have bounded admissible ranges:

$$\begin{aligned} \mathcal{V}_{\text{alt}} &= [3000, 10000]\ \text{ft}, & \mathcal{V}_{\text{ias}} &= [50, 140]\ \text{kt}, \\ \mathcal{V}_{\text{wind}} &= [0, 15]\ \text{kt}, & \mathcal{V}_{\text{gust}} &= [0, 12]\ \text{kt}, \\ \mathcal{V}_{\text{yaw}} &= [0, 0.5], & \mathcal{V}_{\text{pwr}} &= [0, 2]\ \text{s}. \end{aligned}$$

Table 1 Action space and constraint contracts

Component	Type	Constraint/contract
Focus keyword	Discrete	$k \in \mathcal{K}_{\text{allowed}}$; inadmissible keywords are excluded via profile projection
Repeat maneuver (repeat_stall)	Binary	$\in \{0, 1\}$; monotonic constraint: lower competence ⇒ higher repetition probability
Wind-speed lever	Continuous	wind_speed_kt $\in [0, 15]$ kt (absolute); policy outputs are applied additively to the base value and the result is clipped to $\mathcal{V}_\ell$ via $\mathcal{P}_{\text{params}}$
Scenario profile (difficulty class)	Ordinal	$\in \{\text{strong}, \text{moderate}, \text{stable}\}$; derived from threshold-based policy $c(k, s)$
Location/scenario selection	Categorical	Restricted to predefined scenario set; no free generation
Projection operator $\mathcal{P}$	Deterministic mapping	$\mathcal{P} : \mathcal{A} \times Q \to \mathcal{A}_{\text{allowed}} \cup \{\emptyset\}$; enforces bounds, focus admissibility, and phase gating
Phase constraint (DFA)	Structural	Actions only executed for $q \in Q_{\text{adapt}}$

Structural levers have $\mathcal{V}_\ell = \{0, 1\}$. Policy outputs are interpreted as additive offsets from the scenario base value, and the resulting absolute values are clipped to $\mathcal{V}_\ell$ via the projection operator $\mathcal{P}_{\text{params}}$ defined below. The scenario-adaptation set referenced in the problem formulation coincides with the admissible action space:

$$\Delta_{\text{scenario}} = \mathcal{A}_{\text{allowed}}. \tag{11}$$

The contracts summarized in Table 1 ensure that all executed adaptations satisfy structural, operational, and instructional constraints by construction. They are directly implemented in the policy execution layer via deterministic projection and bounded parameter updates.

Rule-based baseline policy. The rule-based baseline policy maps the conceptualized learner state to admissible actions in three substeps. First, up to $\kappa_{\max}$ focus keywords are selected from the admissible keyword set, ranked by ascending focus score:

$$K^*(s_i) = \text{argmin}^{(\kappa_{\max})}_{k \in \mathcal{K}_{\text{allowed}}} w_k(s_i), \tag{12}$$

where $\text{argmin}^{(\kappa_{\max})}$ returns the $\kappa_{\max}$ keywords with the smallest focus scores. The focus score $w_k(s_i)$ is a weighted combination of the per-keyword KSA components:

$$\begin{aligned} w_k(s_i) = w_K^{(k)}\, \alpha_K^{(k)}(s_i) + w_S^{(k)}\, \alpha_S^{(k)}(s_i) \\ + w_A^{(k)}\, \alpha_A^{(k)}(s_i), \end{aligned} \tag{13}$$

with keyword-specific weights $(w_K^{(k)}, w_S^{(k)}, w_A^{(k)})$ and default (0.4, 0.4, 0.2) when no keyword-specific weighting

is defined. Lower focus scores indicate higher adaptation priority. The weighting scheme is keyword-specific and reflects the didactic structure of the underlying competence: knowledge-dominant keywords such as *stall-recovery* or *energy-management* weight the theory-based component more strongly (e.g., $w_K^{(\text{stall-recovery})} = 0.45$, $w_S = 0.40$, $w_A = 0.15$), while skill-dominant keywords such as *altitude-control* weight simulator-based skill evidence higher (e.g., $w_S^{(\text{altitude-control})} = 0.55$, $w_K = 0.25$, $w_A = 0.20$). This realizes the deep-linking principle operationally: the relative weight of theory-derived and simulator-derived evidence adapts to the cognitive structure of each competence. The focus score $w_k(s_i)$ is distinct from the aggregated competence score $\alpha_{\text{total}}^{(k)}(s_i)$ used in the class assignment of Eq. (14): focus selection prioritizes by the knowledge-skill-attitude profile, class assignment thresholds the aggregated competence level. In the current implementation, $\kappa_{\max} = 2$. This reflects the operational reality of stall training, where the adaptation is often driven jointly by multiple competence dimensions, for example power-management together with altitude-control during the approach phase, or energy-management together with stall-recovery during recovery, rather than by a single dominant focus.

Second, a policy class is determined using threshold-based competence evaluation:

$$c(k, s_i) = \begin{cases} \text{strong}, & \alpha_{\text{total}}^{(k)}(s_i) < \theta_1, \\ \text{moderate}, & \theta_1 \le \alpha_{\text{total}}^{(k)}(s_i) < \theta_2, \\ \text{stable}, & \alpha_{\text{total}}^{(k)}(s_i) \ge \theta_2, \end{cases} \tag{14}$$

with $\theta_1 = 0.50$ and $\theta_2 = 0.75$ in the current implementation.

Third, the per-keyword deltas are computed and additively combined into the final scenario update:

$$\pi_{\text{rule}}(s_i) = \mathcal{P}_{\text{params}}\left(\bigoplus_{k \in K^*(s_i)} \Gamma\big(k,\, c(k, s_i)\big) \right), \tag{15}$$

where Γ produces a typed delta vector per keyword-class pair, typed superposition combines these deltas by component-wise addition on numeric levers and logical OR on structural levers ($\bigoplus$), and $\mathcal{P}_{\text{params}}$ clips each resulting absolute value to $\mathcal{V}_\ell$. In the stall/FPM example, this may yield maneuver repetition together with adjusted environmental conditions, with both contributions originating from distinct focus keywords, e.g., *repeat_stall* from power-management, *tighten_tolerances* from altitude-control.

