AI-ASSISTED BRIDGE CONCEPTUAL DESIGN: A HYBRID KNOWLEDGE- AND LEARNING-BASED APPROACH TO AUTOMATED SPAN PLANNING

Han Qian[1], Tim Noack[1], Mengyan Peng[1], Marx Steffen[1] and Chongjie Kang[1]
[1]Technische Universität Dresden, Dresden, Germany

Abstract

Bridge conceptual design is challenged by heterogeneous boundary conditions, strict engineering constraints, and large solution spaces. This paper presents an AI-assisted pipeline for automated bridge conceptual design, focusing on early-stage span layout planning. The proposed approach integrates ontology-based semantic knowledge representation with reinforcement learning to enable automated decision-making while preserving engineering consistency. By embedding engineering constraints into the learning environment, the method supports efficient exploration of feasible span configurations. The results indicate that the proposed pipeline provides a robust and extensible foundation for AI-supported bridge planning in realistic engineering scenarios.

Introduction

Bridge conceptual design is a pivotal early-stage task in which key decisions such as span layout and structural system selection are made under heterogeneous boundary conditions, such as site conditions of terrain and crossed obstacles, clearance requirements, environmental restrictions, and project-specific constraints (Troitsky, 2019). While decisions at this stage strongly influence feasibility, cost, and downstream design flexibility, the workflow is still dominated by manual iterations and experience-driven heuristics. In today's digitalized and automation-oriented construction industry, this creates two persistent challenges: (i) early-stage information is often incomplete, symbolic, and distributed across different sources, and (ii) the design space is combinatorial, yet must remain consistent with engineering constraints.

Artificial Intelligence (AI) is increasingly adopted across civil engineering and construction to improve efficiency, consistency, and scalability of decision-making, which can be a viable path to address the fragmented nature of early-stage design information. Machine Learning (ML) has been used to support structural engineering beyond inspection, including design-related prediction and rapid evaluation. Supervised Learning (SL) and Deep Learning (DL) models are often adopted as surrogate predictors to approximate the mapping between design boundary conditions and design variables (Qian et al., 2024), or between design variables and structural responses (Zheng et al., 2020; Balmer et al., 2024). These ML-based methods enable efficient screening of conceptual alternatives and reduce repeated high-cost simulations. Recent work also explores transfer learning to improve robustness of learned predictors under limited data, which is common in civil engineering applications (Zhang et al., 2025). Beyond prediction, Reinforcement Learning (RL) is being investigated for automating design decisions that can be formulated as sequential optimization, e.g., learning to automatically design reinforced concrete structures (Jeong et al., 2021), and handling construction design coordination tasks such as automated reinforcement clash resolution in Building Information Modeling (BIM) (Liu et al., 2020). Nevertheless, purely ML-based approaches (SL/DL or RL) often lack explicit mechanisms to guarantee strict feasibility under hard engineering constraints and may remain difficult to interpret and justify in safety-critical conceptual design.

To address these limitations, knowledge-based and rule-based systems have been developed to preserve engineering consistency and enable transparent reasoning. Ontology-driven approaches can formalize domain entities, constraints, and relationships, thereby supporting deterministic inference and validation. Ontology- and rule-based systems have been used to encode engineering knowledge in an explicit, machine-interpretable form, enabling early constraint checking and reducing iterative design cycles (Cao et al., 2022). In bridge engineering, most ontology efforts have focused on rehabilitation and maintenance, e.g., knowledge bases for maintenance decision support, while comparatively few studies address ontology integration in bridge design workflows (Ren et al., 2019; Zhang et al., 2022; Aqlan et al., 2025). Recent work also emphasizes that systematic rule structuring and ontology-based knowledge organization remain important for transparency and traceable rule management in digital construction (Wen et al., 2024). However, knowledge-based systems alone are often conservative in practice. While they are strong at enforcing hard constraints and providing explainable logic, they typically lack flexible optimization capabilities when multiple soft objectives (e.g., economy, trade-offs between competing objectives) must be balanced under diverse boundary conditions.

These complementary strengths and weaknesses motivate hybrid AI strategies that combine knowledge-based constraint enforcement with learning-based exploration.

Accordingly, this paper proposes a hybrid AI-assisted framework for bridge conceptual design, with a focus on automated span planning. In contrast to purely constraint-guided RL and existing bridge layout automation methods, it uses ontology-based reasoning to translate boundary conditions into a structured feasibility space and employs RL within this constrained environment to explore and optimize span configurations under soft engineering preferences. By integrating explicit knowledge with adaptive decision-making, the framework aims to improve automation while maintaining engineering consistency, transparency, and extensibility for realistic bridge planning scenarios.

Methods

AI-assisted pipeline for bridge conceptual design

In bridge conceptual design, the involved tasks differ substantially in their data availability, constraint structure, and decision complexity, making the use of a single AI technique across all stages neither efficient nor reliable. This work therefore adopts a task-oriented AI-assisted pipeline in which different design subtasks are supported by AI methods that match their intrinsic characteristics.

