International Journal of
Network Dynamics and Intelligence (IJNDI)

Article

A Grasping and Placement Learning Framework for Workpieces Packing in Complex Scenes

Zhilong Liu, Feiyue Wu, Qinghui Pan, and Jie Lian *

Key Laboratory of Intelligent Control and Optimization for Industrial Equipment, Ministry of Education, and the School of Control Science and Engineering, Dalian University of Technology, Dalian 116024, China

* Correspondence: jielian@dlut.edu.cn

Received: 1 May 2026; Revised: 16 June 2026; Accepted: 25 July 2026; Published: 17 August 2026

Abstract: Online 3D workpiece packing is crucial in logistics, but achieving high space utilization without prior knowledge of incoming item sequences remains challenging. Current methods struggle to capture dynamic 3D spatial topologies and heavily rely on rigid heuristics, which limits exploration diversity and causes space fragmentation. A deep reinforcement learning framework integrating a spatially weighted graph attention network and a diffusion model is proposed. The network accurately extracts the global topological dependencies and local geometric constraints of the packing state. Concurrently, the diffusion model synthesizes highly diverse, physically feasible candidate placement poses, fundamentally expanding the agent's exploration boundary. Experiments demonstrate that our method significantly outperforms baselines in space utilization and efficiency. Its effectiveness and robustness are further validated in a real robotic system, offering a practical hybrid reinforcement learning solution for online 3D workpiece packing.

Keywords: online 3D workpiece packing; deep reinforcement learning; graph attention network; diffusion model

1. Introduction

Industrial robots have automated labor-intensive supply chain tasks like order packing. The online 3D workpiece packing is a key logistics challenge, aiming to pack objects into a fixed-size container to maximize space utilization. In real-world online scenarios, complete incoming item sequences are unavailable, requiring instantaneous decisions. The rapid expansion of e-commerce has created a pressing need for versatile and robust online 3D workpiece packing strategies.

Current methods face significant limitations. Traditional heuristic approaches fail to capture dynamic 3D spatial topologies. Meanwhile, existing deep reinforcement learning methods rely on rigid rules. These rules severely restrict exploration and trap policies in local optima. To bridge these gaps, this study proposes a deep reinforcement learning framework. It integrates a spatially weighted graph attention network (SW-GAN) and a diffusion model. Specifically, SW-GAN accurately extracts 3D placement features using a tree structure. The diffusion model then combines these features with temporal embeddings. It generates diverse packing actions through multi-step denoising. This process fundamentally expands the exploration space. Consequently, the integration reduces ineffective trials and avoids local optima. Finally, the learned strategies are validated on physical robotic systems to evaluate their real-world feasibility and robustness.

The primary contributions of this study are outlined as follows:

- A graph attention network based on a positional penalty is proposed. It explicitly embeds a Euclidean distance penalty term into the attention weights. This enables the network to prioritize spatially adjacent nodes, thereby enhancing container utilization.
- A diffusion model is introduced to generate diverse candidate placement positions through multi-step denoising. This helps the deep reinforcement learning agent develop optimal placement strategies and significantly enhances its exploration capability.

Multiple experiments and ablation analyses are conducted in both simulated and real-world environments to evaluate the effectiveness of the proposed method. The results indicate that the model trained in the simulated environment can be directly transferred to actual placement tasks without the need for further fine-tuning.

2. Related Work

This section presents a concise review of the literature addressing the 3D bin packing problem, including heuristic methods and reinforcement learning-based methods.

2.1. Heuristic Methods

Early approaches to the 3D bin packing problem focus on developing heuristic algorithms. These algorithms are derived from manually summarized packing experiences and favored for their simplicity [1]. Additional methods are refined based on human expertise. The empty maximal space method introduces the concept of extreme points, enhancing container loading efficiency by identifying new potential points [2]. The internal corner point method aims to maximize the use of gap spaces typically overlooked by existing algorithms, thereby identifying optimal item placements to improve packing efficiency [3]. These methods seek to identify potential free spaces for item placement to enhance container utilization.

Ha et al. [4] propose an online bin packing heuristic for two-dimensional packing, selecting an empty maximal space to minimize the edge difference between the item's surface and the empty maximal space surface, improving space utilization and reducing item blockage. Hu et al. [5] introduce the maximizing accessible convex space method, designed to enhance reserved space utilization for efficiently loading larger future items. Nguyen et al. [6] develop a space partitioning technique, employing primary and secondary data structures to track available and unavailable spaces for effective searching. Chen et al. [7] focus on large-scale 3D packing scenarios with robot constraints and propose corresponding optimization algorithms.

