MC-GOPT: Multi-Bin and Curriculum Extensions of GOPT for Generalizable 3D Bin Packing

Naixiang Gao
Department of Mechanical Engineering
Stanford University
ngao4@stanford.edu

Alex Qiu
Department of Mechanical Engineering
Stanford University
aqiu34@stanford.edu

Abstract

With increasing popularity of robotics applications in warehouses, the 3D Online Bin Packing Problem (3D-BPP) has become a focused area of interest. Bin packing is an NP-hard problem and has started to garner more interest from roboticists. Deep Reinforcement Learning has emerged as a promising solution to this problem, with Xiong et al. [2024] developing GOPT as a generalizable approach to 3D-BPP. While their model has shown impressive performance and generalization capabilities, we aim to explore the full potential of this model via various training schemes. We propose more diverse training and curriculum learning approaches to improve generalization. Through extensive experiments, we show that curriculum learning based off a custom packing difficulty heuristic beats GOPT performance on all of our test configurations. The source code will be publicly available at https://github.com/N9nGe/MC-GOPT.

1 Introduction

Warehouse automation has become a major application for robotics within the past decade, as purchasing consumer products for delivery has become the most commonly form of consumerism. The use of robots within these warehouses expands across a vast range of tasks, such as sorting, packing, and inventory management (Wang and Hauser [2021]). In particular, bin packing is one task that can benefit greatly from automation, as this can be both a tedious and labor-intensive task for humans to constantly complete. Bin packing is a classic NP-hard problem, with the robotics variant being 3D-BPP. What makes this problem more challenging is the fact that in real-world scenarios, we do not know the exact location and placement of each item in the bin, and can only attempt to pack items under the observation of one incoming item. This is referred to as the online 3D-BPP problem. To solve this problem, the field has started to explore the application of Deep Reinforcement Learning (Zhao et al. [2021], Xiong et al. [2023], Yang et al. [2021]). While these approaches have displayed strong performance, they sometimes struggle with convergence and generalization across different bin sizes. To address this issue, Xiong et al. [2024] proposed GOPT, a generalizable approach to the online 3D-BPP problem via transformer based Deep Reinforcement Learning. Their method claims to achieve relatively high space utilization across various bin sizes in both simulation and in the real world. While these results are impressive, we would like to explore the full capabilities and potential of this approach. Therefore, the remainder of this paper will explore how well GOPT is able to generalize and if improving the training pipeline can further improve generalization as well as overall performance.

Our main contributions are as follows:

1. We exhaustively evaluate the GOPT model on its generalization capabilities beyond the paper's experiments.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

2. We develop new training regimes for GOPT in attempt to improve upon the baseline performance, these include the following:
   (a) Multi-Bin Training Set
   (b) Curriculum Training Packing Difficulty Progression
   (c) Curriculum Training Bin Size Progression
   (d) Combining Packing Difficulty Curriculum Training with Multi-Bin Set

2 Related Work

2.1 Heuristic Methods

Classical 3D bin packing has long been dominated by heuristic and rule-based strategies, which aim to efficiently approximate feasible placements in combinatorial spaces. Early methods such as first-fit (FF) (Dósa and Sgall [2013]), best-fit (BF) (Dósa and Sgall [2014]), and first-fit decreasing (FFD) (Johnson et al. [1974]) establish simple policies based on sorted item orders, offering computational efficiency but limited adaptivity. More advanced geometric heuristics, including extreme point (EP) (Crainic et al. [2008]) methods and empty maximal space (EMS) (Parreño et al. [2008]), explicitly track free subspaces within the container and select placements that heuristically minimize fragmentation. Despite their wide adoption in industrial packaging, logistics, and automated warehousing, these heuristics rely heavily on human-crafted rules, making them difficult to scale to heterogeneous item distributions, diverse container geometries, or online decision settings where items arrive sequentially (Zhao et al. [2021]). Their lack of learning capability also limits performance in scenarios requiring generalization to unseen configurations. These limitations have motivated a shift toward learning-based methods, particularly reinforcement learning, which can implicitly learn placement strategies from interaction data.

