nature computational science

Article

https://doi.org/10.1038/s43588-023-00503-5

Spatial planning of urban communities via deep reinforcement learning

In the format provided by the authors and unedited

Contents

Supplementary Notes

1 Overview of the method

As illustrated in Figure 1, community spatial planning is a two-stage sequential Markov decision process (MDP). In the first stage of land use planning, the agent decides the locations of different functionalities one at a time, and the result of land use planning becomes the initial condition of the next stage. In the second stage of road planning, the agent selects one land use boundary at each step and builds it into a road. At the end of each stage, a reward regarding the efficiency of the corresponding spatial layout is calculated.

To address the irregular conditions of urban planning, we construct a contiguity graph to represent a community, with urban geographical elements as nodes and spatial contiguity relationships as edges. In this way, non-rectangular land blocks and non-grid road networks can be expressed as vector representations of nodes in the graph. Meanwhile, the edges in the graph capture the neighbor information which is critical to the community plan. As illustrated in Figure 2, through the graph modeling, urban planning is reformulated as a sequential MDP of making choices on a dynamic graph, where the agent selects edges in land use planning and selects nodes in road planning. These actions in the graph space decide the locations of land use in the original geographic space, and the graph also evolves accordingly. To overcome the challenge of huge action space and avoid unreasonable spatial plans, we impose a series of constraints in the transformed sequential MDP. Specifically, we fix the planning order and devise a block-dividing method based on domain knowledge, making the agent focus on the core task of selecting locations. In addition, we block out unreasonable locations in the action space by imposing action constraints with a mask indicating feasible locations. With the above design, our agent learns the skills of urban planning in a purely data-driven way, gradually achieving better layout efficiency through a massive number of trials.

In the sequential MDP, the agent goes through multiple steps to achieve the final spatial plan, and the complete trajectory of one plan is called an episode. As shown in Figure 3 bottom, at each step t of the episode, the agent receives the state s_t, outputs the action a_t based on its policy π, transits to the next state s_{t+1} and obtains a reward r_t. In our proposed framework, the state s summarizes the current conditions of the spatial plan, which is represented by the constructed urban contiguity graph with rich node features. The policy π first encodes the current state with a graph neural network (GNN), learning representations for nodes, edges, and the whole graph (Figure 3 top). Two separate policy networks are designed to take action for the two stages respectively. Specifically, the land use policy network scores each edge via an edge-ranking MLP, with the score serving as the selection probability, and similarly, a node-ranking MLP is adopted to score each node in the road policy network (Figure 3 middle). All the intermediate steps except for the last step of each stage have a reward of 0. At the last step of land use planning, a reward considering the layout efficiency is computed, which is a weighted sum of service accessibility and ecology friendliness. Meanwhile, at the last step of road planning, a reward regarding traffic efficiency is returned which combines the density and connectivity of the planned roads. We also adopt a value network to predict the effect of spatial plans, supervised by the calculated rewards from the environment. During model training, we collect thousands of episodes in each iteration, and use PPO[1] to update the parameters of policy and value networks.

With the graph reformulation of the land use and road layout for the community and the sequential MDP, we can now frame the adopted community spatial planning model as an *optimization*-like problem. The descriptions of the optimization process

are as follows,

$$
\begin{aligned}
&\underline{\text{Land use planning:}} && (1)\\
&\quad \underset{a_t}{\textbf{maximize}} \quad \alpha\texttt{Service}(D_T)+\texttt{Ecology}(D_T), && (2)\\
&\quad \textbf{subject to} \quad \sum_{(i,j)\in E} a_t(i,j)=1, && (3)\\
&\qquad\qquad a_t(i,j)=\{0,1\},\forall(i,j)\in E, && (4)\\
&\quad \textbf{where} \quad D_t=(V_t,E_t),t\le T && (5)\\
&\qquad\qquad D_{t+1}=\Phi(D_t,a_t),t\le T-1 && (6)\\
&\underline{\text{Road planning:}} && (7)\\
&\quad \underset{a_t}{\textbf{maximize}} \quad \texttt{Traffic}(D_{T'}), && (8)\\
&\quad \textbf{subject to} \quad \sum_{i\in N} a_{t'}(i)=1, && (9)\\
&\qquad\qquad a_{t'}(i)=\{0,1\},\forall i\in N, && (10)\\
&\quad \textbf{where} \quad D_{t'}=(V_{t'},E_{t'}),t'\le T' && (11)\\
&\qquad\qquad D_{t'+1}=\Psi(D_{t'},a_{t'}),t'\le T'-1 && (12)\\
& && (13)
\end{aligned}
$$

where D_t and $D_{t'}$ are the land use and road designs at the t-th or t'-th step, a_t and $a_{t'}$ are the decisions made for land use and road planning which are selections of edges and nodes, Φ and Ψ denote the transition of adding new land use and road segments, and `Service`, `Ecology` and `Traffic` are quantitative objective functions defined in Methods.

2 Experiment details

2.1 Experimental setup

We experiment on one synthetic community and two real-world communities. For the synthetic community (see Figure 1), we use the basic grids as the initial trunk road conditions and all the blocks enclosed by roads are vacant lands to be planned. The agent needs to first lay out land use, and then design branch roads with the two planning stages in Figure 1. In addition, we also experiment on two real-world communities, Huilongguan CP-02 (HLG) and Dahongmen (DHM) in Beijing, to perform community renovation. Specifically, we replicate the real-world road conditions, reserve residential areas, and let the agent plan all the community facilities (see Figure 4a and Supplementary Figure 11a. In the real-world experiment setting, the roads are already built and the agent only needs to accomplish the first stage of land use planning. We trained our model on a single Nvidia GeForce RTX 4090 graphics processing unit (GPU) for 1000 iterations per stage, which took about 72 h during which the agent learned to optimize the spatial layout through over 1 million episodes of planning.

2.2 Baseline approaches

We implement multiple existing computational baseline approaches, which are all integrated into our software system. **Centralized heuristic.** We first implement a centralized rule-based heuristic which serves as a proxy for the traditional planning philosophy which tends to concentrate various functions in the center and depend on long-distance commutes by automobiles. Specifically, the probability of selecting a vacant land block for the current planning step is inversely proportional to the distance between the vacant land and the community center. From the perspective of our graph modeling framework, one edge is selected in each step of land use planning stage, and the score of each edge in the centralized heuristic is calculated as follows,

$$
\begin{aligned}
&X(e_{ij})=\frac{1}{2}(X(v_i)+X(v_j)), && (14)\\
&s(e_{ij})=-\text{EucDis}(X(e_{ij}),(0,0)), && (15)
\end{aligned}
$$

where $X(e_{ij})$ denotes the coordinates of the edge which is the middle point of its two endpoints, and we add a minus sign in front of the Euclidean distance between the edge and the community center to make the score vary inversely with distance. The sampling probability is obtained through normalization by Equation (12) of Methods. For road planning in the synthetic grid scenario, since there is no appropriate existing heuristic, we adopt a density-first approach, *i.e.*, the sampling probability of selecting one land use boundary to construct a road is proportional to the length of the land use boundary so as to build denser

roads in a fixed number of steps. In our graph modeling framework which samples one node in each step, the score of each node is thus calculated as follows,

$$s(v_i) = \text{Length}(v_i). \tag{16}$$

Similarly, the score is normalized by Equation (14) of Methods for sampling.

Decentralized heuristic. The above centralized algorithm makes community services less accessible for residents living in marginalized areas of the community. To improve the service efficiency, we further implement a decentralized rule-based heuristic, which plans a land use type at locations that are far from those already planned sites of the same type. Specifically, the probability of selecting a vacant land block is proportional to the distance between the vacant land and the same type but already planned land block. Formally, the score of each edge is calculated as follows,

$$X(e_{ij}) = \frac{1}{2}(X(v_i) + X(v_j)), \tag{17}$$

$$s(e_{ij}) = \frac{1}{n_{T_c}} \sum_{T_k = T_c} \text{EucDis}(X(e_{ij}), X(v_k)), \tag{18}$$

where n_{T_c} is the number of already planned land use of the current type T_c, and we take the average on the distance between the edge coordinates to those nodes v_k that represent land use of type T_c. The sampling probability is computed as Equation (12) of Methods. Meanwhile, we adopt the same density-first road planning strategy as Equation (16) for the decentralized heuristic in the synthetic grid scenario.

Genetic Algorithm (GA). The above two heuristics are based on fixed rules and these rules are not optimized for the specific planning tasks which can lead to sub-optimal performance. Therefore, we further introduce an optimizable GA baseline for comparison. Specifically, we use two linear layers to score the edges and nodes respectively as follows,

$$s(e_{ij}) = \langle w_{land}, \frac{v_i^0 + v_j^0}{2} \rangle, \tag{19}$$

$$s(v_i) = \langle w_{road}, v_i^0 \rangle, \tag{20}$$

where w_{land} and w_{road} are the two linear layers for the two stages, and $\langle \cdot, \cdot \rangle$ represents the inner product between the linear layer and corresponding edge/node embedding. The sampling probability is obtained in the same manner by normalization as Equation (12) and (14) of Methods. We define the gene as the concatenation of the two linear layers,

$$\text{gene} = w_{land} \| w_{road}, \tag{21}$$

which is optimized by the GA. Edges and nodes are selected according to the sampling probability obtained with the gene, and we calculate the final planning performance by Equation (1)-(2) as the fitness score for the gene, which is used for parent selection in GA. We use uniform sampling to randomly initialize different weights as the initial populations which is a common practice in continuous optimization[2,3]. The weight value is sampled from the range -5 to 5, and the population size is set as 20. We use single-point crossover of two parents and random mutation of genes across generations of populations. Multiple parameters of GA are searched, including the population size, the number of generations, and the mutation probability. To accelerate experiments, we stop the evolution if the performance saturates for 10 consecutive generations. Finally, the optimal individual solution in the whole evolution process is retained to evaluate the performance.

GSCA. We implement a geometric set-coverage problem with single-step adaptations. As the candidate locations for land use are not static and are continuously changing with newly added facilities (each planned facility will cut off a parcel from an existing land block, creating new vacant lands of different shapes and exact locations), the 15-minute city planning problem cannot be addressed by a standard geometric set-coverage model. However, within each planning step, the problem can be adapted and transformed into a geometric set-coverage-like problem with reasonable approximations by maximizing the coverage of the given facility type under the current planning conditions. Specifically, we have the following adapted optimization problem in each step,

GSCA: (22)

$$\underset{x_i}{\textbf{maximize}} \quad \sum_{i=1}^{n_{RZ}} s^t_{RZ_i}, \tag{23}$$

$$\textbf{subject to} \quad \sum_{i=1}^{n_{FA}} x_i = 1, \tag{24}$$

$$x_i = \{0,1\}, \forall i \in \{1,2,\cdots,n_{FA}\}, \tag{25}$$

(26)

where x_i denotes the selection of candidate location, n_{FA} is the total number of candidate locations for the specific facility, $s^t_{RZ_i}$ means whether the i-th residential area can be served within the 15-minute life circle after the t-th planning step.

