

Intelligent Cruise Cabin Layout Design based on Reinforcement Learning

Jing Xu, Kefeng Yan

Shandong Jiaotong University, Weihai 264200, China

Abstract

This study focuses on the layout optimization of passenger spaces on cruise ships and proposes a reinforcement learning-based method for compartment arrangement in digital ship design. Traditional layout planning relies mainly on manual experience and repeated adjustments, making it difficult to respond efficiently to changes in deck boundaries, structural conditions, and functional requirements. The proposed method considers deck area, compartment boundaries, structural constraints, circulation, and functional relationships, and uses reinforcement learning to generate and adjust layout schemes through continuous interaction with the design environment. A diversity mechanism is introduced to reduce repeated layouts and provide more alternatives for designers. The results show that the method can generate feasible passenger compartment layouts under different design conditions while meeting the main spatial and functional requirements. The study provides a practical approach for improving the efficiency of cruise ship compartment planning and promoting the application of intelligent methods in digital ship design.

Keywords

Cruise Cabin Layout; Intelligent Generation; Layout Optimization; Digital Ship Design.

1. Introduction

The digital transformation of marine engineering has increased interest in automated spatial planning as a supplement to conventional vessel design. Manual preparation and revision of deck-level layout drawings require considerable engineering time, while automated layout generation can reduce repetitive work and provide more alternatives for design comparison^[1,2]. For cruise ships, passenger cabins are the main living spaces onboard, and their arrangement affects both space utilisation and passenger comfort^[3]. Conventional cabin layout still depends largely on repeated manual adjustment, resulting in long design cycles. To improve this process, Wang et al.^[4] developed an intelligent design framework combining deck-region modelling with heuristic optimisation. Li et al. used evolutionary algorithms to consider equipment relationships and vessel stability in cabin equipment arrangement. Meng et al.^[5] combined two-dimensional bounding boxes with path-search methods to check accessibility during layout optimisation. Louvros et al.^[6] used integer programming and multi-objective optimisation to consider adjacency, noise and evacuation requirements. Su et al.^[7] optimised rectangular compartment coordinates with overlap penalties to obtain compact layouts. Cai et al.^[8] incorporated daylighting and comfort into cruise balcony suite layout evaluation, while Bao et al. developed a mathematical model with potential-energy evaluation for automated cabin arrangement. Although these studies have improved layout automation, their optimisation results still depend strongly on predefined parameters and have limited adaptability. Similar problems exist in automatic floor planning, where data-driven methods learn from annotated drawings but often depend on fixed network structures. Changes in design variables may require model reconstruction and retraining, limiting their flexibility in practical engineering applications.

This paper investigates a collaborative agent-based optimisation paradigm for coping with variable spatial boundaries and personalised requirements in cruise passenger cabin arrangement. The proposed framework adapts to changing spatial environments and satisfies multiple customised design demands, assisting engineers in producing vectorised cabin layouts. Two core technical contributions are presented. First, a multi-level evaluation function classifies planning tasks into standard and customised categories, with six regular assessment criteria and two personalised metrics guiding the agents to follow marine design specifications and established design principles, thereby balancing layout rationality and customisation. Second, a collaborative information-sharing mechanism strengthens communication among distributed agents, facilitates the exchange of design knowledge, and prevents the optimisation process from converging on highly similar configurations, thereby enriching the diversity of candidate layout outputs.

2. Proposed Method

In the present work, the spatial arrangement task for cruise passenger compartments is reconstructed as a sequential decision-making optimisation problem that mimics human-led design workflows. Under this computational paradigm, independent computational entities iteratively position various functional components inside compartment boundaries. Subsequent parts of this section offer detailed descriptions regarding problem modelling, observation-decision dimensions, objective evaluation criteria, as well as the iterative optimisation pipeline.

2.1 State Space and Action Space

The state space contains the information required by an agent to characterize the current cabin configuration. To represent both facility attributes and the evolution of available space, facility descriptors and spatial occupancy information are jointly encoded in the state vector.