2.2 Deep Reinforcement Learning (DRL) Based Methods

DRL has emerged as a promising approach for addressing 3D bin packing tasks due to its ability to learn sequential decision strategies directly from interaction data (Murdivien and Um [2023], Zhou et al. [2024]). Unlike heuristic approaches that rely on manually crafted placement rules, DRL formulates packing as a Markov decision process, where states encode the partial bin configuration and actions correspond to selecting item positions or orientations (Xiong et al. [2024]). Early neural combinatorial optimization methods such as Pointer Networks and attention-based policies demonstrated strong performance in permutation-structured problems (Bello et al. [2016]). More recent work has applied DRL to packing specifically, showing that actor–critic and policy-gradient frameworks can outperform heuristic baselines by learning implicit geometric reasoning (Zhao et al. [2021]). However, DRL models still face challenges such as sample inefficiency, training instability, and limited generalization to unseen bin geometries or item distributions—problems that are further amplified in online packing settings where items arrive sequentially. These issues motivate integrating structure-aware training mechanisms such as curriculum learning to improve convergence and robustness.

2.3 Curriculum learning in DRL

Curriculum Learning was first introduced by Bengio et al. [2009], who were inspired by animal shaping and human education. The main idea behind curriculum learning is that instead of utilizing randomly selected samples during training, we purposely train on easier samples first and then gradually increase the difficulty. This progressive training strategy has been shown to ease optimization and reduce the exploration burden in long-horizon or sparse-reward environments (Florensa et al. [2017], Wang et al. [2022]). Within robotics and control, curriculum-based DRL methods have demonstrated substantial benefits in locomotion, manipulation, navigation, and dexterous control, where gradually expanding the state or goal space leads to better convergence and generalization. Representative examples include reverse curriculum generation, which automatically constructs easier initial states to bootstrap policy learning (Florensa et al. [2017]), and automatic domain randomization strategies that adapt the task distribution during training to improve robustness (Portelas et al. [2020]). However, curriculum learning remains comparatively underexplored in geometric reasoning and combinatorial decision-making problems such as 3D bin packing. Designing curricula that progressively increase item complexity, bin configurations, or placement constraints could significantly enhance the sample efficiency and generalization capabilities of DRL-based packing policies.

3 Method

We aim to test GOPT's peformance in simulation and hope to improve the generalization and bin packing performance capabilities through various methods. The authors trained GOPT on a RS dataset Zhao et al. [2021] with a set bin size of 10 x 10 x 10. They then test generalization on 30 x 30 x 30, 50 x 50 x 50, and 100 x 100 x 100 sized bins. The authors have released source code to train and test GOPT, which we plan to utilize and modify as needed (https://github.com/Xiong5Heng/GOPT). Below are our main objectives:

1. We will first test the generalization capabilities of GOPT on non-cubic bin sizes (e.g. 8x18x15), which would be considered out-of-distribution from the training dataset.
2. We will then attempt to modify the training dataset by adding multiple cubic bin-sizes(e.g. 30x30x30) instead of just a singular bin size. With this modification, we will train a new model and test its zero-shot performance compared to the baseline model.
3. We will develop a curriculum training framework, that will include packing difficulty heuristics to allow for the model to start training with easy instances and gradually move to harder instances. We will manually design the packing difficulty heuristic.
4. We will implement curriculum training for multi-bin training, without packing difficulty. The curriculum will be based off container size alone.
5. Finally, we will also combine packing difficulty curriculum training with multiple bin sizes in order to test the potential benefits of both curriculum and multi-bin training.

