# Stability Ensured Deep Reinforcement Learning for Online Bin Packing

Ziyan Gao and Nak Young Chong

*Abstract*— **The Online Bin Packing Problem (OBPP) aims to determine the optimal loading position for each incoming item to maximize bin utilization, a critical challenge in various industrial applications. While many studies have focused on learning-based policies and heuristic approaches to enhance packing efficiency, stability constraints have largely been overlooked. In this work, we propose a computationally efficient method to validate stable loading positions for incoming items without requiring exact knowledge of their physical properties, such as mass. Our approach leverages the concept of Load-Bearable Convex Polygons (LBCPs), which provide substantial support forces to ensure structural stability. We further integrate our static stability validation framework into a state-of-the-art deep reinforcement learning (DRL) model, guiding it to learn physics-feasible packing strategies. Experimental results demonstrate that our stability-aware DRL model achieves comparable packing efficiency while ensuring robust bin stability, offering a significant advancement in practical OBPP applications.**

## I. Introduction

The Bin Packing Problem (BPP) is a classic optimization problem in computer science and operations research. It involves packing a set of items of varying sizes into a finite number of bins or containers of fixed capacity, while trying to minimize the number of bins used. It assumes that all items are known beforehand before packing begins. On the other hand, the Online Bin Packing Problem (OBPP) presents unique challenges compared with conventional BPP as items arrive sequentially, requiring immediate decisions without knowledge of future items. This complexity makes traditional heuristic-based approaches insufficient for optimal packing [1]. As a result, reinforcement learning (RL) approaches have gained traction as a promising alternative for optimizing bin utilization.

Recent research [2]–[6] has demonstrated that reinforcement learning (RL) techniques can outperform heuristic methods by learning the packing policies in a trial-and-error fashion. These models iteratively improve their decision-making process, enabling them to adapt to diverse item distributions and packing constraints. For example, Xiong et al. [2] introduced GOPT, a transformer-based RL framework that significantly improves bin utilization compared to traditional heuristic-based approaches.

While maximizing bin utilization is crucial, another critical aspect of OBPP is ensuring bin stability during and after each packing operation. In industrial applications, unstable packing configurations can lead to safety risks, inefficient space usage, and logistical challenges. Despite the importance of stability, a robust yet computationally efficient solution remains an open problem in 3D online bin packing. Zhao *et al* [3] conducted a representative study where they used a set of rules designed to validate the geometric relationship between a newly packed item and its supporting items, to assess if the newly packed item leads to bin collapse. However, none of the rules considered in [3] guarantees the stability of the bin. Recent works have started exploring stability constraints beyond simple geometric fitting. Some approaches leverage physics-based modeling, where physical constraints such as friction, weight distribution, and structural load-bearing properties are explicitly formulated to assess stability [7]. While these methods show promise, they often suffer from high computational costs, limiting their practicality for large-scale real-time applications.

In this paper, we investigate the challenges inherent to bin stability and propose a novel learning-based solution that integrates deep reinforcement learning with explicit stability validation mechanisms. Our approach seeks to bridge the gap between efficient bin utilization and stable packing configurations, improving the bin utilization while ensuring the structural integrity of the stacked items.

To achieve this, we introduce a stability validation method based on the concept of Loading Bearable Convex Polygons (LBCPs). LBCPs are defined as convex polygons parallel to the ground level, located at different height levels inside the bin. Under the rigid body assumption, any point within an LBCP can bear an infinite gravitational force, ensuring that the packed items remain stable even as additional items are introduced. By leveraging LBCPs, our approach provides a lightweight yet effective method for assessing stability while maintaining computational efficiency.

The assumptions made in this works are listed bellow.

- All items are cuboidal objects.
- All items are rigid, which means no deformation occurs due to the external force.
- No lateral forces exist between items, ensuring that stability is solely determined by vertical load-bearing properties.

## II. Related Work

Early research on bin stability validation primarily focused on mechanical equilibrium and vertical support constraints. One of the foundational studies by Ramos et al. [8] introduced a set of static mechanical equilibrium conditions applicable to three fundamental scenarios: when an item rests directly on the level ground, when it is fully supported by another item beneath it, and when it is partially supported by multiple underlying items. These early methods provided a theoretical basis for evaluating structural integrity, ensuring that every

This work was supported by JSPS KAKENHI Grant Number JP23K03756.