As shown in Figure 1, the proposed pipeline decomposes bridge conceptual design into three main stages: (i) interpretation and structuring of boundary conditions, (ii) planning of bridge spans, and (iii) planning of superstructure cross-sections. Boundary conditions originate from diverse sources, including site conditions, regulatory requirements, and project-specific constraints. These inputs are predominantly semantic and rule-driven rather than data-intensive, and are therefore represented using ontology-based knowledge models. Ontologies enable consistent semantic interpretation of boundary conditions, support rule-based reasoning, and provide structured inputs for subsequent design steps. Bridge span planning constitutes the central layout decision within the pipeline. It involves selecting the number of spans, their lengths, and pier locations under complex geometric and functional constraints. This task exhibits a large combinatorial search space and strong interdependencies between design variables, making it unsuitable for purely rule-based enumeration. At the same time, strict engineering constraints must be satisfied to ensure feasibility. For this reason, span planning is addressed using a hybrid approach that combines symbolic design rules with RL, allowing automated exploration of feasible layouts while preserving engineering validity. Once a span layout is determined, the pipeline proceeds to the planning of superstructure cross-sections. In contrast to span planning, this task benefits from the availability of existing bridge data and relatively stable input-output relationships between boundary conditions and structural dimensions. As a result, supervised learning techniques are employed to predict key geometric parameters of superstructure components, while rule-based methods are applied where normative prescriptions are explicit and deterministic.

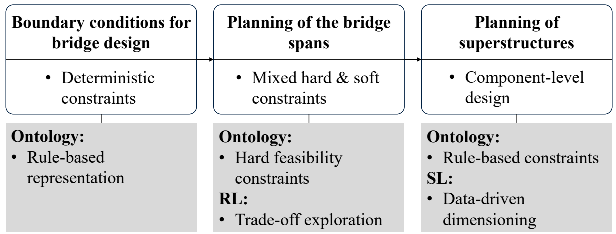


Figure 1: AI-assisted pipeline for bridge conceptual design

Hybrid methodology for bridge span planning

Within the AI-assisted conceptual design pipeline, bridge span planning represents the most critical decision-making task. The selection of span numbers, span lengths, and pier locations determines not only the overall structural layout but also construction feasibility, cost efficiency, and the quality of downstream design tasks.

Purely rule-based approaches have traditionally been used in early-stage bridge layout planning to ensure compliance with standards and engineering best practices (Troitsky, 2019). While such approaches provide strong guarantees of feasibility, they tend to be conservative and lack flexibility when confronted with complex or non-standard boundary conditions. Conversely, learning-based optimization techniques, such as RL, offer the ability to explore large design spaces and identify efficient solutions, but cannot inherently guarantee adherence to engineering rules and safety constraints. In bridge span planning, violations of such constraints are unacceptable, even at the conceptual design stage.

To address these complementary limitations, the proposed hybrid methodology combines rule-based reasoning with RL in a tightly coupled hybrid framework, as illustrated in Figure 2. The first step is to simplify the complex side conditions from three-dimensional (3D) to 2D topography. Explicit design rules and boundary conditions are then formalized using an ontology-based representation and used to encode domain knowledge and constraint logic, defining permissible, restricted, and forbidden regions for pier placement. RL is employed to perform sequential decision-making within this feasibility space defined by the ontology and further encoded design rules, because local support adjustments influence subsequent feasible actions and the overall quality of the final span layout. It enables automated exploration and optimization while maintaining engineering validity. Therefore, rule-based reasoning and RL fulfill distinct but interdependent roles. Semantic rules in Category A (see Table 1) act as constraints that shape the environment in which exploration takes place, reducing the effective search space and preventing infeasible solutions. RL, in turn, provides adaptability by optimizing layout decisions under these constraints and resolving trade-offs between competing objectives based on the rules in Category B (see Table 1), such as minimizing pier numbers while respecting economic span ranges. Rather than encoding all constraints only as reward penalties, the proposed framework externalizes hard, boundary-condition-dependent feasibility constraints through ontology-based reasoning and uses RL to optimize softer layout preferences within the admissible design space.

The hybrid methodology is structured to support modular extension and transparent interpretation. The following sections detail the categorization of design rules, their ontology-based representation, and the RL model used to realize this hybrid approach.

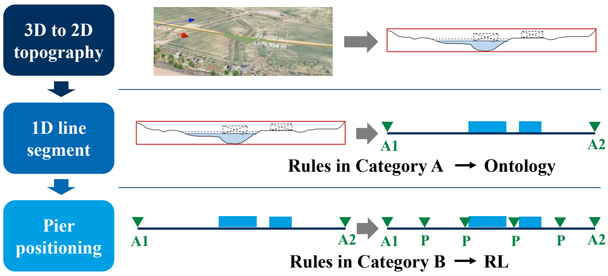


Figure 2: Hybrid methodology for bridge span planning

Design rules for bridge span planning

Design rules form the backbone of engineering knowledge in bridge span planning and play a central role in constraining and guiding the hybrid methodology. In the context of conceptual design, they are particularly important as detailed structural verification is not yet available, making explicit knowledge-based constraints essential for ensuring feasibility. For this work, span planning rules were systematically collected from multiple sources, such as German design standards, project-specific requirements, and engineering practice. The collected rules were then analyzed and structured with respect to their functions, rather than their original textual form. This functional analysis enables a clear distinction between rules that directly restrict geometric feasibility and those that provide qualitative or economic guidance for span layout decisions.

