Agent-based Reinforcement Learning to Increase Housing Density in London

Lewen Zhang[1], Robin Song[1], Wenshuo Zhang[1]

[1] University College London, London, United Kingdom

Keywords: Agent-based, Depth-First Search, Housing Density, Land Use, London, Reinforcement Learning, Sunlight, Urban Planning, Unity

1 Introduction

London, a metropolitan city and one of the world's most significant financial, cultural, and educational hubs, has continually attracted domestic and international migrants in the contemporary era. Consequently, its housing prices rose 130 percent between January 2005 and January 2023. With an affordability Ratio below three being considered cost-effective, such a ratio in London denotes a span of housing expenditure from 7 to 13 times the average salary, which established significant housing unaffordability amongst Londoners (Treadwell, 2024).

Yet, the city also suffers from a low delivery rate of housing construction, meeting only ⅓ of its original expansion plan in 2023. The government has published corresponding policies in response to the housing shortage, such as the Permitted Development Rights (PDR) for commercial-to-residential conversions in locations once deemed suitable only for non-residential land use. However, most estates constructed under this policy are smaller and more expensive (by sqm) than average, with less accessible green spaces and higher levels of air pollution (Chng et al., 2023).

Another obstructive factor against London's housing expansion is the Metropolitan Green Belt (MGB), which protects about one-fifth of its occupiable land. Treadwell (2024) claimed that MGB leads to the inflation of the actual housing price by four and a half times and 15 times for the land price.

With these concerns, we came up with our research interest: How can we streamline the housing design workflow for a metropolitan city like London, in a computational way? As such, this project aims to automate the process of urban housing densification for stakeholders, delivering a more intelligent and systematic approach to relieve the urban housing crisis and prototyping the practice in London.

2 Literature Review

2.1 Existing Urban Planning Tool

There are several urban design tools available for a metropolitan city like London. Prism (2025), an integrated open-source online design tool for urban planners developed by Bryden Wood and sponsored by the Mayor of London, aims to enhance the design process for Precision Manufactured Housing and Modern Methods of Construction. Its advantages include automation through the use of open-source and real-time data specific to London, the ability to facilitate precise online housing design activities, and the provision of dashboard visuals that summarize the status of current housing availability and distribution. However, the tool has limitations, such

as the lack of support for computational design features and outdated dashboard data, with information that is not current and dated before 2018.

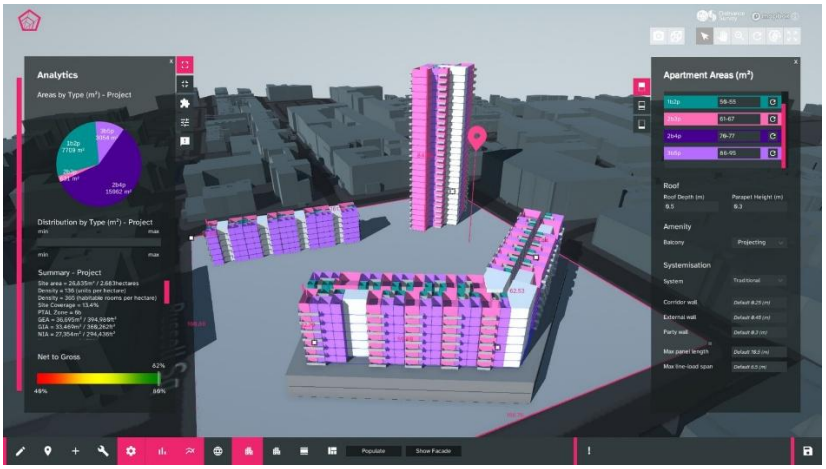

Fig. 1A. Prism by Bryden Wood

2.2 Generative Design Example: Townscaper

We also look into 3D design generation tools, such as Townscaper by Stålberg (2021), which automates building configuration in a gaming environment. The interactive prototype application enables users to design freely with just a few clicks. Utilizing model synthesis through Wave Function Collapse for procedural content generation, it features a non-orthogonal base map that allows for designs from various heights and angles, facilitating the generation of features for the corresponding placement of blocks. Its advantages are a minimal user learning curve and the automation of site-specific designs with identifiable block features. However, the tool is constrained by the shape of the base map, and the outcomes are not transferable to real-life contexts.

Fig. 1B. Townscaper by Stålberg

2.3 Algorithms

Kimm (2024) develops a comprehensive classification of AI tools in architectural design and planning. The technologies are organized into categories of tools, techniques, and methods according to their roles in the design workflow of practitioners. It summarizes the current AI tools, models, and prototypes while providing a detailed breakdown of the agent-based simulation technique. The study points out that agent-based modeling produces complex systems through autonomous agent interactions, and learning agents enhance performance from learning experience, making it suitable for real-time design with freedom.

Qian et al. (2023) introduce a Consensus-based Multi-Agent Reinforcement Learning framework to facilitate real-world land use readjustment, thereby enhancing participatory urban planning and automating the complex, manual, experience-driven alteration planning process. This approach flattens the planning decision-making process by employing cooperatively rewarded agents, with top-down agents mimicking the behaviors of planners and developers. At the same time, bottom-up groups simulate behaviors emerging from different economic bracket layers, approaching actual design workflow systematically. Additionally, the framework includes real-life applications to validate model performance. Similarly, Yu et al. (2023) establish a Deep Reinforcement Learning (DRL) model for multifunctional urban planning. By comparing the generative results of the model with manual designs created by human experts, the research concludes that the agent-based DRL model outperforms traditional planning workflows. The performance evaluation highlights the advantages of DRL over human urban planners, particularly in implementing smart fill techniques for optimizing functioning zones.

Within the domain of agent-based reinforcement learning, research by Simoniti (2022) into reinforcement learning algorithms focuses on developing the Proximal Policy Optimization (PPO) technique, which employs a clipped surrogate objective function that quantifies and maps the advantage function to assess an agent's actions under a given policy. This approach effectively stabilizes the learning process's efficiency by constraining the gradient of policy updates, making it widely applicable across various training scenarios, particularly for single-agent reinforcement learning. PPO balances sample complexity, simplicity, and wall time, contributing to its effectiveness and popularity in the field.