All authors are with the School of Information Science, Japan Advanced Institute of Science and Technology, Nomi, Ishikawa 923-1292, Japan {ziyan-g, nakyoung}@jaist.ac.jp

placed object had sufficient support to prevent instability. Expanding on this work, Gzara et al. [9] incorporated vertical support constraints, emphasizing the necessity for items to be adequately supported at their corners or across a significant portion of their base. Their study also introduced a graph-based load tracing approach, which enabled a more systematic weight distribution across pallets, improving real-world stability assessments.

As the complexity of bin packing problems increased, researchers developed more advanced mathematical models to enforce stability constraints in 3D bin packing. Zhu et al. [10] proposed a stack-based integer programming approach that enforced strict support constraints to prevent tipping and improve stability. This method effectively enhanced the structural integrity of packed items by ensuring each item was placed in a stable equilibrium. Similarly, Liu et al. [11] introduced force-balancing equations for structural stability in block stacking, accurately identifying weak points in stacked assemblies, such as Lego constructions. While these mathematical approaches significantly improved stability analysis, they required high computational costs, limiting their scalability in real-time applications.

With the rise of machine learning and reinforcement learning (RL) techniques, researchers began to explore how to learn optimal stacking strategies while maintaining stability constraints. Wu et al. [12] proposed an iterative action masking learning approach for RL-driven palletization, allowing robotic systems to optimize stacking decisions in real-time. However, despite its effectiveness, this approach exhibited difficulties in generalizing to out-of-distribution scenarios, making it less reliable for diverse packing conditions. Building on this, Zhang et al. [13] introduced a reinforcement learning framework with online masking inference, training models to dynamically adjust stacking configurations based on gravity and rigidity constraints. By integrating physical constraints directly into the RL training process, this method improved stacking adaptability compared to previous learning-based solutions.

Beyond learning-based approaches, recent research has integrated real-time physics simulations to provide more accurate and dynamic stability assessments. Mazur et al. [7] introduced physics-based simulations for cargo loading stability, demonstrating how traditional static models often overestimate or underestimate actual stability when compared to dynamic evaluations. Their findings highlighted a crucial limitation in previous approaches: static stability checks alone may not be sufficient to ensure the long-term integrity of a packed bin under real-world conditions.

Structured data-driven approaches have also gained traction in stability validation research. Zhao et al. [14] proposed a constrained deep reinforcement learning framework that leveraged an adaptive stacking tree to recursively update mass distribution and validate the stability of each packed item. While this method achieved high performance in simulations, it relied on the assumption that the mass of each item is known and evenly distributed, which may not hold in real-world scenarios. Zhou *et al* [15] proposed the concept of "empty map", sharing the same dimension as the heightmap, to record if there exists the wasted spaces for each pixel positions. The pixel positions of no wasted spaces are expected to offer the support forces to stabilize the upcoming item. The authors experimentally evaluated their stable checking method in simulation and found it outperforms the baseline method [8]. In comparison to the approach introduced in this study, this method adopts a conservative stance, which limits learning efficiency.

## III. Structural Stability Based Packing

This section first outlines the problem state. It then introduces load-bearable convex polygons (LBCPs) for stability validation. Then, it presents a stability-ensured deep reinforcement learning (DRL) framework that optimizes packing policy while ensuring safe loading.

### A. Problem Statement

Imagine a container with specified dimensions $W \times D \times H$ and a set of items, each characterized by dimensions $\{\mathcal{O}_1, \mathcal{O}_2, ..., \mathcal{O}_i, ..., \mathcal{O}_m\}, \mathcal{O}_i = (w_i, d_i, h_i), w_i < W, d_i < D, h_i < H$. These items are cuboidal in shape with varying sizes, and although their weights are unknown, the maximum displacement of each item's center of gravity (CoG) from its geometric center is known. An agent observes only one item and is able to pack single item each time. Following the setting described in [2], items can be rotated around the $z$ axis by $90^\circ$. The loading position of the $i$th item is noted as $\mathcal{L}_i = (x_i, y_i, z_i)$, and $\mathcal{I}_i = (\mathcal{O}_i, \mathcal{L}_i)$ is used to describe the item's state inside the bin.