Table 1: Economical span ranges for different bridge types

Category A	Topography-dependent rules
Rule sources	• German design standards (e.g., DIN, RiZ-ING) • site-specific planning documents • engineering practice
Rule characteristics	• Hard constraints • Deterministic • Directly linked to terrain and obstacles
Typical rules	• Required clearance heights and widths for waterways and roads • Restrictions on pier placement in water bodies • Restrictions on pier placement in built-up areas and private property
Category B	Topography-independent rules
Rule sources	• Engineering handbooks • Guidelines • Expert knowledge
Rule characteristics	• Soft constraints • Heuristic and preference-based • Potentially conflicting
Typical rules	• Symmetry requirements of pier layout • Main span ≥ side spans • Trade-off between span length and number of piers • Economical span ranges for different bridge types

Based on this analysis, the design rules are categorized into two main groups according to their dependency on boundary conditions, which is demonstrated in Table 1. Category A rules are strongly dependent on topography and crossed obstacles, and define spatial constraints on span planning, such as clearance requirements over waterways and roads, and geometric limitations imposed by terrain conditions. These rules primarily determine whether a pier location is permissible, restricted, or forbidden along the bridge axis. Conversely, Category B rules are largely independent of local topography and describe general engineering principles for bridge span layout, e.g., symmetry requirements, and economical span length ranges for different bridge types. Unlike Category A rules, these rules do not impose absolute feasibility constraints but influence the quality and efficiency of candidate span layouts.

The separation of design rules into Category A and Category B supports their differentiated use within the hybrid span planning framework. Category A rules are primarily used to define hard constraints that shape the feasible design space, while Category B rules provide soft constraints and optimization guidance during layout generation. This structured rule system forms the basis for both ontology-based reasoning and RL, enabling explicit integration of engineering knowledge into automated bridge span planning.

Ontology-based rule representation and reasoning

To enable systematic and machine-interpretable application of topography-dependent design rules, an ontology-based representation is adopted for bridge span planning. Ontologies provide a formal and extensible framework for structuring heterogeneous engineering knowledge and are particularly suitable for representing explicit boundary conditions and hard feasibility constraints. In the proposed methodology, ontology-based reasoning is deliberately applied to the Category A rules.

The ontology used in this work is called Bridge Core Ontology (BCore), an extension of the Bridge Topology Ontology (BROT) proposed by Hamdan and Scherer (2020) which is again based on the Building Topology Ontology (BOT) described by Rasmussen et al. (2020). BCore was created with the ontology editor Protégé in the Web Ontology Language (OWL 2). BCore has multiple purposes in the field of bridge engineering, amongst them the semantic description of bridge boundary conditions. The Category A rules are implemented using the Semantic Web Rule Language (SWRL). Based on the bridge boundary conditions, they determine whether it is possible, inadvisable, or not possible to place a support (pier or abutment) at a specific point of the bridge axis.

The outputs of ontology-based reasoning consist of structured feasibility representations that define the admissible design space for bridge span planning. These representations serve as validated inputs to the RL environment, where Category B rules are subsequently integrated as optimization guidance. In this way, the ontology acts as a semantic filter that enforces hard boundary-condition-driven constraints, while leaving efficiency-oriented layout principles in Category B to be handled by learning-based decision making.

Learning-based decision making using RL

Based on the ontology-derived feasibility space, RL is used to generate bridge span layouts under complex boundary conditions. While ontology-based reasoning enforces hard constraints via Category A rules, Category B rules are embedded in the learning process to guide optimization toward efficient and balanced configurations. Operating within this rule-constrained environment, RL resolves trade-offs that are difficult to encode deterministically, enabling flexible yet engineering-consistent span planning with interpretable decision logic at the conceptual stage.

Problem formulation

Bridge span planning is formulated as a sequential decision-making problem in a custom Gymnasium (Towers et al., 2024) environment and modeled as a Markov Decision Process (MDP) (Garcia et al., 2013), defined by the tuple

$$M = \langle S, A, P, R, \gamma \rangle \quad (1)$$

where S denotes the state space, A the action space, P the state transition dynamics, R the reward function, and $\gamma \in (0,1)$ the discount factor. The problem formulation in RL is illustrated in Figure 3.