For facility i , the state descriptor is $f_i = (x_i, y_i, w_i, h_i, c_i)$, where (x_i, y_i) is the lower-left coordinate, w_i and h_i are the facility width and height, and c_i identifies the facility category. The remaining available cabin space is represented by the binary occupancy matrix M_t :

$$M_t(x, y) = \begin{cases} 1, & (x, y) \in \Omega_t^{\text{occ}}, \\ 0, & (x, y) \in \Omega_t^{\text{free}}. \end{cases} \quad (1)$$

The occupancy matrix is updated after each placement to reflect the evolving use of spatial resources. The complete system state is therefore defined as:

$$\mathbf{s}_t = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_{n_t}, M_t]. \quad (2)$$

This representation provides the agents with both inter-facility spatial relationships and real-time information on the remaining feasible region, thereby supporting subsequent placement decisions.

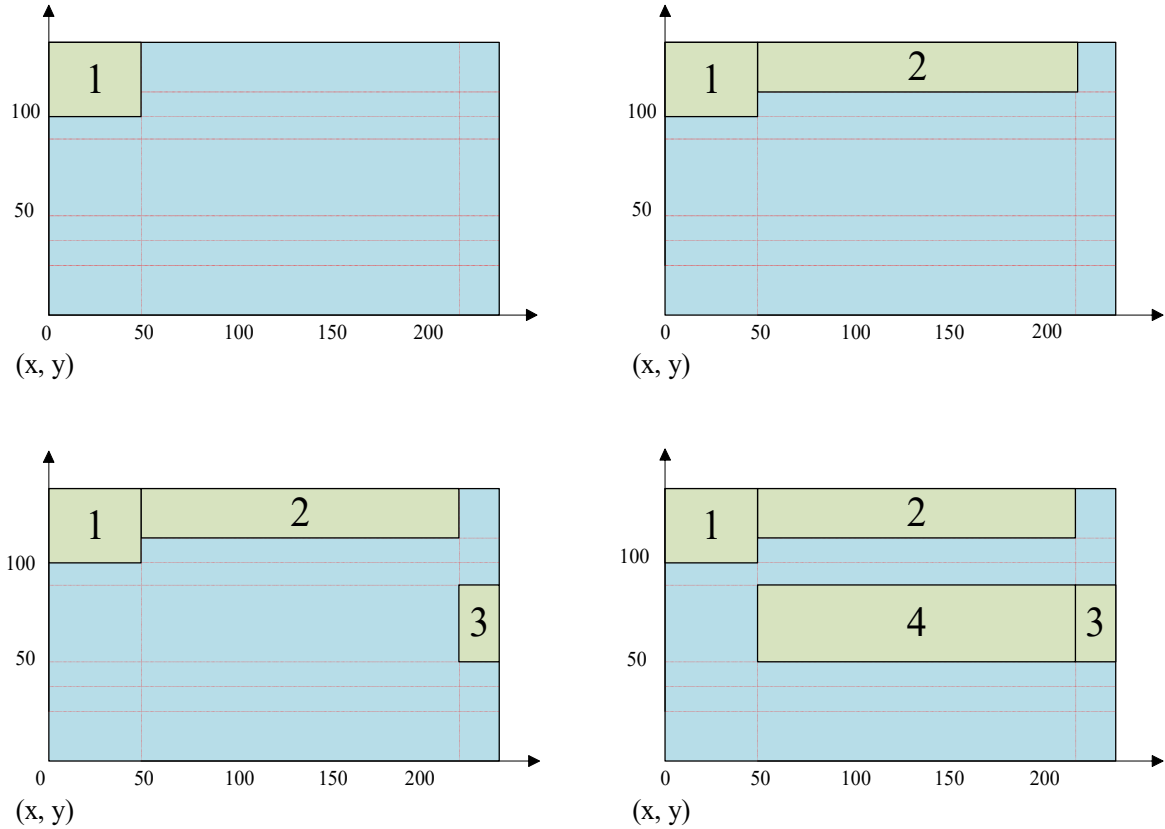


Fig. 1. Illustration of the dynamic facility-size selection procedure. Candidate dimensions are derived from the remaining boundary-aligned grid and updated sequentially as facilities are placed.