In the present implementation, π_{rule} additionally guarantees a minimum number of two adapted levers per case through deterministic keyword-aware fallback rules; this ensures meaningful scenario perturbation even when individual keyword classifications produce sparse deltas. The fallback mechanism remains within $\mathcal{A}_{\text{allowed}}$ and does not affect formal constraint compliance.

The constrained policy execution is itself implemented as a deterministic procedure. Algorithm 2 presents the pseudocode of the rule-based baseline policy.

Algorithm 2 ConstrainedAdapt: Rule-based constrained scenario adaptation

Require: $s_i, q_i, \kappa_{\max}$
Ensure: a_i
1: **if** $q_i \in Q_{\text{adapt}}$ **then**
2: $K^*(s_i) \leftarrow \operatorname{argmin}_{k \in \mathcal{K}_{\text{allowed}}}^{(\kappa_{\max})}$
3: $w_k(s_i)$
4: $a_i \leftarrow \mathbf{0}$
5: **for all** $k \in K^*(s_i)$ **do**
6: $c_k \leftarrow c(k, s_i)$
7: $a_i \leftarrow a_i \oplus \Gamma(k, c_k)$
8: **end for**
9: $a_i \leftarrow \mathcal{P}_{\text{params}}(a_i)$
10: **return** a_i
11: **else**
12: **return** $\varnothing$
13: **end if**

Line 1: Adaptation is only executed if the current DFA state belongs to the subset of adaptation-enabled instructional phases.

Line 2–3: The $\kappa_{\max}$ admissible focus keywords with smallest focus scores are selected. The set may contain fewer than $\kappa_{\max}$ keywords when the admissible keyword set is smaller.

Lines 4–8: The accumulator a_i is initialized to the neutral delta and iteratively updated by superposition of the per-keyword deltas. For each selected keyword, the threshold-based policy class c_k determines the adaptation strength, and Γ produces the corresponding typed delta vector.

Line 9: The accumulated delta is clipped to the admissible parameter ranges $\mathcal{V}_\ell$ via $\mathcal{P}_{\text{params}}$.

Lines 11–13: If adaptation is not admissible in the current phase, the empty action is returned.

The constrained adaptation model is therefore defined by three interacting restrictions: deterministic instructional control via the DFA, admissible action generation via scenario contracts and safety limits, and structured rule-based policy execution via the mapping Γ. The adaptation procedure is phase-gated and contract-compliant by construction, since actions are only generated in adaptation-enabled instructional phases and are always projected onto the admissible action space, satisfying (P3) for the rule-based baseline.

This formulation directly addresses the core of RQ2. It defines a bounded and formally structured state-action architecture within which any later ML approximation must operate. The separation between control logic, decision logic, and execution ensures that data-driven components can be integrated without compromising determinism, safety, or auditability.

3.3 Policy approximation by machine learning

Building on the constrained adaptation model, ML is introduced as a bounded approximation of the existing rule-based policy. Figure 5 illustrates the resulting architecture: the conceptualized learner state serves as input to the constrained policy approximation, which maps it to an admissible adaptation class. The objective is therefore not to replace the architecture by an unrestricted data-driven predictor, but to approximate validated rule-based behavior within the same formal constraints.

The ML policy is defined as

$$\begin{aligned} \pi_{\mathrm{ML}}(s,q) &\approx \pi_{\mathrm{rule}}(s,q) \quad \text{on } \mathcal{S}\times Q_{\mathrm{adapt}}, \\ \pi_{\mathrm{ML}} &= (\pi_{\mathrm{profile}}, \pi_{\mathrm{repeat}}, \pi_{\mathrm{wind}}, \dots). \end{aligned} \tag{16}$$

where π_{rule} denotes the deterministic rule-based baseline policy, $s \in \mathcal{S}$ the conceptualized learner state, $q \in Q_{\mathrm{adapt}}$ the current adaptation-enabled instructional phase, and π_{profile}, π_{repeat}, π_{wind} are sub-policies for focus-score prediction, maneuver repetition, and wind-speed adaptation respectively.

The approximation is understood in terms of the empirical risk defined below, rather than exact functional equivalence.

Here, π_{profile} approximates the per-keyword focus-score profile defined in Eq. (13), π_{repeat} and π_{wind} predict maneuver repetition and bounded wind-speed deltas. Additional sub-policies (e.g., π_{tighten}, π_{altitude}, π_{power}) share the same constraint mechanisms and extend componentwise. The decomposition into several supervised learning tasks reflects the structure of the policy implementation and avoids the assumption of a single monolithic unconstrained model.

The supervised learning objective is formulated as

$$\min_{\pi\in\Pi_c} \mathbb{E}_{(s,q)\sim\mathcal{D}}\left[L(\pi(s,q), \pi_{\mathrm{rule}}(s,q))\right], \tag{17}$$

where Π_c denotes the constrained policy space. Let $\mathcal{C}_{\mathrm{mono}} \subseteq \{1,\dots,d\} \times \mathcal{O}_{\mathrm{ML}} \times \{-1,+1\}$ denote the set of (state-feature index, sub-policy output, sign) triples subject to monotonicity, where $\mathcal{O}_{\mathrm{ML}} = \{\pi_{\mathrm{repeat}}, \pi_{\mathrm{wind}}, \pi_{\mathrm{profile},k}, \dots\}$ contains the scalar-valued sub-policy outputs. The constrained policy space is defined by three conditions:

$$\Pi_c = \{\pi \mid \mathrm{M}(\pi) \wedge \mathrm{A}(\pi) \wedge \mathrm{G}(\pi)\}, \tag{18}$$

where M denotes monotonicity, A action admissibility, and G phase gating, $\mathcal{S}$ is the conceptualized learner state space, and Q the set of instructional phases, defined as follows.