DRL w/ MLP. To investigate what role RL and GNN play in our model respectively, we develop a basic RL model which replaces the GNN state encoder in our method with a simple multi-layer perceptron (MLP). Specifically, this baseline directly utilizes MLP without message passing and neighbor aggregation as the state encoder, thus it ignores the spatial topology of the community. Formally, the score of each edge or node is calculated as follows,

$$s(e_{ij}) = \mathrm{MLP}_{land}(\frac{v_i^0 + v_j^0}{2}), \tag{27}$$

$$s(v_i) = \mathrm{MLP}_{road}(v_i^0), \tag{28}$$

where MLP_{land} and MLP_{road} are two separate MLP models for land use and road planning stages respectively. On the one hand, by comparing DRL w/ MLP against our proposed approach, we can investigate the effect of GNN in modeling the contiguity relations between various urban geographical elements. On the other hand, DRL w/ MLP can be regarded as optimizing the genes (parameters of MLP) with RL instead of evolution, thus we also compare DRL w/ MLP against the GA baseline to study the advantage of RL over traditional approaches in handling the complicated planning task, especially the effect of function approximation of values and efficient exploration in the huge action space.

2.3 Comparison with baseline methods

We implement the above baseline approaches, and evaluate their performance in multiple scenarios including both one synthetic grid planning task and two real-world community renovation planning tasks to compare with our proposed method. Specifically, for the two heuristics, we replace the policy network of our framework with manually defined rule-based policies. For the GA method, we implement the algorithm with PyGAD[4] and integrate it with our framework. For the DRL w/ MLP baseline, we change the state encoder in our framework from GNN to MLP. To achieve a fair comparison, all the methods interact with the same spatial planning environment as our proposed DRL approach. After we obtain the generated plans, we compute the spatial efficiency metrics to compare their planning performance.

As introduced in the paper, Table 1 shows the planning performance on three planning scenarios where our DRL method outperforms all the baseline methods with significant improvements on all metrics. Supplementary Figure 3-4 demonstrate the generated spatial plans from all the methods for the HLG and DHM community respectively, as well as the quantitative performance evaluation. We can observe that baseline methods tend to yield more clustering land use functions, such as the recreation cluster in Supplementary Figure 3a and the school cluster in Supplementary Figure 4c. On the contrary, as shown in Supplementary Figure 3e and Supplementary Figure 4e, the spatial plans generated by our approach successfully avoid clusters of the same land use function. Particularly, we illustrate the planning process of our framework for the two real-world communities in Supplementary Figure 1 and Supplementary Figure 2 respectively. We can find that the DRL agent lays out different land use functions according to the predefined planning order introduced in Section M3.3, with each land use function arranged in a dispersing manner. Experimental results compared with baseline methods verify that our DRL approach learns the decentralized skill of urban planning, which achieves better spatial efficiency with respect to traffic, service and ecology.

2.4 Comparison with professional human designers

As most of the spatial plans are designed by human experts currently, we also conduct experiments to compare our DRL framework with human experts. We recruit 8 professional human designers in both Britain and China to accomplish the planning tasks for the two real-world communities, HLG and DHM. Specifically, the 8 participants are all experienced designers, working for top planning institutes including University College London, Beijing Tsinghua Tongheng Urban Planning & Design Institute, Chongqing Planning & Design Institute, and School of Architecture and Urban Planning, Chongqing University. Each participant starts planning from the same initial conditions as our DRL framework, and accomplishes the planning task according to the same planning needs and planning requirements as our DRL framework with the help of CAD and GIS software. For the spatial plans generated by human designers, we compute the spatial efficiency based on the same definitions of service and ecology metrics.

Supplementary Figure 5 demonstrates the spatial plans for the real-world HLG community generated by human experts and our DRL method, as well as their corresponding spatial efficiency performance. With respect to service efficiency, our DRL method outperforms 7 out of 8 professional designers, and achieves the same performance as the best designer. Specifically, as in Supplementary Figure 5j, our method and designer H7 attains 0.71 in service efficiency, which significantly outperforms the expert average of 0.64 with relative improvements about 9.86%. With respect to ecology efficiency, our DRL method outperforms all 8 professional designers with significant improvements, 17.74% higher than the best designer and 57.41% higher than the designer average. The service efficiency metric is the average accessibility over five different basic services, and we also inspect the accessibility for each of the services in Supplementary Figure 5k. We can observe that our DRL agent

generates a plan with a more balanced accessibility of different services, while it is challenging for human designers to achieve a balanced trade-off between different objects. Similarly, Supplementary Figure 6 illustrates the results on the DHM community by human experts and our DRL method. Results show that our method beats all 8 professional designers on both service and ecology metrics. Specifically, as in Supplementary Figure 6j, our DRL method improves the service efficiency by 13.64% and 19.52% against the best and average designer performance respectively. For ecology efficiency, relative improvements against the best and average designer are 15.38% and 59.65%. Supplementary Figure 6k illustrates the specific accessibility of five basic services, where our DRL method still achieves a much more balanced performance, with 3 out of 5 services ranking the highest against all human designers. Human designers accomplish the planning task based on experience and subjective intuition, which can be influenced by the traditional centralized planning concept, yielding clusters of the same land use function, as shown in Supplementary Figure 5a and Supplementary Figure 6b. Different from human designers, DRL method guided by quantitative reward on spatial efficiency can optimize for higher efficiency in a data-driven way, which can get rid of the influence of traditional planning concepts, achieving more decentralized planning solutions.

2.5 More experimental results on 15-minute city

As introduced in Section M3.2, we conduct experiments under different planning requirements, *i.e.*, different numbers of service facilities, and investigate the accessibility of services in such different service provisions. In actual community spatial planning, more facilities mean it is easier to achieve 15-minute city. However, the community may not be able to support so many facilities due to many conditions, resulting in problems such as low utilization rates and poor accessibility. On the other hand, fewer facilities may lead to high utilization rate while reducing accessibility. The exact number of facilities depends on the actual situation, and we study the planning performance under both low and high service needs in our experiments. Specifically, Figure 4 shows the planning performance of our DRL method in achieving 15-minute city for the HLG community, and the corresponding generated spatial plans under given different service needs are illustrated in Supplementary Figure 9. We can observe that even with a low or medium needs of facilities, our DRL model learns to layouts these facilities in different areas of the community. As the needs increase, the spatial plan is getting more decentralized.

We also conduct experiments on 15-minute city for the DHM community renovation task. As shown in Supplementary Figure 11a, we reserve most of the residential blocks and a few large-area facilities, and re-design the remaining areas to improve service and ecology efficiency. Similar to the experiments on the HLG community, we utilize a well-trained model and conduct model inference under multiple settings with different service needs (low, medium, high and mix). Supplementary Figure 11b-c illustrate the specific needs and the resulting accessibility of different service types. As the needs change, our method can generate spatial plans with varying service accessibility accordingly. And the accessibility of different service facility types can be customized by feeding a mixed needs to our model, *e.g.*, see the needs of school and hospital and the corresponding accessibility of education and medical care. Supplementary Figure 11d-g show the generated spatial plans under different service needs, where a consistent decentralized planning strategy can be observed which guarantees superior performance in achieving 15-minute city.

2.6 Details of model transferability

We study model transferability from two perspectives. The first one is the transfer between different scales, from a small community to a large community. The second one is the transfer between different forms, from simple road and land conditions to complicated ones. These two types of transferability enable us to train a model in a small-scale synthetic environment and apply it to generate plans for different kinds of large-scale real-world communities. The GNN state encoder and policy networks in our model operate on every node and edge embedding separately, thus the number of model parameters is independent of both the graph size (community scale) and the graph topology (community form). Owing to the generality of the graph modeling (Figure 2), our model is naturally suited for transfer between different planning environments, since we can finetune a pretrained model in new communities without introducing new model parameters.

To study model transferability, we first construct a simulated grid scenario with only 3×3 blocks which is smaller than the 4×4 blocks in Figure 1. We pretrain our DRL model in this small-scale synthetic environment for 300 iterations which take about 24h, and utilize the pretrained model as the start point for transfer. For the first case of transfer between different scales, we load the parameters of the pretrained model and finetune it in the 4×4 blocks grid community, with the community area increased by 44% and the episode length increased by nearly 100%. For the second case of transfer between different forms, we use the same pretrained model from 3×3 blocks grid community, and finetune it in the real-world renovation scenario in Figure 4a, where the road and land use conditions become much more complicated, *i.e.*, the roads change from simple grid form to irregular form, and the land use conditions change from all vacant lands to existing residential blocks. For both cases, we compare the planning performance of finetuning the pretrained model against training a model from scratch with randomly initialized model parameters.

2.7 Training with different data volume

Our model is primarily a data-driven method, which depends on extensive interactions with the environment to learn urban planning skills. The data volume, *i.e.* the total number of interactions between our model and the environment during the training process, serves as a key factor in the final planning performance. During the model training process, the agent goes through hundreds of iterations, and accomplishes thousands of episodes in each iteration, with about 100 steps per episode (one episode is the generation of a complete spatial plan starting from the same initial condition), meaning that the overall data volume easily exceeds one million. Such a large amount of data is necessary for the agent to perform sufficient exploration in the huge action space, so as to accurately predict the value of different spatial plans and finally obtain a decent policy. Supplementary Figure 12 shows the episodic reward achieved under different scales of training samples. In the first half of model training (before 250 iterations), the agent just warms up and explores different strategies, thus the performance oscillates under both larger (7.5 million) and smaller (3.75 million) training samples. As model training continues, the difference between large and small sample scales starts to emerge, with the agent's performance under large data volume gradually improving while the performance under small data volume remains in oscillation. Results in Figure 12 demonstrate the essential role of data volume in urban planning with reinforcement learning, and suggest that larger quantitative training data will eventually lead to better spatial efficiency.

In fact, the volume of data represents the extent to which the DRL agent explores the action space. Since the action space is extremely large, the agent needs to interact with the environment extensively to obtain sufficient training data, which guarantees enough exploration and finally achieves layouts with higher spatial efficiency. To inspect how the agent learns the skills of spatial planning, we visualize the learning process of our DRL method by investigating the spatial plans obtained at different iterations in the whole training procedure.

