Gradual Receptive Expansion Using Vision Transformer for Online 3D Bin Packing

Minjae Kang[1], Hogun Kee[1], Yoseph Park[1], Junseok Kim[1], Jaeyeon Jeong[1], Geunje Cheon[2], Jaewon Lee[1], and Songhwai Oh[1,2]

***Abstract*— The bin packing problem (BPP) is a challenging combinatorial optimization problem with a number of practical applications. This paper focuses on online 3D-BPP, where the packer makes immediate decisions for a loading position as items continually arrive. We propose a novel reinforcement learning algorithm, GREViT, which utilizes a vision transformer to tackle online 3D-BPP for the first time. By introducing the gradual receptive expansion technique, GREViT overcomes the limitations inherent in learning-based methods that only excel in their trained bins. As a result, GREViT surpasses existing BPP algorithms in packing ratio across various bin sizes. The effectiveness of GREViT in real-world scenarios is validated by its successful demonstrations using a real robot for solving 3D-BPP. The attached video demonstrates GREViT undertaking 3D-BPP in both simulated and real-world environments.**

I. Introduction

The bin packing problem (BPP), a combinatorial optimization problem that demands the placement of items with varying sizes into separate bins without exceeding the maximum utilization of bins, remains one of the challenging problems. There are different variations of BPP [1]–[4], including 3D-BPP [5], which deals with packing objects with volume, commonly encountered in real-world situations, such as warehouse management, pallet loading, and cargo packing. In this paper, we focus on loading items into a 3D bin with limited space as shown in Figure 1.

In recent years, many studies have been conducted to adapt standard 3D-BPP to real-world scenarios. To this end, two practical assumptions have been integrated into 3D-BPP. First, the packer only observes the specifics (size, weight, or density) of a fixed number of items, not all items. This problem setting, termed online BPP [6], requires immediate decision-making as items arrive continuously without advance knowledge of future items. Second, the packer must consider the stability of the loaded items [7]. Unlike 1D and 2D-BPP, 3D items occupy space and form a physical structure, so the loading state could collapse if the stability is not maintained. By considering the above assumptions, we address the 3D-BPP setting applicable in real-world tasks.

Previous research has explored heuristic algorithms as potential solutions for online 3D-BPP [5], [8]–[11]. These methods offer consistent rules inspired by the strategies of human packers. For instance, the corner-point (CP) algorithm [5] finds corner positions of the loaded area that cover all items already packed, aiming to place new items close to existing ones. Another approach, the empty-maximal-space (EMS) algorithm [9], places the item at a point where the largest empty orthogonal space remains after loading to generate promising loading states. Since most heuristic algorithms are designed without presuming bin size or item size distributions, they are easily applicable across various 3D-BPP settings. However, developing a universal and effective rule for diverse bin packing scenarios is challenging, leading heuristics to generally achieve lower packing ratios.

Fig. 1: Performing online 3D-BPP in NVIDIA Isaac Sim. The robot observes the current loading state and next item and decides the optimal loading position in the bin.

On the other hand, deep reinforcement learning (RL)-based methods for learning a packing policy have widely been studied. [12]–[14] reconfigure the loading state of the bin into a 2D grid map and utilize a discrete action set made up of grid coordinates. [12]–[15] use the actor-critic algorithm wherein the critic network is trained to estimate the final packing ratio when a block is loaded on a specific grid coordinate. Typically, learning-based methods surpass heuristics in performance, but their performance depends heavily on assumptions such as the bin size, size distribution of boxes, and the number of possible orientations of a box. If training and evaluation settings diverge, the loading performance can be significantly degraded.

To overcome this limitation, we propose gradual receptive expansion using vision transformer (GREViT), a novel RL algorithm for online 3D-BPP. To the best of our knowledge, GREViT is the first algorithm to utilize a vision transformer (ViT) [16] for BPP. GREViT employs Tsallis actor-critic (TAC) [17] to capture the multi-modal optimal packing locations of BPP and incorporates ViT as a critic network. Further, by proposing a gradual receptive expansion (GRE) technique that progressively broadens the receptive area on the bin, GREViT not only amplifies BPP performance but also establishes an advantage of robust training across various bin sizes. In the experiment, GREViT outperformed existing heuristics and learning-based methods for all item

[1] Department of Electrical and Computer Engineering and ASRI, Seoul National University, Seoul, Korea (e-mail: {minjae.kang, hogun.kee, yoseph.park, junseok.kim, jeayeon.jeong, jaewon.lee}@rllab.snu.ac.kr, songhwai@snu.ac.kr), [2] Graduate School of Artificial Intelligence (GSAI) and ASRI, Seoul National University, Seoul, Korea (e-mail: geunje.cheon@rllab.snu.ac.kr). *(Corresponding author: Songhwai Oh.)*

This work was supported by the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2022-0-00480, Development of Training and Inference Methods for Goal-Oriented Artificial Intelligent Agents).

sequences. Also, GREViT achieved an average enhancement of packing ratio of 6.2 percentage points over heuristics across diverse bin sizes not used for training. In addition, we propose a multi-step loading scheme designed to effectively incorporate GREViT in circumstances where multiple items are previously observed. Finally, by successfully loading boxes using a real robot, UR5, we have validated the applicability of GREViT in real-world scenarios.