Monotonicity requires that increasing a competence-related input feature does not decrease the corresponding sub-policy output:

$$\begin{aligned} \mathrm{M}(\pi):\quad & \forall (j,o,\sigma) \in \mathcal{C}_{\mathrm{mono}},\ \forall s,s' \in \mathcal{S},\ \forall q \in Q \\ & \left(s_j \le s'_j \wedge s_{\setminus j} = s'_{\setminus j}\right) \\ & \Rightarrow \sigma\, \pi_o(s,q) \le \sigma\, \pi_o(s',q), \end{aligned} \tag{19}$$

Action admissibility requires that all policy outputs lie within the admissible action space:

$$\begin{aligned} \mathrm{A}(\pi):\quad & \pi(s,q) \in \mathcal{A}_{\mathrm{allowed}} \\ & \forall (s,q) \in \mathcal{S}\times Q_{\mathrm{adapt}}, \end{aligned} \tag{20}$$

Phase gating requires that adaptation is suppressed outside adaptation-enabled instructional phases:

$$\begin{aligned} \mathrm{G}(\pi):\quad & \pi(s,q) = \emptyset \\ & \forall (s,q) \in \mathcal{S}\times (Q \setminus Q_{\mathrm{adapt}}). \end{aligned} \tag{21}$$

This definition makes explicit that admissible policies are restricted by three distinct conditions: order-preserving monotonicity at the feature level, admissibility at the action level, and phase gating at the instructional control level. Each condition is enforced by construction and none is left to training pressure.

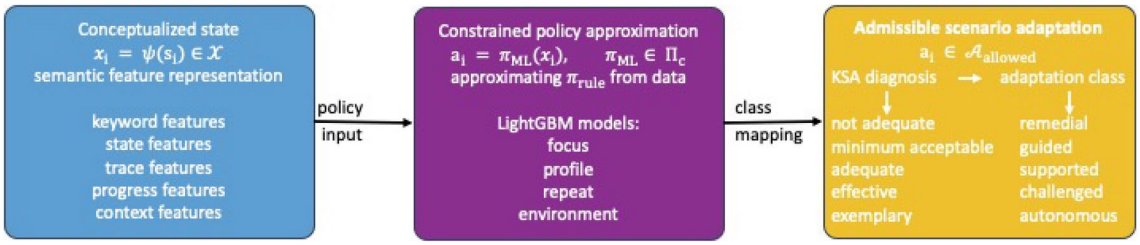


Fig. 5 Constrained policy learning. Left: conceptualized state representation derived from heterogeneous CPS evidence. Center: constrained LightGBM-based policy approximation mapping the conceptualized state to an ordered adaptation class. Right: admissible scenario adaptation expressed as an ordered mapping from KSA diagnosis to adaptation class

To ensure that the learned policy remains consistent with the formal structure of the system, constraints are enforced at two complementary levels.

First, constraints are incorporated at the learning level. For selected competence-related features, monotonicity constraints are imposed on the ML model as order relations on the predicted output. For example, let $\alpha_{\mathrm{SR}}(s) := \alpha_{\mathrm{S}}^{(\text{stall-recovery})}(s)$ denote the stall-recovery skill component. The *repeat_stall* sub-policy is required to be monotonically non-increasing in stall-recovery skill:

$$\begin{aligned} &\alpha_{\mathrm{SR}}(s) \leq \alpha_{\mathrm{SR}}(s') \\ &\quad \Rightarrow \pi_{\mathrm{repeat}}(s) \geq \pi_{\mathrm{repeat}}(s'). \end{aligned} \tag{22}$$

Here, all other coordinates of s and s' are assumed to coincide. This enforces a monotonic relationship in which increasing competence reduces the likelihood of remedial adaptation. Such monotonic constraints encode domain knowledge directly into the learning process and reduce structurally implausible model behavior.

Second, constraints are enforced at the inference level by projecting model outputs onto the admissible action space. Since lower focus scores indicate higher adaptation priority, inadmissible focus keywords must not be mapped to low scores. For a raw focus-score profile $\hat{p} \in \mathbb{R}^{|\mathcal{K}|}$ predicted by π_{profile} and an admissible keyword set $\mathcal{K}_{\mathrm{allowed}}$, the constrained (projected) profile $\tilde{p}$ is defined as

$$\tilde{p}_k = \begin{cases} \mathrm{clip}(\hat{p}_k, 0, 1), & k \in \mathcal{K}_{\mathrm{allowed}}, \\ 1, & k \notin \mathcal{K}_{\mathrm{allowed}}. \end{cases} \tag{23}$$

The constrained focus decision is then computed by selecting the $\kappa_{\max}$ admissible keywords with smallest predicted profile scores:

$$K^*(s_i) = \mathrm{argmin}_{k \in \mathcal{K}_{\mathrm{allowed}}}^{(\kappa_{\max})} \tilde{p}_k. \tag{24}$$

Similarly, executable adaptation parameters are clipped or projected to predefined admissible ranges. The executable policy is obtained by applying a deterministic projection operator

$$\pi_{\mathrm{exec}}(s, q) = \mathcal{P}\big(\pi_{\mathrm{ML}}(s, q), q\big), \tag{25}$$

which decomposes into three independently defined components applied in sequence:

$$\mathcal{P}(a, q) = \big(\mathcal{P}_{\mathrm{phase}} \circ \mathcal{P}_{\mathrm{focus}} \circ \mathcal{P}_{\mathrm{params}}\big)(a, q), \tag{26}$$

where $\mathcal{P}_{\mathrm{params}}$ clips each numeric lever to its admissible range $\mathcal{V}_\ell$ and rounds structural levers to $\{0, 1\}$, $\mathcal{P}_{\mathrm{focus}}$ enforces $k \in \mathcal{K}_{\mathrm{allowed}}$ on the focus-keyword component via the profile-clipping rule above, and $\mathcal{P}_{\mathrm{phase}}$ returns $\emptyset$ for $q \notin Q_{\mathrm{adapt}}$ and the identity otherwise. Each component is deterministic and idempotent. Their composition guarantees that the projection maps any action-phase pair to an admissible output:

$$\mathcal{P}(\mathcal{A} \times Q) \subseteq \mathcal{A}_{\mathrm{allowed}} \cup \{\emptyset\}. \tag{27}$$

The phase projection is applied last so that non-admissible instructional phases override all previously computed model outputs. Consequently, the executed policy satisfies

$$\pi_{\mathrm{exec}}(s, q) \in \mathcal{A}_{\mathrm{allowed}} \quad \text{for } q \in Q_{\mathrm{adapt}}, \tag{28}$$

and

$$\pi_{\mathrm{exec}}(s, q) = \emptyset \quad \text{for } q \notin Q_{\mathrm{adapt}}. \tag{29}$$

The projection guarantees the admissibility and phase-gating components of Π_c for the executable policy. The monotonicity component is enforced separately at model level through the constrained learning procedure. Together, both mechanisms ensure that the deployed policy satisfies the enforceable structural requirements imposed by Π_c, thereby ensuring constraint compliance by construction and satisfying (P3) for the ML policy.