The primary objective is to determine a sequence of loading positions $\{\mathcal{L}_1, \mathcal{L}_2, ..., \mathcal{L}_i, ..., \mathcal{L}_m\}$ that maximizes the utilization of the container while ensuring the stability of the assembled load to prevent collapse. This problem entails sequential decision-making under uncertainty, balancing optimal space utilization with the rigorous constraints imposed by load stability.

### B. Structural Stability

*1) Static Equilibrium:* Given the item $\mathcal{O}_i$ with loading position $\mathcal{L}_i$, the item geometric center *w.r.t* the bin coordinate system can be calculated and is represented by $\mathbf{g}_i$. We assume that the center of mass $\mathbf{c}_i$ of the item is not overlaid with the $\mathbf{g}_i$, and the deviation from geometric center is proportional to the item dimension. Mathematically, if the max deviation ratio is $\delta_{CoG}$, then, the range of possible locations of the center of mass is represented by Eq.

$$\mathcal{C}_i = \{\mathbf{g}_i + \begin{bmatrix} \delta_w w_i \\ \delta_d d_i \\ \delta_h h_i \end{bmatrix}, |\delta_w|, |\delta_d|, |\delta_h| \leq \delta_{CoG}\} \tag{1}$$

Stemming from Newton's laws of motion and classical static equilibrium, in the case that all support forces are parallel to each other and perpendicular to the horizontal plane, item $\mathcal{I}_i$ is considered statically stable if its $\mathbf{c}_i$ is located within the support polygon [8]. This conclusion can be easily extended to our assumption such that item $\mathcal{O}_i$ is considered statically stable if the bounds of CoG is inside the support polygon.

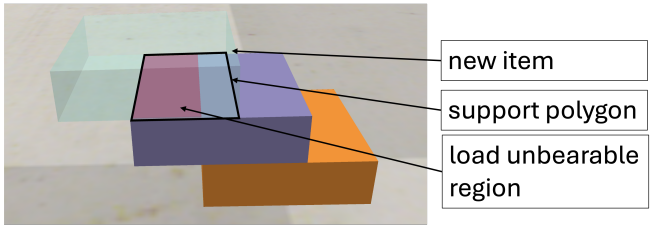


Fig. 1. Illustration of instable case: the item on top may cause the stack collapsing due to the lack of substantial support beneath the region marked as load unbearable region.

Please note that the deviation of CoG of the item along $z$-axis does not affect the stability.

To calculate the support polygon for $\mathcal{I}_{new}$, we need to find all items that support item $\mathcal{I}_{new}$, we use $\mathcal{B}_t = \{\mathcal{I}_1, \mathcal{I}_2, ..., \mathcal{I}_i, ..., \mathcal{I}_m, \}$ to represent the current bin state. Then, all regions of support for the new item $\mathcal{I}_{new}$ is represented by $\{\mathcal{I}_{new} \cap \mathcal{I}_j | z_j + h_j = z_{new}, \mathcal{I}_j \in \mathcal{B}_t\}$. The support polygon is the region inside the smallest convex hull encompassing all regions of support, which can be represented by $\mathbf{CH}(\{\mathcal{I}_{new} \cap \mathcal{I}_j | z_j + h_j = z_{new}, \mathcal{I}_j \in \mathcal{B}_t\}\})$.

However, the implicit assumption of this method is that any point inside the support polygon can resist any magnitude of gravitational forces. This assumption can be easily violated in multi-layer (more than two) stacking scenario. Fig. 1 illustrates the case that the support polygon may fail to bear the gravitational forces by the new item since there is no structural support beneath the item in the middle. Thus, the geometric intersection alone cannot guarantee true load-bearing capacity, and thus the item can topple despite appearing stable in a purely geometric analysis.