Supplementary Figure 13 and Supplementary Figure 14 demonstrate the generated spatial plans at different iterations for the real-world HLG and DHM communities, respectively. At iteration 1, the agent has not yet learned the skills of urban planning, thus it just expands in a local area, which results in all the land use functions of the same type arranged next to each other (see Supplementary Figure 13a and Supplementary Figure 14a). As the training progresses, the agent starts to discover that decentralized planning can achieve higher rewards during the exploration, so some land use functions begin to disperse. For example, the business and recreation area in Supplementary Figure 13c and the school area in Supplementary Figure 14c all begin to be arranged at different locations in the community. However, without sufficient explorations which usually take about 400 iterations, the generated spatial plans still contain a certain number of clusters of the same land use function, *e.g.*, clinic at the southern (bottom) area of HLG community in Supplementary Figure 13c and recreation at the northeastern (top right) area of DHM community in Supplementary Figure 14d. After training with enough samples that usually take over 500 iterations, the agent becomes able to make more decent arrangements for various land use functions, and different services are laid out in a decentralized way (see Supplementary Figure 13f and Supplementary Figure 14f). From the results in Supplementary Figure 13g and Supplementary Figure 14g, we can observe that the two spatial efficiency metrics are continuously improved as the training progresses. Through the visualization of the learning process, we further verify the necessity of large-scale training samples for DRL models to master the skills of urban planning.

Due to the limitation of computing resources, all the experiments are conducted on a single Linux server with one single GPU, and the training for spatial layout of one community usually takes about 48h. After the model converges, the evaluation performance on a community can be obtained through model inference within less than 10s. We believe that better performance can be achieved if larger computing resources are adopted, such as collecting training samples using distributed clusters and training the model with multiple GPUs on multiple servers.

2.8 Hyper-parameter study

In this section, we investigate the effect of several critical hyper-parameters in our framework, including reward weight, GNN layer, and GNN node dimension.

In land use planning, the reward function is a comprehensive evaluation of spatial efficiency, which is a weighted sum of service and ecology. As shown in equation (1) of the paper, we introduce a hyper-parameter α to adjust the ratio between the importance of the two aspects, with larger α emphasizing more on service efficiency and smaller α emphasizing more on ecology efficiency. We train three different models with the ratio between service and ecology set as $2:1$ to $4:1$ for the HLG community, and evaluate their corresponding planning performance. Supplementary Figure 15 demonstrates the generated spatial plans under different reward weight ratios, as well as the service and ecology metrics. As shown in Supplementary Figure 15a, we can observe that with lower α, *i.e.*, ecology is more important, the DRL agent learns to leave vacant lands at different locations of the community that will finally be filled as open space, which in turn improves the ecology metric. As we increase the value of α, *i.e.*, service becomes more important, and the DRL agent learns to arrange these previous vacant lands as service facilities to improve their accessibility. Particularly, as shown in Supplementary Figure 15c, the DRL agent leaves many vacant lands next to the existing central park , while filling these areas as open space. These open spaces do not

bring much improvements to the ecology metric since their ecological serving range (ESR, see equation (19) of Methods) is largely overlapped with the ESR of the center park. Supplementary Figure 15d-e show the service and ecology metrics of the generated plans under different reward weight ratios, where we can observe that the service metric increases and the ecology metric decreases as the reward weight ratio changes from $2:1$ to $4:1$. Experiments on different reward weight ratios validate the flexibility of our framework, where we can adjust the value of α to realize spatial plans with different emphasis on service or ecology.

The GNN state encoder is a crucial component of our framework, which learns representations for different geographical elements in the community, and supports both value prediction and action selection. Particularly, two hyper-parameters of GNN, the number of GNN layers and the GNN node dimension, determine the topological modeling ability and expressive power of GNN. We investigate the planning performance of our framework under different numbers of GNN layers and node dimensions, and the results are shown in Figure 16. Specifically, as shown in Figure 16a, setting GNN layer as 0 (no message passing, *i.e.*, a trivial MLP model) achieves much worse performance (-8.38% reward) than the models with at least one GNN layer, proving the essential role of message passing and neighbor aggregation in GNN. Meanwhile, too few GNN layers (*e.g.*, only 1 layer) makes the perceptive neighbor field not large enough to acquire effective topological information; while too many GNN layers (*e.g.*, 3 layers) can lead to over-smoothing which deteriorates the planning performance. Therefore, setting the number of GNN layer as 2 achieves the best reward. Similarly, as shown in Figure 16b, too low node dimension (e.g., 4) provides insufficient expressive power of GNN, which results in inferior performance. Increasing the node dimension to 8 and 16 can significantly improve the expressive ability and makes much progress in spatial efficiency (+9.31% reward). Further increasing the node dimension to 32 can overfit the model to the noise in the training samples and achieve worse performance.

2.9 Demonstration of human-AI collaborative workflow

We have shown that AI can outperform professional human designers in optimizing spatial efficiency in a huge solution space. But human designers are good at abstract prototyping, thus we propose a human-AI collaborative workflow to take advantage of their respective expertise, as illustrated in Supplementary Figure 7a. Conceptual planning described by center and axis is first provided by human designers. Then we train DRL models to generate spatial plans that satisfy the planning concepts and maximize spatial efficiency. Human designers only need to adjust the plans generated by AI without changing the full layout, *e.g.*, adjusting the shapes of a few blocks. To verify the effectiveness of this workflow, we also compare it with a full human labor workflow. We invite 5 professional human designers to accomplish the planning task, who start planning given the same initial conditions, constraints, and planning concepts as our DRL model. For our DRL model, we take the generated plans from the best 5 model checkpoints. We conduct a comprehensive evaluation of the spatial plans generated by professional human designers and AI, including both objective metrics and subjective blind tests. For the objective metric, we calculate the efficiency of service and ecology as equations (18) and (22) of Methods. For subjective evaluation, we invite 100 post-graduate level human designers to participate in the evaluation, where they choose one of two spatial plans based on their subjective preference. The evaluation is conducted in blind manner, *i.e.*, the participants are unaware of whether the spatial plan is generated by human designers or AI.

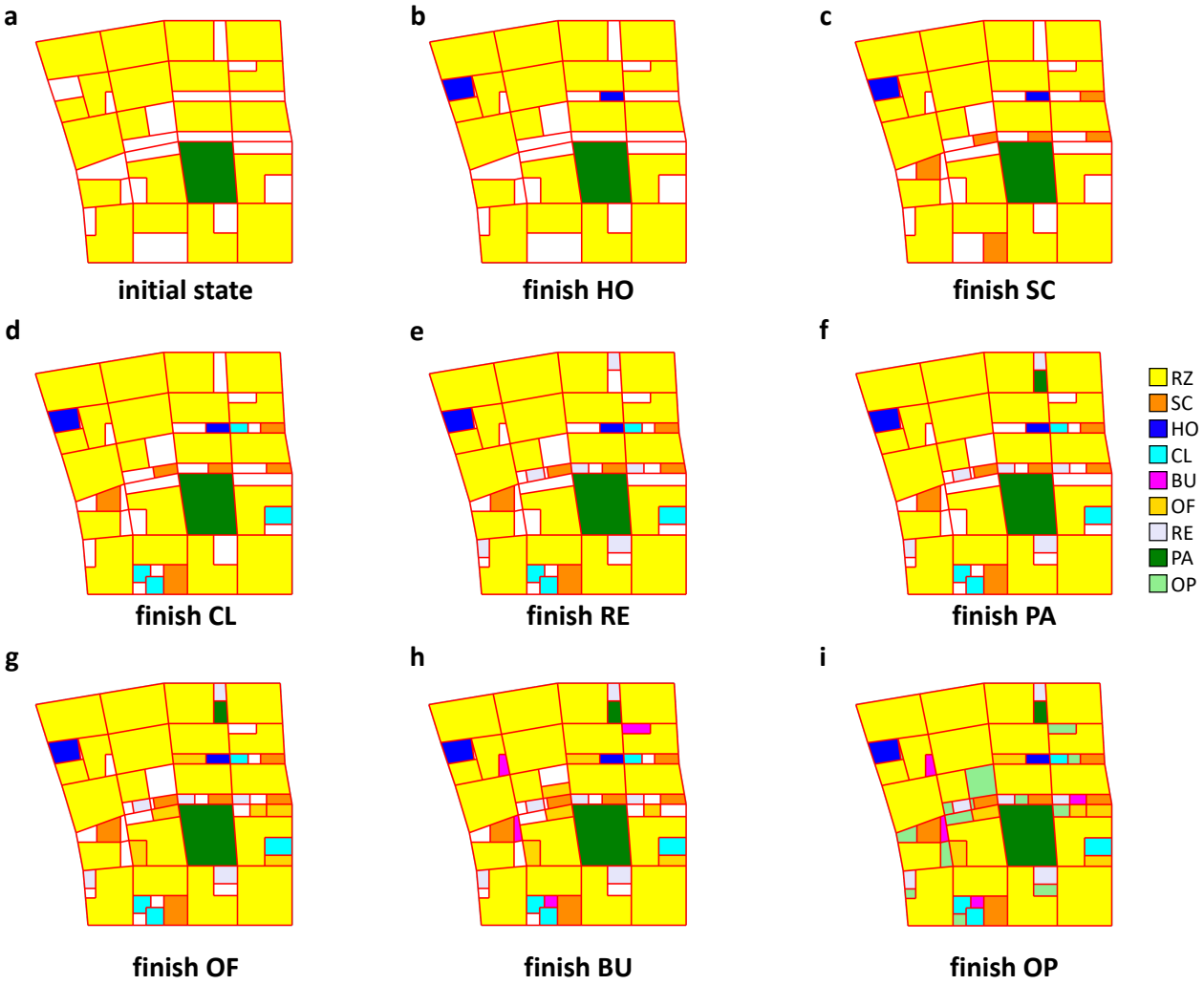
Supplementary Figure 8 shows the generated spatial plans by human designers and our DRL model for the two real-world communities. We calculate the spatial efficiency of these plans, and Supplementary Table 5 shows the results as well as the time cost for training and planning. We can observe that our DRL model achieves competitive performance in objective metrics. Specifically, for the DHM community, the best DRL solution achieves Pareto optimal against all spatial plans with significant improvements (service +12.3%, ecology +5.0%). For the HLG community, no single spatial plan attains Pareto optimal, while our DRL model achieves comparable performance to professional human designers, and improves the ecology efficiency by 14.3%. Nevertheless, a human designer requires about ten years of professional training, and spends about 50-100 minutes to accomplish the two planning tasks. On the contrary, it only takes about two days of training to obtain an AI model with superior performance, which can generate decent spatial plans in less than 1 second. We can conclude that AI can achieve comparable and even better performance on objective metrics than human designers in the heavy and specific planning step, and greatly help human designers to improve their planning effect and efficiency. The blind test is also conducted on the 10 groups of spatial plans by 100 post-graduate level human designers, and the results are shown in Figure 7c-d. Contrary to intuition, professional human designers did not beat the AI model. There is no clear preference in most cases, where AI tends to gain slightly more votes. In a few cases, AI wins much more votes than human designers, *e.g.*, group 2 and group 3 of HLG community. Through evaluation of both objective metrics and the subjective blind test, we verify the feasibility of our proposed human-AI collaborative workflow, in which AI can significantly improve the productivity of human urban designers. It is worthwhile to notice that we only conduct a simplified demo of the workflow, whereas practical planning can be more complicated, such as more diverse planning concepts, opinions from stakeholders, and multiple rounds of revisions. Fortunately, the high flexibility of our DRL framework, *e.g.* the customized reward functions, can help to extend the simplified workflow to real-world applications of human-AI collaborative urban planning.