Thus, the learned policy is not only trained as an approximation of the rule-based baseline, but is explicitly constrained during both learning and inference.

Figure 5 visualizes the transition from conceptualized learner state to admissible symbolic scenario adaptation. The figure emphasizes that the ML component operates within a formally restricted architecture, where conceptualization, policy approximation, and action execution are clearly separated.

In the operational system, the ML-based policy π_{ML} replaces or augments individual components of Algorithm 2. However, its outputs are deterministically transformed into the executable policy π_{exec} by projection onto the admissible action space. This ensures that learned adaptations remain compliant with the same structural constraints as the rule-based baseline.

Consequently, the ML component is not an unconstrained predictor, but a constraint-compliant approximation embedded in a formally structured CPS. This provides the ML part of the answer to RQ2: data-driven adaptation is enabled, while determinism, safety constraints, and instructional control are preserved by construction.

3.4 Summary of the solution

The proposed solution integrates a semantically grounded state space, deterministic control logic, a constrained action space, and a bounded ML-based policy into a unified cyber-physical training architecture. RQ1 is addressed through deterministic state-space construction from heterogeneous evidence, RQ2 through constrained policy integration within a formal instructional and action framework, and RQ3 through architectural interpretability and auditability by construction.

4 Verification

The verification proceeds in two parts. First, the formal system properties (P1)–(P4) are analytically verified by construction, independently of any specific dataset. Consistent with the correct-by-construction paradigm discussed in Sect. 2, each property is treated as a universally quantified correctness predicate over its respective input domain. Thus, $\forall x \in \mathcal{D}_P : P(x)$ is established by architectural construction rather than by induction over individual execution steps. The respective domains are $\mathcal{D}_{\text{raw}}$ for P1, $\mathcal{S}$ for P2, $\mathcal{A} \times Q$ for P3, and the space of decision traces τ for P4. Second, the operationalizability of the resulting architecture is demonstrated empirically as a proof-of-concept on N=160 policy cases.

4.1 Analytical verification

The formal system properties follow from the architectural design independently of the empirical dataset.

Determinism (P1). The mappings χ, η, the aggregation procedure, the materialization ψ, the DFA transition function δ, the rule-based mapping Γ, and the projection operator $\mathcal{P}$ are deterministic by construction. Identical raw evidence therefore yields identical diagnostic states, keyword assignments, conceptualized learner states, feature snapshots, phase transitions, and executed actions:

$$\begin{aligned} d^{(1)} = d^{(2)} &\Rightarrow \chi(d^{(1)}) = \chi(d^{(2)}) \\ &\Rightarrow s^{(1)} = s^{(2)} \\ &\Rightarrow \psi(s^{(1)}) = \psi(s^{(2)}). \end{aligned} \tag{30}$$

For a fixed trained model and fixed system version,

$$\pi_{\text{exec}}(s_t, q_t) = \mathcal{P}\big(\pi_{\text{ML}}(s_t, q_t), q_t\big) \tag{31}$$

is deterministic as well. This verifies (P1) and ensures that semantically grounded learner representations and constrained policy execution remain reproducible under identical system conditions.

Semantic grounding (P2). The state space $\mathcal{S}$ contains no latent or unconstrained components. Every dimension of s_i is defined as a function of Deep-Linking Keywords, KSA scores, maneuver-state abstractions, trace-derived features, or context variables, all expressible in terms of $\mathcal{K}$. The operational feature space $\mathcal{X} = \psi(\mathcal{S})$ inherits this structure, since ψ is a deterministic projection that preserves the semantic typing of $\mathcal{S}$. Every intermediate representation produced by Φ therefore remains traceable to $\mathcal{K}$. This verifies (P2) and ensures that all internal states of the system are interpretable in terms of semantically defined concepts.

Constraint compliance (P3). The projection operator $\mathcal{P} : \mathcal{A} \times Q \to \mathcal{A}_{\text{allowed}} \cup \{\emptyset\}$ defined in Eq. (26) is total and deterministic. The executable policy therefore satisfies

$$\pi_{\text{exec}}(s, q) \in \mathcal{A}_{\text{allowed}} \quad \text{for } q \in Q_{\text{adapt}}, \tag{32}$$

and

$$\pi_{\text{exec}}(s, q) = \emptyset \quad \text{for } q \notin Q_{\text{adapt}}. \tag{33}$$

Constraint compliance is therefore not an empirical property of the learned model alone but an architectural invariant enforced through admissible action contracts, phase gating, and deterministic projection. Monotonicity constraints additionally restrict the learned sub-policies at model level. Together, these mechanisms satisfy the structural requirements imposed by Π_c. This verifies (P3) and ensures that executed adaptations remain bounded within formally specified safety and instructional constraints.

Reconstructability (P4). Each executed decision is associated with a persistent trace

$$\begin{aligned} \tau = (&d,\, z,\, s,\, x,\, q,\, \pi_{\text{ML}}(s, q), \\ &\pi_{\text{exec}}(s, q),\, \Delta_{\text{scenario}},\, \text{meta}). \end{aligned} \tag{34}$$

If the input evidence, intermediate representations, model outputs, executed actions, projection results, and policy metadata are logged, each adaptation decision is reconstructable from τ. Interpretability and auditability are therefore architectural properties: interpretability follows from semantically grounded features, auditability from deterministic state construction and constrained policy execution. This verifies (P4).