With interest in urban zoning and classification, Lagonigro's article discusses quadtree mapping as a method for visualizing local population data while preserving accuracy and maintaining privacy (2017). It compares this approach to a traditional model, AZTools, which utilizes a bottom-up methodology to aggregate data points into uniform areas based on defined constraints. The study concludes that the hierarchical squared structure of the quadtree offers more comparable spatial and temporal statistics. Key advantages include effective data visualization on an urban scale and a layered structure that enhances the depth of analysis.

We also explore the Depth-First Search (DFS) Algorithm for identical building labeling tasks. Berman (2024) highlights its role as a fundamental and versatile tool for graph pathfinding and cycle detection. The search mechanics are based on a recursive binary tree structure, allowing the algorithm to traverse branches before backtracking and enabling it to detect cycles to the greatest depth possible. Key features of DFS include depth-sensitive searching and its effectiveness as a data segmentation and labeling algorithm.

3 Methodology

3.1 Data simplification

In order to make the real-world constraints into information that can be recognized by the reinforcement learning model, we abstracted the present information and behaviors, extracted their features, simplified the information into 3D-maps and 2D-maps to be stored in arrays, and simplified the densification process into two behaviors: adding in the search for an open space and adding on a roof.

3.1.1 3D-Map

To better analysis and summarize the urban information, we collected the urban data from OSM in London. (see Fig. 3A). It shows different categories of land use and building types.

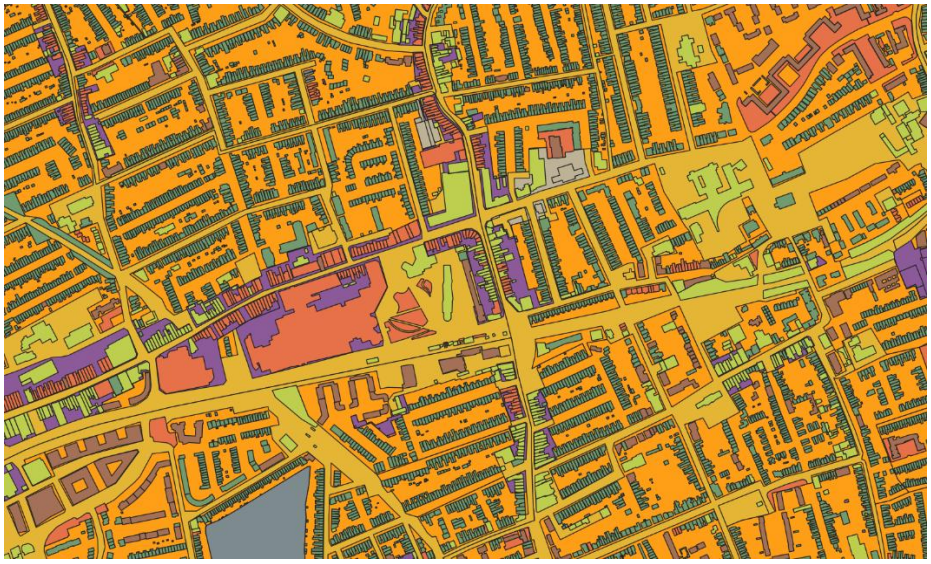

Fig. 3A. A figure shows the different land use and building type in London

To simplify the problems, we extracted 4 most important land use and 2 building types into our study as 6 cell types, which include residential area, street, commercial area, green area, residential building and commercial building. (see Fig 3B)

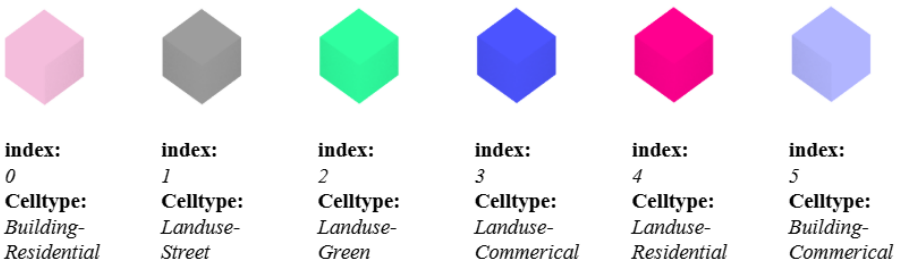

Fig. 3B. Category of 6 different cell types.

Accordingly, with these 6 cell types we transformed the urban information into voxel and storage the cell type information in a 3D Array. (see Fig.3C)

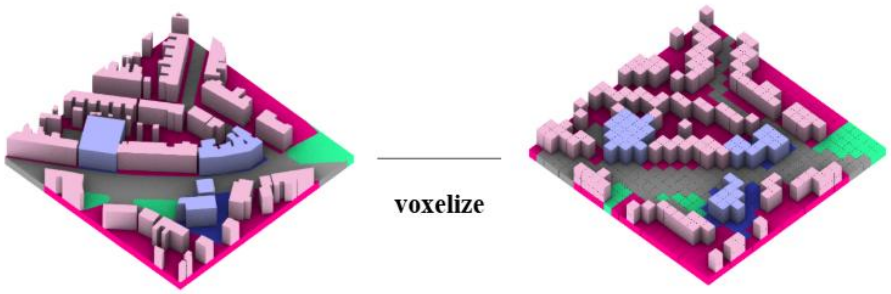

Fig. 3C. Voxelization of the urban plot.

3.1.2 2D-Map

2D-map is used as a constrain information in this study to support the training process. We transformed four important city figure into 2D-map and storage in 2D Array which including residential building index, façade solar exposure, ground solar radiation and empty space.

- **Residential Building index**

The residential building index is used to give a unique index to each of the residential building for further training process. (see Fig 3D). In this 2D-map, residential building cells are recognized as integers larger than zero while other cells area recognized as zero.