*2) Load Bearable Convex Polygon:* In this work, we introduce the concept of load bearable convex polygons (LBCP)s. We use $\mathcal{P}_t$ to refer the set of LBCP in the current bin state $\mathcal{B}_t$. Basically, LBCPs is a set of convex polygons that are parallel to the horizontal plane and located at different height level inside the bin. The height level of the LBCP is at the top surface of each item. We use $(\mathbf{P}_i^{\triangle}, h_i^c)$ to represent a specific LBCP where $h_i^c = z_i + h_i$. If the ground of the bin is homogeneous, then, the ground can be considered as one specific LBCP and located at the bottom of the bin. The LBCP at the level ground is represented by $(\mathbf{P}_0^{\triangle}, 0)$. Excluding the LBCP at the level ground, the number of LBCPs is equal to the number of packed items, in other words, new LBCP will be generated after packing the new item.

An essential characteristic of LBCPs is their ability to support any gravitational forces at any point within these polygons. Apart from the LBCP at ground level $(\mathbf{P}_0^{\triangle}, 0)$, other LBCPs are formed based on one or more existing LBCPs to maintain this property.

*Lemma 3.1:* If a cuboidal item is placed stably on level ground, then its entire top surface is a load-bearable polygon.

*Proof:* Firstly, an item is considered statically stable if its (CoG) lies within the support polygon. For an item resting on the level ground, the support polygon is simply the item's base in full contact with the ground plane. Since the base fully contacts the ground, any load placed on the item's top surface is transmitted vertically downward through the item to the ground. The ground, in turn, provides equal and opposite reaction forces that prevent tipping or toppling. In other words, placing a load on any point of the top surface will not shift the item's CoG outside its support polygon. Thus, the top face can fully bear that load without overturning. Hence, the item's top face is indeed a valid load-bearable polygon. ■

In this work, the item stability validation method is also based on the inclusiveness relationship between the CoG of the new item and its support polygon, however, the substantial difference with [8] is that the support polygon is calculated based on LBCPs, which can be represented by Eq. 2

$$\mathbf{P}_{new}^{\triangle} = \mathbf{CH}(\{\mathcal{I}_{new} \cap \mathbf{P}_i^{\triangle} | z_{new} = h_i^c, \mathbf{P}_i^{\triangle} \in \mathcal{P}_t\}) \quad (2)$$

where, $\mathcal{P}_t = \{\mathbf{P}_0^{\triangle}, \mathbf{P}_1^{\triangle}, \mathbf{P}_2^{\triangle}, ..., \mathbf{P}_i^{\triangle}, ..., \mathbf{P}_m^{\triangle}\}$.

*Theorem 3.2:* The support polygon calculated based on the load bearable convex polygons is also a load bearable convex polygon.

*Proof:* Since the support polygon is calculated based on all intersected regions between $\mathcal{I}_{new}$ and LBCPs. Based on the definition of LBCPs, any point inside the intersected regions can bear any magnitude of gravitational forces. Meanwhile, since the resultant force of two support forces is located in the line segment between these two lines of forces and parallel to these forces, the net force is always located inside the smallest convex hull of all intersected regions. ■

Therefore, given the object dimension $\mathcal{O}_{new}$, loading position $\mathcal{L}_{new}$ and maximal deviation of the CoG of the item, the item is considered as stable if the support polygon calculated based on $\mathcal{P}_t$ contains all deviations of the CoG.

If the new item is packed, a new LBCP will be created, sharing the same convex hull as the support polygon, and being located on the top surface of the new item. As illustrated in Fig. 2, the quantity of LBCPs grows with an increase in the number of items packed.

### C. Stability Ensured Deep Reinforcement Learning

In this work, we utilize GOPT [2], a state-of-the-art model known for its superior space utilization, to address the stability-ensured online bin packing problem. Specifically, GOPT is an actor-critic deep reinforcement learning (DRL) model that incorporates two key components: the packing heuristics (refers to as the packing generator in [2]), which generates a fixed-length set of Empty Maximal Spaces (EMS) [16] to define feasible placements, and the Packing Transformer, which serves as the backbone of the DRL framework by capturing spatial relationships among EMSs and item.