4.2 Empirical demonstration

The empirical part is positioned as a proof-of-concept and not as a comprehensive statistical evaluation. The dataset of N=160 policy cases is sufficient to demonstrate that

Table 2 Dataset characteristics

Property	Value
Cases	162 exported; 160 used after focus-class filtering
Features	87 (after preprocessing)
Valid focus classes	5 (speed-control, altitude-control, power-management, stall-recovery, energy-management)
CV strategy	Group-wise $k = 5$ (no learner leakage)
Targets	Focus keyword, profile scores, binary decisions, scenario deltas

Table 3 Mean ± std across folds

Metric	Value
Focus classification accuracy	0.631 ± 0.080
Majority-class baseline	0.288 ± 0.042
Lift over baseline	+ 0.344
Profile MAE	0.126 ± 0.024
Profile argmin accuracy	0.619 ± 0.109
repeat_stall accuracy	0.794 ± 0.098
tighten_tolerances accuracy	0.831 ± 0.132
Wind-speed delta MAE (kt)	0.670 ± 0.111
Rule policy decision margin	0.130
ML policy decision margin	0.113

Baseline: majority-class classifier

the conceptualized state space is operationalizable with recorded training data and that the constrained policy approximation converges within the formally bounded space Π_c. The dataset does not support claims of statistical generalization across cohorts, maneuver categories, or operator populations. A multi-cohort generalization study is identified as the natural next step (cf. Sect. 5).

Within this scope, three aspects are demonstrated: operationalizability of the state-space construction with recorded CPS data, learnability of the rule-based policy structure within Π_c, and the semantic explanation mechanism enabled by the conceptualized state space.

The dataset comprises 162 exported policy cases. After filtering cases without a valid focus class, 160 cases remain for model training and demonstration. A group-wise k=5 cross-validation split by learner identity prevents information leakage. After preprocessing, 87 usable features remain. Dataset characteristics and cross-validated results are summarized in Tables 2 and 3.

State-space operationalizability. The conceptualized state space is reproducibly constructible from recorded CPS data and serves directly as input for policy learning. After preprocessing, 87 semantically defined features remain. These features are derived from keyword-level KSA aggregates, maneuver-level trace descriptors, instructional progress variables, and scenario context variables. Relative to the heterogeneous raw evidence streams, this finite feature set is compact because it reduces theory scores, maneuver-level telemetry descriptors, instructional progress, and scenario context to semantically typed variables. No latent or unconstrained feature representation is introduced. This demonstrates that the conceptualized state space $\mathcal{S}$ can be operationalized as an ML feature space $\mathcal{X}$ while preserving semantic traceability to the underlying Deep-Linking Keywords.

Constrained policy learnability. The cross-validated results indicate that the constrained policy approximation recovers relevant parts of the deterministic rule-based decision structure. The focus-classification model achieves an accuracy of 0.631 ± 0.080 compared with a majority-class baseline of 0.288 ± 0.042, corresponding to a lift of +0.344. The profile model achieves a MAE of 0.126 ± 0.024, indicating that the continuous rule-based focus-score structure is approximated within a narrow error range. The executable adaptation components are also recovered with high accuracy for *repeat_stall* (0.794 ± 0.098) and *tighten_tolerances* (0.831 ± 0.132), while the wind-speed delta is approximated with a MAE of 0.670 ± 0.111 kt.

The rule-policy decision margin (0.130) and the ML-policy decision margin (0.113) indicate that the learned approximation preserves a similar separation structure between competing focus decisions. Within the proof-of-concept scope, these results demonstrate that the rule-based policy structure is learnable from the conceptualized state representation without leaving the formally bounded policy space.

Semantic policy-space analysis.

The following semantic policy-space analysis links the three empirical perspectives. It examines whether the conceptualized learner-state space is operationally usable (RQ1), whether the constrained policy structure is learnable (RQ2), and whether the resulting model behavior remains interpretable in terms of semantic competence variables (RQ3).

Table 4 relates explicit rule-based focus selection to SHAP-based feature relevance within the learned policy approximation. The analysis therefore does not only count how often a keyword is selected as an adaptation target, but evaluates how strongly its semantic state variables shape the learned approximation of the policy structure.

The results show a clear decoupling between direct rule priority and global feature relevance. Energy-management is selected as explicit focus keyword in only 7 of 160 cases (4.4%) and has the highest mean rule score among eligible focus classes (0.622), indicating low direct adaptation priority. Nevertheless, it contributes the largest SHAP share in the profile model (33.0%, rank 1). Thus, energy-management acts primarily as a policy-shaping semantic state variable rather than as a frequent explicit adaptation target. This supports the interpretation that the constrained policy approximation captures semantically meaningful

Table 4 Keyword focus coverage and SHAP-based feature relevance in the policy approximation

Keyword	Focus N	Focus %	Mean rule	SHAP rank	Best $\lvert\phi\rvert$	SHAP sum	SHAP share
Eligible focus keywords after filtering (sorted by SHAP share ↓)							
Energy-management†	7	4.4	0.622	1	0.092	0.121	33.0
Altitude-control‡	47	29.4	0.583	3	0.018	0.045	12.4
Speed-control†	52	32.5	0.408	4	0.017	0.039	10.6
Stall-recovery	19	11.9	0.544	2	0.021	0.037	10.0
Power-management	35	21.9	0.512	9	0.010	0.023	6.3
Not retained as valid focus class after filtering							
Tolerance-discipline	0	0.0	0.740	12	0.009	0.013	3.4
Flight-stability*	0	0.0	1.000	10	0.009	0.011	3.0
Control-coordination	0	0.0	1.000	19	0.006	0.007	1.8
Heading-control	0	0.0	1.000	–	–	0.000	0.0

Results are reported for N=160 policy cases and k=5 group-wise cross-validation. Markers indicate three structural findings: † decoupling of rule priority from SHAP relevance; ‡ broadest SHAP footprint (3 features in global top-6); * cross-keyword predictor (SHAP rank 10, never selectable as focus)

competence interactions rather than merely reproducing isolated rule-triggered focus labels.

Altitude-control shows the broadest SHAP footprint, with three independent altitude-related features in the global top-6 ranking. This indicates a distributed contribution across several competence-related feature dimensions rather than a single dominant variable.

Flight-stability is never selectable as explicit focus keyword, but still contributes measurable SHAP relevance (rank 10, share 3.0%). This supports the interpretation that non-selectable semantic state variables can act as cross-keyword predictors for other adaptation decisions. The policy should therefore not be interpreted as a conventional keyword classifier, but as operating on a semantically structured state space with coupled competence dimensions.