3.1 Baseline Generalization Test

We first trained a baseline GOPT model to reproduce the paper's results. The GOPT baseline model trains only on a 10x10x10 container size. For evaluation, Xiong et. al tested their model on container sizes 10x10x10, 30x30x30, 50x50x50, and 100x100x100. To further test generalization, we added non-cubic containers in the evaluation set, including: 5x8x10, 8x18x15, 12x15x20, 15x25x30, 20x35x40.

3.2 Multi-Bin Training Set

We decided to test two different multi-bin training sets to see if the GOPT model would benefit from more container size diversity as opposed to just training on a single container size. The first training set utilized three bins sizes: 10x10x10, 15x15x15, and 20x20x20. The second training set added five extra bin sizes for a total of nine different bin sizes. Some of the extra bin sizes added were non-cubic. The eight bin sizes are as listed: 10x10x10, 10x15x12, 20x20x20, 20x25x30, 25x35x30, 30x30x30, 40x45x50, and 50x50x50. It should be noted that while these training sets are listed as though they are a pre-collected offline dataset, they are in fact generated online during training. The multi-bin training set was implemented by randomly choosing a container size out of a list of available sizes for each training example.

3.3 Curriculum Training Packing Difficulty Progression

To implement curriculum training, we implement a packing difficulty heuristic. We formalize the packing difficulty based on the box-to-bin volume ratio. For a box with dimensions (l_b, w_b, h_b) and container with dimensions (L, W, H), the dimensional ratio is:

$$r_{dim}(b) = \sqrt[3]{\frac{l_b}{L} \cdot \frac{w_b}{W} \cdot \frac{h_b}{H}}$$

Higher dimensional ratios correspond to easier packing tasks, as larger items relative to the container size result in: (1) Fewer placement decisions per episode. (2) More forgiving positioning errors. (3) Simpler spatial reasoning requirements. Thus this provides us with a simple packing difficulty heuristic that can guide our learning curriculum.

3.4 Curriculum Training Bin Size Progression

We additionally test curriculum training on GOPT based off bin size, with smaller bin sizes considered easier than larger bin sizes. We test this on a progression of 3 different bin sizes: 10x10x10, 12x12x12, and 15x15x15. We utilize a curriculum length of 700 epochs out of a total 1000 epochs of training, using a linear progression of bin sizes.

3.5 Combining Packing Difficulty Curriculum Training with Multi-Bin Set

Finally, we combine our packing difficulty curriculum training with multiple bin sizes. This entails sampling easier dimensional ratios of box sequences at the start of training but with randomized bin sizes per training instance. The three bin sizes available were 10x10x10, 15x15x15, and 20x20x20.

4 Experiments

4.1 Training Details

For training, we utilized a single NVIDIA GeForce RTX 4090. Each model was trained to 1000 epochs, with parallel processing set to 24. Training each model depending on the experiment, took anywhere from 24 to 96 hours of training.

4.2 Results

For all models evaluated, a total of 1000 episodes were run. We report average space utilization (Uti), number of boxes packed (Num), and the standard deviation of the space utilization (Std) for the 1000 evaluation episodes. These metrics are the same metrics used by Xiong et al. [2024] for their experiments. Average space utilization

4.2.1 Baseline Generalization Test

The results for our baseline generalization test are shown in Table 1. On top of the reproduced results from the experiments that Xiong et al. [2024] conducted, we added five more configurations that are non-cubic. As can be seen in the table, the baseline GOPT model struggles to generalize to these non-cubic containers and show generally similar Uti performance.

Table 1: Generalization performance of the baseline model trained on 10×10×10 containers.