2.2 Reward Design

During the layout generation process, different reward functions are used to guide and constrain each action taken by the agent. After placing a facility, the environment evaluates the current layout and assigns a corresponding reward. Actions that satisfy the cabin boundary, facility dimensions, spatial relationships, and functional requirements receive positive rewards, while actions that cause overlap, exceed the cabin boundary, or violate layout requirements are penalized. Through the combined effect of these reward functions, the agent gradually adjusts the position and arrangement of each facility. In this way, the final cabin layout can meet the basic requirements of ship design while gradually approaching the target arrangement.

The reward guides the agent using the standard inputs of cabin boundary, facility configuration, and entrance location. Four criteria are considered: boundary containment, non-overlap, entrance clearance, boundary adjacency. The overall regular-task reward is the sum of the six components:

$$R_t^{\text{reg}} = r_t^{\text{inside}} + r_t^{\text{overlap}} + r_t^{\text{entrance}} + r_t^{\text{adjacency}} \quad (3)$$

The four terms denote the in-boundary, non-overlap, entrance-clearance, adjacency. **Fig. 2** provides a graphical summary of these components.

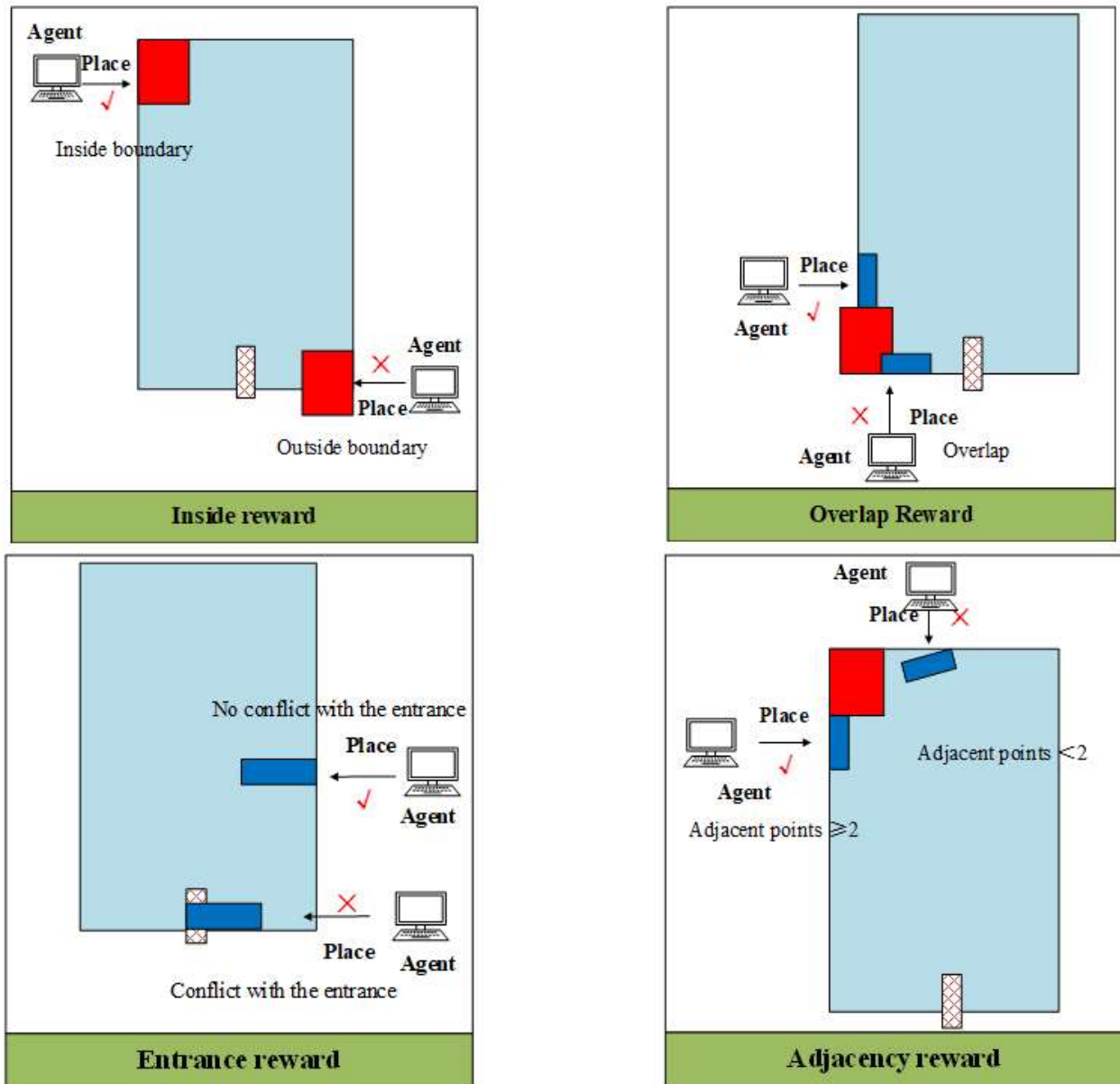


Fig. 2. Graphical definitions of the six regular-task rewards: in-boundary placement, non-overlap, entrance clearance, boundary adjacency, entrance-distance preference, and preservation of a central activity area.

In-boundary reward. Because an agent freely selects facility positions and dimensions, a candidate facility may extend outside the cabin polygon. The four candidate vertices are therefore tested using a ray-casting method. A positive reward is assigned only when every vertex lies inside the cabin boundary; otherwise, the action is penalized:

$$r_t^{\text{inside}} = \begin{cases} 10, & \prod_{k=1}^4 I_B(p_{t,k}) = 1, \\ -10, & \text{otherwise.} \end{cases} \quad (4)$$

$$I_B(p) = \begin{cases} 1, & p \in B, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

where $I_{B(p)}$ is the indicator of whether point p lies inside the cabin polygon B .

Non-overlap reward. A valid layout requires pairwise separation of all facilities. The total pairwise intersection area is evaluated; zero intersection yields a positive reward, whereas any overlap is penalized:

$$r_t^{\text{overlap}} = \begin{cases} 10, & \sum_{i=1}^{n_t-1} \sum_{j=i+1}^{n_t} |F_i \cap F_j| = 0, \\ -10, & \text{otherwise.} \end{cases} \quad (6)$$

where $|F_i \cap F_j|$ denotes the intersection area between facilities i and j .

Entrance-clearance reward. Facilities must not obstruct the entrance region. The reward is determined from the intersection between the newly placed facility F_t and the rectangular entrance zone E . A conflict-free placement is rewarded, whereas a nonzero intersection is penalized:

$$r_t^{\text{entrance}} = \begin{cases} 10, & |F_t \cap E| = 0, \\ -10, & \text{otherwise.} \end{cases} \quad (7)$$

where E is the entrance rectangle centered at (x_e, y_e) , with width w_e and height h_e .

Adjacency reward. To improve space utilization and visual coherence, each facility is encouraged to share at least one edge with the cabin boundary. Operationally, at least two facility vertices must lie on boundary segments. A placement is rewarded when this condition is satisfied; otherwise, it is penalized:

$$r_t^{\text{adjacency}} = \begin{cases} 10, & \sum_{k=1}^4 I_{\partial B}(p_{t,k}) \geq 2, \\ -10, & \text{otherwise.} \end{cases} \quad (8)$$

$$I_{\partial B}(p) = \begin{cases} 1, & p \in \overline{v_m v_{m+1}} \subset \partial B, \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

where $I_{\partial B}(p)$ indicates whether point p lies on a boundary segment joining consecutive vertices v_m and v_{m+1} .