However, these methods rely heavily on pre-defined human decision rules to guide solutions, resulting in limited exploration spaces. Fixed prior methods risk missing optimal solutions. Algorithm adjustments are required for new scenarios, demanding extensive expertise and engineering effort. In this study, heuristic methods are introduced as constraints during the early stages of reinforcement learning training, successfully preventing model convergence issues.

2.2. Reinforcement Learning-Based Methods

Deep reinforcement learning has shown impressive performance in various tasks [8]. Data-driven reinforcement methods have been explored to address the bin packing problem. Liu et al. [9] propose a fusion-perception-to-action Transformer model, which uses 3D visual fusion attention and proprioception to enhance the precision of robotic manipulation. A deep reinforcement learning model with a Transformer network is developed by Que et al. [10] to solve the variable-height packing problem. Effective optimization is achieved, but flexibility in strategy adjustment is limited by offline processing in dynamic environments.

Deep reinforcement learning has been applied to tackle the online 3D workpiece packing problem in recent studies. However, deep reinforcement learning's exploratory learning is not always stable, and effective strategies are sometimes challenging to acquire. A constrained deep reinforcement learning model is developed by Zhao et al. [11]. A prediction mask module is integrated to ensure feasible actions during training, improving learning efficiency and strategy performance. A packing configuration tree is proposed in subsequent work as a structured representation of packing states and action spaces, simplifying deep reinforcement learning strategy learning [12].

A reinforcement learning-based robotic packing system is proposed by Xiong et al. [13]. Candidate maps are constructed to balance exploration and exploitation, identifying feasible placements in discrete action spaces, achieving high performance and reliability in online 3D workpiece packing. Wang et al. [14] further proposed a hierarchical reinforcement learning method for robotic bin packing to optimize task execution efficiency.

In summary, the fundamental shortcoming of existing heuristic methods lies in their inability to adapt to complex spatial topologies, while current deep reinforcement learning methods are bottlenecked by restricted exploration spaces derived from rigid rules. In contrast to these approaches, our proposed method eliminates the reliance on manually designed placement rules. By leveraging SW-GAN to dynamically capture 3D geometric constraints and utilizing a diffusion model to generate highly diverse candidate actions, our framework fundamentally expands the exploration boundary and achieves superior space utilization compared to existing baselines.

3. The Proposed Method

This section outlines the methodology for online 3D workpiece packing. An object is grasped from complex environments, and SW-GAN extracts features for candidate positions to train the reinforcement learning model. The high-quality experiences generated are stored in the replay buffer to update the diffusion model and incorporate candidate positions. The process is shown in Figure 1, which illustrates the complete workflow, including robotic grasping, feature extraction via SW-GAN, candidate position generation by the diffusion model, decision making and experience storage.

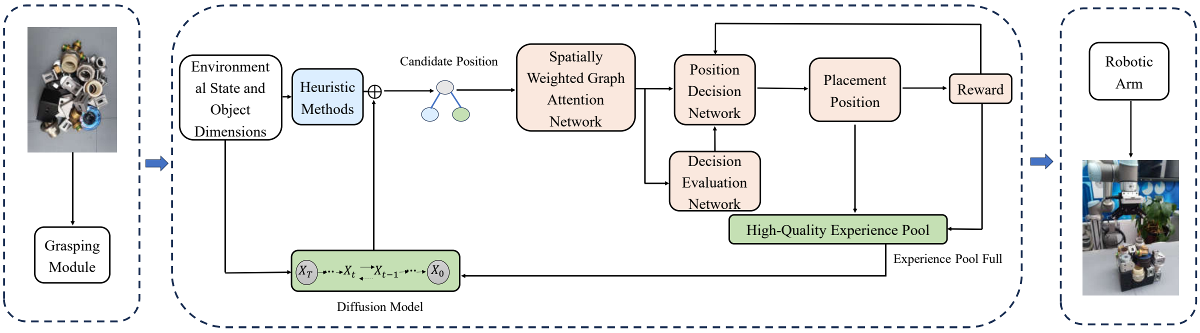

Figure 1. Overall framework of the proposed method.