Test Container Size	Uti	Num	Std
Cubic Containers			
10×10×10	74.8%	29.0570	0.0769
30×30×30	74.8%	29.0740	0.0751
50×50×50	75.0%	29.1420	0.0739
100×100×100	75.0%	29.1420	0.0739
Non-Cubic Containers			
5×8×10	47.1%	8.020	0.2007
8×18×15	41.7%	8.432	0.1592
12×15×20	44.9%	8.579	0.1604
15×25×30	45.3%	8.083	0.1889
20×35×40	43.4%	8.138	0.1778

4.2.2 Multi-Bin Training Set

The results of both the 3-bin and 8-bin model are shown in Table 2 and Figure 1, where they are compared to the performance of the baseline model. The bolded values in Table 2 represent the strongest performing model for that test configuration. As can be seen, the baseline model outperforms 3-bin and 8-bin in all cubic categories. For non-cubic containers however, we found that the

3-bin model outperformed baseline and 8-bin for 4/5 of the non-cubic containers, showcasing an improvement in generalization. This indicates that added more training diversity can yield better performance, however in the case of the 8-bin model the amount of diversity negatively impacted performance likely due to insufficient training time. We hypothesize that training for much longer than 1000 epochs on 3-bin and 8-bin can yield much better results.

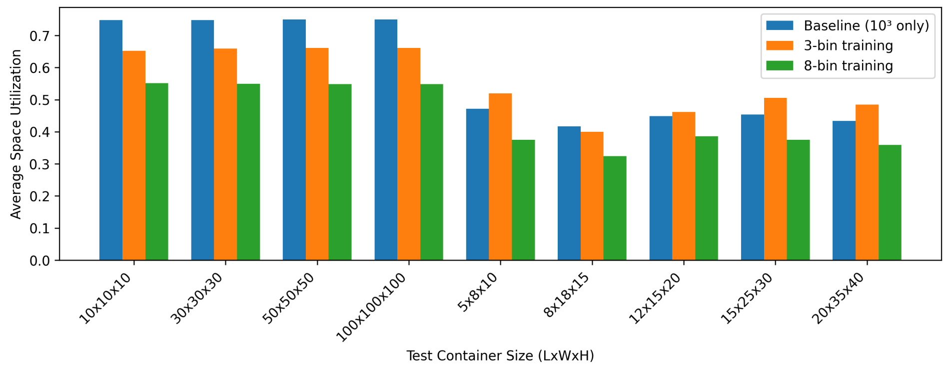


Figure 1: Utilization Comparison Across Different Multi-bin Size Training Strategies

Table 2: Performance of different multi-bin training strategies across all 9 test container sizes.

Method	Cubic Containers								Non-Cubic Containers									
	$10\times10\times10$		$30\times30\times30$		$50\times50\times50$		$100\times100\times100$		$5\times8\times10$		$8\times18\times15$		$12\times15\times20$		$15\times25\times30$		$20\times35\times40$	
	Uti	Num	Uti	Num	Uti	Num	Uti	Num	Uti	Num	Uti	Num	Uti	Num	Uti	Num	Uti	Num
Baseline (10^3 only)	**74.8%**	**29.06**	**74.8%**	**29.07**	**75.0%**	**29.14**	**75.0%**	**29.14**	47.1%	8.02	**41.7%**	**8.43**	44.9%	8.58	45.3%	8.08	43.4%	8.14
3-bin training	65.2%	25.58	65.9%	25.85	66.1%	25.91	66.1%	25.91	**52.0%**	**8.68**	40.0%	7.96	**46.1%**	**8.78**	**50.6%**	**8.77**	**48.5%**	**8.84**
8-bin training	55.2%	21.87	54.9%	21.80	54.8%	21.77	54.8%	21.77	37.5%	6.58	32.4%	6.73	38.6%	7.52	37.5%	6.80	35.9%	6.82

4.2.3 Curriculum Training Packing Difficulty Progression

For packing difficulty curriculum training, we trained four separate models. Two models were trained purely on 10x10x10 container sizes and the other two were trained purely on 15x15x15 container sizes. Within these pairs, each has a model trained with a curriculum length of 400 epochs and a length of 700 epochs. Curriculum length refers to how long it takes for the training stage to get to the highest difficulty level. After the curriculum ends, the training examples are all hard. Figure 2 showcases a comparison of these pairs of models. We find that for the 10x10x10 pair of models, the 400 epochs model outperforms the 700 epochs model for almost all configurations. The reverse was true for the 15x15x15 pair of models.