2.10 Integration with manually defined rules

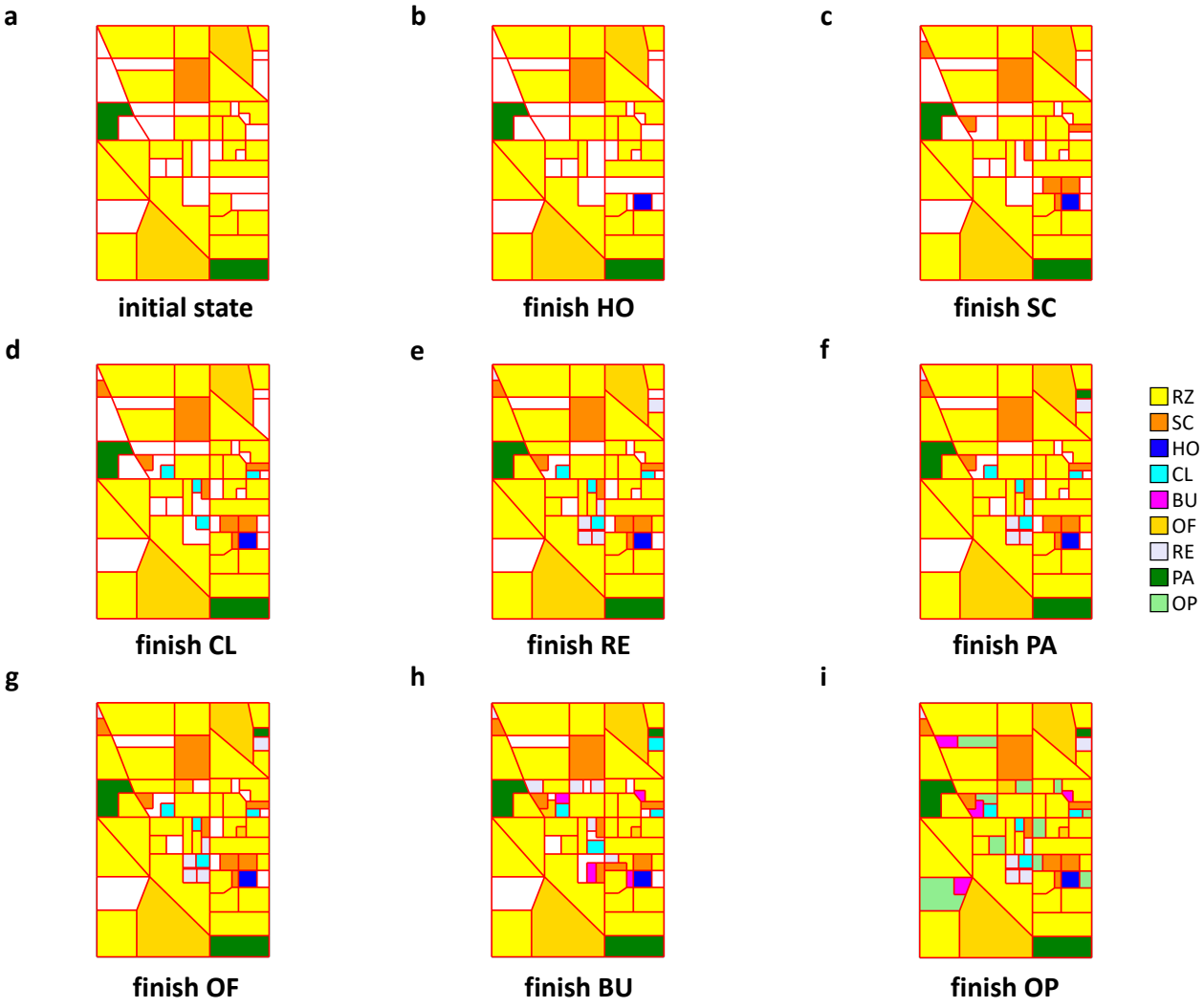
Despite planning concepts and styles that have been investigated in the paper, urban planning in practice can be more complicated since there are other planning rules or restrictions to be considered. These rules from political realities are usually expressed as spatial relationships, *e.g.*, some land use types are not suitable to be arranged next to each other, such as neighboring hospital and school will be detrimental to students' physical and mental health. In fact, experienced human designers can list dozens or hundreds of rules, which need to be considered carefully in practical spatial planning, and these manually defined rules are a part of domain knowledge. Fortunately, our framework is flexible and fully compatible with manually defined rules, and these proposed rules from political realities can be well received and easily incorporated. Specifically, as introduced in the model design, we add a mask in the action space to indicate all feasible actions and avoid unreasonable actions, thus actions that do not satisfy planning requirements are blocked out and will never be chosen by the agent. In our experiments, we set the action mask as False for edges except for L-J edges linking a vacant land and a junction in land use planning task. In the road planning task, we set the action mask as *False* except for S nodes of land use boundaries. Similarly, we can implement the above manually defined rules by adding extra action masks.

We implement the above school-and-hospital rule by action mask to demonstrate the effect of integrating manually defined rules with our framework. Specifically, we take a model that is trained without the rule inserted. Therefore, as shown in Supplementary Figure 10a, although the generated community plan achieves decent performance in spatial efficiency (service=0.6390, ecology=0.7147), it violates the above rule, *i.e.*, dashed black boxes show the positions where the DRL agent lays out a school and a hospital/clinic together. We then finetune this model and add a rule-aware action mask that indicates the above school and hospital rule during finetuning. In other words, when planning a school, we set the mask values of land blocks as false if they are near a hospital. As shown in Supplementary Figure 10b, the final generated spatial plan is fully consistent with the planning rule, where schools and hospitals/clinics are separated by other land use functionalities. Meanwhile, the service efficiency is not getting worse (service=0.6400, +0.001), and the ecology efficiency gets even better (ecology=0.7487, +4.76%). In this experiment, we only add one single rule to showcase the integration of planning rules, and this approach can be easily extended to hundreds of planning rules in practice. For example, when planning business areas, we can set the mask values as true for vacant blocks that are within a certain distance of a subway station, and thus we can place business zones near subway stations to maximize their economic benefits. It is worthwhile to notice that planning rules can be combined with planning concepts in Supplementary Figure 7b, and it can also be introduced into the proposed workflow. Human designers can collaborate with our proposed AI model by defining planning rules and designing prototypes, and let the AI agent accomplish the heavy work of generating spatial layouts.

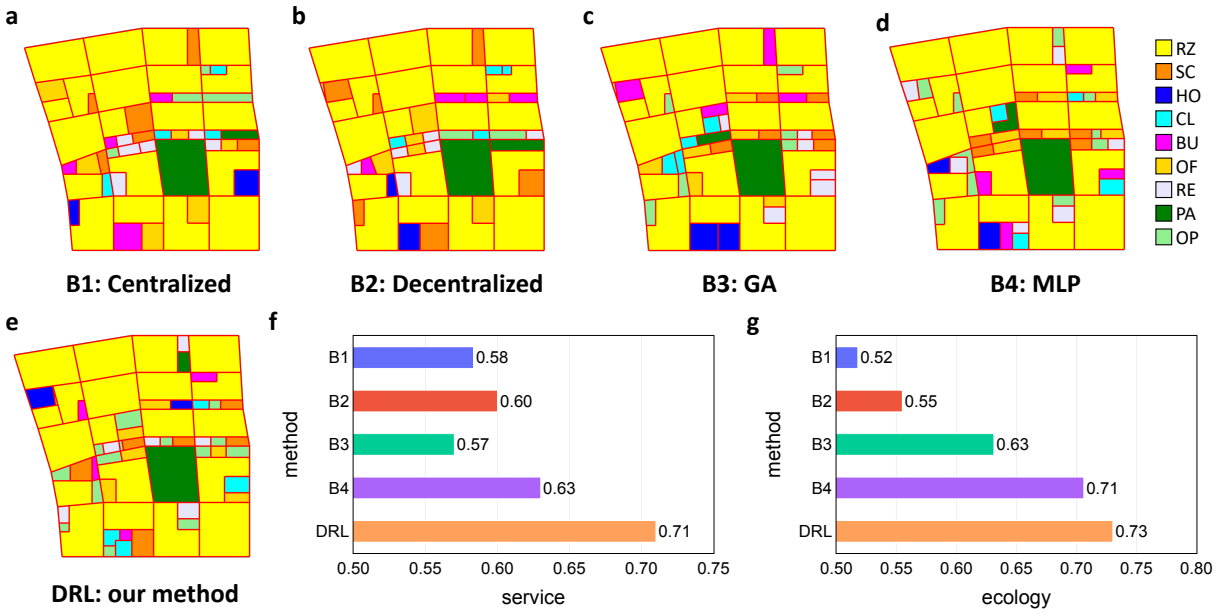
Supplementary Figures



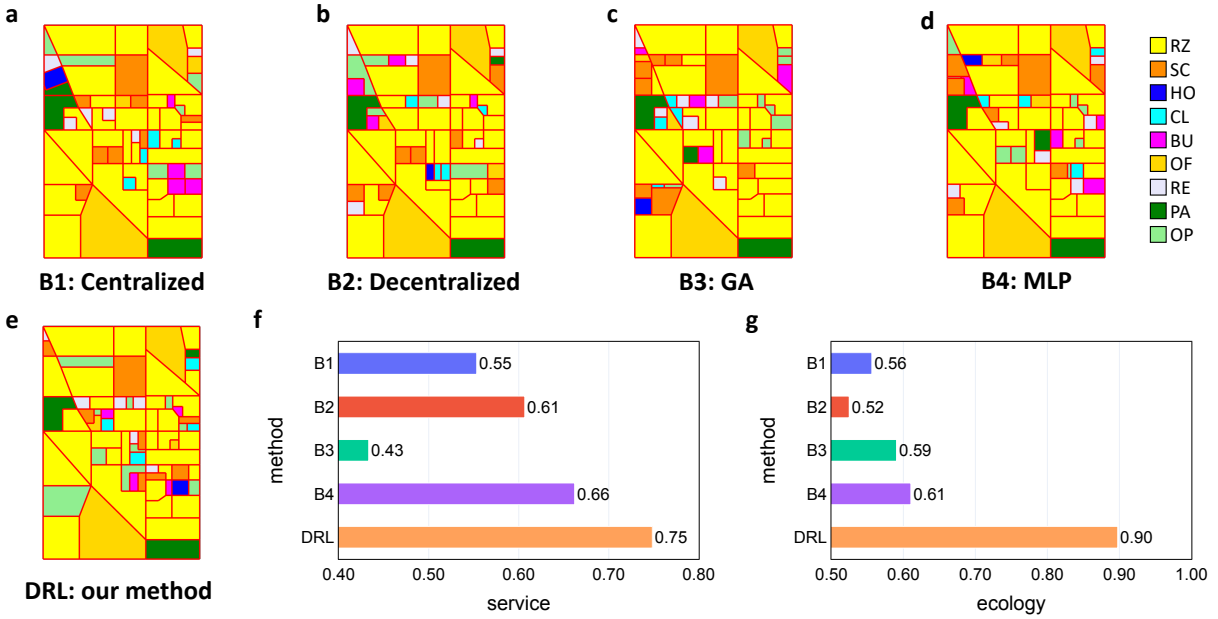
Supplementary Figure 1. Demonstration of the spatial planning process of our DRL approach for the real-world HLG community. We show the snapshots of the spatial plan at those steps when the DRL agent finishes each land use function in the planning episode. **a**, Initial state of HLG community. **b-h**, Snapshots when the DRL agent finishes the layout of hospital, school, clinic, recreation, park, office, business and open space. **i**, fill the remaining vacant land as open space.