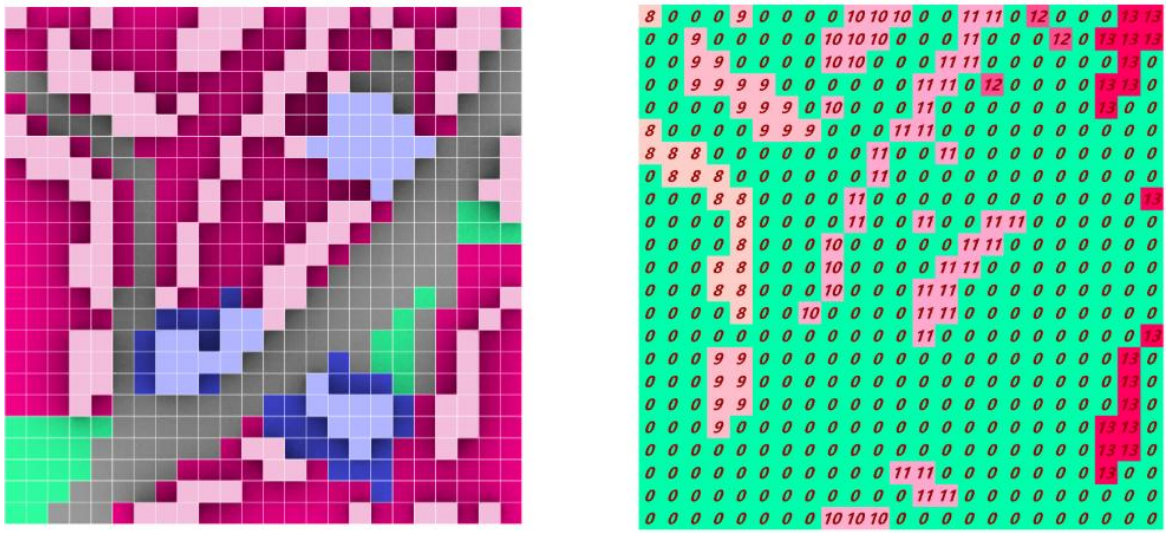

Fig. 3D. left image is the top projection of the city plot right image shows the residential building with unique figure.

- **Façade solar exposure**

In this study, we simplify the façade solar exposure problems into distance problem. If there is enough space between residential buildings, it will assure the adequate sunlight. Hence, we use a distance map to shows the area which can offers more façade solar exposure to the buildings. **(see Fig 3E)**. In this map, 0 represent the buildings, 2 represent 1 cell away from the buildings 4 represent 2 cells away from the building, 6 represent else space.

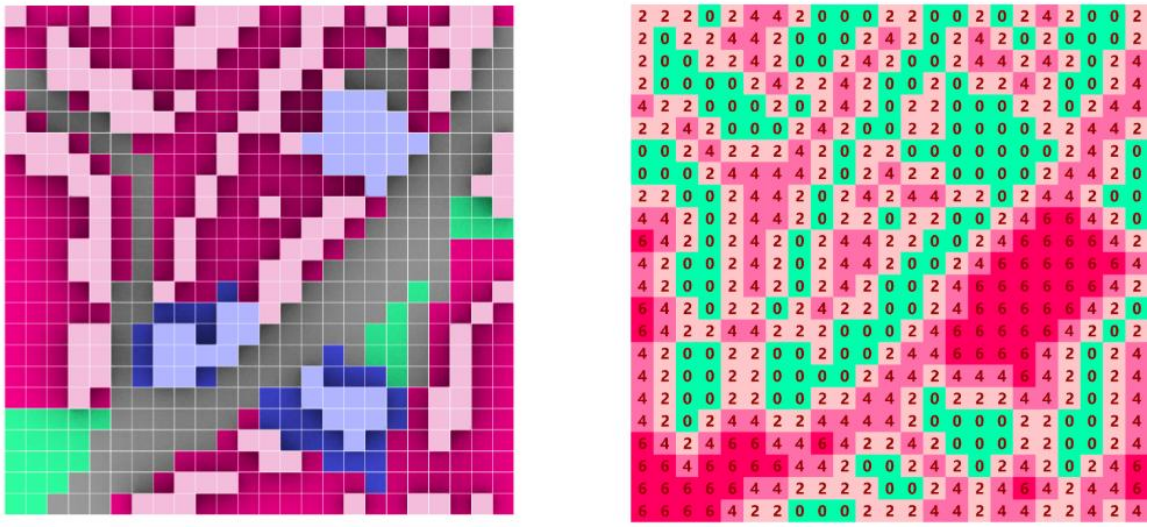

Fig. 3E. left image is the top projection of the city plot right image shows the distance to the buildings

- **Ground solar radiation**

For ground solar radiation, we calculate daylight hours on the winter solstice, when daylight hours are shortest, to ensure that the community has enough daylight hours. **(see Fig 3F).** In this map, -1 represent the buildings, 1-7 represent the radiation hour in the community in winter solstice.