II. Related Work

A. Online 3D Bin Packing Algorithms

Recent studies addressing online 3D-BPP can be categorized into two main approaches. The first approach is heuristic methods that draw inspiration from human bin packing techniques to decide suitable loading positions. To compare performance, we implement and evaluate a variety of heuristics such as corner-point (CP) [5], extreme-point (EP) [8], empty-maximal-space (EMS) [9], heightmap-minimization (HM) [10], and deep-bottom-left (DBL) [11].

On the other hand, the second approach employs RL algorithms. [12] and [13] train a policy to select loading positions from all stable positions within the bin, while [14] and [15] consider only positions suggested by heuristics. Furthermore, there are strategies that involve allocating additional space or time to enhance the packing ratio. For instance, [13] uses extra space for temporary block placement, and [14] takes additional space and time by relocating already loaded boxes. We have selected [12] and [15] as baseline models due to the availability of their officially published codes.

B. Tsallis Actor-Critic

TAC [17] is an off-policy actor-critic RL algorithm that updates the actor and critic networks to maximize both the cumulative rewards and the Tsallis entropy of the output of the policy. In contrast, soft actor-critic (SAC) [18], a widely used maximum entropy RL method, utilizes the Shannon-Gibbs entropy, which is a specific case of the Tsallis entropy under a certain parameter setting. As a result, TAC offers a wider range of options for RL compared to SAC.

For a Tsallis Markov decision process, the Bellman equation for state value function V and state-action value function Q is as follows:

$$\begin{aligned} \ln_q(x) &\stackrel{\text{def}}{=} (x^{q-1}-1)/(q-1), \\ Q^\pi(s,a) &= \mathop{\mathbb{E}}_{s'\sim P(\cdot|s,a)}\big[r(s,a,s') + \gamma V^\pi(s')\big], \\ V^\pi(s) &= \mathop{\mathbb{E}}_{a\sim\pi}\big[Q^\pi(s,a) - \alpha \ln_q(\pi(s))\big], \end{aligned} \tag{1}$$

where π is a policy, $P(s'|s,a)$ is the state transition probability, r is the reward function, γ is a discount factor, and α is an entropy temperature. Based on the above equations, TAC induces objectives for π, Q, and V as follows:

$$\begin{aligned} J_\pi &= \mathop{\mathbb{E}}_{s_t\sim\mathcal{D}}\Big[\mathop{\mathbb{E}}_{a\sim\pi(\cdot|s_t)}\big[\alpha\ln_q(\pi(a|s_t)) - Q(s_t,a)\big]\Big], \\ J_Q &= \mathop{\mathbb{E}}_{(s_t,a_t,r_t,s_{t+1})\sim\mathcal{D}}\big[(Q(s_t,a_t) - r_t - \gamma V(s_{t+1}))^2/2\big], \\ J_V &= \mathop{\mathbb{E}}_{(s_t,a_t)\sim\mathcal{D}}\big[(Q(s_t,a_t) - \alpha\ln_q(\pi(a_t|s_t)) - V(s_t))^2/2\big], \end{aligned} \tag{2}$$

where $\mathcal{D}$ is a replay buffer. However, as TAC is originally designed for continuous action spaces, we have modified it to suit the discrete actions. In Section V-A, we present the modified Bellman equation and objectives for discrete TAC.

III. Problem Statement

This section defines online 3D-BPP that we address in this paper. We stack cuboid-shaped blocks b_i with sizes $(x_{b_i}, y_{b_i}, z_{b_i}) \in \mathbb{R}^3_+$ along x, y, z axes on one cuboid-shaped bin $\mathcal{C}$ with sizes $(x_\mathcal{C}, y_\mathcal{C}, z_\mathcal{C}) \in \mathbb{R}^3_+$ in an axis-aligned manner. To simplify the problem definition, we denote the smallest unit of length l and assume all lengths of the bin and blocks to be multiples of l, i.e., b_i and $\mathcal{C}$ can be represented as $(\bar{x_{b_i}}, \bar{y_{b_i}}, \bar{z_{b_i}}) \in \mathbb{Z}^3_+$ and $(\bar{x_\mathcal{C}}, \bar{y_\mathcal{C}}, \bar{z_\mathcal{C}}) \in \mathbb{Z}^3_+$, where $\bar{x}$ is the quotient of x divided by l. To prevent simple packing scenarios, we set each length of the block to be less than or equal to half the minimum length of the bin, i.e., $max(\bar{x_{b_i}}, \bar{y_{b_i}}, \bar{z_{b_i}}) \leq min(\bar{x_\mathcal{C}}, \bar{y_\mathcal{C}}, \bar{z_\mathcal{C}})/2$.