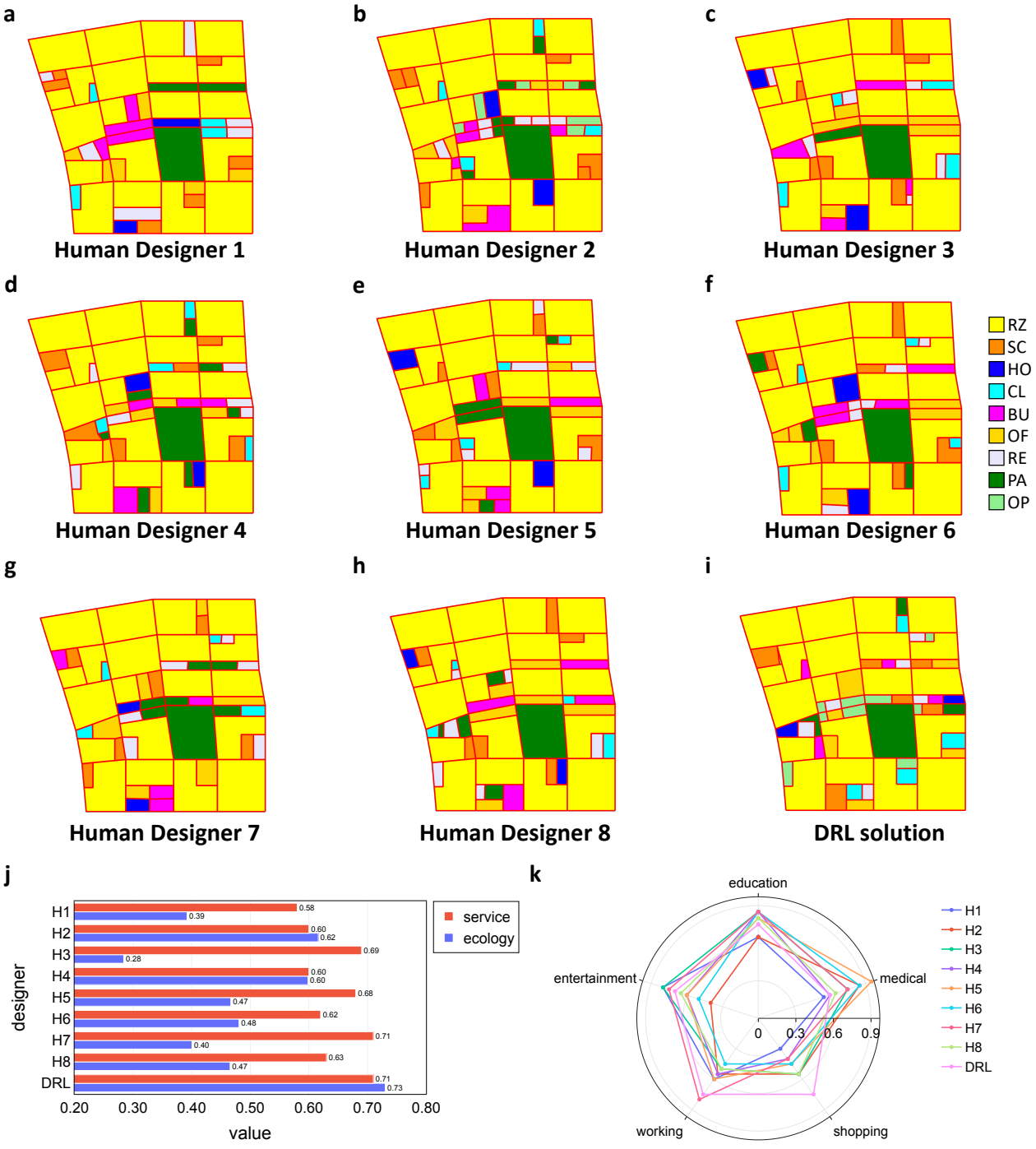
Supplementary Figure 2. Demonstration of the spatial planning process of our DRL approach for the real-world DHM community. We show the snapshots of the spatial plan at those steps when the DRL agent finishes each land use function in the planning episode. **a**, Initial state of DHM community. **b-h**, Snapshots when the DRL agent finishes the layout of hospital, school, clinic, recreation, park, office, business and open space. **i**, fill the remaining vacant land as open space.



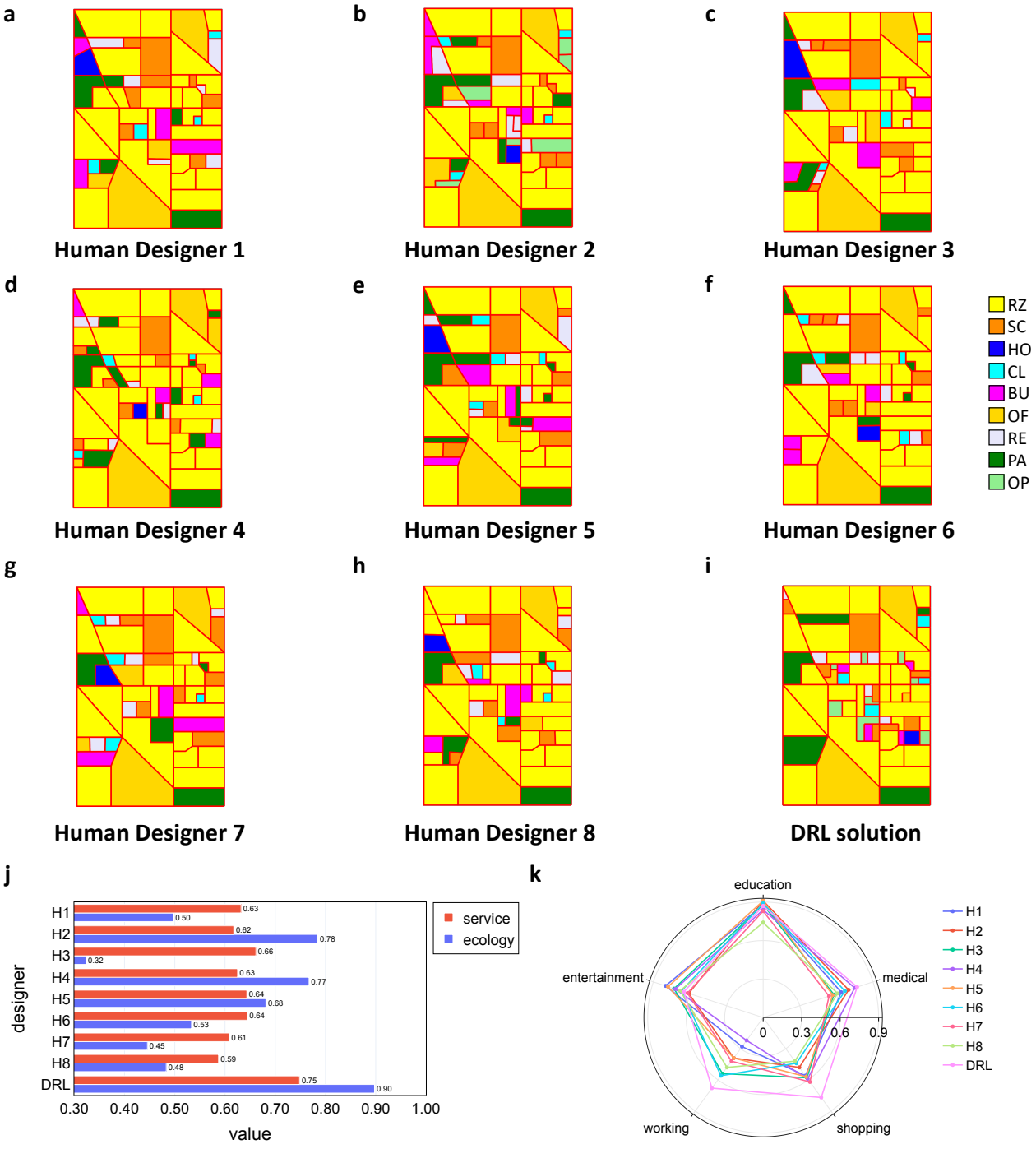
Supplementary Figure 3. Demonstration of generated plans of different methods for the HLG community. a-e, generated plans of four baseline methods and our DRL method. **f**, service efficiency metric of all the methods. **f**, ecology efficiency metric of all the methods.