5 Discussion

The verification confirms the feasibility of the proposed architecture along all three research questions. The analytical verification establishes that the formal properties (P1)–(P4) hold by construction, independently of dataset size. The empirical demonstration shows that the architecture is operationalizable on recorded training data and that the constrained policy approximation recovers the rule-based decision structure within the formally bounded space Π_c. The contribution lies not primarily in unrestricted adaptivity but in defining formally specified and analytically verified conditions under which ML-based adaptation becomes structurally compatible with safety-critical cyber-physical training systems.

A direct numerical comparison with prior adaptive systems is structurally limited due to different task definitions, conceptual representations, and adaptation objectives. Existing work often focuses either on adaptive capability or on semantic interoperability. The present architecture combines deterministic learner-state construction, constrained policy execution, and semantically grounded adaptation within a unified CPS framework. The rule-based policy π_{rule} therefore serves as the domain-specific reference baseline.

As shown in Table 4, the dominance of energy-management features in the semantic policy-space analysis reflects the interaction between the conceptualized state space and the rule-based policy. Keywords with limited label diversity tend to exhibit less stable decision boundaries, indicating a structural property of the current scenario design rather than a modeling deficiency.

Two limitations require explicit acknowledgment. First, the empirical demonstration is restricted to N=160 policy cases drawn from a single competence-relevant maneuver context with five eligible focus classes. This sample size is sufficient to demonstrate operationalizability and constraint-compliant approximation as a proof-of-concept, but it does not support claims of statistical generalization across cohorts, maneuver categories, or operator populations. Second, the architectural scope of the present contribution is the formal verification of the proposed adaptation pattern and its empirical operationalization. Broad maneuver coverage and alternative evaluation paradigms, such as reinforcement-learning-based comparison, would require methodologically distinct setups and remain beyond the present scope.

The transferability of the conceptualization principles to further safety-critical maneuver categories is an open research question and depends on the availability of comparable multi-scenario data. The formal properties (P1)–(P4) themselves are not bound to the stall case. Their re-instantiation in further competence contexts is a direction for subsequent work.

6 Conclusion

This paper presented a formally constrained CPS for scenario-based flight simulator training that embeds ML-based adaptation into a deterministic, interpretable, and auditable architecture. The mapping $\Phi : \mathcal{D}_{\text{raw}} \to \Delta_{\text{scenario}}$ is decomposed into formally defined substeps that jointly satisfy the structural properties of determinism (P1), semantic grounding (P2), constraint compliance (P3), and reconstructability (P4). With respect to RQ1, the conceptualized state-space construction based on Deep-Linking Keywords was verified to satisfy (P1) and (P2) by construction and demonstrated to be operationalizable and compact on the present dataset. With respect to RQ2, the constrained policy approximation was shown to satisfy the structural requirements of Π_c through architectural projection and model-level constraints, thereby satisfying (P3). Its operationalizability was demonstrated as a proof-of-concept on N=160 policy cases, with a cross-validated profile MAE of 0.126 ± 0.024 and a focus-classification accuracy lift of +0.344 over the majority-class baseline. With respect to RQ3, interpretability and auditability follow architecturally from (P2) and (P4); semantically grounded features and persistent decision traces ensure that adaptation decisions are reconstructable post-hoc.

Open research directions include multi-cohort generalization studies beyond the present proof-of-concept scope, the formal characterization of conditions under which adaptation decisions are correct and admissible, the completeness limits arising from the interplay of determinism and explainability, and the structural trade-offs between adaptivity, explainability, and optimization within the constraint architecture.

Author contributions H.L. conceptualized the study, developed the methodological framework, implemented the prototype, conducted the experiments, analyzed the results, prepared the figures, and wrote the main manuscript text. O.N. contributed to the methodological framing of the machine learning and explainable AI components and reviewed the manuscript. T.N. supervised the work, provided aeronautical and engineering guidance, and reviewed the manuscript. All authors approved the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL. No project-specific external funding was received.

Data availability The datasets generated and analyzed during the current study are not publicly available because they originate from a prototype-specific implementation of the cyber-physical training system and include implementation-specific policy cases, traces, and scenario adaptation data. The data are available from the corresponding author upon reasonable request.

Code availability A repository was prepared for reproducibility assessment during the peer-review process and contains the ML training and validation pipeline, experiment configuration, evaluation scripts, and reproducibility artifacts used in this study. The repository further includes validation tests, feature and target documentation, and reference experiment outputs required to reproduce the reported results.

Declarations

Conflict of interest The authors declare that they have no relevant financial or non-financial interests to disclose.

Ethics approval The study was conducted using anonymized simulator-session and training-performance data within an educational and research context. No personally identifiable information was used in the reported analysis. All data processing was performed in accordance with applicable institutional and data-protection regulations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. European Union Aviation Safety Agency and Ines Berlenga: Ethics for artificial intelligence in aviation—aviation professionals survey results 2024/2025 (2025). ISBN: 978-92-9210-289-0. https://doi.org/10.2822/7047867, https://www.easa.europa.eu/en/document-library/general-publications/ethics-artificial-intelligence-aviation
2. MLEAP Consortium: EASA Research – Machine Learning Application Approval (MLEAP) final report (2024). https://www.easa.europa.eu/sites/default/files/dfu/mleap-d4-public-report-issue01.pdf
3. European Union Aviation Safety Agency (EASA): EASA artificial intelligence concept paper, issue 2: guidance for level 1 & 2 machine-learning applications (2024). https://www.easa.europa.eu/en/document-library/general-publications/easa-artificial-intelligence-concept-paper-issue-2
4. International Air Transport Association. Competency-Based Training and Assessment (CBTA) Center Program Manual, Edition 2. International Air Transport Association, Montreal - Geneva, 2024. ISBN: 978-92-9272-044-5 https://www.iata.org/contentassets/907edf7b75b34704ae4241285f024601/cbtapm-2-en-final.pdf
5. European Union Aviation Safety Agency (EASA): EBT manual version 2.1 (2023). https://www.easa.europa.eu/en/downloads/137656/en
6. International Civil Aviation Organization (ICAO): Procedures for Air Navigation Services - Training. 3rd Edn. International Civil Aviation Organization (2020). ISBN: 978-92-9265-187-9. https://www.icao.int/sites/default/files/APAC/RASG/SafetyTools/19-Procedures-for-Air-Navigation-Services-Training-PANS-TRG-Doc-9868.pdf
7. International Air Transport Association (IATA): Evidence-Based Training Implementation Guide. 2. International Air Transport