The loading state of the bin can be depicted as a 2D grid map $s_t \in \mathbb{Z}_+^{\bar{x_\mathcal{C}} \times \bar{y_\mathcal{C}}}$, with each grid cell having an integer value ranging from 0 to $\bar{z_\mathcal{C}}$, representing the loading height. The agent uses the current loading state s_t and block b_t as an observation to determine the action a_t, which is a three-dimensional integer vector. The first index of the action determines the orientation of the block, while the remaining two integers indicate the grid coordinates where the left-bottom corner of the block is placed. Once the location of the block is determined, its height is automatically set, making two-dimensional coordinates adequate for the action representation. The reward for RL corresponds to the loading rate increase, which is proportional to the volume of the currently placed block. The objective of BPP is to maximize the packing ratio within the bin, serving as the metric for evaluating the performance of BPP algorithms. Each BPP episode ends if the current block cannot be accommodated within the bin or the loading state becomes unstable.

For the implementation of online 3D-BPP in a real-world environment, we consider several aspects. 1) Blocks must be stably packed in the bin. We adopt stability conditions proposed by [12] to check the stability of the loading state. 2) Blocks can be reoriented for effective packing. We consider a total of six block orientations, a common setting in 3D-BPP. 3) k blocks can be previewed. We assume $k = 1$ for training, though the trained agent can accommodate instances when $k > 1$. This assumption is practical in a real-world factory, allowing for the possibility of a k-step packing plan, thus improving the packing ratio. 4) The relocation policy for loaded blocks and buffers for temporarily placing blocks are not considered as these demand extra time and space, detrimental to real-time packing operations in a factory setting. Based on the above assumptions, we describe RL for online 3D-BPP in detail in the subsequent sections.

IV. Q-Value Estimation via Vision Transformer

We propose a novel actor-critic off-policy RL algorithm, **G**radual **R**eceptive **E**xpansion using **Vi**sion **T**ransformer (GREViT), which utilizes ViT for solving online 3D-BPP, as depicted in Figure 2. GREViT employs a ViT-based model as a critic network and predicts Q-value, $Q(s_t, b_t, a_t)$, indicating the cumulative sum of future rewards afforded when the block b_t is placed by performing the action a_t within the loading state s_t. Consequently, Q-value becomes the expected future packing ratio of the bin. This section describes the Q-value inference process that utilizes ViT, while the next section elaborates on the entire GREViT algorithm, including the ViT-based critic network.

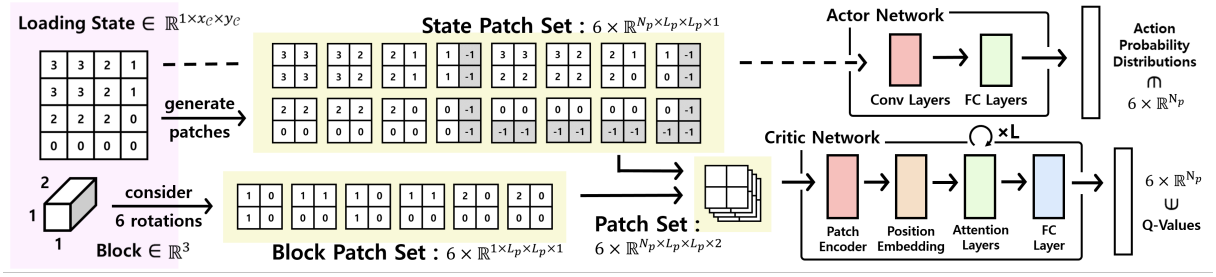

Fig. 2: Overview of the GREViT structure. (Left) The loading state and block are converted into a set of fixed-sized patches for input of ViT. (Right-Bottom) The patch set is fed into the critic network to output the expected packing ratio. (Right-Top) The actor network takes the loading state and block as input and deduces the probability of loading a block at each position.

A. Patch Generation for Vision Transformer

In online 3D-BPP, the loading state $s_t \in \mathbb{Z}_+^{\bar{x_{\mathcal{C}}}\times\bar{y_{\mathcal{C}}}}$ and the block $b_t \in \mathbb{Z}_+^3$ serve as an observation. As depicted in Figure 3(a), each observation is transformed into six patch sets, and each patch set $P \in \mathbb{Z}^{N_p\times L_p\times L_p\times C_p}$ is used as input of ViT, where N_p is the number of generated patches, L_p is the size of the patch, and C_p is the number of patch channels. Six patch sets are generated from a state patch set P_s and block patch set P_b, i.e., we assign C_p to two, with one channel designated for the loading state and the other for the block.