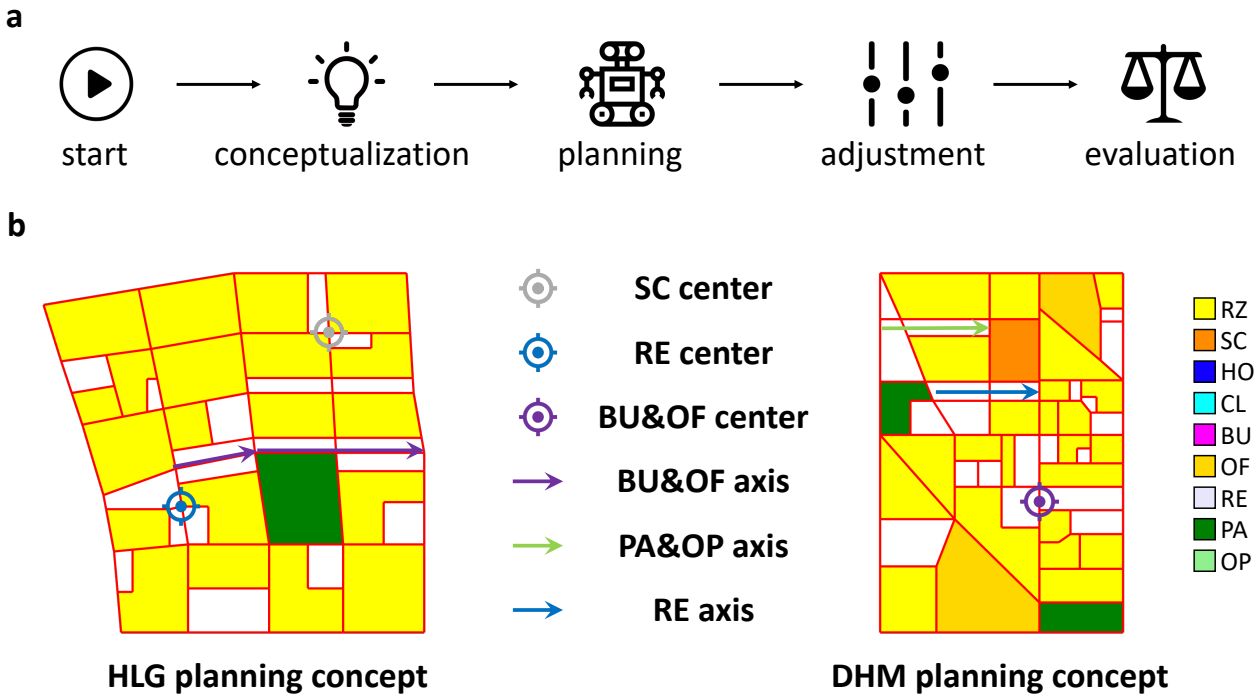
Supplementary Figure 4. Demonstration of generated plans of different methods for the DHM community. a-e, generated plans of four baseline methods and our DRL method. **f**, service efficiency metric of all the methods. **f**, ecology efficiency metric of all the methods.



Supplementary Figure 5. Spatial plans for HLG community designed by 8 professional human designers and our DRL method and their corresponding planning performance. a-h, the spatial plans generated by human designers. **i**, the spatial plan generated by our DRL method. **j**, service and ecology efficiency performance comparison between 8 human designers and our DRL method. **k**, service accessibility comparison of five basic residential needs between 8 human designers and our DRL method.

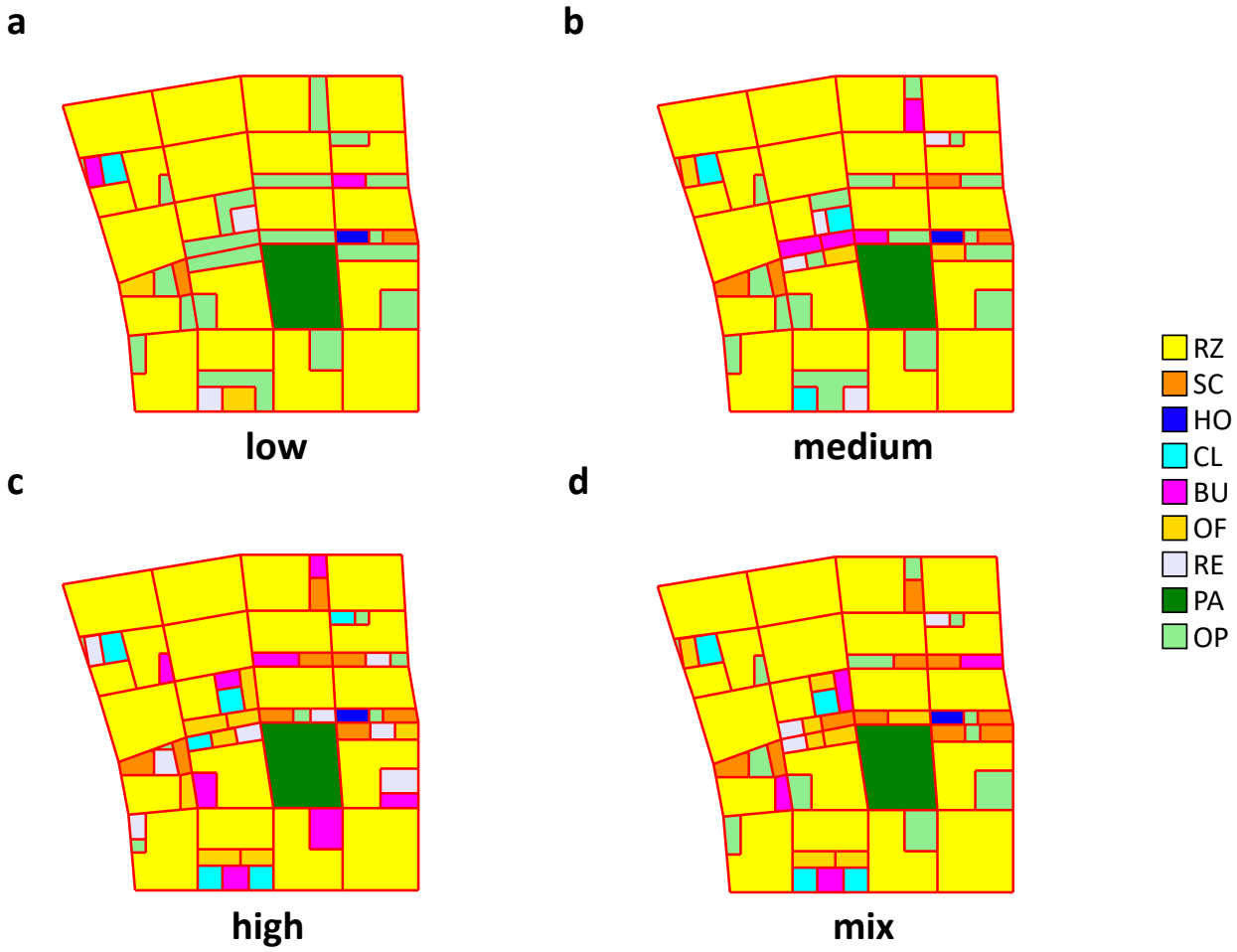


Supplementary Figure 6. Spatial plans for DHM community designed by 8 professional human designers and our DRL method and their corresponding planning performance. a-h, the spatial plans generated by human designers. **i**, the spatial plan generated by our DRL method. **j**, service and ecology efficiency performance comparison between 8 human designers and our DRL method. **k**, service accessibility comparison of five basic residential needs between 8 human designers and our DRL method.

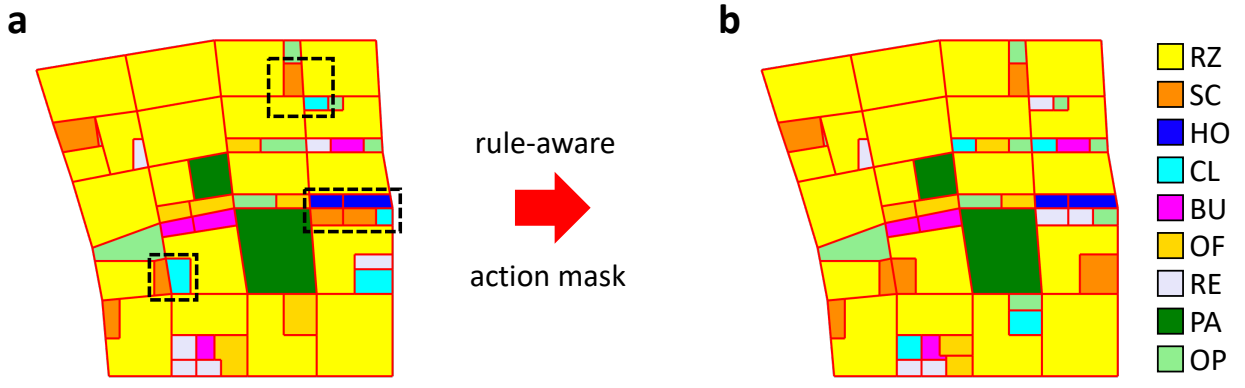


Supplementary Figure 7. Demonstration of incorporating AI into the workflow of urban planning. a, the diagram of human-AI incorporating workflow. **b**, the predefined planning concept of the HLG and DHM community.

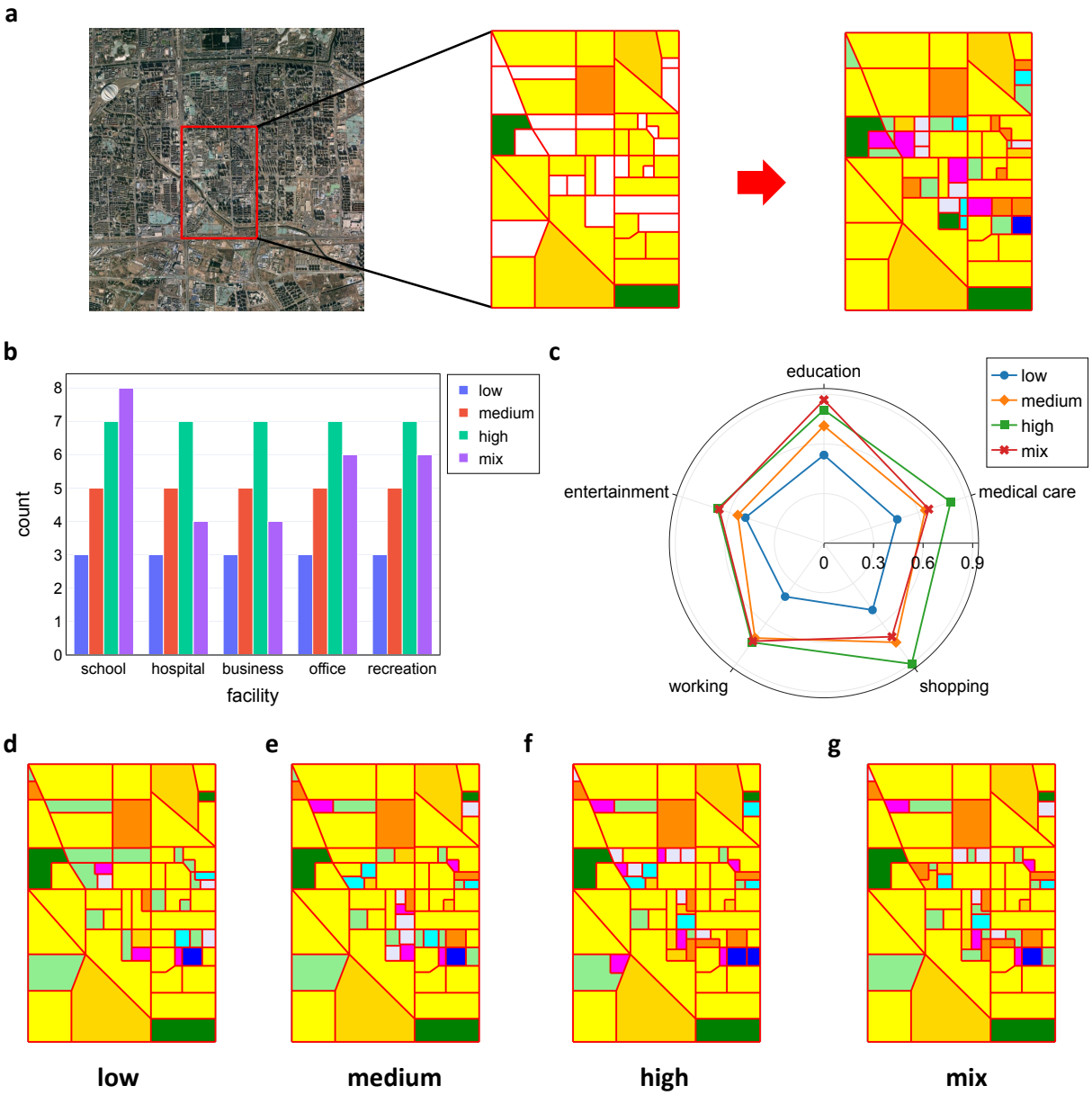
Supplementary Figure 8. **Spatial plans with planning concept designed by professional human designers and our DRL method.**