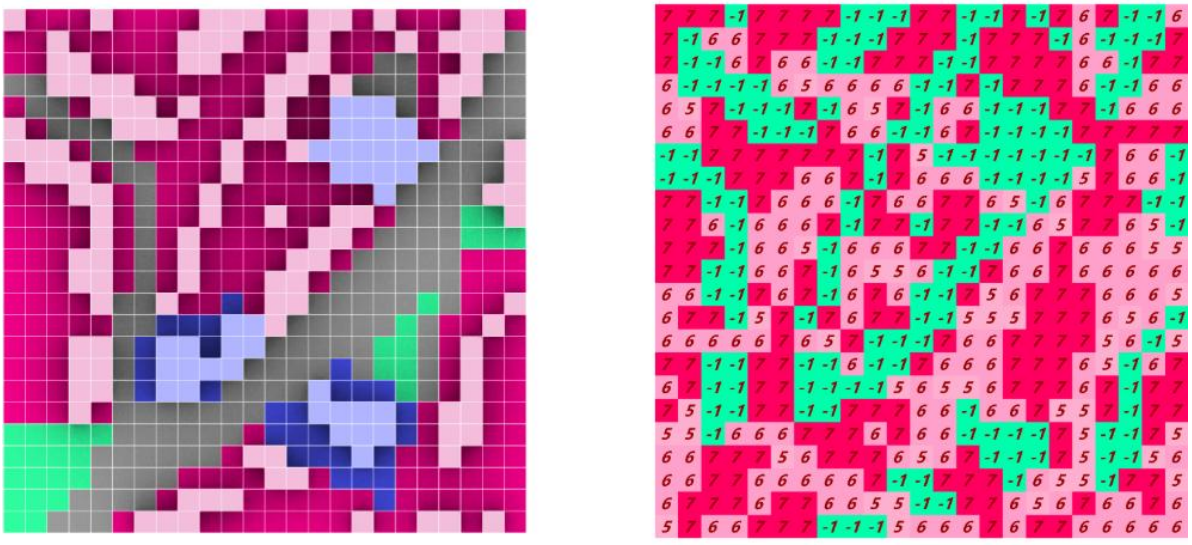

Fig. 3F. left image is the top projection of the city plot right image shows the ground solar radiation map.

- **Empty space**

For empty space, our goal is to spot the empty space with largest area. Hence, we introduced quadtree searching method into our study.

First, we preprocess the city plot into a 2D map which shows every empty area in the residential land use **(see Fig 3G)**

Fig. 3G. right image shows the preprocessed empty space map.

Second, we input the preprocessed maps into a quadtree to identify regions with a density lower than 0.2. Within these selected regions, we prioritize cells at lower levels and include their surrounding cells in our selection list to minimize the risk of missing relevant areas. **(see Fig 3H).** In this two quadtree maps, light pink represent the cell with the density lower than 0.2, dark pink represent the cell with lower levels.

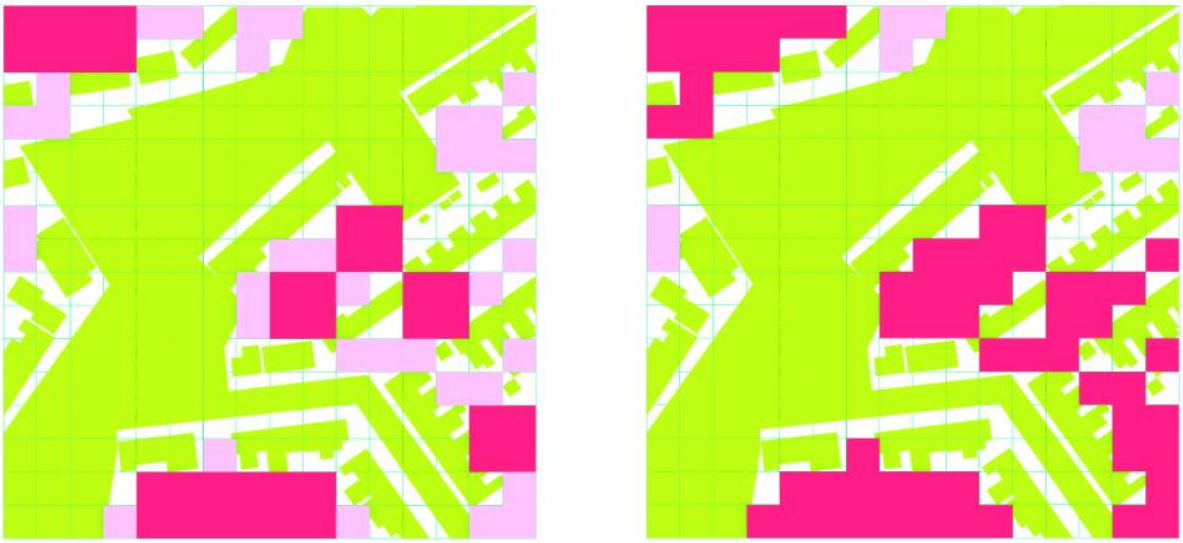

Fig. 3H. The figure shows the selecting process through quadtree map.

Finally, we transformed this quadtree map into 2D-map. **(see Fig 3I).**

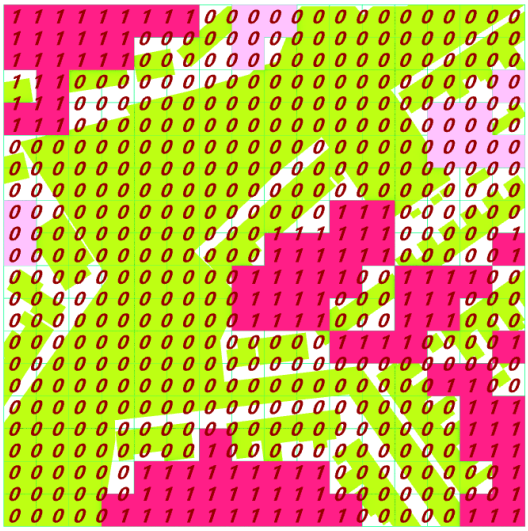

Fig. 3I. The figure shows the quadtree 2D map.

3.1.3 Behaviors

In this study, the framework simulates two real-world densification strategies: adding extensions to existing buildings or constructing new ones on vacant land.

3.2 Reinforcement Learning

In our procedural building generation system, we implement a multi-agent reinforcement learning approach with specialized agent types for different aspects of construction.

3.2.1 Architecture

Our system employs two distinct agent types:

1. Ground Agent: Responsible for the primary building construction and establishing foundations.
2. Roof Agent: Modified from the Ground Agent, specializing in roof structure formation with adjusted reward parameters.

Both agents operate within a three-dimensional voxel-based environment, where each voxel represents a potential building element. The agents make sequential decisions based on current observations to maximize cumulative rewards aligned with architectural constraints.

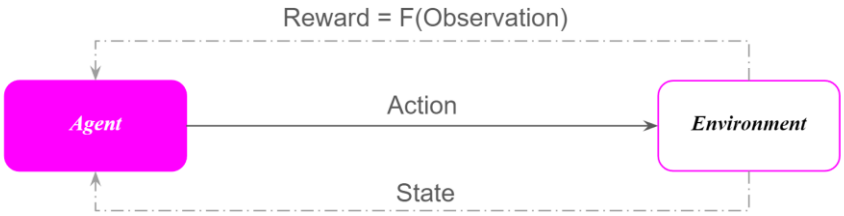

Fig. 3J. Image shows a simple architecture of reinforcement learning.