The loading state s, represented by a 2D grid map, is segmented into fixed-sized state patches as depicted in Figure 3(b). Contrary to the conventional ViT, however, the patches for GREViT are overlapped. The state patches are generated by moving the receptive area like a filter of the convolutional layer with one stride. If the receptive area deviates from the bin, we fill deviated grids with -1 as in the last case of Figure 3(b). Therefore, the number of state patches is determined as $N_p = \bar{x_{\mathcal{C}}} \times \bar{y_{\mathcal{C}}}$, which is equivalent to the number of grids in the bin. Next, the ViT patch size, L_p, is established as half of the longer length among $\bar{x_{\mathcal{C}}}$ and $\bar{y_{\mathcal{C}}}$ of the bin, i.e., $L_p = \lfloor \frac{\max\{\bar{x_{\mathcal{C}}},\bar{y_{\mathcal{C}}}\}}{2} \rfloor$, where $\lfloor x \rfloor$ is the biggest integer less than or equal to x. Finally, the loading state s is converted to the state patch set $P_s \in \mathbb{Z}^{N_p\times L_p\times L_p\times 1}$.

The block patch conversion is executed separately for each block rotation as shown in Figure 3(c). The block b with a rotation r is transformed into single block patch $P_{b^r} \in \mathbb{Z}^{1\times L_p\times L_p\times 1}$. First, a block patch is crafted by filling a grid map of the same sizes as the state patch with zeroes. After this, we select left-bottom grids on the zero patch as much as the bottom area of the rotated block by r. Then, we complete the block patch P_{b^r} by filling the chosen grids with the height of the rotated block. Since we consider six block orientations, the block patch set P_b consists of six block patches.

Finally, the state patch set P_s and block patch set P_b are tiled to fit the dimensions of each other and concatenated along the last dimension (channel dimension) to produce six patch sets as shown in Figure 3(a). ViT processes these six patch sets as independent inputs, so they do not affect each other in the Q-value estimation.

B. Gradual Receptive Expansion

The critic network of GREViT adopts the ViT structure, utilizing converted patch sets to estimate Q-values that indicate the expected bin packing ratio. To employ ViT, we first encode patch sets using a patch encoder and add positional embedding vectors to identify patch locations.

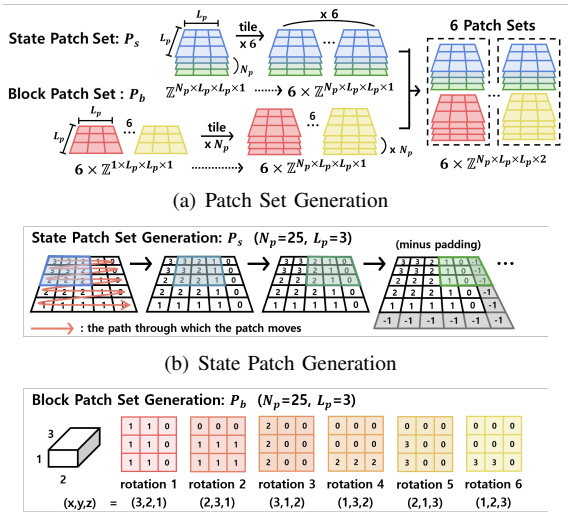

Fig. 3: Patch generation for ViT. (a) Six patch sets are generated from one loading state and block. (b) For state patches, the loading state is divided by moving the fixed-size receptive area with one stride. (c) Since six block rotations are considered in BPP, the block is converted into six patches.

Then, embedded patches are fed to the attention layer, which consists of the layer normalization and multi-head self-attention layer, and this process is repeated L times.

At each repetition, GREViT expands the receptive area for the Q-value estimation by masking non-neighboring patches. As shown in Figure 4, when predicting Q-value of position $(1, 0)$ on the bin, GREViT conducts self-attention mechanisms with patches corresponding to $(1, 0)$ and five adjacent positions. For the first attention, the receptive area for $(1, 0)$ covers 12 grids. After performing the attention, each embedded patch contains information about its neighboring patches. Therefore, at the second attention mechanism, the remaining four grids are added into the receptive area. To summarize, when inferring Q-value of a single position, the receptive area of GREViT initiates from the local loading area and gradually expands to a more global region with each attention. This process is referred to as **G**radual **R**eceptive **E**xpansion (GRE). Note that for GRE to encapsulate all grids on the bin, a sufficient number of attention layers should be used according to the bin size. Finally, the embedded vectors that have passed through all attention layers are projected to Q-values ranging from 0 to 1 via fully connected layers.