3.1. Problem Description

The robotic arm performs grasping and placement of objects from a cluttered environment, with full item information unavailable beforehand [15]. Objects vary significantly in size, with few identical dimensions. Packing is carried out instantaneously based on the order in which items are grasped, and the strategies are predicted using limited prior box information. A camera measures rectangular item dimensions (l_i, w_i, h_i), and items are placed into a fixed-size container (L, W, H) to maximize space utilization. Stacking follows six predefined orientation constraints for stability, as shown in Figure 2. Boundary, parallel, and separation constraints ensure items remain within container limits, align parallel to inner walls, and avoid overlap for optimal space use and damage prevention. The optimization goal is to maximize container space utilization, measured by the space utilization rate S_p, calculated as

$$S_p = \frac{1}{L \cdot W \cdot H} \sum_{i=1}^{N} l_i \cdot w_i \cdot h_i \times 100\%, \quad i \in \{1, 2, 3, \ldots, N\}.$$

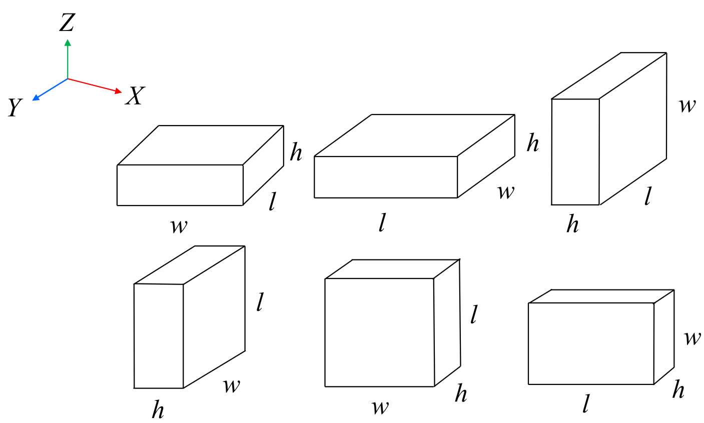

Figure 2. Schematic diagram of item placement orientation.

When a rectangular item n_t is placed at position (p_x, p_y, p_z) at time t, new potential positions are generated for subsequent items. Candidate positions and orientations are determined using alignment direction $o_t \in \Omega$ and existing item positions, as shown in Figure 3. The packing process replaces the node n_t with an empty node, creating new child nodes for potential placements, forming a dynamic packing configuration tree T_t. Internal nodes $B_t \subseteq T_t$ represent placed items' layout, while leaf nodes $L_t \subseteq T_t$ indicate available positions. Infeasible leaf nodes due to obstruction are removed from L_t. Packing ends when no leaf node meets the placement constraints for item n_t.

Two distinct experiments are employed to evaluate item stability and orientation constraints during the packing process. In experiment 1, items are restricted to two horizontal orientations ($|O| = 2$), permitting only orthogonal rotations. In experiment 2, items are allowed to rotate and flip in any direction ($|O| = 6$).

3.2. Reinforcement Learning Formulation

The placement process is formulated as a Markov Decision Process and employ a deep reinforcement learning approach to optimize the placement strategy. The state $s_t = (\mathcal{T}_t, n_t)$ is represented by a packing configuration tree $\mathcal{T}_t$, where internal nodes B_t encode the spatial configurations of packed items, and leaf nodes L_t denote the candidate placements for the current item n_t. The action a_t is defined as the index of a selected leaf node $l \in L_t$.

The deep reinforcement learning model leverages feature extraction networks to extract spatial relationship features from tree structure and feeds them into an actor-critic architecture.

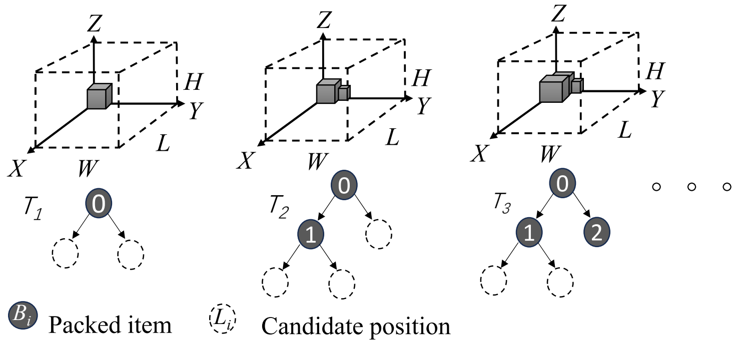

Figure 3. Tree structure growth diagram.