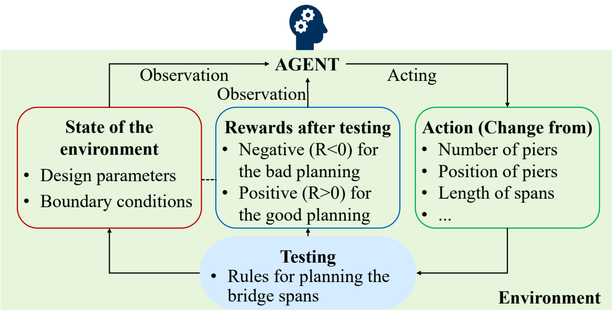


Figure 3: RL framework for bridge span planning

At each step, the learning agent interacts with an environment that represents the current bridge layout under given boundary conditions. The environment state $s \in S$ includes structured information derived from ontology-based reasoning, such as total bridge length L, admissible regions and forbidden zones that encode hard feasibility constraints (Category A rules). Additional state variables describe the current number of piers n, span lengths l_i, and pier positions x_i along the bridge axis. All quantities are normalized with respect to the total bridge length to ensure numerical stability during learning. The action space A comprises discrete layout edits, primarily moving piers forward or backward by a fixed step, and adding or removing piers within admissible regions. At each step, the environment executes the selected action, updates the state, computes a composite reward, and checks termination. State transitions P are deterministic given the selected action and current layout configuration. As shown in Figure 3, by observing a current state representing the bridge layout and boundary conditions, the agent is guided by current learned policy and selects an action that modifies pier positions or span layout and aims to maximize the cumulative reward. Then a testing procedure is applied in the environment, which evaluates the action against predefined rules and feasibility constraints. Valid actions lead to deterministic updates of the bridge layout and derived state variables, resulting in the next state. Actions violating hard feasibility constraints are rejected or penalized, so that infeasible states are not generated during learning.

Integration of design rules into the learning process

Category B rules encode soft principles and are incorporated into the RL framework as optimization guidance. Due to their qualitative nature and potential conflicts, these rules are not deterministic but are reflected in the reward formulation.

The reward function is defined as a weighted aggregation of multiple rule-related components:

$$R(s,a) = \sum_{i=1}^{N} \omega_i R_i(s,a) \quad (2)$$

where $R_i(s,a)$ denotes an individual reward component under specific state s after action a associated with a specific layout principle, and ω_i is the corresponding weighting factor. This formulation allows different engineering preferences to jointly influence the optimization process while preserving flexibility in resolving trade-offs between competing objectives.

In the current implementation, these reward components capture span validity, symmetry, uniformity, and span proportion preferences derived from Category B rules. The reward shaping provides continuous feedback that guides the learning agent toward high-quality span layouts. The reward function is designed to penalize layouts while rewarding solutions that achieve feasible span distributions. In the following, one representative reward, the component symmetry reward R_{sym}, is presented to illustrate how the Category B rule is translated into quantitative learning signals.

Symmetry of pier placement is evaluated based on the spatial distribution of pier positions along the bridge axis. For layouts with an odd number of piers, an additional requirement is imposed on the central pier. Let x_c denote the position of the middle pier, then its deviation from the bridge center is measured as:

$$e_{\text{center}} = |x_c - L/2| \quad (3)$$

For all layouts, symmetry of pier pairs is evaluated by comparing the sum of symmetric pier positions with the bridge length. For each mirrored pier pair (x_i, x_{n+1-i}), the pairwise symmetry error is defined as:

$$e_i = |(x_i + x_{n+1-i}) - L| \quad (4)$$

The overall symmetry error is computed as:

$$e_{\text{sym}} = \begin{cases} \frac{1}{\lfloor n/2 \rfloor} \left(\sum_{i=1}^{\lfloor n/2 \rfloor} e_i + e_{\text{center}} \right), & \text{if n is odd} \\ \frac{1}{\lfloor n/2 \rfloor} \sum_{i=1}^{\lfloor n/2 \rfloor} e_i, & \text{if n is even} \end{cases} \tag{5}$$

where $\lfloor n/2 \rfloor$ denotes the number of mirrored pier pairs. This formulation ensures that both mirrored pier alignment and, if applicable, the central pier position contribute to the symmetry test, while normalization prevents bias toward layouts with more piers.

Based on the computed symmetry error e_{sym}, the symmetry reward is defined as:

$$R_{\text{sym}}(e_{\text{sym}}) = \left(1 - \frac{e_{\text{sym}}}{T_{\text{sym}}}\right) R_{\text{sym,max}} \tag{6}$$

where T_{sym} represents symmetry threshold defining "acceptable" asymmetry and $R_{\text{sym,max}}$ denotes maximum achievable symmetry bonus magnitude. This reward provides a graded positive bonus when the layout is sufficiently symmetric (error below threshold) and a proportional penalty once the symmetry error exceeds the threshold, thereby encoding symmetry as a soft optimization objective rather than a hard constraint.

Beyond reward shaping, Category B rules are also used to define episode termination. An episode ends when a feasible span layout is achieved (success) or when the maximum step limit is reached. In the implementation, a solution is marked as DONE when all spans fall within the prescribed range and either (i) the layout meets a uniformity threshold, or (ii) it satisfies a symmetry threshold with the center span being the largest. This criterion reflects, that acceptable conceptual solutions may arise through different quality pathways rather than a single rigid condition.