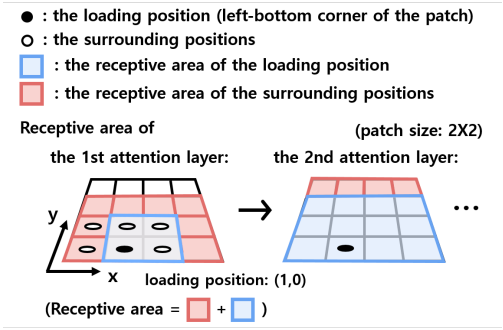

Fig. 4: Visualization for the GRE process. Through GRE, the receptive area for the Q-value estimation expands each time the attention layer repeats.

There are benefits of employing GRE. First, we focus more on the area close to the loading position, a crucial region when predicting Q-value. Second, the bin size less influences Q-learning by gradually expanding the receptive area rather than referring to the loading state all at once. As a result, GREViT displays a generalized performance for smaller bins even when trained with a larger bin. In Section VI-B, we have confirmed that GRE improves results with varied bin sizes.

V. Reinforcement Learning for Bin Packing

GREViT employs the discrete Tsallis actor-critic (TAC) algorithm for online 3D-BPP. Since TAC effectively learns multi-modal behaviors [17], it encourages the placement of blocks in a multitude of suitable loading positions. In this section, we implement a discrete version of TAC, which is originally introduced for the continuous action space. In addition, we enhance the sample efficiency and RL performance by executing BPP-specific data augmentation. Furthermore, we describe a packing planning method for handling cases where lookahead observed blocks exist (i.e., $k > 1$) using multi-modal actions learned by discrete TAC.

A. Discrete Tsallis Actor-Critic

Given the assumption of a discrete action space in our problem definition, the original TAC is modified to discrete TAC. The process is conducted similarly to [19], which applies SAC to a discrete action set. The discrete TAC structure has an actor network π_ϕ and a critic network Q_θ, and a model for estimating the state value is not required.

Due to the actions being finite, the output of the policy is presented as a probability value rather than a density function. This means that the output of the policy does not need to be a probability distribution over the action space displayed through mean and variance. Instead, direct probabilities for all actions are outputted. The actor network uses the state as input and produces the selection probability of each action. Therefore, the objective of the policy π_ϕ of discrete TAC is as follows:

$$J_\pi(\phi) = E_{s_t\sim\mathcal{D}}\big[\pi_\phi(s_t)^T[\alpha \ln_q(\pi_\phi(s_t)) - Q_\theta^{\pi_\phi}(s_t)]\big], \tag{3}$$

where $\mathcal{D}$ is a replay buffer, and α is a entropy temperature. In other words, in (2), the objective of the policy is estimated by sampling actions in a continuous space, but in (3), the objective is calculated using all actions. To prevent unstable loading in BPP, the probability of selecting an unstable position is amended to zero from the output of the policy.

Next, the critic network of discrete TAC also takes the state as input and outputs a vector of the same dimension as the number of actions. Each element of the vector indicates Q-value when the corresponding action is performed in the state. As a result, the state value Tsallis Bellman equation of discrete TAC is changed to:

$$V^\pi(s_t) = \pi(s_t)^T\big[Q^\pi(s_t) - \alpha \ln_q(\pi(s_t))\big]. \tag{4}$$

That is, in (1), state value V^π is estimated by sampling actions from the policy, but in (4), V^π is calculated directly using all state-action values and action probabilities. Therefore, a parameterized model for estimating V^π is not required. The other equations and objectives for discrete TAC are the same as TAC in (1) and (2).

B. Bin Packing Data Augmentation

Due to the properties of 3D-BPP, the loading state can be transformed into another feasible state via rotational changes. For instance, rotating the loading state 90 degrees generates a new loading state. Furthermore, since the loading state is represented as a 2D grid map, it is possible to extract a partial area to use as another loading state. For example, four 9×9 grid maps can be derived from a 10×10 grids, and the extracted grids with zero padding becomes a new loading state. In summary, we utilize rotation-type and extraction-type data augmentation for BPP. The extraction augmentation can be interpreted as a sample loaded on a smaller bin, which aids general learning related to the bin size.

Data augmentation for RL needs to be considered not only for loading states but also for actions. In rotation-type augmentation, the rotation index of the action is determined according to the block orientation after rotation. Simultaneously, new loading coordinates are obtained by applying the same rotational transformation. In extraction-type augmentation, while the rotation index is fixed, the coordinate indexes undergo the same translation transformation as the extracted loading state. Augmented data enhances the sample efficiency and robustness in relation to the bin size of RL, which is further discussed in Section VI-B.

C. Monte Carlo Tree Search Using Lookahead Blocks

When there are no lookahead blocks, the RL agent places the block at the grid with the highest Q-value. Contrarily, in scenarios where $k > 1$, it becomes possible to explore better loading positions by utilizing information about the next blocks. We propose a Monte Carlo tree search (MCTS)-based planning method with GREViT to investigate multi-step loading schemes. Note that BPP is a suitable task for MCTS application, given the characteristic of the discrete action set and deterministic state transitions.