The stability of each placement is achieved in this way: firstly, the set of placement candidates is extracted based on the packing heuristics, all candidates will pass to the static stability validation module to generate the stability mask. Meanwhile, all candidates along with the dimension of the new item will be forward into GOPT. Then, the distribution of actions for packing is derived through the softmax function,

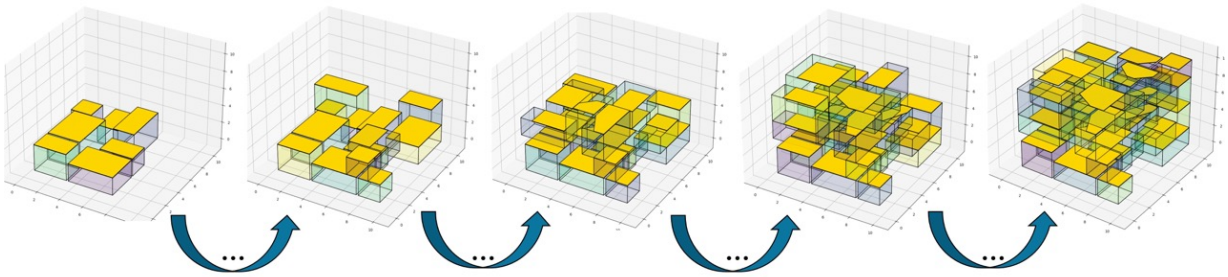

Fig. 2. Illustration of Loading Bearable Convex Polygons $\mathcal{P}_t$ (The ground is also a LBCP but hidden for clarity purpose). With the increase of the number of the packed items, $\mathcal{P}_t$ increases incrementally.

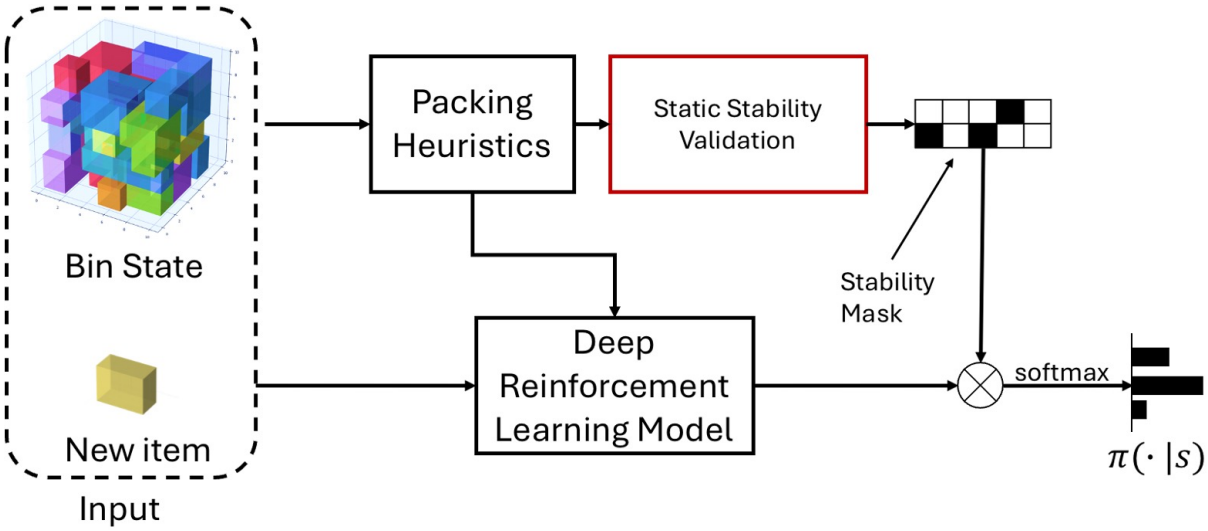


Fig. 3. Proposed framework for stability ensured online bin packing. The guarantee on bin stability is through the structural stability validation module that filter instable placements.

which relies on the element-wise product of the GOPT output and the stability mask. Finally, new LBCP will be appended to $\mathcal{P}_{t+1}$ after finishing the packing operation.

## IV. Experiment

We first conduct the experiment validating the efficiency of the proposed item stability validation method. Then performance of the trained DRL model is demonstrated on the RS Dataset [3].

### A. Structural Stability

In this investigation, our goal is to assess the efficiency of the proposed approach. The data set is derived from the RS dataset [3] along with a modified version referred to as RS dataset (same height). In the RS dataset, each sequence of items is selected without replacement from a collection of 64 items with varying sizes. In contrast, in the RS dataset (same height), the sequences are also sampled without replacement, but from a subset of the RS dataset where all items have an identical height, specifically $0.12$ meters.