Deep Q-Network (DQN) architecture and training

A DQN framework in RL (Mnik et al., 2015) is adopted to approximate the optimal action-value function $Q^*(s, a)$ for bridge span planning. Instead of explicitly modeling transition probabilities, learning is performed based on sampled transitions generated through interaction with the environment. Accordingly, the Bellman optimality principle is expressed in its sample-based form:

$$Q^*(s, a) = \mathbb{E}[r_t + \gamma \max Q^*(s_{t+1}, a') \,|s_t, a_t] \tag{7}$$

where s_t denotes the current state representing the bridge layout and boundary conditions at decision step *t*, while s_{t+1} is the subsequent state, a_t is the selected action modifying the current layout, r_t is the immediate reward obtained after applying action a_t, a' denotes an admissible action in the next state, and γ is the discount factor weighting future rewards.

In practice, the expectation is approximated using an experience replay buffer. During training, a separate target network with parameters θ is maintained to compute the temporal-difference target Q-values:

$$y_t = r_t + \gamma \max Q(s_{t+1}, a'; \theta) \tag{8}$$

where $Q(\cdot; \theta)$ denotes the target network used for stable target estimation. The main network parameters are updated by minimizing the squared temporal-difference error between the predicted Q-value and the target y_t. Then the target network parameters are periodically synchronized with the main network. This delayed update mechanism improves convergence robustness.

The network maps the environment state to Q-values for all admissible actions and is implemented as a multilayer perceptron (MLP) with two fully connected hidden layers and ReLU activations. Action selection follows an epsilon-greedy policy with probability ε, a random admissible action is chosen for exploration, otherwise, the action with the highest estimated Q-value is selected. The exploration rate ε is gradually decayed over training episodes. Training is conducted across diverse boundary conditions to promote generalization and to reliably produce feasible and efficient span layouts.

Results and case study

Results of ontology-based rule representation and reasoning

The ontology-based knowledge representation provides a structured and interpretable foundation for bridge span planning under heterogeneous boundary conditions. Design rules related to topography, crossed obstacles, clearance requirements, and pier feasibility are formalized as semantic entities and relations, enabling explicit reasoning over permissible (support-possible), restricted (support-not-recommended), and forbidden (support-not-possible) regions along the bridge axis.

The important classes of BCore are displayed in Figure 4. Every *Bridge carriesRoute InfrastructureRoute* which is composed of *RoutePoint*s. These points, being a *rdfs:subClassOf LocationPoint*, can be *locatedInZone. Zone*s can be, amongst others, a *Forest, WaterBody, Estate, Building* or another *Bridge*. Important *DataProperties* (attributes) in BCore are the coordinates of *RoutePoint* (*LocalX*, *LocalY*, *LocalZ*, *Station*; all of *Datatype xsd:decimal*) and its *xsd:boolean*-type properties *SupportNotPossible* and *SupportNotRecommended*. An *InfrastructureRoute* has the *DataProperties InfrastructureName* (*xsd:string*) and *InfrastructureWidth* (*xsd:decimal*).

With these semantics, the bridge's boundary conditions can be represented, as well as the Category A rules. Two examples of these rules are shown and explained in Table 2 and Table 3. They serve to ascertain that a pier or other bridge support cannot be placed on another *InfrastructureRoute*. All rules are implemented in SWRL and are thus hard *if-then*-rules. Other Category A rules include different types of *Zone*s, where a *RoutePoint* is located in. Depending on the type of *Zone*, either a support is not possible or not recommended at the respective *RoutePoint*. For example, in a *WaterBody*, a support is not recommended, while in a *Building*, it is not possible.

After creating instances of the *Bridge*, the carried route(s), surrounding *Zone*s and crossing routes, the bridge axis (= carried route) is divided into a high number of sections

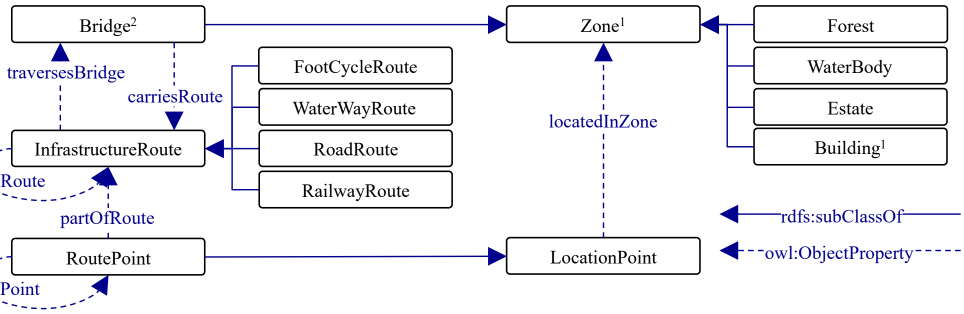


Figure 4: Diagram of BCore parts adapted from Rasmussen et al. (2020) for Zone [1] and Hamdan and Scherer (2020) for Bridge [2]

of the same length. A section length between 10 cm and 1 m is recommended. At the end of each section, a *RoutePoint* is created with its respective x-, y- and z-coordinates and its *Station* (kilometrage). In the proximity of the bridge, the other routes need their name, width, and *RoutePoint*s with their coordinates. Each *RoutePoint* that is located in a *Zone* needs to be connected to that *Zone* via the *locatedInZone*-property. After creating all these instances, a SWRL-capable reasoner (e.g., *Pellet*) needs to be run. For every point inside a zone or on a crossing route, the values of properties *SupportNotPossible* and *SupportNotRecommended* are inferred.