3.2.2 Actions

Agents have two primary action capabilities:

1. Movement: Agents can navigate in six directions within the 3D environment (forward, backward, left, right, up, down).
2. Occupation: At each position, agents decide whether to occupy the current voxel with building material.

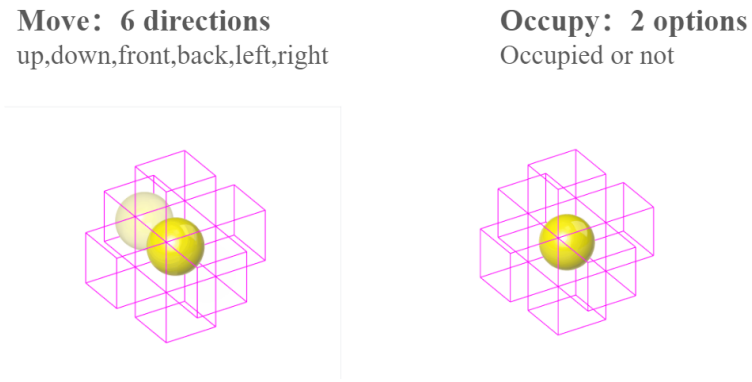

Fig. 3K. Image shows two branches of action.

These binary decisions (move direction + occupy/not occupy) form the action space that agents learn to optimize through reinforcement.

3.2.3 Observations

Agents receive the following observations from the environment:

1. Coordinate Observations: Relative position within the construction plot.
2. Neighborhood Observations:
    - Lower layer: 25 surrounding positions
    - Middle layer: 8 surrounding cubes
    - Upper layer: 1 position directly above
3. Cell Type Observations: Material types of neighboring cells.
4. Building Metrics:

- Current house size (occupied cubes)
- Maximum house size (target constraint)
- Normalized house size ratio (0-1 range)

5. Spatial Measurements:

- Current mass diagonal
- Maximum diagonal (diagonal of the plot)
- Normalized diagonal ratio

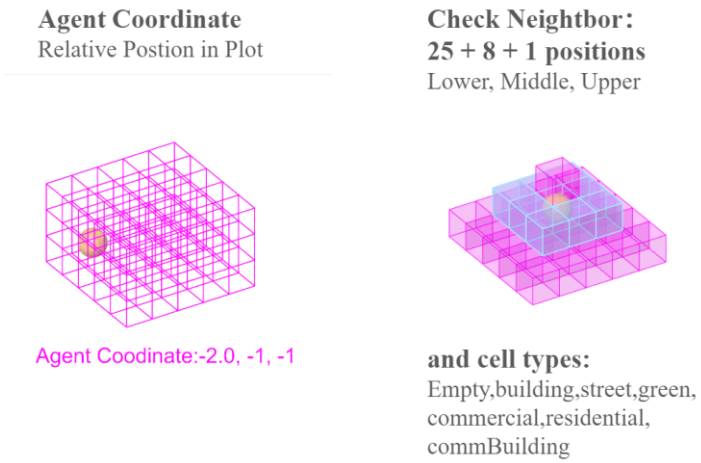

Fig. 3L. Image shows the checking coordinate and neighbors.

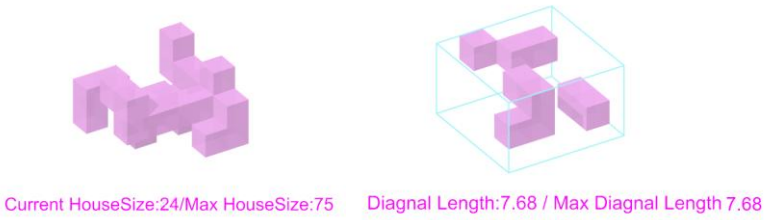

Fig. 3M. Image shows the spatial measurements.

3.2.4 Rewards

The Ground Agent's behavior is guided by seven reward components:

1. Neighborhood Reward: Awarded when agents maintain proximity with neighbors.
2. Stable Base Reward: Higher rewards for positions directly above existing structures; penalties otherwise.
3. Continuity Reward: Encourages continuous structures by rewarding adjacency to existing building elements.
4. Residential Area Reward: Incentivizes building within designated zones.
5. Boundary Reward: Distance-based calculations to maintain appropriate plot boundaries.
6. Ground Stability Reward: Promotes stable foundation at ground level.
7. Compactness Reward: Calculated as (volume ratio - diagonal ratio) to encourage efficient space utilization.

Environmental Constraint: Solar Index Reward evaluates sunlight exposure using ray-casting based on London winter solstice light vectors.

The Roof Agent uses modified reward parameters including:

- "Inside Residential" replaced with "On Existing Building"
- "Away from Building" changed to "Close to Building"
- Additional reward for maintaining single-building structures