The objective of the training process is to maximize the expected cumulative discount reward expressed as

$$J(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t, a_t)\right],$$

where $\gamma \in [0, 1)$ denotes the discount factor, and the reward function $R(s_t, a_t)$ is defined as

$$R(s_t, a_t) = \alpha \cdot S_p - \beta \cdot S_w,$$

where α and β are both hyperparameters, S_w denotes the volume of wasted space before and after placing the current item.

The placement decision network generates the policy distribution $\pi(a_t|s_t)$ by assigning probability weights to each leaf node $l \in L_t$, focusing on relevant candidate nodes through a pointer mechanism. The optimal placement action $a_t = \text{index}(l)$ is determined by selecting the leaf node with the highest probability. This process considers feasible placement positions based on the learned spatial relationships between objects, ensuring efficient decision-making. The feasibility is further assessed through constraint verification. For all $i \neq j$ and $d \in \{x, y, z\}$, the non-overlapping constraint is defined as

$$p_i^d + s_i^d \leq p_j^d + S^d(1 - e_{ij}^d),$$

where p_i^d is the coordinate of the lower-left-front corner of item i along axis d, s_i^d is its length in that direction, S^d is the corresponding bin dimension, and $e_{ij}^d \in \{0, 1\}$ equals 1 if item i precedes item j along axis d and 0 otherwise. The containment constraint $0 \leq p_i^d \leq S^d - s_i^d$ ensures that every item is fully inside the bin.

3.3. A Feature Extraction Network

A spatially weighted graph attention network is proposed to extract features from the tree structure, where spatial relationships are critical for packing stability and utilization. The standard graph attention network is purely data-driven and may fail to capture accurate 3D geometric relations if spatial information is insufficient. To overcome this issue, a 3D Euclidean distance penalty term is introduced into the attention score, which is defined as

$$e_{ij} = \text{LeakyReLU}\left({a_m}^\top [W_a h_i \| W_a h_j] - \lambda \cdot \frac{\|p_i - p_j\|_2}{d_\text{m}}\right),$$

where W_a is the weight matrix, a_m represents the weighting coefficient, p_i and p_j are 3D coordinates of nodes, $\|\cdot\|_2$ denotes the Euclidean distance, d_m is the bin's maximum dimension for normalization, and λ balances the spatial penalty.

This mechanism assigns higher attention to spatially adjacent nodes, enhancing the perception of local geometric constraints and improving stacking stability. The attention weights are normalized by the softmax function:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=0}^{\mathcal{N}(i)} \exp(e_{ik})},$$

where $\mathcal{N}(i)$ represents the neighbor set of node i. The updated node feature is computed as

$$h_i' = \sigma\left(\sum_{j=0}^{\mathcal{N}(i)} \alpha_{ij} W_a h_j\right),$$

where σ denotes the nonlinear activation function.

After acquiring refined node features h_i', the global context feature of the packing scene is computed using global average pooling:

$$\bar{h} = \frac{1}{N_B}\sum_{i=1}^{N_B} h_i',$$

where N_B is the total number of nodes in the current tree topology. The obtained global feature $\bar{h}$ serves as a conditional prior to guide the diffusion model.

3.4. A Diffusion Model

To overcome limitations in action exploration of traditional deep reinforcement learning, a diffusion model is introduced to generate diverse, stable packing actions, thereby enhancing candidate position quality and improving reinforcement learning training efficiency. Traditional deep reinforcement learning methods that rely on heuristic rules exhibit limited exploration and a single candidate approach. The diffusion model uses multi-step denoising to fit complex action distributions, producing high-quality actions from Gaussian noise, enhancing diversity and packing state relevance for the online 3D workpiece packing. The input combines SW-GAN's global feature h_i' with a time embedding, mapping discrete time steps $t \in [1, T]$ to an D_t dimensional vector via an embedding matrix $E \in \mathbb{R}^{T\times D_t}$, initialized from a normal distribution and optimized during training. A multilayer fully connected network processes the input, mapping it with ReLU activation into N_a candidate actions with coordinates and dimensions.