Association, Montreal - Geneva (2024). https://www.iata.org/contentassets/632cceb91d1f41d18cec52e375f38e73/ebt-implementation-guide.pdf
8. Marques, E., Carim Junior, G., Campbell, C., Lohmann, G.: Ab initio flight training: a systematic literature review. Int. J. Aerosp. Psychol. **33**(2), 99–119 (2023). https://doi.org/10.1080/24721840.2022.2162405
9. European Union Aviation Safety Agency (EASA): Easy access rules for organisation requirements for aircrew (part-ORA) (2020). https://www.easa.europa.eu/en/downloads/84907/en
10. European Union Aviation Safety Agency (EASA): Easy Access Rules for Aircrew. Regulation (EU) No 1178/2011. (2020). https://www.easa.europa.eu/en/downloads/115485/en
11. European Union Aviation Safety Agency (EASA): Flight examiners' manual (FEM) (2023). https://www.easa.europa.eu/en/document-library/general-publications/flight-examiners-manual-fem
12. European Union Aviation Safety Agency (EASA): Airline transport pilot (ATP) integrated course manual (2024). https://www.easa.europa.eu/en/document-library/general-publications/airline-transport-pilot-atp-integrated-course-manual
13. Niggemann, O., Biswas, G., Kinnebrew, J.S., Hranisavljevic, N., Bunte, A.: Konzeptualisierung als Kernfrage des Maschinellen lernens in der Produktion. In Handbuch Industrie 4.0: Band 2: Automatisierung, pp. 631–649. Springer (2024). ISBN: 978-3-662-58528-3 https://doi.org/10.1007/978-3-662-58528-3_73
14. Lilla, H., Niggemann, O., Netzel, T.: CPS prototype development for AI-based scenario adaptation in flight simulator training. CEAS Aeronaut. J.17, 855–868 (2026). https://doi.org/10.1007/s13272-025-00870-x
15. Sun, H., Zhou, X., Zhang, P., Liu, X., Lu, Y., Huang, H., Song, W.: Competency-based assessment of pilots' manual flight performance during instrument flight training. Cognit. Technol. Work **25**(4), 345–356 (2023). https://doi.org/10.1007/s10111-023-00737-3
16. Cross, J.I., Ryley, T.: Assessing evidence-based training in a collaborative virtual reality flight simulator. Aeronaut. J. **129**(1332), 261–281 (2025). https://doi.org/10.1017/aer.2024.113
17. Hwang, E.S., Kim, J.K.: Success factors of evidence-based training for pilots. Inf. Dev. **41**(2), 320–332 (2023). https://doi.org/10.1177/02666669231159100
18. Lilla, H., Tenberg, R.: KI- und LA-basierte Systeme in der Aus- und Weiterbildung von Pilot/-innen und luftfahrttechnischem Personal. In: Deutscher Luft- und Raumfahrtkongress 2022. Deutsche Gesellschaft für Luft- und Raumfahrt – Lilienthal-Oberth e.V., Bonn (2023). https://doi.org/10.25967/570300
19. Farelo, D.G., Bracaglia, L., Dailey, P., Tamrakar, S., Palladino, A., Carroll, M., Valenti, A.: An adaptive learning model for predicting and analyzing student performance on flight training tasks. In: Signal Processing, Sensor/Information Fusion, and Target Recognition XXXI, vol. 12122, pp. 278–286. SPIE (2022). https://doi.org/10.1117/12.2619068
20. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. stat. **29**(5), 1189–1232 (2001). https://doi.org/10.1214/aos/1013203451
21. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016). https://doi.org/10.1145/2939672.2939785
22. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y.: LightGBM: a highly efficient gradient boosting decision tree. Adv. Neural Inf. Proc. Syst. 30, 3146–3154 (2017). https://papers.nips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf
23. Wang, Q., Liu, C., Liu, S., Zhang, Y., Yang, J.: Flight maneuver evaluation method based on similarity metric via triplet network. IEEE Access Pract. Innov. Open Solut. **13**, 144456–144466 (2025). https://doi.org/10.1109/ACCESS.2025.3598248
24. Xu, G.J., Pan, S., Sun, P.Z., Guo, K., Park, S.H., Yan, F., Gao, M., Wanyan, X., Cheng, H., Wu, E.Q.: Human-factors-in-aviation-loop: multimodal deep learning for pilot situation awareness analysis using gaze position and flight control data. IEEE Trans. Intell. Transp. Syst. **26**(6), 8065–8077 (2025). https://doi.org/10.1109/TITS.2025.3558085
25. Guevarra, M., Das, S., Wayllace, C., Demmans Epp, C., Taylor, M., Tay, A.: Augmenting flight training with AI to efficiently train pilots. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37(13), 16437–16439 (2023). https://doi.org/10.1609/aaai.v37i13.27071
26. Li, F., Xu, X., Li, J., Hu, H., Zhao, M., Sun, H.: Wind Shear operation-based competency assessment model for civil aviation pilots. Aerospace **11**(5), 363 (2024). https://doi.org/10.3390/aerospace11050363
27. Dapica, R., Hernández, A., Peinado, F.: Who trains the trainers? Gamification of flight instructor learning in evidence-based training scenarios. Entertain. Comput. **43**, 100510 (2022). https://doi.org/10.1016/j.entcom.2022.100510
28. Källström, J., Granlund, R., Heintz, F.: Design of simulation-based pilot training systems using machine learning agents. Aeronaut. J. **126**(1300), 907–931 (2022). https://doi.org/10.1017/aer.2022.8
29. Caballero, W.N., Gaw, N., Jenkins, P.R., Johnstone, C.: Toward automated instructor pilots in legacy air force systems: physiology-based flight difficulty classification via machine learning. Expert Syst. Appl. **231**, 120711 (2023). https://doi.org/10.1016/j.eswa.2023.120711
30. Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., Liang, P.: Concept bottleneck models. In: International Conference on Machine Learning, pp. 5338–5348. PMLR 119 (2020). http://proceedings.mlr.press/v119/koh20a/koh20a.pdf
31. Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., Sayres, R.: Interpretability beyond feature attribution: quantitative testing with concept activation vectors (TCAV) . In: Proceedings of the 35th International Conference on Machine Learning, PMLR 80, 2668–2677 (2018). https://proceedings.mlr.press/v80/kim18d.html
32. Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. Adv. Neural Inf. Process. Syst. **30**, 4765–4774 (2017). https://papers.nips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf
33. Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.-I.: From local explanations to global understanding with explainable AI for trees. Nat. Mach. Intell. **2**(1), 56–67 (2020). https://doi.org/10.1038/s42256-019-0138-9
34. Ebensperger, N., Razzaghi, P., Wei, P., Kalyanam, K.: Enhancing runway configuration assistant model: the role of explainable AI for model interpretability. In: AIAA Aviation Forum and Ascend 2025-3517 (2025). https://doi.org/10.2514/6.2025-3517
35. Degas, A., Islam, M.R., Hurter, C., Barua, S., Rahman, H., Poudel, M., Ruscio, D., Ahmed, M.U., Begum, S., Rahman, M.A., Bonelli, S., Cartocci, G., Di Flumeri, G., Borghini, G., Babiloni, F., Aricó, P.: A survey on artificial intelligence (AI) and eXplainable AI in air traffic management: current trends and development with future research trajectory. Appl. Sci. **12**(3), 1295 (2022). https://doi.org/10.3390/app12031295
36. Bernard, D., Perello-March, J., Solis-Marcos, I., Cooper, M.: A user-centered ontology for explainable artificial intelligence in aviation. In: Proceedings of the 2nd International Conference on Cognitive Aircraft Systems - ICCAS, pp. 88–94. SciTePress/INSTICC (2024). ISBN: 978-989-758-724-5. https://doi.org/10.5220/0012978600004562