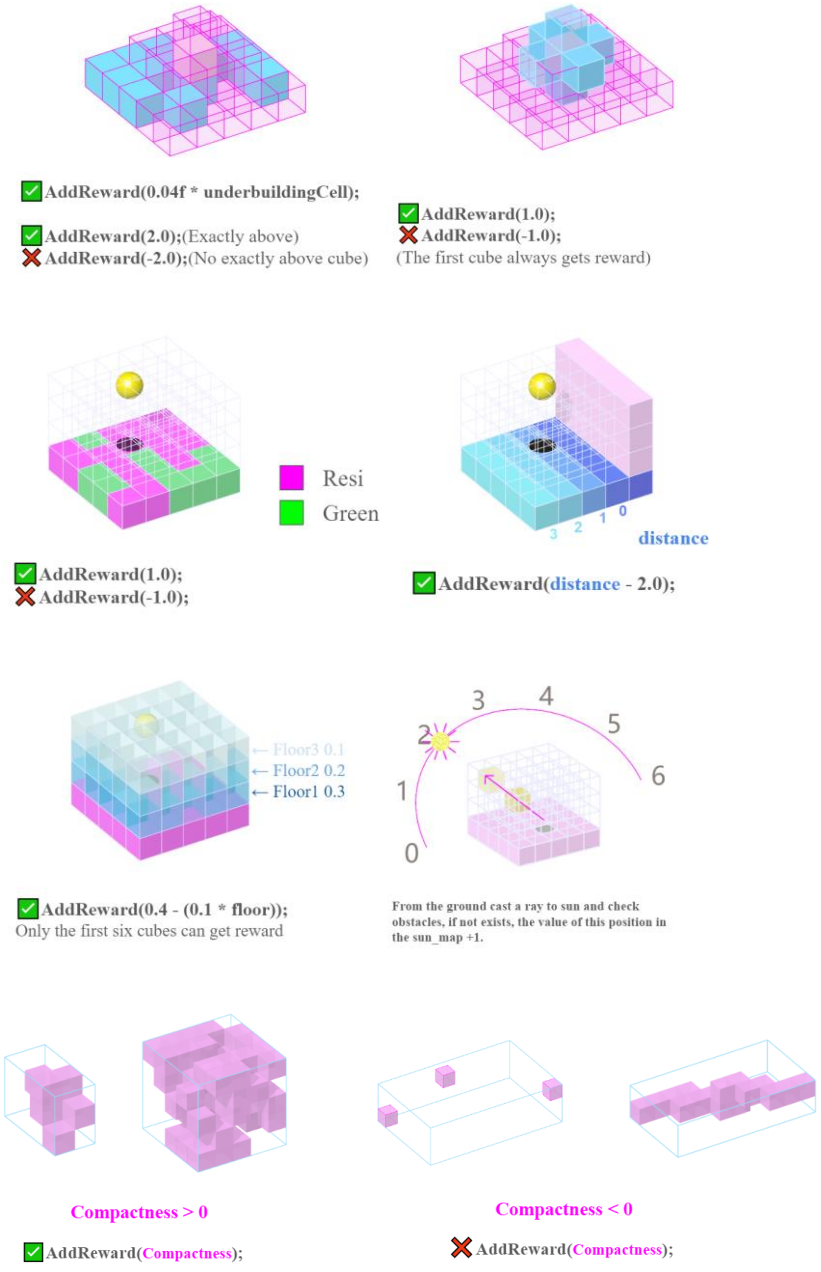

Fig. 3N. These images show the different types of reward.

3.2.5 Episode

Each training episode follows a structured process:

Initialization:

- Plot environment setup
- Agent position pre-processing using quad tree mapping
- Agent instantiate

Progression:

- Sequential agent actions and environment updates
- Reward accumulation
- State observation

Termination Conditions:

- Target Achievement: Building reaches specified target size
- Agent Entrapment: Agent has no valid movement options

This episodic structure enables agents to learn effective building strategies through repeated interaction with the environment.

4 Results

4.1 Prototype Result

To validate our model setting, we trained our agent in three basic prototypes.

- **Prototype-1: Ground agent on orthogonal building plot**

The objective of the agent in prototype-1 is to find the empty space in the 20*20 plot, avoiding occupied cells on the green area and keep distance with the existing building. The results are shown in **Fig.4A-1**. Cumulated reward is shown in **Fig.4A-2.** The outcomes demonstrate the effectiveness of the building placement strategy.

Fig. 4A-1. The figure shows the model performance before and after training.

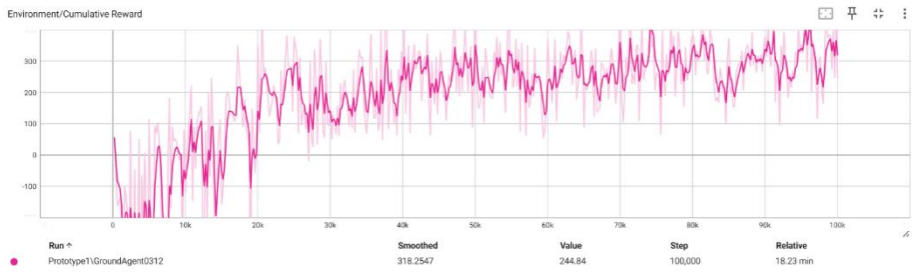

Fig. 4A-2. The figure shows the cumulated reward through the training process.

- **Prototype-2: Ground agent on non-orthogonal building plot**

The objective of the agent in prototype-2 is to find out if the model works in protoype-1 will works as well in the non- orthogonal plot. The results are shown in **Fig.4B-1**. Cumulated reward is shown in **Fig.4B-2.** The outcomes demonstrate the effectiveness.

Fig. 4B-1. The figure shows the model performance before and after training.

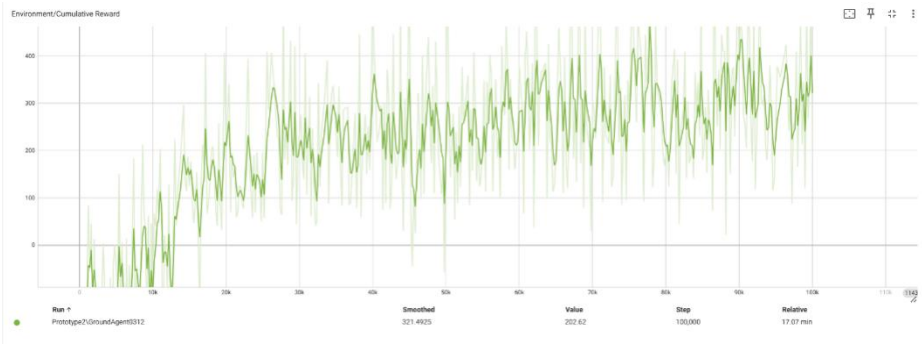

Fig. 4B-2. The figure shows the cumulated reward through the training process.

- **Prototype-3: Roof agent on orthogonal building plot**

5 Results(500-700)

5.1 Prototype Result

To validate our model setting, we trained our agent in three basic prototypes.