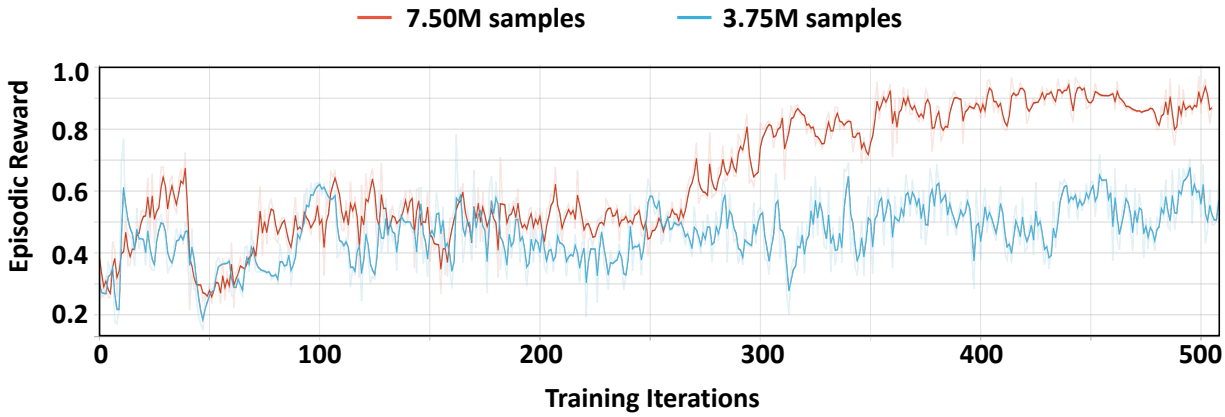
Supplementary Figure 9. Demonstration of generated community renovation plans under different needs of facility types. We vary the needs of five different facility types (school, hospital and clinic, business, office, recreation) which correspond to the five basic services (education, medical care, shopping, working, entertainment). We investigate (a) low needs (2 per facility), (b) medium needs (4 per facility), (c) high needs (8 per facility) and (d) mix needs (10, 5, 4, 8, 3 for the five facility types). The service accessibility of these plans are shown in Figure 4d.



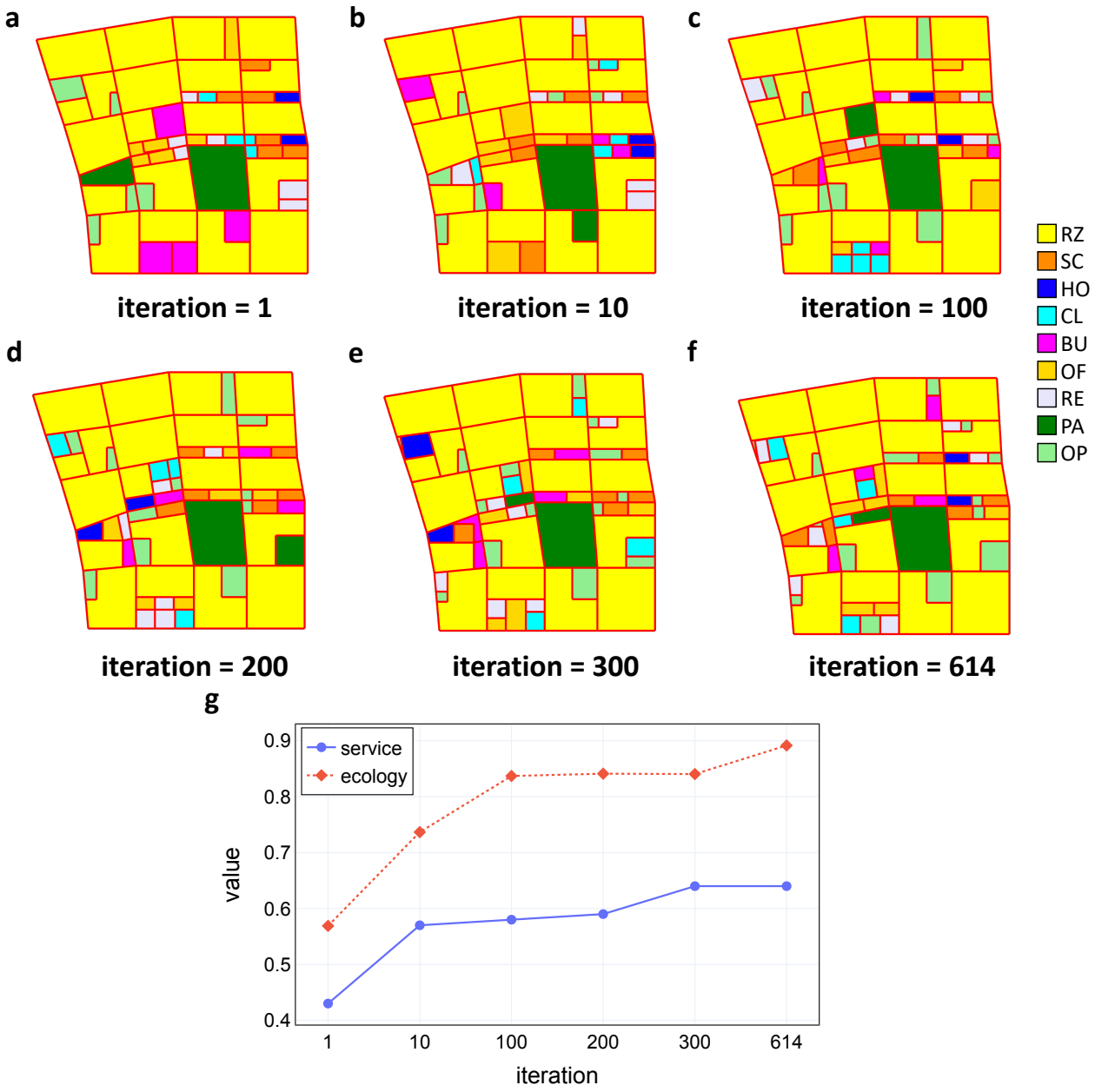
Supplementary Figure 10. Integration with the rule-aware action mask for the HLG community. We demonstrate the generated spatial plans **(a) before** and **(b) after adding the rule-aware action mask.**



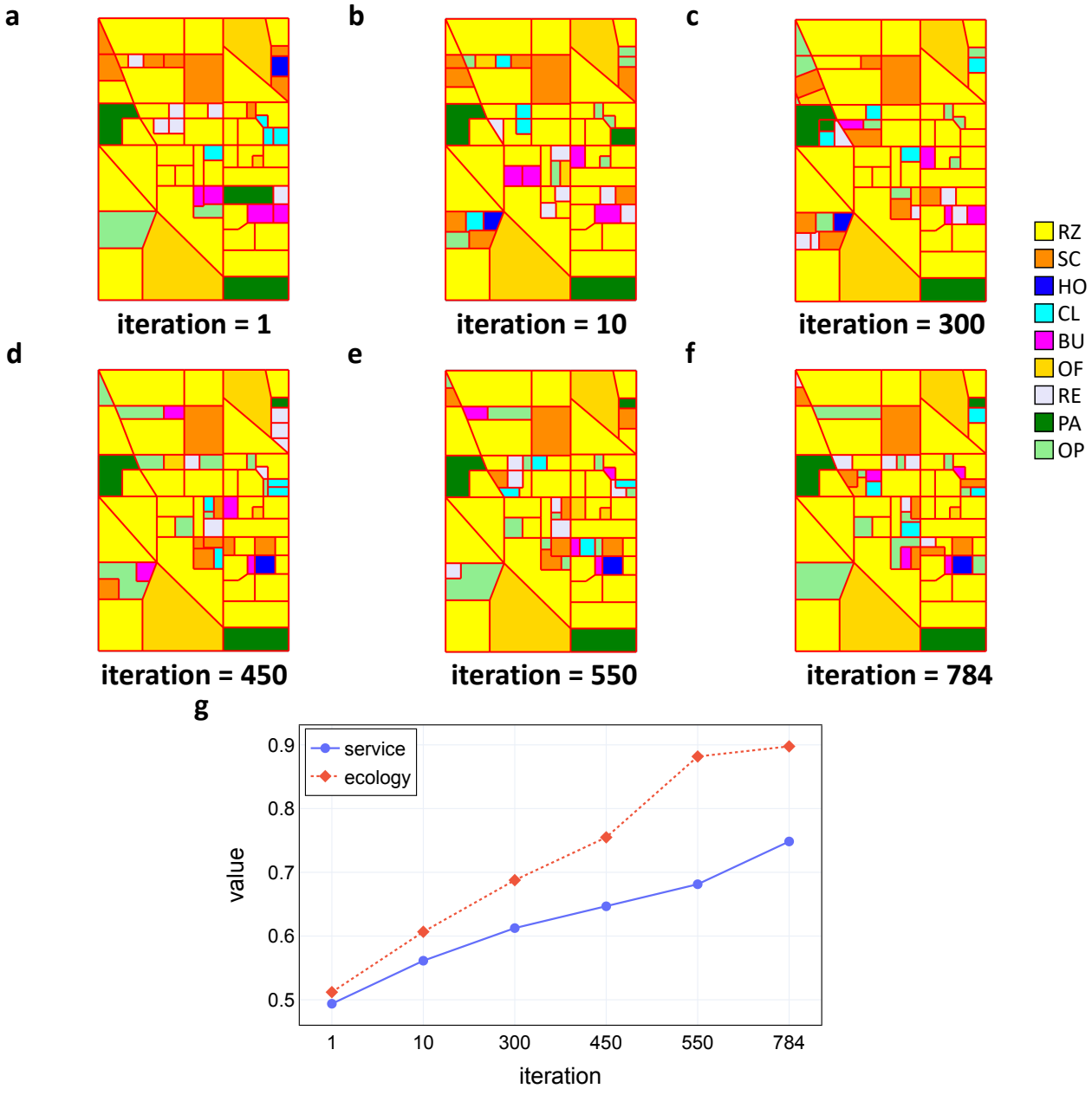
Supplementary Figure 11. Demonstration of community renovation and 15-minute city planning for the DHM community. a, Community renovation. We replicate the roads, residential blocks and large-area facilities of the DHM community, and leave other areas as vacant lands for renovation. The agent places different types of facilities to improve the accessibility of service for residents in the community. **b, Facility needs.** We vary the needs of five different facilities (school, hospital, business, office, recreation) which correspond to the five basic services (education, medical care, shopping, working, entertainment). We investigate low needs (3 per facility), medium needs (5 per facility), high needs (7 per facility) and a mix needs (8, 4, 4, 6, 6 for the five facilities). **c, Service accessibility performance under different needs.** We show the 15-minute circle index for the five basic needs of the generated community plan under different facility needs. The radical value means the proportion of residential blocks that can access the corresponding service with 15 minutes. **d-g, Generated spatial plans under different needs.** We demonstrate the generated community renovation plans under different needs of facility types.