2.3 Training Procedure

The objective of multi-agent reinforcement learning is to learn effective individual and collective decisions through interaction with the environment and with other agents. In the proposed framework, each agent acts as an independent designer and employs a separate deep Q-network (DQN) to select facility dimensions and placement actions. At each step, an agent selects an action using an epsilon-greedy policy, executes it in the layout environment, and receives the reward and next state. The minimum LS between the current layout and the layouts generated by the other agents is then evaluated. A transition is retained only when the layout is sufficiently distinct, i.e., when the minimum LS is not less than τ_{SEA} . Mini-batches are sampled from the retained experiences to train the online network, while the target network provides stable target values. For a terminal state, the bootstrap term is suppressed by the done indicator:

$$y_t = r_t + \gamma(1 - d_t) \max_{a'} Q_{\bar{\theta}}(s_{t+1}, a'). \quad (10)$$

The loss function measures the discrepancy between the online-network prediction and the target value. For each sampled mini-batch, gradients are backpropagated to update the online-network

parameters. Progressive reduction of the loss indicates improved value estimation and contributes to stable convergence of the layout policy:

$$\mathcal{L}(\theta) = \mathbb{E}[(Q_{\theta}(s_t, a_t) - y_t)^2]. \quad (11)$$

3. Experimental Setup

All experiments were performed on a server equipped with 18 virtual CPU cores of an AMD EPYC 9754 128-core processor and an NVIDIA RTX 4090D GPU with 24 GB of memory. The algorithms were implemented in Python. Cabin boundaries, entrance locations, and facility configurations extracted from selected accommodation drawings of the 800K W-354-00QD vessel were used as design inputs. The main experimental parameters are summarized in Table 1.

Table 1. DQN core parameters

Hyperparameter	Value
Number of agents	4
Training episodes	2000
Learning rate	0.0005
Exploration rate	0.05
Mini-batch size	64
Discount factor γ	0.95
Replay batch size	128
SEA threshold	0.5
Experience replay buffer	50000

3.1 Evaluation Metrics

The generated layouts are quantitatively evaluated using cumulative reward (CR), area-consistency score (ACS), gangway-width reward, graph edit distance (GED), and layout-constraint score (LCS). CR is used primarily to assess training convergence and is defined as:

$$CR_i = \sum_{e=1}^E R_{i,e}. \quad (12)$$

where $R_{i,e}$ is the cumulative reward obtained by agent i in training episode e , and E is the number of training episodes.

ACS measures consistency between the target facility areas specified in the input and the areas of the generated facilities:

$$ACS = \frac{1}{n} \sum_{i=1}^n \frac{A_i'}{A_i}. \quad (13)$$

where n is the number of input facilities, A_i is the prescribed area of facility i , and A_i' is its generated area.

LCS quantifies the overall satisfaction of the six regular design constraints and is used as an indicator of layout rationality:

$$LCS = \frac{1}{6}(c_{\text{inside}} + c_{\text{overlap}} + c_{\text{entrance}} + c_{\text{adjacency}} + c_{\text{distance}} + c_{\text{activity}}). \quad (14)$$

$$c_{\text{activity}} = 1 - \frac{\sum_{i=1}^n |F_i \cap L|}{|L|}. \quad (15)$$

The four terms represent satisfaction of the in-boundary, non-overlap, entrance-clearance, adjacency. Binary terms take a value of 1 when the corresponding constraint is satisfied and 0 otherwise; the activity-space term is evaluated from the ratio between the total facility–activity-region intersection area and the activity-region area.

3.2 Experiments

Experiments use only the cabin boundary, prescribed facility areas, and entrance location as inputs, without additional customized requirements. The facilities comprise a bed, desk and chair, sofa, balcony, and bathroom, selected from predefined dimension sets and placed sequentially. The principal facility configurations are listed in **Table 2**.