In terms of performance compared to the baseline GOPT model, Table 3 shows average Uti performance comparisons across cubic and non-cubic configurations. The 10x10x10-400 epoch model shows the best performance for both cubic and non-cubic configurations. We hypothesize this model was so successful due to the fact that it utilized a similar base training set as the baseline model, however it greatly benefited from a fast curriculum and extended post-curriculum training on hard examples (600 epochs worth).

Method	Cubic	Non-cubic
Baseline	74.9%	44.5%
Curriculum 10^3–400ep	**75.2%**	**51.9%**
Curriculum 10^3–700ep	74.4%	45.0%
Curriculum 15^3–400ep	71.9%	45.0%
Curriculum 15^3–700ep	71.9%	46.9%

Table 3: Average space utilization of different curriculum training strategies with packing difficulty heuristic on cubic, and non-cubic. The detailed experiment data for all container sizes is in the Appendix A Table 6.

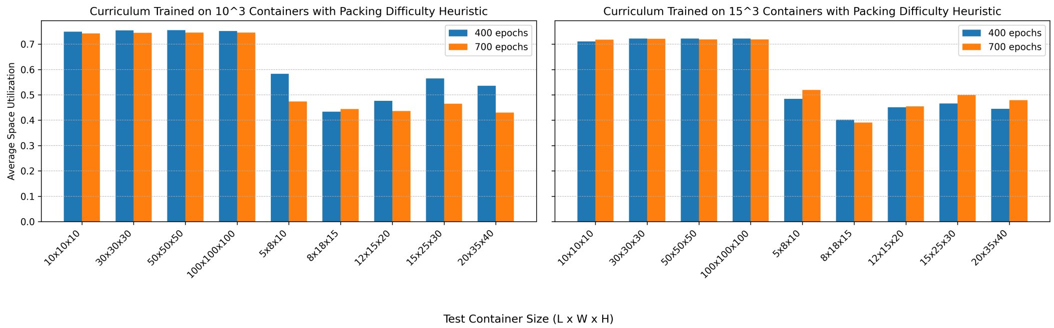


Figure 2: Utilization Comparison Across Different Curriculum Training Strategies with Packing Difficulty Heuristic

4.2.4 Curriculum Training Bin Size Progression

We only trained a single model for the bin size curriculum training approach. The results of its evaluation can be seen in Figure 3 and Table 4. The baseline model greatly beats our methods model for cubic containers, but shows similar performance for non-cubic containers. We hypothesize this approach was not successful due to the weak correlation between container size and "difficulty". The difficulty of packing a 10x10x10 container might not necessarily be much easier than a 20x20x20 container for the model, which would then make the curriculum invalid.

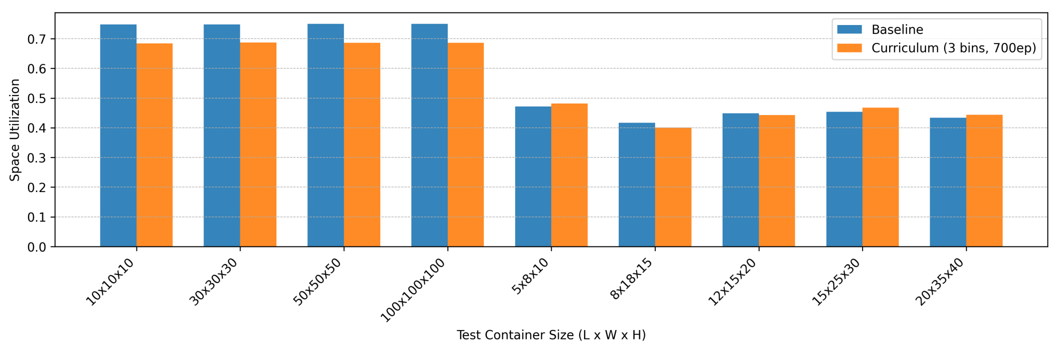


Figure 3: Utilization Comparison between Baseline and Curriculum Training with Bin Size Heuristic