The points and their respective support-conditions can be exported or queried from the ontology.

Table 2: Rule route crossing point: description and SWRL-implementation

If two points from different routes have the same x- and y-coordinates, they are "equal" and their routes cross.

```
InfrastructureRoute(?r1) ^InfrastructureRoute(?r2)
^InfrastructureName(?r1,?n1) ^swrlb:notEqual(?n1,?n2)
^InfrastructureName(?r2,?n2) ^RoutePoint(?rp1)
^partOfRoute(?rp1,?r1) ^RoutePoint(?rp2)
^partOfRoute(?rp2,?r2) ^LocalX(?rp1,?x) ^LocalY(?rp1,?y)
^LocalX(?rp2,?x) ^LocalY(?rp2,?y)
-> equalsRoutePoint(?rp1,?rp2) ^crossesRoute(?r1, ?r2)
```

Table 3: Rule route crossing support: description and SWRL-implementation

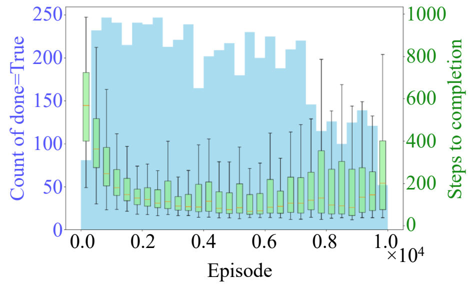
If a route point is located in the area of another route, a support is not possible at this point.

```
InfrastructureRoute(?r1) ^InfrastructureRoute(?r2)
^crossesRoute(?r1,?r2) ^RoutePoint(?rp1)
^partOfRoute(?rp1,?r1) ^RoutePoint(?rp2)
^partOfRoute(?rp2, ?r2) ^equalsRoutePoint(?rp1,?rp2)
^RoutePoint(?rp3) ^partOfRoute(?rp3,?r1)
^Station(?rp1,?s1) ^Station(?rp3,?s3)
^swrlb:subtract(?diff,?s1,?s3) ^swrlb:abs(?abs,?diff)
^InfrastructureWidth(?r2,?b) ^swrlb:divide(?bhalf,?b,2)
^swrlb:greaterThanOrEqual(?bhalf,?abs)
-> SupportNotPossible(?rp3,true)
```

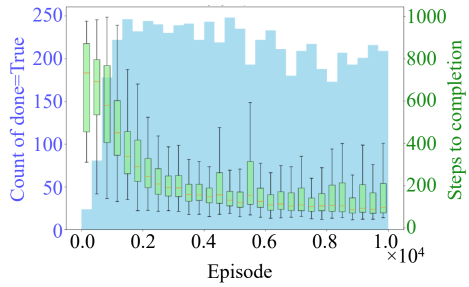
RL training and hyperparameter analysis

To achieve stable and efficient learning behavior in bridge span planning, a systematic hyperparameter tuning process was conducted. A broad range of parameters was explored, including training-related parameters (e.g., number of episodes and maximum steps per episode), DQN architecture parameters (e.g., network depth, neuron numbers, and replay memory size), and environment parameters (e.g., forbidden zone configurations and action step lengths), and reward-related parameters reflecting engineering objectives.

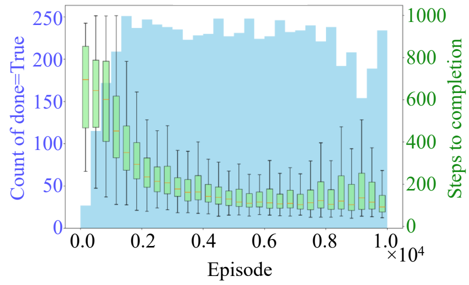
Special attention in this paper was given to parameters governing the exploration-exploitation balance, namely the epsilon target percent (etp) and epsilon decay percent (edp). These parameters define the decay schedule of the ε-greedy exploration strategy, where ε decays to etp within the first edp portion of training episodes. To evaluate training performance, two complementary metrics were adopted: (i) count of done, representing the number of episodes that successfully reached a valid span configuration within fixed episode intervals, and (ii) steps to completion, indicating the number of action steps required to reach termination in each episode. Together, these metrics provide insight into both learning effectiveness and planning efficiency.