Table 2. Different sizes of main facilities in the cabin

Type	Facility configurations					
	Bed	Sofa	Bathroom	Desk	Wardrobe	Balcony
T1	3.6m ²	0.78m ²	2.867m ²	1.8m ²	0.752m ²	4.35m ²
T2	3.6m ²	/	3.06m ²	/	1.2m ²	/
T3	3.6m ²	/	2.419m ²	/	1.11m ²	/
T4	3.6m ²	0.91m ²	3.9m ²	1.05m ²	0.84m ²	4.8m ²

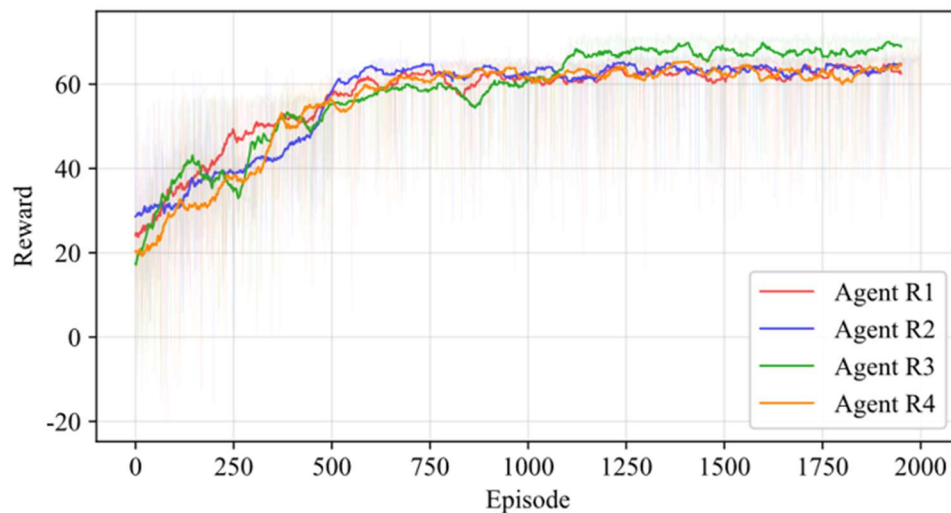


Fig. 3. Reward trajectories of the four agents during training

Fig. 3 shows the reward curves of the four agents. For the two customized tasks, the agents show different learning speeds and reward fluctuations because each agent explores the layout space independently. Despite these differences, the reward values of all four agents gradually increase during training and become relatively stable in the later stages. This suggests that the agents can gradually adjust their placement strategies through repeated interaction with the environment and finally obtain feasible layout policies that meet the corresponding customized objectives.

3.3 As-built Cabin Layouts

A compatibility experiment was also carried out using the boundary, entrance, and facility arrangement of an as-built cabin as the input conditions. The four agents were trained for 4000 episodes, as shown in **Fig. 4**. During training, the generated layouts gradually became closer to the reference cabin arrangement as the placement policies were updated. At episode 1187, the generated layout was nearly identical to the as-built drawing. This result shows that, when the input conditions are consistent with an existing cabin, the trained policy can produce a layout close to the actual engineering arrangement, indicating that the method is capable of reproducing practical layout decisions under the corresponding design conditions.