Each node in MCTS presents a loading state while the action corresponds to the orientation and loading coordinates of the current block. MCTS iteratively executes loading schemes utilizing the tree policy. The tree policy selects the action with the highest upper confidence bounds value U, which is calculated as follows:

$$U(s_t, b_t, a_t) = R(s_t, b_t) + w_q Q(s_t, b_t, a_t) + w_c\sqrt{\frac{\log N(s_t, b_t)}{M(s_t, b_t, a_t)}},$$

where $N(s, b)$ is the number of visits for the node with the state (s,b), $M(s, b, a)$ is the number of selections for the

action a in the state (s,b), and w_q and w_c are constant. The first term, R, represents the accumulated rewards to reach the current node from the root node. The second term, Q, is the predicted Q-value of GREViT. The composite of R and Q assists in the exploitation of the loading scheme with weight w_q set to be less than 1, prioritizing R, the sum of certainly obtained rewards. The last term signifies the visit rate of the child node, prompting exploration of the loading scheme. Upon reaching the leaf node during a tree search, the value of the leaf node needs to be determined. This measurement is taken using a weighted sum of Q-value and previously collected rewards without performing rollout steps. By repeating the above process, MCTS previews various loading schemes to find an optimal loading plan.

VI. Experiment

A. Performance for Online 3D Bin Packing

In the experiments, we evaluate the performance of online 3D-BPP. We utilize a $(10, 10, 10)$ sized bin and blocks with lengths ranging from 2 to 5 and consider six block rotations. Also, we use the random, CUT-1 and CUT-2 datasets as block sequences. The random sequence is made up of blocks whose side lengths are randomly determined. Conversely, the block sequence in the CUT-1 and CUT-2 datasets can fully pack the bin. While CUT-1 comprises a sequence sorted in low order according to the z-coordinate of the left-bottom vertex of the block, CUT-2 is a sequence generated by randomly selecting blocks that can be stably placed without height considerations. These block sequences are commonly used in existing BPP studies [12]–[15].

We use the packing ratio as a metric for BPP performance. The packing ratio denotes the proportion of the total volume stably occupied by loaded blocks. If there is not adequate space for the current block or if the loading structure becomes unstable, an episode is concluded. The stability of the loading state is decided using stability conditions proposed by [12]. To visualize the BPP task, we developed a 3D-BPP environment with a UR10 robot based on bin stacking Cortex, which is a robotic system of NVIDIA Isaac Sim [20] as illustrated in Figure 1. The attached video shows demonstrating online 3D-BPP in the Isaac Sim environment.

In the experiment, we use heuristic methods and learning-based methods to compare performance with GREViT. As heuristics, we implement various methods such as CP [5], EP [8], EMS [9], HM [10], and DBL [11]. On the other hand, we evaluate learning-based algorithms, [12] and PCT [15], by re-training with the same problem setting in this paper. Each RL agent is trained for one million steps and evaluated with 500 episodes using the same block sequences.

The bin packing results are shown in Table I. First, heuristic algorithms generally demonstrate poor performance. The HM algorithm shows the highest packing ratio among heuristics. However, all learning-based methods outperform the HM algorithm under the same block sequence. Out of all box sequences, GREViT achieves the best packing ratio compared to the existing state-of-the-art baselines.

B. General Performance for Smaller Bins

We conduct an ablation study for GREViT with three distinct configurations. First, the Base algorithm is a discrete TAC employing standard ViT without GRE and data augmentation. Second, GRE$*$ utilizes the GRE technique with Base. The last algorithm, GREViT, incorporates both GRE and data augmentation into Base. To verify the generalized performance of GREViT across various bin sizes, we measure the packing ratio for smaller bins using the same model trained within a $(10, 10, 10)$ sized bin. In addition, for performance comparisons, we deploy heuristic baselines that demonstrate optimal performance irrespective of bin sizes.

Algorithm	Random	CUT-1	CUT-2
CP [5]	0.469	0.501	0.501
EP [8]	0.510	0.532	0.541
EMS [9]	0.462	0.496	0.503
HM [10]	0.611	0.636	0.649
DBL [11]	0.560	0.593	0.603
Zhao et al. [12]	0.662 $\pm$ 0.024	0.675 $\pm$ 0.012	0.670 $\pm$ 0.019
PCT [15]	0.687 $\pm$ 0.003	0.703 $\pm$ 0.001	0.679 $\pm$ 0.005
GREViT	**0.704 $\pm$ 0.004**	**0.731 $\pm$ 0.013**	**0.730 $\pm$ 0.004**

TABLE I: Online 3D-BPP results using a $(10, 10, 10)$ sized bin. All results are an average of 500 episodes for fixed box sequences. Learning-based methods use three random seeds.