Based on the denoising diffusion probabilistic model framework, the model recovers actions via multi-step denoising. The forward process adds Gaussian noise to true action x_0, generating noisy data x_t, with its distribution defined as

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I),$$

where $\alpha_t = 1-\beta_t, \bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$, β_t denotes the variable linearly increasing from 0.0001 to 0.02, I is the identity matrix. The process of reverse denoising involves predicting the noise $\epsilon_\theta(x_t, t)$ to recover x_0. The mean is calculated as follows

$$\mu_\theta(x_t, t, \bar{h}) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t, \bar{h})\right),$$

where $\mu_\theta(x_t, t, \bar{h})$ represents the predicted mean after denoising given the noisy data x_t at time step t, $\epsilon_\theta(x_t, t, \bar{h})$ denotes the model's prediction of the noise. The training objective is to minimize the mean squared error, which is defined as

$$L = \mathbb{E}_{t,\epsilon}\left[\left\|\epsilon - \epsilon_\theta(x_t, t, \bar{h})\right\|^2\right],$$

where ϵ represents the ground-truth noise.

The weights of the multi-layer perceptron and the time embedding matrix E are updated through end-to-end backpropagation. After reverse denoising, a set of candidate actions is generated, ensuring a high correlation between the actions and the features of SW-GAN, which is suitable for online 3D workpiece packing requirements. To further enhance the model's generalization ability, experiences stored in the experience replay buffer are shuffled before training, breaking the temporal correlations and improving the model's robustness.

4. Experiments

4.1. Comparative Experiments

A container size of $100 \times 100 \times 100$ mm^3 grid units is established to simulate an online heterogeneous 3D workpiece packing scenario. To evaluate the proposed method, representative baseline methods from publicly available implementations are selected. The random algorithm randomly selects feasible loading positions within the container. The remaining methods are grouped into two categories: heuristic methods (OnlineBPH [1], LSAH [16], MACS [5]) and reinforcement learning-based methods (Zhao et al. [11], PCT [12], Xiong et al. [13]).

The quantitative comparison results in Table 1 demonstrate that the proposed method significantly outperforms all baseline algorithms in both key metrics: average space utilization (Uti) and average number of packed items (Num). Uti is defined as the mean ratio of total packed volume to container volume over $N = 30$ runs:

$$Uti = \frac{1}{NV_C} \sum_{k=1}^{N} \left(\sum_{i=1}^{\mathrm{Num}_k} v_{k,i} \right), \tag{1}$$

where V_C is the container volume, Num_k is the number of items in run k, and $v_{k,i}$ is the volume of item i. The average number of packed items Num is computed as

$$Num = \frac{1}{N} \sum_{k=1}^{N} Num_k. \tag{2}$$

Higher values of both metrics indicate better space allocation and filling performance. The results confirm that the proposed method achieves superior packing efficiency and container space utilization. This advantage stems from the capacity of learning-based methods to extract complex packing patterns from extensive training data, which enables adaptation to diverse packing scenarios. In contrast, heuristic methods rely on fixed rules, limiting their generalization and leading to inferior performance in complex or non-standard packing demands.

Table 1. Comparison of different methods.

Methods		Experiment 1		Experiment 2	
		Uti	**Num**	**Uti**	**Num**
Random		35.7%	14.9	38.1%	15.7
Heuristic	OnlineBPH	50.4%	20.1	60.2%	24.4
	LSAH	51.5%	20.8	64.5%	25.6
	MACS	57.7%	22.6	50.8%	20.1
Learning-based	Zhao et al.	49.1%	19.6	56.5%	22.6
	PCT	70.8%	27.5	77.8%	29.6
	Xiong et al.	72.6%	28.3	75.6%	28.9
	Ours	75.3%	28.8	78.1%	30.1

Nevertheless, learning-based approaches remain influenced by heuristic design and are subject to limitations imposed by rule constraints. Experimental results indicate that the high-quality features extracted by SW-GAN, combined with the diverse candidate actions generated by the diffusion model, ensure rational stacking and higher space utilization. As shown in Figure 4, these examples show that Traditional methods easily produce excessive gaps and unstable stacking structures, leading to low space utilization.