Test Container Size	**Baseline Uti**	**Curriculum Uti**
Cubic Containers		
10x10x10	**74.8%**	68.4%
30x30x30	**74.8%**	68.7%
50x50x50	**75.0%**	68.6%
100x100x100	**75.0%**	68.6%
Non-Cubic Containers		
5x8x10	47.1%	**48.2%**
8x18x15	**41.7%**	40.1%
12x15x20	**44.9%**	44.3%
15x25x30	45.3%	**46.8%**
20x35x40	43.4%	**44.4%**

Table 4: Space utilization of the baseline model and the curriculum model (bin size heuristic: 3 bin sizes, 700 epochs) on all test containers.

4.2.5 Combining Packing Difficulty Curriculum Training with Multi-Bin Set

Finally, we showcase our results for a combined training approach in Figure 4 and Table 5. This approach showed the worst performance across all training approaches, with no improvement in performance for any test configuration. We speculate that this is because our training is severely under sampled for our complex training regime. Therefore with more extensive training resources, we could potentially see improvement for this training approach.

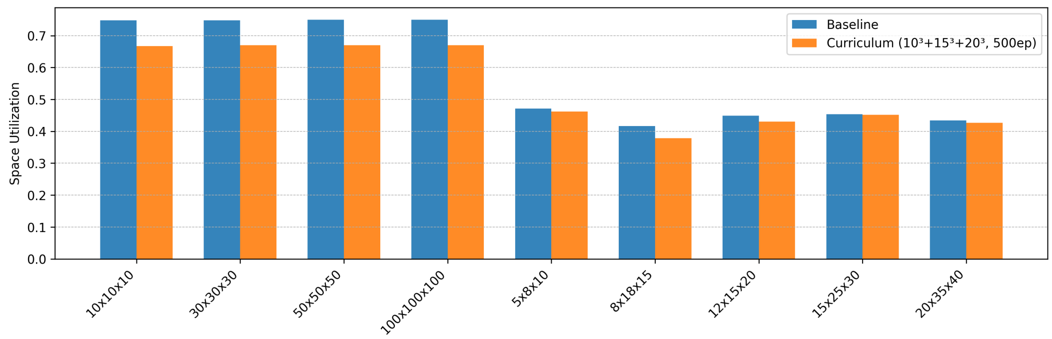


Figure 4: Utilization Comparison between Baseline and Multi-bin size + Curriculum Training with Packing Difficulty Heuristic

Test Container Size (L×W×H)	Baseline Uti	Combined Uti
Cubic Containers		
10x10x10	**74.8%**	66.7%
30x30x30	**74.8%**	67.0%
50x50x50	**75.0%**	67.0%
100x100x100	**75.0%**	67.0%
Non-Cubic Containers		
5x8x10	**47.1%**	46.2%
8x18x15	**41.7%**	37.8%
12x15x20	**44.9%**	43.1%
15x25x30	**45.3%**	45.2%
20x35x40	**43.4%**	42.7%

Table 5: Space utilization of the baseline model and the multi-size (10^3, 15^3, 20^3) + curriculum model with packing difficulty heuristic (500 epochs) evaluated on all 9 test container sizes

5 Conclusion

In this work, we extend the GOPT algorithm by exploring more diverse and scalable training strategies beyond the original single-bin setting of 10^3. We introduce multi–bin-size training, two forms of curriculum-based training, and a combined regime, and systematically evaluate how these strategies influence generalization across cubic and non-cubic container geometries. Our results reveal a distinct trade-off: multi-bin training reduces performance on cubic containers but markedly improves accuracy on non-cubic ones, indicating stronger adaptability to heterogeneous and realistic packing environments. Both curriculum strategies—difficulty ordering based on item-to-container ratios and a staged expansion of bin sizes during training—further enhance performance across container types, demonstrating that structured learning progressions provide more stable optimization.