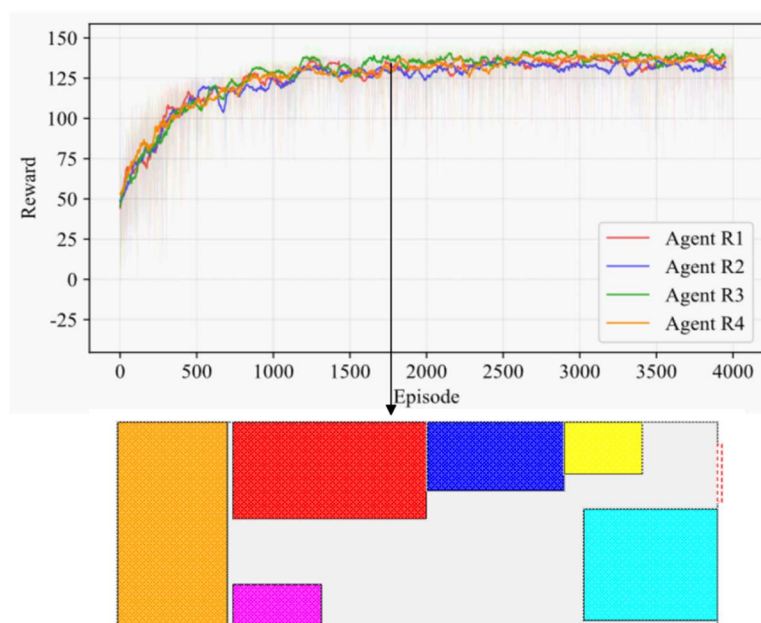


Fig. 4 Compatibility evaluation against an as-built cruise-cabin layout.

4. Conclusion

This study developed a reinforcement learning (RL)-based method for automated cruise-cabin layout design. The agent interacts with the layout environment step by step and adjusts the positions of cabin facilities according to the current layout state. In this way, the method can generate feasible cabin arrangements while reducing repeated manual adjustments. The multi-agent framework allows different agents to explore the same design conditions separately, providing several layout alternatives for comparison.

A controllable cabin-layout framework was established to meet the basic requirements of cruise ship interior design. Reward functions were used to guide facility placement and dimension selection during the layout process. Several cabin types from the 800K W-354-00QD vessel were used for testing. The results show that the proposed method can generate feasible layout schemes under the given cabin boundaries and design requirements. Overall, the study indicates that RL can improve the efficiency of automated cruise-cabin layout and provide additional layout options for ship designers.

Acknowledgments

Jing Xu: Experiment, Investigation, Methodology, Conceptualization, Writing - Original Draft, Writing - Review & Editing. **Kefeng Yan:** Investigation, Data Curation.

References

- [1] Hu, R., Huang, Z., Tang, Y., van Kaick, O., Zhang, H., & Huang, H. (2020). Graph2Plan: Learning floorplan generation from layout graphs. *ACM Transactions on Graphics*, 39(4), 118:1–118:14.
- [2] Li, J., Guo, H., Zhang, S., Wu, X., & Shi, L. (2019). Optimum design of ship cabin equipment layout based on SLP method and genetic algorithm. *Mathematical Problems in Engineering*, 2019, Article 9492583.
- [3] Louvros, P., Boulougouris, E., & Coraddu, A. (2022). Multi-objective optimisation as an early design tool for smart ship internal arrangement. *Ships and Offshore Structures*, 17(6), 1392–1402.
- [4] Meng, X., Sun, H., & Kang, J. (2021). Equipment layout optimization based on human reliability analysis of cabin environment. *Journal of Marine Science and Engineering*, 9(11), 1263.
- [5] Nauata, N., Chang, K.-H., Cheng, C.-Y., Mori, G., & Furukawa, Y. (2020). House-GAN: Relational generative adversarial networks for graph-constrained house layout generation. In *Computer Vision – ECCV 2020* (pp. 162–177). Springer.
- [6] Su, S., Zheng, Y., Xu, J., & Wang, T. (2020). Cabin placement layout optimisation based on systematic layout planning and genetic algorithm. *Polish Maritime Research*, 27(1), 162–172.
- [7] Zhang, H., Yu, Y., Song, Z., Han, Y., Yang, Z., & Ti, L. (2024). Method for collaborative layout optimization of ship equipment and pipe based on improved multi-agent reinforcement learning and artificial fish swarm algorithm. *Journal of Marine Science and Engineering*, 12(7), 1187.
- [8] Heinbach, B., Burggräf, P., & Wagner, J. (2023). Deep reinforcement learning for layout planning: An MDP-based approach for the facility layout problem. *Manufacturing Letters*, 38, 40–43.