37. Flores, A., Paselk, A.: Enhancing pilot training performance: a scoping review of artificial intelligence technologies. Train. Educ. Learn. Sci. **193**, 1–19 (2025). https://doi.org/10.54941/ahfe1006641
38. Fenn, J., Nicholson, M., Pai, G., Wilkinson, M.: Architecting safer autonomous aviation systems. Preprint at arXiv:2301.08138 (2023). https://arXiv:2301.08138
39. Brosowsky, M., Keck, F., Dünkel, O., Zöllner, M.: Sample-specific output constraints for neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35(8) 6812–6821 (2021). https://doi.org/10.1609/aaai.v35i8.16841
40. Goyal, K., Dumancic, S., Blockeel, H.: DeepSaDe: learning neural networks that guarantee domain constraint satisfaction. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38(11), 12199–12207 (2024). https://doi.org/10.1609/aaai.v38i11.29109
41. Wabersich, K.P., Zeilinger, M.N.: A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. Automatica **129**, 109597 (2021). https://doi.org/10.1016/j.automatica.2021.109597
42. Hildebrandt, C., Köcher, A., Küstner, C., López-Enríquez, C.-M., Müller, A.W., Caesar, B., Gundlach, C.S., Fay, A.: Ontology building for cyber-physical systems: application in the manufacturing domain. IEEE Trans. Autom. Sci. Eng. **17**(3), 1266–1282 (2020). https://doi.org/10.1109/TASE.2020.2991777
43. Kabashkin, I.: Ontology-driven digital twin framework for aviation maintenance and operations. Mathematics **13**(17), 2817 (2025). https://doi.org/10.3390/math13172817
44. Zhou, Y.-F., Yang, H.-L., Li, J.-J., Lin, Y.-D.: Automata for knowledge assessment based on the structure of observed learning outcome taxonomy. Inf. Sci. **659**, 120058 (2024). https://doi.org/10.1016/j.ins.2023.120058
45. Waterman, M.W., Frezzo, D.C., Wang, M.X.: Adaptive learning using finite state machine logic. In: Proceedings of the Seventh ACM Conference on Learning@ Scale, pp. 237–240 (2020). https://doi.org/10.1145/3386527.3406720
46. Sarkar, P., Kar, C.: Adaptive e-learning using deterministic finite automata. Int. J. Comput. Appl. 97(21), 14–17 (2014). https://doi.org/10.5120/17130-7712
47. Dowling, C.E., Hockemeyer, C.: Automata for the assessment of knowledge. IEEE Trans. Knowl. Data Eng. **13**(3), 451–461 (2001). https://doi.org/10.1109/69.929902
48. Weiss, G., Goldberg, Y., Yahav, E.: Extracting automata from recurrent neural networks using queries and counterexamples. In: International Conference on Machine Learning, pp. 5247–5256. PMLR 80 (2018). http://proceedings.mlr.press/v80/weiss18a/weiss18a.pdf
49. Merrill, W., Tsilivis, N.: Extracting finite automata from RNNs using state merging. Preprint at arXiv:2201.12451 (2022). https://arxiv.org/abs/2201.12451
50. Umili, E., Capobianco, R., DeepDFA: automata learning through neural probabilistic relaxations. In: ECAI 2024, Frontiers in Artificial Intelligence and Applications, vol. 392, pp. 1051–1058. IOS Press (2024). https://doi.org/10.3233/FAIA240596
51. Diedrich, A., Moddemann, L., Niggemann, O.: On validating propositional logic system descriptions for fault diagnosis. Eng. Appl. Artif. Intell. **165**, 113379 (2026). https://doi.org/10.1016/j.engappai.2025.113379
52. Kontos, G., Lehmann, O.: Supporting a flight instructor for customizing training scenarios. An evidence based approach. In: Vrochidis, S., Ducq, Y., Gialampoukidis, I., Hribernik, K. (eds) Enterprise Interoperability XI, pp. 79–90. Springer Nature Switzerland, Cham (2025). ISBN: 978-3-031-93995-2. https://doi.org/10.1007/978-3-031-93995-2_7

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.