

# Adaptive 3D UI Placement in Mixed Reality Using Deep Reinforcement Learning

Feiyu Lu\*  
feiyu.lu@jpmchase.com  
JPMorgan Chase & Co.  
New York, NY, USA

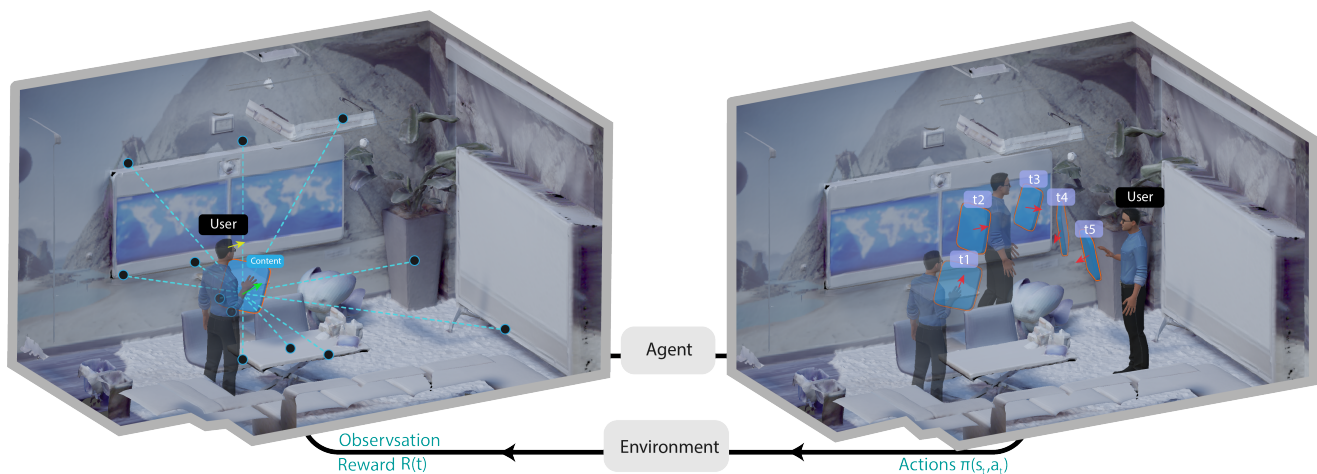
Pranav Deshpande  
pranav.deshpande@jpmchase.com  
JPMorgan Chase & Co.  
New York, NY, USA

Mengyu Chen\*  
mengyu.chen@jpmchase.com  
JPMorgan Chase & Co.  
New York, NY, USA

Cheng Yao Wang  
chengyao.wang@jpmchase.com  
JPMorgan Chase & Co.  
New York, NY, USA

Hsiang Hsu  
hsiang.hsu@jpmchase.com  
JPMorgan Chase & Co.  
New York, NY, USA

Blair MacIntyre  
blair.macintyre@jpmchase.com  
JPMorgan Chase & Co.  
New York, NY, USA



**Figure 1: A demonstration of our reinforcement learning (RL) framework to assist 3D UI placement in dynamic mixed reality environments. According to a RL agent’s observations of user and environmental states (i.e., distance to surrounding environment, the user, and the user’s pose), the RL agent generates a force vector (see red arrow) to maneuver the content in 3D via a physics simulation, with the goal of maximizing the accumulated reward for users in mobile scenarios.**

## ABSTRACT

Mixed Reality (MR) could assist users’ tasks by continuously integrating virtual content with their view of the physical environment. However, where and how to place these content to best support the users has been a challenging problem due to the dynamic nature of MR experiences. In contrast to prior work that investigates optimization-based methods, we are exploring how reinforcement learning (RL) could assist with continuous 3D content placement that is aware of users’ poses and their surrounding environments. Through an initial exploration and preliminary evaluation, our results demonstrate the potential of RL to position content that

\*Both authors contributed equally to this research.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).  
CHI EA '24, May 11–16, 2024, Honolulu, HI, USA  
© 2024 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-0331-7/24/05.  
<https://doi.org/10.1145/3613905.3651059>

maximizes the reward for users on the go. We further identify future directions for research that could harness the power of RL for personalized and optimized UI and content placement in MR.