- **Prototype-1: Ground agent on orthogonal building plot**

The objective of the agent in prototype-1 is to find the empty space in the 20*20 plot, avoiding occupied cells on the green area and keep distance with the existing building. The results are shown in **Fig.4A-1**. Cumulated reward is shown in **Fig.4A-2.** The outcomes demonstrate the effectiveness of the building placement strategy.

Fig. 4A-1. The figure shows the model performance before and after training.

Fig. 4A-2. The figure shows the cumulated reward through the training process.

- **Prototype-2: Ground agent on non-orthogonal building plot**

The objective of the agent in prototype-2 is to find out if the model works in protoype-1 will works as well in the non- orthogonal plot. The results are shown in **Fig.4B-1**. Cumulated reward is shown in **Fig.4B-2.** The outcomes demonstrate the effectiveness.

Fig. 4B-1. The figure shows the model performance before and after training.

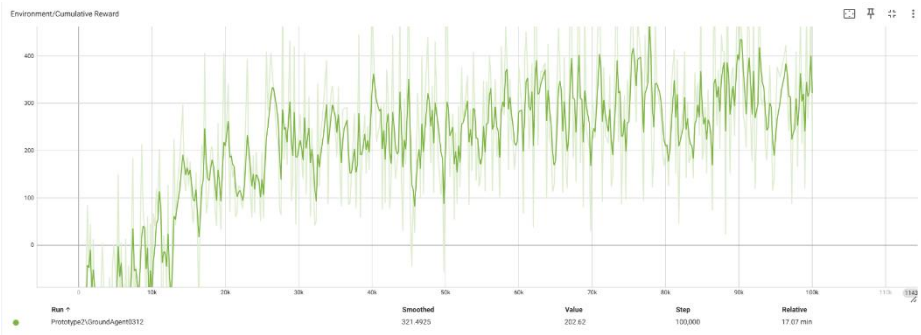

Fig. 4B-2. The figure shows the cumulated reward through the training process.

- **Prototype-3: Roof agent on orthogonal building plot**

Fig. 4A-3. The figure shows the model performance before and after training.

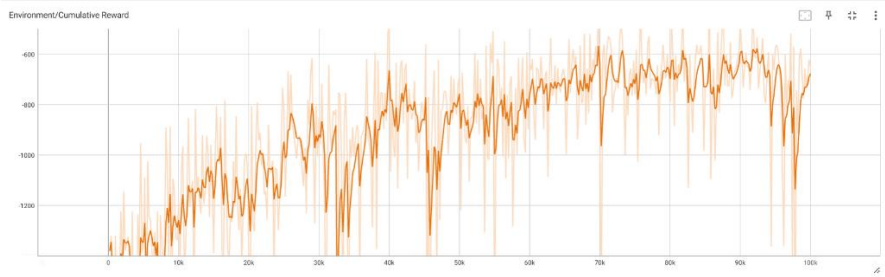

Fig. 4B-3. The figure shows the cumulated reward through the training process.

5.2 Application Robin

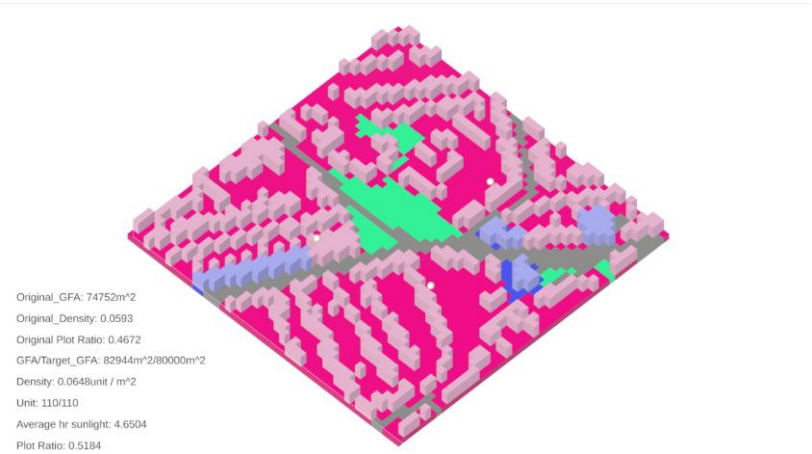

Fig. 4C. The figure shows the final result on real site.

Based on the provided data, the project has achieved significant value growth. The Gross Floor Area (GFA) increased by 8,192 square meters, the number of units expanded by 110 voxels, while the plot ratio improved by 11% and density grew by 9%. This comprehensive enhancement across all metrics reflects optimized spatial utilization efficiency, achieving both building area and functional unit expansion while maintaining reasonable density growth. This balanced development not only increases the overall value of the project but also establishes a solid foundation for future sustainable development.

6 Discussion

Throughout the study, we identified the housing crisis in London and proposed a new way to densify the residential area in the initial design phase. In this study, we envisioned a method to simplify the complicated real world constrain into quantitative data. We successfully built up a multi-agents reinforcement learning framework to automatically spot the available space in the given city plot and densify the plot under the constrain of land using, sunlight, communities continuity etc. Our innovative approach offers a powerful reference tool for urban densification processes.

6.1 Strength

- **Innovative Methodology:**
  The integration of agent-based reinforcement learning (RL) into urban planning offers a novel approach to addressing London's housing crisis.
- **Multidimensional Data Integration:**
  The digitalization process incorporates diverse factors (e.g., land use, solar exposure, green spaces, neighborhood continuity, compactness). This holistic approach ensures realistic constraints are modeled, enhancing the practical relevance of the solutions.
- **Two ways of densification'**
  The framework differentiates between two agent behaviors. Ground-level agents focus on densifying vacant residential spaces, while rooftop agents target the expansion of existing buildings. These two approaches reflect real-world densification strategies: new constructions and rooftop additions.
- **Scenario-Specific Validation:**
  Prototype training was conducted in varied environments and real-world testing in London's Waltham Forest demonstrated measurable improvements (e.g., increased Gross Floor Area/GFA).