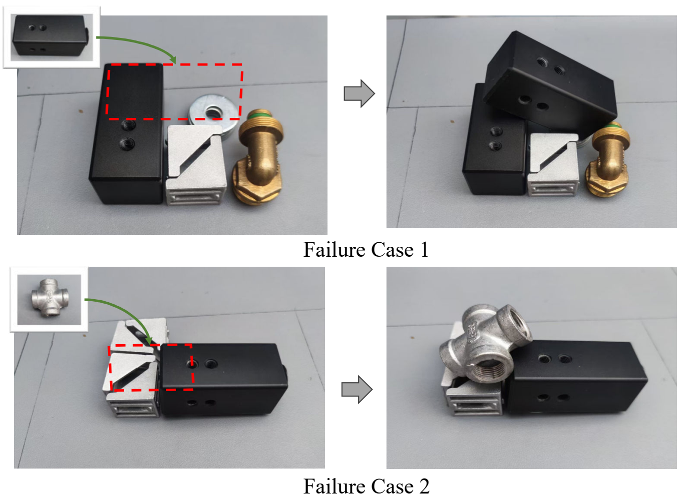

Figure 4. Packing failure case illustration.

4.2. Ablation Experiments

To evaluate the contributions of SW-GAN and the diffusion model in 3D workpiece packing tasks, ablation experiments are conducted under two settings.

(1) The diffusion model without SW-GAN

When the SW-GAN feature extractor is removed, the diffusion model relies solely on raw environmental signals as input. As shown in Figure 5, its space utilization curve remains consistently lower than that of the full model, with a notable gap in the steady state. The convergence is also visibly slower: the curve rises more gradually and requires considerably more training steps to approach saturation.

This contrast confirms the importance of SW-GAN in capturing structured spatial-geometric dependencies among stacked workpieces. Without the refined spatial features from SW-GAN, the diffusion model cannot develop effective environmental awareness. This limitation restricts the diversity of the generated candidate poses and, in turn, reduces the final packing efficiency.

(2) Traditional deep reinforcement learning without SW-GAN and the diffusion model

Ablation experiments are conducted under two settings to evaluate the contributions of SW-GAN and the diffusion model. When both modules are discarded, the framework reduces to a conventional heuristic-driven deep reinforcement learning method. Its training curve shows the worst performance among all variants: the steady-state space utilization is significantly lower, and the convergence rate drops drastically.

This result confirms the essential exploration capability provided by the diffusion module. Without the diverse candidate actions generated by diffusion, the policy is restricted to a limited set of placements dictated by rigid heuristics. Such constrained sampling leads to insufficient exploration of feasible packing configurations and ultimately inferior optimization performance.

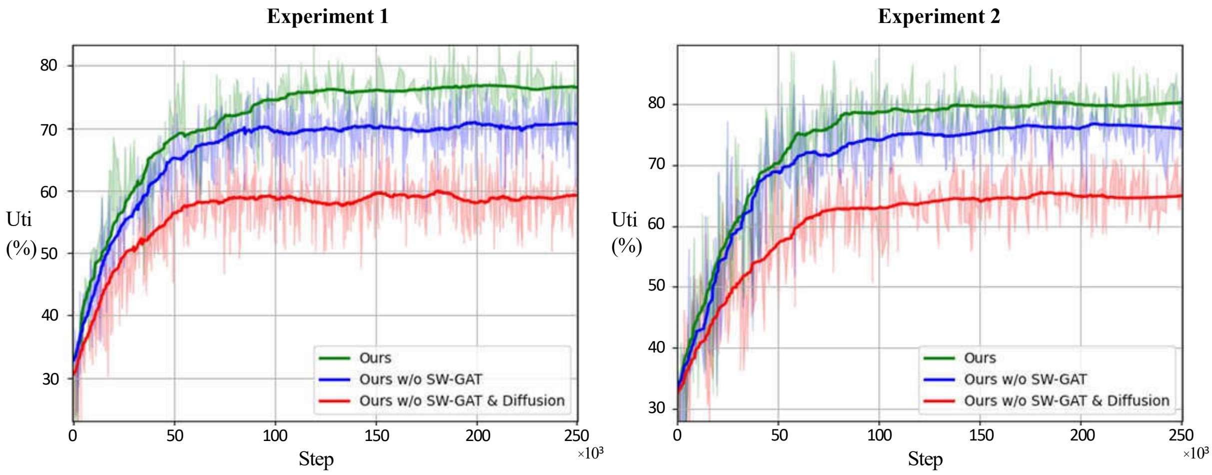

Figure 5. Training curves of average space utilization rate across different model variants.

Table 2 reports the average space utilization on the two datasets CUT-2 and RS [11]. It can be seen that under both experimental settings, our method achieves the highest average space utilization on every dataset, reaching 75.3% and 76.6% on CUT-2, demonstrating strong transferability.

Table 2. Comparative ablation study on average space utilization.