(a) Average packing ratio with the 10×10 bin

Dataset	CP [5]	EP [8]	EMS [9]	HM [10]	DBL [11]	Base	GRE$*$	GREViT
Random	0.469	0.510	0.462	0.611	0.560	0.698	0.682	**0.704**
CUT-1	0.501	0.532	0.496	0.636	0.593	0.679	0.717	**0.731**
CUT-2	0.501	0.541	0.503	0.649	0.603	0.672	0.714	**0.730**

(b) Average packing ratio with the 9×9 bin

Dataset	CP [5]	EP [8]	EMS [9]	HM [10]	DBL [11]	Base	GRE$*$	GREViT
Random	0.464	0.516	0.466	0.609	0.583	0.582	0.636	**0.698**
CUT-1	0.491	0.526	0.487	0.640	0.602	0.587	0.652	**0.719**
CUT-2	0.484	0.530	0.490	0.637	0.610	0.608	0.650	**0.707**

(c) Average packing ratio with the 8×8 bin

Dataset	CP [5]	EP [8]	EMS [9]	HM [10]	DBL [11]	Base	GRE$*$	GREViT
Random	0.462	0.508	0.458	0.615	0.571	0.610	0.630	**0.647**
CUT-1	0.502	0.539	0.493	0.651	0.623	0.608	0.690	**0.721**
CUT-2	0.499	0.547	0.496	0.640	0.621	0.625	0.692	**0.713**

(d) Average packing ratio with the 7×7 bin

Dataset	CP [5]	EP [8]	EMS [9]	HM [10]	DBL [11]	Base	GRE$*$	GREViT
Random	0.496	0.522	0.498	0.596	0.617	0.579	0.607	**0.639**
CUT-1	0.528	0.581	0.539	0.656	0.621	0.603	0.711	**0.728**
CUT-2	0.524	0.578	0.544	0.656	0.619	0.605	0.698	**0.705**

TABLE II: Packing results in bins of various sizes. All results are an average of 500 episodes for fixed box sequences. Learning-based methods use three random seeds.

Table II shows the packing results for bins of diverse sizes. First, GREViT outperforms all heuristics for all bin sizes and datasets. It means that GREViT overcomes the limitations of learning-based methods, whose performance is typically reliant on the bin sizes used for training. Next, we also analyze the ablation study results for GREViT. The GRE$*$ method achieves higher performance than Base in most instances. This result indicates that GRE not only improves the performance of bin packing but also induces robust learning with regard to bin sizes. Additionally, the performance of GREViT surpasses GRE$*$, suggesting that data augmentation aids learning about the multi-modal loading positions of BPP.

C. Loading Planning Using Lookahead Items

In this experiment, we address lookahead blocks by exploring multi-step loading schemes via MCTS as described in Section V-C. We aim to verify how much more GREViT enhances the packing ratio by using information about lookahead blocks. A GREViT agent trained with a $(10, 10, 10)$

Num. of Lookahead Items	Random	CUT-1	CUT-2
1	0.704 ± 0.004	0.731 ± 0.013	0.730 ± 0.004
3	0.725 ± 0.012	0.775 ± 0.006	0.762 ± 0.003
5	0.735 ± 0.006	0.788 ± 0.006	0.780 ± 0.004
8	**0.736 ± 0.011**	**0.789 ± 0.005**	**0.784 ± 0.006**

TABLE III: Multi-step loading planning results with various numbers of lookahead blocks using a $(10, 10, 10)$ sized bin. The results are an average of 500 episodes for fixed box sequences with three random seeds.

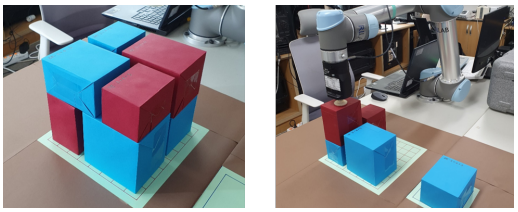

Fig. 5: Real robot experiment. (Left) There are eight blocks in each sequence. (Right) We experiment online 3D-BPP using a UR5 robot with a Robotiq Epick vacuum gripper.

sized bin is used.

The results of the multi-step packing are depicted in Table III. The experimental results are an average of 500 BPP episodes with three random seeds. For all block sequences, GREViT shows a progressively enhanced packing ratio as the number of lookahead items increases. Finally, GREViT improves the packing ratio by 4.8 percentage points on average using eight lookahead blocks. It indicates that the MCTS-based planning with GREViT can leverage the information about lookahead items, and GREViT enables the proposal of a variety of suitable loading locations. In particular, higher performance improvements have been made on CUT-1 and CUT-2 sequences, which are datasets that completely fill the bin, so the multi-step planning is applied more effectively.