6.2 Limitation

- **Limitations of Generalizability:**
  The training results are constrained by the predefined observation space, making them inapplicable outside the designated plot.
- **Simplified Assumptions:**
  The extraction of information into 2D and 3D maps may overlook certain spatial complexities inherent to real-world 3D environments.
  The Quadtree method, due to its inherent mechanism, may fail to capture some effective plots, potentially reducing accuracy.
- **Policy and Social Disconnectedness:**

The study overlooks key policy constraints (e.g., Metropolitan Green Belt restrictions) and community dynamics (e.g., public resistance to densification), limiting its practical applicability.

- **the leak of integration of training result**

  While the training successfully meets its predefined targets, there is room for improvement in the overall coherence and adaptability of the model's output format.

6.3 Future Work

- **Generalizing:**

  In future work, we plan to adopt a relative coordinate system and utilize raycasting to estimate distances between elements. This approach could improve the model's adaptability and enable it to perform more robustly in diverse spatial contexts.
- **Detailing:**

  Rather than massing, we plan to retouch the details on generated buildings, including functional areas, interior circulation, locating openings for window to maximize ventilation. These could potentially be achieved by further CFD analysis, 3D-level sunlight analysis, etc.

7 Conclusion

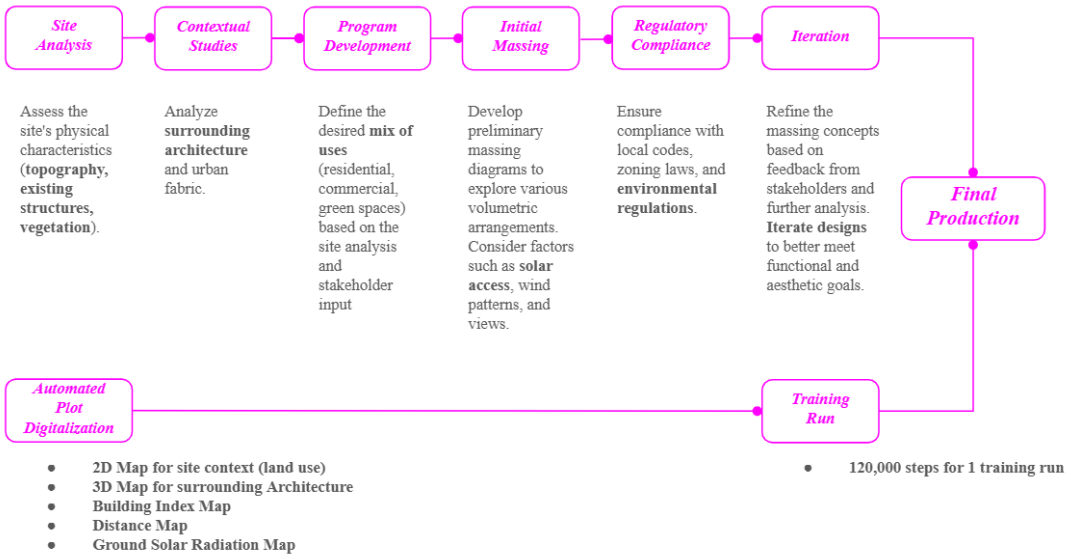

Fig.7. The figure shows the cumulated reward through the training process.

Fig 7 showcases the encapsulated workflow in our Reinforcement Learning framework by doing pairwise comparison against human designers and urban planners. With all these strategies cumulatively applied, our agent-based RL model successfully streamlines the framework of housing design for a metropolitan city prototype.

References

[1] As, I. and Basu, P. (2022). Artificial Intelligence in Urban Planning and Design : technologies, implementation, and Impacts. San Diego: Elsevier.

[2] Berman, E. (2024). Depth-First Search (DFS) Algorithm - Eli Berman - Medium. [online] Medium. Available at: https://medium.com/@that-software-PM/depth-first-search-dfs-algorithm-201dc95e524.

[3] British Politics and Policy at LSE. (2024). Forget the Green Belt, London Needs More Houses. [online] Available at: https://blogs.lse.ac.uk/politicsandpolicy/labour-needs-to-build-houses-in-london/.

[4] Chng, I., Reades, J. and Hubbard, P. (2023). Planning Deregulation as Solution to the Housing crisis: the affordability, Amenity and Adequacy of Permitted Development in London. Environment and Planning A: Economy and Space, 56(3). doi:https://doi.org/10.1177/0308518x231209982.

[5] Cocho-Bermejo, A. and Navarro-Mateu, D. (2019). User-centered Responsive Sunlight Reorientation System Based on Multiagent Decision-making, UDaMaS. Blucher Design Proceedings, pp.695–704. doi:https://doi.org/10.5151/proceedings-ecaadesigradi2019_358.

[6] Lagonigro, R., Oller, R. and Martori, J.C. (2017). A Quadtree Approach Based on European Geographic grids: Reconciling Data Privacy and Accuracy. Sort-statistics and Operations Research Transactions, 41(1), pp.139–158. doi:https://doi.org/10.2436/20.8080.02.55.

[7] Prism-app.io. (2025). PRiSM. [online] Available at: https://www.prism-app.io/configApp/index.html [Accessed 25 Mar. 2025].

[8] Qian, K., Mao, L., Liang, X., Ding, Y., Gao, J., Wei, X., Guo, Z. and Li, J. (2023). AI Agent as Urban Planner: Steering Stakeholder Dynamics in Urban Planning via Consensus-based Multi-Agent Reinforcement Learning. ArXiv (Cornell University). doi:https://doi.org/10.48550/arxiv.2310.16772.

[9] Simonini, T. (2022). Proximal Policy Optimization (PPO). [online] huggingface.co. Available at: https://huggingface.co/blog/deep-rl-ppo.

[10] Townscaper. (2021). Townscaper. [online] Available at: https://www.townscapergame.com/.

[11] Yu, Z., Lin, Y.-M., Zhao, L., Wu, T., Jin, D. and Li, Y. (2023). Spatial Planning of Urban Communities via Deep Reinforcement Learning. Nature Computational Science, 3(9), pp.748–762. doi:https://doi.org/10.1038/s43588-023-00503-5.