## KEYWORDS

Mixed reality, adaptive user interface, reinforcement learning, mobile scenarios

### ACM Reference Format:

Feiyu Lu, Mengyu Chen, Hsiang Hsu, Pranav Deshpande, Cheng Yao Wang, and Blair MacIntyre. 2024. Adaptive 3D UI Placement in Mixed Reality Using Deep Reinforcement Learning. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '24)*, May 11–16, 2024, Honolulu, HI, USA. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3613905.3651059>

## 1 INTRODUCTION

Mixed Reality (MR) technologies have the potential to assist users’ tasks by integrating digital content pervasively with the users’ view of their physical environment [21]. Users are empowered to rely on the digital information continuously on-the-go in a variety of

everyday tasks. However, determining the optimal placement of a 3D user interface (UI) in the physical space poses a non-trivial challenge due to the increased degree of freedom and dynamic nature of MR use cases. Since virtual information could be displayed anywhere and anytime in 3D space, it may hinder real-world activities by demanding unnecessary user attention and manual interactions (e.g., moving or hiding digital content) if not designed carefully and updated continuously. Such issues become prominent especially when users are mobile, as an initially useful placement may lose its utility after users alter their poses or spatial positions. In order for MR UIs to better support the users, there are two unsolved challenges: (1) they must seamlessly adapt to dynamic contextual changes such as user’s position, pose, and surroundings, which traditional rule-based adaptations fall short of doing [31]; and (2) they must take multiple adaptation goals into consideration in real time, such as visibility, reachability, and comfort [17, 25].

To realize such adaptive MR UI behaviors, recent research formulates the problem as multi-objective optimization [9, 10, 16, 17, 22, 25, 26, 32]. User’s goals are formulated as a set of objective functions and placements are selected that maximize/minimize these objectives. However, there are two theoretical challenges with optimization-based solutions: (1) the optimization needs to be re-solved each time users move themselves in space, which may be computationally costly and inefficient; (2) these solutions may fail to generalize to other environments or user movement patterns, requiring either reconfiguration or modifications of objectives and constraints.

In this work, we take a novel perspective to address the 3DUI placement challenge in MR. Inspired by its successful applications in dynamic real-world environment such as robotics and automated vehicles, deep reinforcement learning (RL) is employed to make decisions about placement of 3D UIs in dynamic scenarios. As compared to optimization-based approaches in which objectives are defined and observed more intuitively, RL demands careful crafting of the reward functions and a large amount of trials and errors during training. In contrast, it could be more advantageous in handling complex and dynamic scenarios with opportunities to incorporate user preferences, especially when decisions need to be made sequentially. By observing the environments and user states, our proposed RL agent learns to take actions, observe the outcome, and formulate a policy that maximizes long-term reward for the users through its own interactive experiences. To our knowledge, this work is the first that employed RL to place virtual information in 3D MR space when users are mobile. We describe in detail the training configuration and the setup of the models. To assess the potential of our RL-driven UI placement approach in MR, we validate the performance and generalizability of our trained model with a preliminary evaluation. Our results demonstrate the potential of RL to address the 3D UI placement challenges in MR environments by helping users decide optimal placements that maximize the accumulative reward on-the-go. Additionally, we highlight future possibilities about how to further improve our proposed method with state-of-the-art RL methodologies.

In summary, the contributions of our work are three-fold: (1) a novel approach proposed to adapt MR UI placement through state-of-the-art RL algorithms; (2) a preliminary simulation-based

evaluation of our proposed method; (3) future directions identified that leverage RL for personalized and adaptive UI placement in MR.

## 2 RELATED WORK

### 2.1 Adaptive user interfaces in mixed reality

As compared to traditional 2D screens, AR/MR opens more possibilities for information displays due to its higher dimensionality and rich interaction paradigms. View management refers to how virtual information should be displayed and managed in the user’s viewport [3]. Research in view management has primarily investigated label placement, which investigates how 2D UIs should be arranged in the screen plane [1, 2, 30, 36, 42], with less emphasis on content registered in the 3D environment. Grubert and colleagues proposed the concept “Pervasive AR”, in which they argue that future AR/MR experiences should be context-aware to continuously adapt to the users’ environments and tasks [21]. However, it has been challenging to develop such interfaces due to the dynamic nature of MR use cases, the frequent conflicts in user goals, and the theoretically infinite possible placements in 3D space.