Method	Experiment 1		Experiment 2	
	RS	CUT-2	RS	CUT-2
w/o SW-GAN & Diffusion	48.7%	56.6%	55.4%	63.7%
w/o SW-GAN	62.1%	70.3%	67.8%	75.3%
Ours	65.3%	75.3%	70.7%	76.6%

4.3. A Real-World Packing System

To transfer the simulation strategy to real-world settings, an experimental platform is built. A UR5 robotic arm, a BY-E140 two-finger gripper, and an Intel RealSense D435i camera are used, as is shown in Figure 6. The system is built to verify the practical performance of the proposed packing strategy. Object grasping is performed in complex scenes. The camera, mounted on the arm, measures object sizes in real time. Placement decisions are generated by the proposed model and converted to reachable positions for the arm. A container of 200 × 200 × 200 mm^3 is defined in front of the arm. After completing the grasp in a complex environment, the robotic arm determines the placement position of the object through the obtained strategy, thereby achieving efficient stacking. The experimental results confirm that the task can be completed successfully.

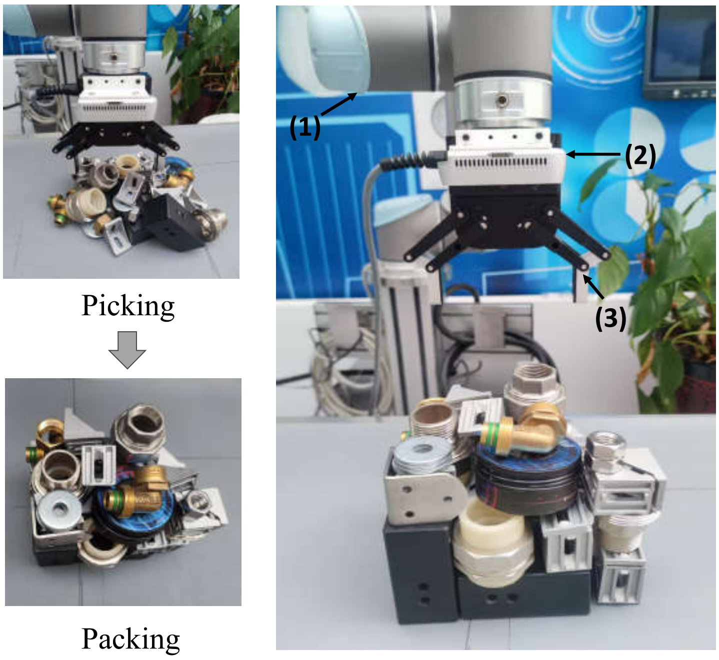

Figure 6. Through deep reinforcement learning training, robot can pick objects from a cluttered pile and complete stacking tasks. The hardware setup includes: (1) a UR5e robotic arm; (2) an Intel RealSense D435i camera and (3) a BY-E140 gripper.

5. Conclusions

A framework integrating SW-GAN with a diffusion model to enhance online 3D workpiece packing is proposed. SW-GAN effectively captures 3D spatial dependencies, while the diffusion model generates diverse and stable placement actions. Experiments demonstrate that our method significantly outperforms baselines in average space utilization and achieves high success rates in zero-shot real-world deployments. Despite these advantages, the multi-step denoising of the diffusion model introduces computational overhead, limiting its application in ultra-high-speed scenarios. Furthermore, the current framework focuses primarily on cuboid objects. Future work will aim to accelerate inference via sampling optimization and extend the method to irregular shapes.

Author Contributions: Z.L.: conceptualization, methodology; F.W.: investigation, data curation, visualization; Q.P.: formal analysis, software; J.L.: supervision, writing—reviewing and editing, project administration.

Funding: This research was supported in part by the National Natural Science Foundation of China under Grant 62533005 and Grant 62573088, and in part by the Ministry of Education of the People's Republic of China under Grant 091B02032305.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Use of AI and AI-Assisted Technologies: No AI tools were utilized for this paper.

References

1. Dósa, G.; Sgall, J. Optimal Analysis of Best Fit Bin Packing. In *Automata, Languages, and Programming, Proceedings of the 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, 8–11 July 2014*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 429–441. https://doi.org/10.1007/978-3-662-43948-7_36.
2. Parreño, F.; Alvarez-Valdés, R.; Tamarit, J.M.; et al. A Maximal-Space Algorithm for the Container Loading Problem. *INFORMS J. Comput.* **2008**, *20*, 412–422. https://doi.org/10.1287/ijoc.1070.0254.