The baseline for validating the stability is based on adaptive stacking tree [3] since it attains the state of the art accuracy while has the computational complexity $O(NlogN)$ where N is the number of the already packed object. Specifically, the Adaptive Stacking Tree is a data structure designed to efficiently update mass distribution. The tree is updated in a top-down fashion, meaning stability updates propagate downward in an efficient manner. Conversely, our approach possesses a computational complexity of $O(N)$. To further enhance the computational efficiency, we align the LBCPs with the height map recorded by the ceiling camera. Then, the support regions are easily determined by: 1. Retrieving all points within the window defined by the loading position and item dimensions from the height map. 2. Identifying the points within that window nearest to the ceiling camera. Consequently, the computational complexity is decreased to $O(k)$, where $k$ is a fixed constant.

We created $1000$ sequences consisting of $500$ items each. A random packing strategy is used, which chooses a stable loading position from a predefined set of possible positions. In a sequence of items, we initially verify the available loading positions for the next item. If there are no stable loading positions, the item is disregarded, and the next item is then checked for stable loading positions. If a stable position is found, a random policy will select one of these positions for packing the item. This routine continues until the entire

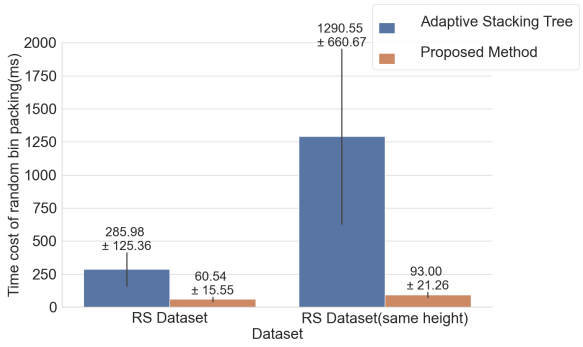


Fig. 4. Time cost for completing the packing of the whole sequence.

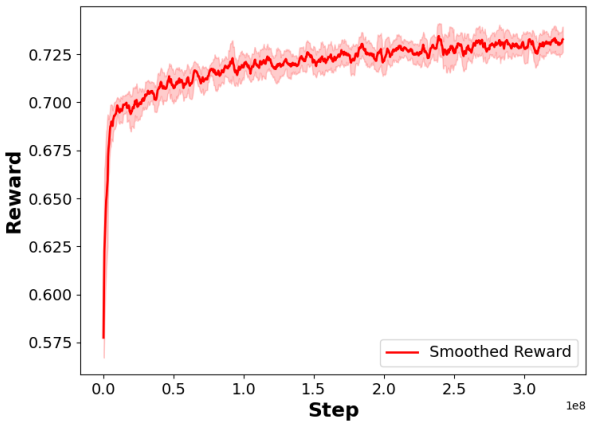


Fig. 5. Reward on the test set that consists of 2000 random sequences of items.

sequence has been processed. The time required to pack the entire sequence is illustrated in Fig. 4.

Overall, the proposed method is at least 4 times faster than the baseline method. The efficiencies of the baseline method for dealing with different dataset are significantly different. The time cost becomes even higher in RS datsaset(same height), while our method is not affected by the variation of the dataset.

### B. DRL Model Performance

The training of the GOPT model with structural stability validation is done by using Tianshou framework [17]. We adopt the same hyperparameter setting as [2], and the dataset for training and testing are based on RS dataset. We generated 2000 sequences of items for testing. Fig. 5 reports the model performance on the test set in the phase of training. The total number of training steps is about 30 million, and finally it achieves the bin utilization 73.6% with standard deviation 7.1%, which is slightly lower than the performance of the original GOPT (76.1%). However, thanks to the stability validation module, every placement guarantees the item stability.

## V. Conclusion

In this paper, we addressed the fundamental challenge of ensuring stability in OBPP by integrating load-bearable convex polygons (LBCPs) into a deep reinforcement learning framework. Unlike traditional heuristic methods, which prioritize space utilization but overlook stability, our approach effectively balances packing efficiency and physical feasibility. By leveraging LBCPs, we provide a lightweight, scalable, and computationally efficient stability validation technique that ensures that each placement decision maintains structural integrity.