D. Real Robot Experiment

To verify the practical applicability of GREViT, we execute a 3D-BPP experiment using a real UR5 robot with a Robotiq Epick vacuum gripper as shown in Figure 5. We evaluate the performance using 20 block sequences, each comprising eight blocks with side lengths ranging from 3 to 5 that perfectly accommodate an $(8, 8, 8)$ sized bin. For the convenience of the robot behavior, we consider only two block orientations. Also, we handle BPP scenarios where there is a no lookahead block, i.e., $k = 1$. Since the size of the block can be simply estimated using RGB-D images, we assume a size estimation module exists. A model trained using the same BPP setting within a simulation environment is employed. As a result, GREViT achieves an average packing rate of 85.4%. The model trained in a simulator is effectively implemented in real-world scenarios, which corroborates that the assumptions for 3D-BPP outlined in this paper are suitable for practical tasks. The robot demonstrations can be found in the attached video.

VII. Conclusion

In this paper, we propose a novel RL algorithm, GREViT, developed to address online 3D-BPP. GREViT employs a discrete TAC algorithm adapted for 3D-BPP and leverages the ViT-based critic network to estimate the expected packing ratio for each loading position. By proposing the GRE technique and data augmentation, GREViT enhances bin packing performance and facilitates robust training across a range of bin sizes. In conclusion, we establish that GREViT can be implemented with a robot to demonstrate bin packing within both a simulator and a real-world environment.

References

[1] J. O. Berkey and P. Y. Wang, "Two-dimensional finite bin-packing algorithms," *Journal of the operational research society*, vol. 38, pp. 423–429, 1987.

[2] A. Lodi, S. Martello, and D. Vigo, "Recent advances on two-dimensional bin packing problems," *Discrete Applied Mathematics*, vol. 123, no. 1-3, pp. 379–396, 2002.

[3] A. E. F. Muritiba, M. Iori, E. Malaguti, and P. Toth, "Algorithms for the bin packing problem with conflicts," *Informs Journal on computing*, vol. 22, no. 3, pp. 401–415, 2010.

[4] M. Delorme, M. Iori, and S. Martello, "Bin packing and cutting stock problems: Mathematical models and exact algorithms," *European Journal of Operational Research*, vol. 255, no. 1, pp. 1–20, 2016.

[5] S. Martello, D. Pisinger, and D. Vigo, "The three-dimensional bin packing problem," *Operations research*, vol. 48, no. 2, pp. 256–267, 2000.

[6] H. I. Christensen, A. Khan, S. Pokutta, and P. Tetali, "Approximation and online algorithms for multidimensional bin packing: A survey," *Computer Science Review*, vol. 24, pp. 63–79, 2017.

[7] J. de Castro Silva, N. Soma, and N. Maculan, "A greedy search for the three-dimensional bin packing problem: the packing static stability case," *International Transactions in Operational Research*, vol. 10, no. 2, pp. 141–153, 2003.

[8] T. G. Crainic, G. Perboli, and R. Tadei, "Extreme point-based heuristics for three-dimensional bin packing," *Informs Journal on computing*, vol. 20, no. 3, pp. 368–384, 2008.

[9] C. T. Ha, T. T. Nguyen, L. T. Bui, and R. Wang, "An online packing heuristic for the three-dimensional container loading problem in dynamic environments and the physical internet," in *Applications of Evolutionary Computation: 20th European Conference, EvoApplications 2017, Amsterdam, The Netherlands, April 19-21, 2017, Proceedings, Part II 20*. Springer, 2017, pp. 140–155.

[10] F. Wang and K. Hauser, "Stable bin packing of non-convex 3d objects with a robot manipulator," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8698–8704.

[11] K. Karabulut and M. M. İnceoğlu, "A hybrid genetic algorithm for packing in 3d with deepest bottom left with fill method," in *International Conference on Advances in Information Systems*. Springer, 2004, pp. 441–450.

[12] 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.

[13] A. V. Puche and S. Lee, "Online 3d bin packing reinforcement learning solution with buffer," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 8902–8909.

[14] 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*, 2023.

[15] H. Zhao, Y. Yu, and K. Xu, "Learning efficient online 3d bin packing on packing configuration trees," in *International Conference on Learning Representations*, 2021.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[17] K. Lee, S. Kim, S. Lim, S. Choi, M. Hong, J. I. Kim, Y.-L. Park, and S. Oh, "Generalized tsallis entropy reinforcement learning and its application to soft mobile robots." in *Robotics: Science and Systems*, vol. 16, 2020, pp. 1–10.

[18] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning, ICML*, Jul 2018, pp. 1856–1865.

[19] P. Christodoulou, "Soft actor-critic for discrete action settings," *arXiv preprint arXiv:1910.07207*, 2019.

[20] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.