3. Agarwal, M.; Biswas, S.; Sarkar, C.; et al. Jampacker: An Efficient and Reliable Robotic Bin Packing System for Cuboid Objects. *IEEE Robot. Autom. Lett.* **2021**, *6*, 319–326. https://doi.org/10.1109/LRA.2020.3043168.
4. Ha, C.T.; Nguyen, T.T.; Bui, L.T.; et al. An Online Packing Heuristic for the Three-Dimensional Container Loading Problem in Dynamic Environments and the Physical Internet. In *Applications of Evolutionary Computation, Proceedings of the 20th European Conference, EvoApplications 2017, Amsterdam, The Netherlands, 19–21 April 2017*; Springer: Cham, Switzerland, 2017; pp. 140–155. https://doi.org/10.1007/978-3-319-55792-2_10.
5. Hu, R.Z.; Xu, J.Z.; Chen, B.; et al. TAP-Net: Transport-and-Pack Using Reinforcement Learning. *ACM Trans. Graph.* **2020**, *39*, 1–15. https://doi.org/10.1145/3414685.3417796.
6. Nguyen, T.H.; Nguyen, X.T. Space Splitting and Merging Technique for Online 3D Bin Packing. *Mathematics* **2023**, *11*, 1912. https://doi.org/10.3390/math11081912.
7. Chen, L.; Zhang, S.; Ding, M.M.; et al. Solving a Large-Scale 3D Packing Problem with Robot Constraints. *Comput. Oper. Res.* **2026**, *188*, 107373. https://doi.org/10.1016/j.cor.2025.107373.
8. Rao, Z.H.; Wu, Y.C.; Yang, Z.F.; et al. Visual Navigation with Multiple Goals Based on Deep Reinforcement Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 5445–5455. https://doi.org/10.1109/TNNLS.2021.3057424.
9. Liu, Y.; Liu, S.; Chen, B.; et al. Fusion-Perception-to-Action Transformer: Enhancing Robotic Manipulation with 3D Visual Fusion Attention and Proprioception. *IEEE Trans. Robot.* **2025**, *41*, 1553–1567. https://doi.org/10.1109/TRO.2025.3539193.
10. Que, Q.Q.; Yang, F.; Zhang, D.F. Solving 3D Packing Problem Using Transformer Network and Reinforcement Learning. *Expert Syst. Appl.* **2023**, *214*, 119153. https://doi.org/10.1016/j.eswa.2022.119153.
11. Zhao, H.; She, Q.J.; Zhu, C.Y.; et al. Online 3D Bin Packing with Constrained Deep Reinforcement Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 2–9 February 2021; Volume 35, pp. 741–749. https://doi.org/10.1609/aaai.v35i1.16155.
12. Zhao, H.; Yu, Y.; Xu, K. Learning Efficient Online 3D Bin Packing on Packing Configuration Trees. In Proceedings of the International Conference on Learning Representations, Online, 25–29 April 2022; pp. 1–18.
13. Xiong, H.; Ding, K.; Ding, W.; et al. Towards Reliable Robot Packing System Based on Deep Reinforcement Learning. *Adv. Eng. Inform.* **2023**, *57*, 102028. https://doi.org/10.1016/j.aei.2023.102028.
14. Wang, B.Y.; Yu, W.; Zhang, X.D.; et al. Robotic Bin Packing via Hierarchical Reinforcement Learning. *IEEE Trans. Autom. Sci. Eng.* **2026**, *23*, 8634–8646. https://doi.org/10.1109/TASE.2026.3678655.
15. Liu, H.T.; Lian, J.; Chen, Y.F.; et al. A Dual-Functional Push-Grasping Strategy for Goal-Agnostic and Goal-Oriented Tasks Based on Deep Reinforcement Learning. In Proceedings of the 2025 Joint International Conference on Automation-Intelligence-Safety (ICAIS) & International Symposium on Autonomous Systems (ISAS), Xi'an, China, 23–25 May 2025; pp. 1–6. https://doi.org/10.1109/ICAISISAS64483.2025.11051742.
16. Hu, H.Y.; Zhang, X.D.; Yan, X.W.; et al. Solving a New 3D Bin Packing Problem with Deep Reinforcement Learning Method. *arXiv* **2017**, arXiv:1708.05930.