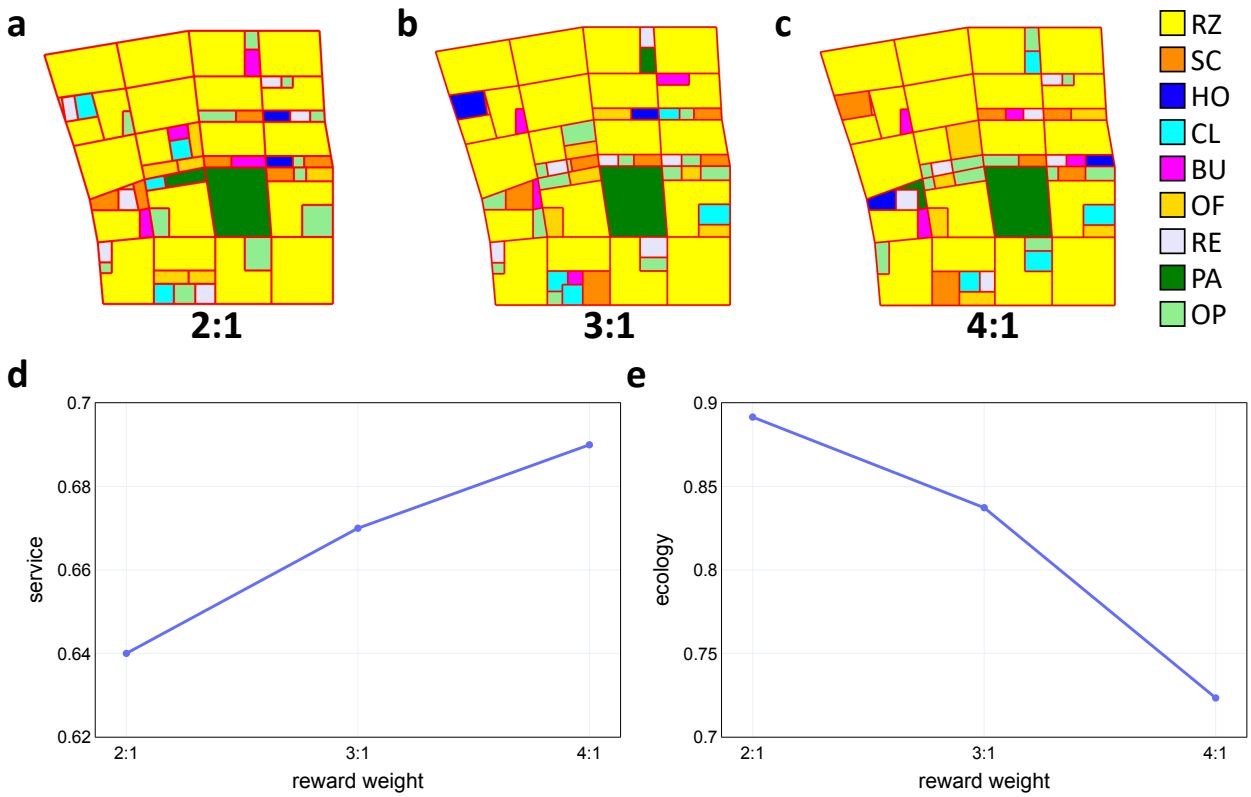
Supplementary Figure 12. Performance of different training samples. The episodic reward of training with 7.50 million samples versus training with 3.75 million samples. Larger data volume plays a critical role in achieving better planning performance.



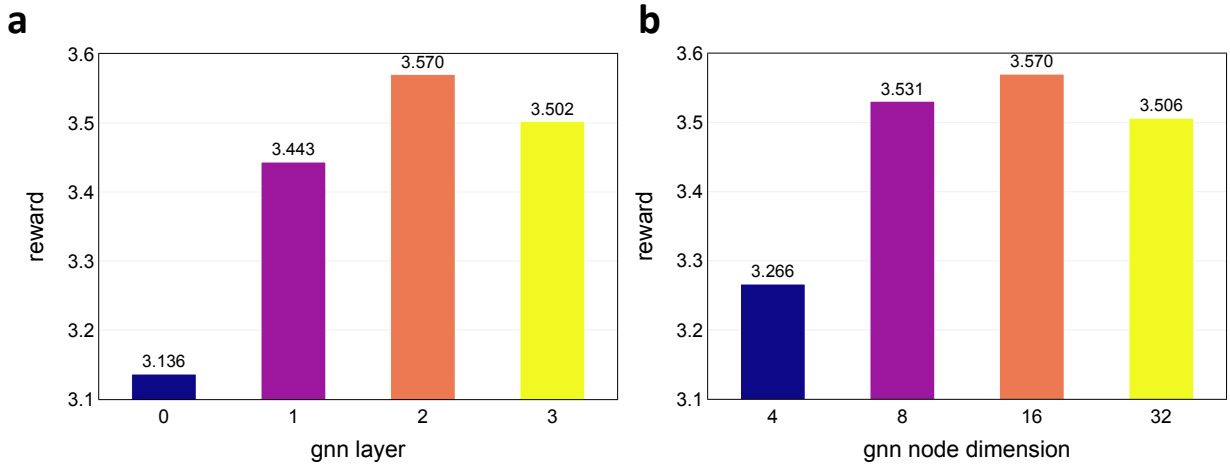
Supplementary Figure 13. Demonstration of generated community renovation plans at different training iterations and their corresponding spatial efficiency performance for the HLG community. a-f, The obtained spatial plans at different iterations. The DRL agent gradually learns to lay out facilities and parks in a more decentralized manner. **g, the corresponding planning performance at different iterations.** We show the corresponding service and ecology metric values for the spatial plans at different iterations. The spatial efficiency with respect to both service and ecology continues to improve during the training process.



Supplementary Figure 14. Demonstration of generated community renovation plans at different training iterations and their corresponding spatial efficiency performance for the DHM community. a-f, The obtained spatial plans at different iterations. The DRL agent gradually learns to lay out facilities and parks in a more decentralized manner. **g, the corresponding planning performance at different iterations.** We show the corresponding service and ecology metric values for the spatial plans at different iterations. The spatial efficiency with respect to both service and ecology continues to improve during the training process.



Supplementary Figure 15. Demonstration of generated community renovation plans with different reward weights for the HLG community and their corresponding planning performance. a-c, The final generated spatial plans under different reward weight ratios. We show the obtained spatial plans trained with reward ratio between service and ecology from 2:1 to 4:1. **d-e, Service and ecology metrics values.** We show the corresponding service and ecology metric values for the spatial plans with different reward weights. The performance can be successfully tuned towards service of ecology by changing their reward weights.



Supplementary Figure 16. Hyper-parameter study. We demonstrate the evaluation reward of our model under different number of **(a) GNN layers** and **(b) GNN node dimensions.**

Supplementary Tables

Supplementary Table. 1. **Abbreviation and color for different land use functions**

Land use function	Abbreviation	Color
Residential	RZ	
School	SC	
Hospital	HO	
Clinic	CL	
Business	BU	
Office	OF	
Recreation	RE	
Park	PA	
Open Space	OP	

Supplementary Table. 2. **Abbreviation and color for road segment and land use boundary**

Segment item	Abbreviation	Color
Road	R	
Boundary	B	

Supplementary Table. 3. **Example of planning needs and requirements**

Name	Residential	School	Hospital	Clinic	Business	Office	Recreation	Park	Open Space
Needs	60%	4	1	3	2	3	3	15%	1
Requirements	20000	10000	10000	2000	10000	10000	2000	15000	2000

Supplementary Table. 4. Information about recruited professional human designers

ID	Educational Background	Registered	Working Years	#Participated Projects
H1	PhD	Yes	9	25
H2	Master	Yes	12	80
H3	Master	Yes	13	100
H4	PhD	Yes	4	10
H5	Master	Yes	3	21
H6	PhD	Yes	9	23
H7	Master	Yes	7	47
H8	Master	Yes	11	18

Supplementary Table. 5. Objective performance in human-AI collaborative workflow

Planner	Real-world HLG		Real-world DHM		Training Time Cost	Planning Time Cost
	Service	Ecology	Service	Ecology		
Human Designer 1	0.60	0.70	0.55	0.80	~10 years	~83 minutes
Human Designer 2	0.70	0.50	0.57	0.76	~10 years	~105 minutes
Human Designer 3	0.70	0.47	0.55	0.72	~10 years	~105 minutes
Human Designer 4	0.66	0.49	0.51	0.60	~10 years	~53 minutes
Human Designer 5	0.67	0.45	0.54	0.47	~10 years	~75 minutes
DRL solution 1	0.67	0.76	0.57	0.79	~2 days	<1 second
DRL solution 2	0.67	0.75	0.63	0.84	~2 days	<1 second
DRL solution 3	0.62	0.80	0.61	0.80	~2 days	<1 second
DRL solution 4	0.62	0.80	0.64	0.84	~2 days	<1 second
DRL solution 5	0.64	0.67	0.61	0.74	~2 days	<1 second

References

[1] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. “Proximal policy optimization algorithms”. In: *arXiv preprint arXiv:1707.06347* (2017).

[2] Leila Kallel and Marc Schoenauer. “Alternative Random Initialization in Genetic Algorithms.” In: *ICGA*. Citeseer. 1997, pp. 268–275.

[3] Jerome Andre, Patrick Siarry, and Thomas Dognon. “An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization”. In: *Advances in engineering software* 32.1 (2001), pp. 49–60.

[4] Ahmed Fawzy Gad. *PyGAD: An Intuitive Genetic Algorithm Python Library*. 2021. arXiv: 2106.06158 [cs.NE].