At the same time, our analysis exposes important limitations. As the environment becomes more complex—larger containers, richer bin-size mixtures, and broader sampling ranges—the fixed epoch budgets inherited from baseline GOPT training become insufficient. Under such constrained horizons, policies often fail to fully converge, resulting in substantial performance drops in the most complex settings.

Overall, our study broadens the design space for GOPT-based packing systems and provides new empirical insights into how container diversity and curriculum structure jointly affect policy generalization, paving the way toward more robust and scalable learning-based packing solutions.

6 Future Work

In future work, we aim to improve the multi–bin-size training regime by scaling the training horizon in proportion to the sampling distribution of container sizes. In our current experiments, the 3-bin and 8-bin settings were trained with only 1000 and 1200 epochs due to limited GPU resources. Such constrained budgets may prevent the policy from fully converging under diverse bin mixtures, resulting in suboptimal generalization. Increasing the number of epochs proportionally to the number and diversity of sampled bins is therefore a natural extension.

We also plan to refine our curriculum strategies for multi-container packing. Our first heuristic—sorting training instances by the ratio between item size and container size—captures only coarse difficulty and cannot model key edge cases such as elongated or flat items. A second heuristic explored in our experiments gradually expands the set of available container sizes during training (e.g., initially training on [10,10,10], then progressively adding larger bins), which offers a staged and controllable learning progression. Designing more principled difficulty metrics and integrating both curriculum types into a unified framework may further stabilize optimization. Finally, combining extended training schedules with improved curriculum design represents a promising direction for enhancing policy robustness and cross-bin generalization.

7 Contributions

Naixiang Gao and Alex Qiu contributed equally to this work. Both authors were involved in all major aspects of the project. For specific implementation details, certain tasks were divided for efficiency: Alex Qiu primarily led the design and execution of the experiments, while Naixiang Gao was responsible for validating and analyzing the experimental results.

References

Irwan Bello, Hieu Pham, Quoc Viet Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

Teodor Gabriel Crainic, Giacomo Perboli, and Roberto Tadei. Extreme point-based heuristics for three-dimensional bin packing. *INFORMS Journal on Computing*, 20(3):368–384, 2008. doi: 10.1287/ijoc.1070.0250.

György Dósa and Jiří Sgall. First fit bin packing: A tight analysis. *LIPIcs*, 20:538–549, 01 2013. doi: 10.4230/LIPIcs.STACS.2013.538.

György Dósa and Jiří Sgall. Optimal analysis of best fit bin packing. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 8573 of *Lecture Notes in Computer Science*, pages 429–441. Springer, 2014. doi: 10.1007/978-3-662-43948-7_38.

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *Conference on robot learning*, pages 482–495. PMLR, 2017.

David Stifler Johnson, Alan Jay Demers, Jeffrey David Ullman, Michael René Garey, and Ronald Lewis Graham. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on Computing*, 3(4):299–325, 1974. doi: 10.1137/0203021.

Shokhikha Murdivien and Jumyung Um. Boxstacker: Deep reinforcement learning for 3d bin packing problem in virtual environment of logistics systems. *Sensors*, 23:6928, 08 2023. doi: 10.3390/s23156928.

Fidel Parreño, Rafael Álvarez-Valdés, José María Tamarit, and José Fernando Oliveira. A maximal-space algorithm for the container loading problem. *INFORMS Journal on Computing*, 20(3): 412–422, 2008. doi: 10.1287/ijoc.1070.0251.

Rémy Portelas, Cédric Colas, Lilian Weng, Katja Hofmann, and Pierre-Yves Oudeyer. Automatic curriculum learning for deep rl: A short survey. *arXiv preprint arXiv:2003.04664*, 2020.

Fan Wang and Kris Hauser. Dense robotic packing of irregular and novel 3d objects. *IEEE Transactions on Robotics*, 38(2):1160–1173, 2021.

Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576, 2022. doi: 10.1109/TPAMI.2021.3069908.

Heng Xiong, Kai Ding, Wan Ding, Jian Peng, and Jianfeng Xu. Towards reliable robot packing system based on deep reinforcement learning. *Advanced Engineering Informatics*, 57:102028, 2023.

Heng Xiong, Changrong Guo, Jian Peng, Kai Ding, Wenjie Chen, Xuchong Qiu, Long Bai, and Jianfeng Xu. Gopt: Generalizable online 3d bin packing via transformer-based deep reinforcement learning. *IEEE Robotics and Automation Letters*, 2024.

Zifei Yang, Shuo Yang, Shuai Song, Wei Zhang, Ran Song, Jiyu Cheng, and Yibin Li. Packerbot: Variable-sized product packing with heuristic deep reinforcement learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5002–5008. IEEE, 2021.

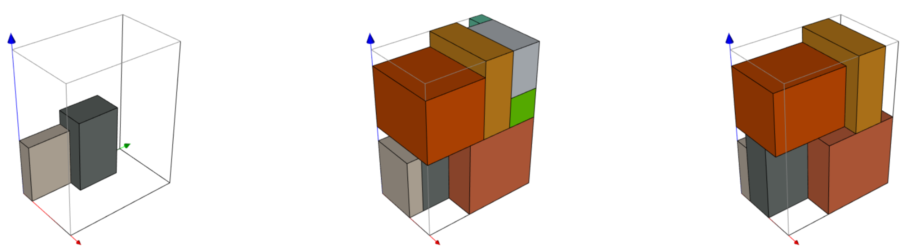
Hang Zhao, Qijin She, Chenyang Zhu, Yin Yang, and Kai Xu. Online 3d bin packing with constrained deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 741–749, 2021.

Peiwen Zhou, Ziyan Gao, Chenghao Li, and Nak Young Chong. An efficient deep reinforcement learning model for online 3d bin packing combining object rearrangement and stable placement, 2024. URL `https://arxiv.org/abs/2408.09694`.

A Detail Experiment Data

Test Container Size	Baseline		10^3–400ep		10^3–700ep		15^3–400ep		15^3–700ep	
	Uti	**Num**	**Uti**	**Num**	**Uti**	**Num**	**Uti**	**Num**	**Uti**	**Num**
Cubic Containers										
10x10x10	0.7475	**29.057**	**0.7491**	29.038	0.7417	28.848	0.7106	27.693	0.7173	27.908
30x30x30	0.7479	29.074	**0.7540**	**29.202**	0.7451	28.975	0.7220	28.106	0.7207	28.058
50x50x50	0.7499	29.142	**0.7550**	**29.226**	0.7453	28.975	0.7224	28.106	0.7183	27.973
100x100x100	0.7499	**29.142**	**0.7516**	29.134	0.7453	28.975	0.7224	28.106	0.7183	27.973
Non-Cubic Containers										
5x8x10	0.4714	8.020	**0.5831**	**9.603**	0.4738	8.088	0.4844	8.210	0.5190	8.727
8x18x15	0.4169	8.432	0.4340	8.637	**0.4437**	**8.808**	0.4019	8.075	0.3908	7.914
12x15x20	0.4491	8.579	**0.4763**	**9.033**	0.4360	8.396	0.4508	8.588	0.4543	8.705
15x25x30	0.4534	8.083	**0.5648**	**9.658**	0.4649	8.300	0.4659	8.241	0.4995	8.750
20x35x40	0.4337	8.138	**0.5363**	**9.688**	0.4302	8.053	0.4454	8.270	0.4794	8.749

Table 6: Full results for all training strategies on all 9 test containers.



(a) Baseline (b) Curriculum in 10^3 (c) Curriculum in 15^3

Figure 5: Visualization of Packing Results under Different Bin Sizes