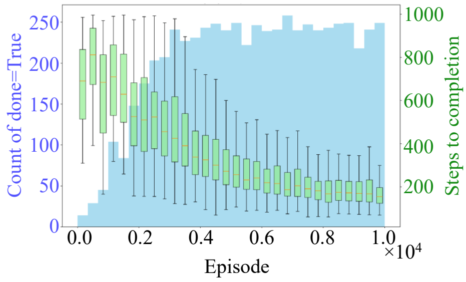
(a) etp = 40 %, edp = 30 %



(b) etp = 40 %, edp = 80 %



(c) etp = 70 %, edp = 30 %



(d) etp = 70 %, edp = 80 %

Figure 5: Training evolution of DQN model with the examples under four representative parameter pairs of etp and edp

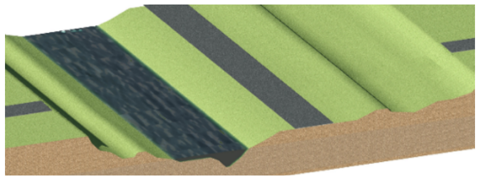
Figure 5 presents the training evolution with the examples under four representative parameter pairs of etp and edp, highlighting the interaction between exploration decay and learning stability. Apparently, for small values of (etp = 40%, edp = 30%) in Figure 5(a), the agent exhibits a rapid increase in successful episodes during early training stages. However, this improvement is not sustained, as the success rate decreases in later stages, and the variance of steps to completion increases. This pattern indicates early-stage overfitting, where the agent prematurely commits to suboptimal policies and loses robustness as training proceeds.

By comparing Figure 5(b) to (a) or Figure 5(d) to (c), increasing the decay period to edp = 80% significantly improves long-term stability. When comparing configurations with identical edp (see Figure 5(a) and (c), Figure 5(b) and (d)), a larger target exploration rate (etp = 70%) consistently yields more stable success rates and lower variance in step counts. These observations demonstrate that slower exploration decay and higher residual exploration are essential for avoiding local optima in bridge span planning tasks with complex constraints. Based on this comparative analysis, the parameter combination (etp = 70%, edp = 80%) was selected for subsequent experiments, as it provides the best balance between convergence speed, solution quality, and robustness.

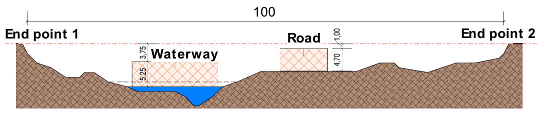
Case study

To demonstrate the practical applicability of the proposed pipeline, a simple case study was conducted, as shown in Figure 6. Starting from a simulated 3D terrain model in Autodesk Revit (Figure 6(a)), the terrain data were first transformed into a 2D longitudinal section (Figure 6(b)), from which ontology-based reasoning identified forbidden zones and feasible support regions considering terrain geometry and crossed obstacles (Figure 6(c)).

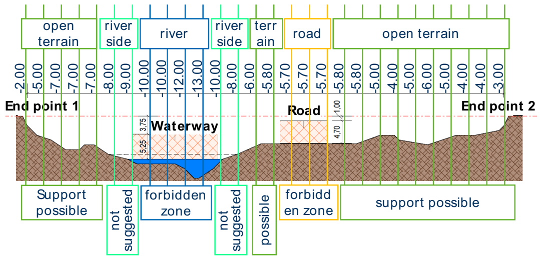
Based on the resulting 1D design space derived from ontology, the trained RL agent was then applied to this scenario to generate an optimized bridge span layout (green marks in Figure 6(d)). The resulting configuration satisfies all feasibility constraints while achieving a balanced span distribution consistent with engineering design principles in Category B. The generated span layout was subsequently mapped back into the 3D terrain model (Figure 6(e)), producing a conceptual bridge span design aligned with the terrain and boundary conditions.



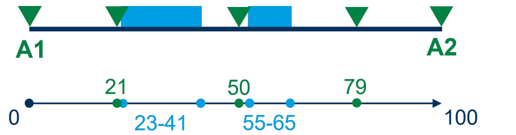
(a) 3D terrain model



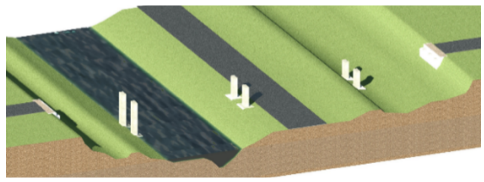
(b) Longitudinal 2D terrain



(c) Schema of ontology-based reasoning results



(d) Generated span layout through trained RL agent



(e) Bridge span layout in the 3D terrain model

Figure 6: A case study of bridge span planning using developed hybrid method

Conclusions and outlook

This paper presents an AI-assisted pipeline for bridge conceptual design, with a focus on automated span planning. Design rules are systematically collected and separated into topography-dependent feasibility rules and general layout principles related to efficiency and economy. The former are formalized through ontology-based reasoning to define the feasible design space, while RL is used to optimize span layouts under softer engineering preferences. The results demonstrate the feasibility of the proposed hybrid reasoning-and-learning framework for AI-assisted bridge span planning.