Experimental results demonstrate that our method achieves comparative bin utilization while satisfying the stability constraint. The proposed stability-aware DRL model offers a practical and generalizable solution for real-world warehousing, logistics, and robotic packing applications.

Future work may explore incorporating additional physical constraints, such as friction and dynamic stability, and extending the model to non-cuboidal objects to further enhance its applicability in industrial automation.

## References

[1] H. I. Christensen, A. Khan, S. Pokutta, and P. Tetali, "Multidimensional bin packing and other related problems: A survey," *Computer Science Review*, 2016.

[2] H. Xiong, C. Guo, J. Peng, K. Ding, W. Chen, X. Qiu, L. Bai, and J. Xu, "Gopt: Generalizable online 3d bin packing via transformer-based deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 10 335–10 342, 2024.

[3] H. Zhao, Q. She, C. Zhu, Y. Yang, and K. Xu, "Online 3d bin packing with constrained deep reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 741–749.

[4] J. Xu, M. Gong, H. Zhang, H. Huang, and R. Hu, "Neural packing: from visual sensing to reinforcement learning," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 6, pp. 1–11, 2023.

[5] S. Yang, S. Song, S. Chu, R. Song, J. Cheng, Y. Li, and W. Zhang, "Heuristics integrated deep reinforcement learning for online 3d bin packing," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 1, pp. 939–950, 2023.

[6] R. Hu, J. Xu, B. Chen, M. Gong, H. Zhang, and H. Huang, "Tap-net: transport-and-pack using reinforcement learning," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 6, pp. 1–15, 2020.

[7] P. G. Mazur, J. W. Melsbach, and D. Schoder, "Physical question, virtual answer: Optimized real-time physical simulations and physics-informed learning approaches for cargo loading stability," *Operations Research Perspectives*, vol. 14, p. 100329, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214716025000053

[8] A. G. Ramos, J. F. Oliveira, and M. P. Lopes, "A physical packing sequence algorithm for the container loading problem with static mechanical equilibrium conditions," *International Transactions in Operational Research*, vol. 23, no. 1-2, pp. 215–238, 2016.

[9] F. Gzara, S. Elhedhli, and B. C. Yildiz, "The pallet loading problem: Three-dimensional bin packing with practical constraints," *European Journal of Operational Research*, vol. 287, no. 3, pp. 1062–1074, 2020.

[10] W. Zhu, S. Chen, M. Dai, and J. Tao, "Solving a 3d bin packing problem with stacking constraints," *Computers Industrial Engineering*, vol. 188, p. 109814, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835223008380

[11] R. Liu, K. Deng, Z. Wang, and C. Liu, "Stablelego: Stability analysis of block stacking assembly," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 9383–9390, 2024.

[12] Z. Wu, Y. Li, W. Zhan, C. Liu, Y.-H. Liu, and M. Tomizuka, "Efficient reinforcement learning of task planners for robotic palletization through iterative action masking learning," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 9303–9310, 2024.

[13] T. Zhang, Z. Wu, Y. Chen, Y. Wang, B. Liang, S. Moura, M. Tomizuka, M. Ding, and W. Zhan, "Physics-aware robotic palletization with online masking inference," *arXiv preprint arXiv:2502.13443*, 2025.

[14] H. Zhao, C. Zhu, X. Xu, H. Huang, and K. Xu, "Learning practically feasible policies for online 3d bin packing," *Science China Information Sciences*, vol. 65, no. 1, p. 112105, 2022.

[15] P. Zhou, Z. Gao, C. Li, and N. Y. Chong, "An efficient deep reinforcement learning model for online 3d bin packing combining object rearrangement and stable placement," in *2024 24th International Conference on Control, Automation and Systems (ICCAS)*, 2024, pp. 964–969.

[16] F. Parreño, R. Alvarez-Valdés, J. M. Tamarit, and J. F. Oliveira, "A maximal-space algorithm for the container loading problem," *INFORMS Journal on Computing*, vol. 20, no. 3, pp. 412–422, 2008.

[17] J. Weng, H. Chen, D. Yan, K. You, A. Duburcq, M. Zhang, Y. Su, H. Su, and J. Zhu, "Tianshou: A highly modularized deep reinforcement learning library," *Journal of Machine Learning Research*, vol. 23, no. 267, pp. 1–6, 2022.