Despite these promising results, several limitations remain. Overall, the present study is intended as a proof-of-concept validation of the hybrid reasoning-and-learning framework rather than a comprehensive benchmark against alternative search strategies. From the ontology perspective, practical adoption is constrained by the effort and expertise required to develop and maintain ontologies with specialized languages and tools, as well as the need for engineers to formalize rules, configure the knowledge base, and interpret outputs. This highlights an important direction for future research, i.e., improving automation and user-friendliness at the knowledge-engineering level. In this context, Large Language Model (LLM)-based agents may help lower this barrier by supporting semi-automatic rule formalization, ontology population, and natural-language interaction with the system. From the RL perspective, the model performance is also sensitive to the formulation of the reward function. Although rule-guided reward shaping enables the incorporation of engineering knowledge into the learning objective, the selection, weighting, and interaction of reward components strongly affect training dynamics and can encode subjective trade-offs, limiting generalization across boundary conditions and bridge typologies. Future work will therefore explore adaptive or hierarchical reward strategies, automated tuning of reward weights, and broader evaluation across unseen boundary-condition scenarios, including comparisons with constrained-search and heuristic baselines, to improve robustness and reduce manual calibration.

In summary, this work demonstrates that combining ontology-based reasoning with reinforcement learning enables a robust and extensible framework for AI-assisted bridge conceptual design. The proposed framework will be extended to the downstream design task of superstructure cross-section design in our research project by combining ontology-based rule reasoning and SL. The pipeline is intended as a decision-support tool that improves efficiency, consistency, and transparency in early-stage bridge planning, while laying the groundwork for more automated and user-friendly design workflows.

Acknowledgments

This research was supported by the HyBridGen project (project number 01FV2067A) funded by the German Federal Ministry of Transport (BMV) within the framework of the mFUND funding program.

References

Aqlan, S., Hamdan, A. H., Marx, S., & Kang, C. (2025, July). Ontology-driven Database for Integrating Bridge Data from Multiple Open Sources. In EC3 Conference 2025 (Vol. 6, pp. 0-0).

Balmer, V., Kuhn, S. V., Bischof, R., Salamanca, L., Kaufmann, W., Perez-Cruz, F., & Kraus, M. A. (2024). Design space exploration and explanation via conditional variational autoencoders in meta-model-based conceptual design of pedestrian bridges. Automation in Construction, 163, 105411.

Cao, J., Vakaj, E., Soman, R. K., & Hall, D. M. (2022). Ontology-based manufacturability analysis automation for industrialized construction. Automation in Construction, 139, 104277.

Garcia, F., & Rachelson, E. (2013). Markov decision processes. Markov Decision Processes in Artificial Intelligence, 1-38.

Hamdan, A.-H. & Scherer, R. J. (2020). Integration of BIM-related bridge information in an ontological knowledgebase. In 8th Linked Data in Architecture and Construction Workshop - Proceedings.

Jeong, J. H., & Jo, H. (2021). Deep reinforcement learning for automated design of reinforced concrete structures. Computer-Aided Civil and Infrastructure Engineering, 36(12), 1508-1529.

Liu, J., Liu, P., Feng, L., Wu, W., Li, D., & Chen, Y. F. (2020). Automated clash resolution for reinforcement steel design in concrete frames via Q-learning and Building Information Modeling. Automation in Construction, 112, 103062.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. nature, 518(7540), 529-533.

Qian, H., Panagiotou, E., Peng, M., Ntoutsi, E., Kang, C., & Marx, S. (2024). A novel dataset and feature selection for data-driven conceptual design of offshore jacket substructures. Ocean Engineering, 303, 117679.

Rasmussen, M. H., Lefrançois, M., Schneider, G. F., & Pauwels, P. (2020). BOT: The building topology ontology of the W3C linked building data group. Semantic Web, 12(1), 143-161.

Ren, G., Ding, R., & Li, H. (2019). Building an ontological knowledgebase for bridge maintenance. Advances in Engineering Software, 130, 24-40.

Towers, M., Kwiatkowski, A., Terry, J., Balis, J. U., De Cola, G., Deleu, T., ... & Younis, O. G. (2024). Gymnasium: A standard interface for reinforcement learning environments. arXiv preprint.

Troitsky, M. S. (2019). Conceptual bridge design. In Bridge engineering handbook (pp. 1-1). CRC Press.

Wen, Y., Wang, M., Ariyachandra, M., Wei, R., & Xiao, I. B. (2024). Knowledge Driven Rule-Based Building Geometric Digital Twin Construction–State of Art Review. EC3 2024 proceeding.

Zhang, P. F., Zhang, D., Zhao, X. L., ... & Zhao, Q. (2025). Natural language processing-based deep transfer learning model across diverse tabular datasets for bond strength prediction of composite bars in concrete. Computer-Aided Civil and Infrastructure Engineering, 40(7), 917-939.

Zhang, Y., Liu, Y., Lei, G., Liu, S., & Liang, P. (2022): An Enhanced Information Retrieval Method Based on Ontology for Bridge Inspection. In: Applied Sciences 12 (20), S. 10599.

Zheng, H., Moosavi, V., & Akbarzadeh, M. (2020). Machine learning assisted evaluations in structural design and construction. Automation in Construction, 119, 103346.