MRTK<sup>1</sup> provides solver components to decide placement of UIs based on one or more objectives. However, the adaptations are mostly rule-based executions. The objectives are solved only sequentially which makes it less applicable in dynamic multi-objective scenarios. Recent work proposes to realize such adaptive behaviors for 3D content in MR through optimization, which have been widely studied in 2D applications to decide the layout of UI elements [14, 37, 40, 44]. For example, Lindlbauer et al. implemented an adaptive UI system in MR using integer programming taking into account the user’s task and cognitive load [32]. Belo et al. explored placement of UIs in MR that maximizes ergonomic comfort for users [16]. Cheng et al. explored automatic placement of MR content while optimizing visibility, consistency, and semantic correlations using linear programming [10]. Similar methods were applied in a recent work by Han et al. to blend MR windows on top of physical object meshes, taking into account the properties of physical objects [22]. However, adaptations in these work mostly took place in a discrete solution space, which may not guarantee optimality when users are moving continuously. To enable adaptations in continuous spaces, Belo et al. explored simulated annealing for solving non-linear objectives for MR UI adaptations. Johns et al. explored genetic algorithms for deriving optimal placements along the pareto frontier [25, 26]. This work demonstrated the potential of optimization-based approach for UI adaptations in MR.

In this work, we take a new perspective by applying RL to assist with content placement in MR environments, by training an AI agent how to “drive” a piece of UI content to maximize user reward.

### 2.2 Reinforcement learning for interface adaptations

RL is a category of machine learning that enables intelligent agents to learn how to take actions in a dynamic environment with the goal to maximize long-term rewards temporally downstream. It has demonstrated great potential in scenarios where decisions need to be planned sequentially, such as robotics [29], automotive vehicles

<sup>1</sup><https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/>

[18, 47], gaming [28], and crowd simulations [6]. We argue that RL could be a suitable approach to address the 3D UI placement challenge in MR due to its dynamic and user-centric nature. Unlike supervised learning which relies on large amount of labelled data, RL learns by observing and interacting with the environment on itself, which makes it powerful (1) for handling complex, unexpected, and dynamic scenarios, (2) when current decisions can influence future states, and (3) when large-scale labelled training datasets are challenging to obtain. Due to such properties, RL has been recently explored as a new strategy to achieve interface adaptations in both 2D and MR spaces. Todi et al. used model-based RL to adapt item ranking in a 2D menu [43]. Yu et al. studied using RL to determine for timing of intelligent assistance during object selection tasks in MR [46]. Gebhardt et al. explored using RL to adapt the visibility of 2D labels in VR based on gaze data [20]. Chen et al. employed RL for AR label placement in the view plane during basketball matches [8], which we consider the most relevant to our work. Their results show promising potential of RL to intelligently assist 2D UI placements in 3D environments when the user’s viewport is static. However, there has been a lack of exploration of how RL could be applied to assist 3D UI placement and adaptation, where (1) UIs could be registered anywhere in the 3D physical world beyond a 2D camera plane and (2) the user’s egocentric viewport dynamically changes in mobile scenarios. Our work sheds light on such possibilities by training a RL agent that is capable of maneuvering UIs in 3D constantly according to its observations of how the users’ and environmental states change.

### 3 PROBLEM FORMULATION

#### 3.1 Environment & User Simulation

To construct a realistic training environment for the RL agent to freely explore and learn, we utilize the Replica dataset [41], which contains high-fidelity 3D scans of indoor scenes with segmented objects and bounding boxes. This grants us full understanding of a simulated physical environment, including position, orientation, and size of floors, walls, and furniture. We picked four scenes from the dataset covering small/large office/living room spaces, two for training and two for validation. In our simulated environment, we replicate the actions of a simulated user traversing the space thoroughly and adequately. The simulated user would make stops at random locations on a 3D Cartesian grid situated inside the room, mimicking a variety of user poses and locations. These include, but not limited to, sitting on a sofa, standing in front of a whiteboard, sitting around a table, all while facing a wall, furniture, or empty space (see Figure 2 (a)). The RL agent is then designed to learn 3D UI adaptations that maximize the reward for the presented simulated user along the horizon.

We understand that such approaches may introduce limitations depending on how well the simulated user behaves in comparison with actual users. We argue that our approach could bring unique benefits since real-world user trajectory data is largely unavailable in well-labeled indoor spaces. Applying synthetic user data could be a scalable strategy to train and gauge the feasibility of RL-based methods. As an initial exploration, we started with one piece of virtual content in the environment, the movement of which the RL agent takes full control of.

#### 3.2 RL Agent

We employ model-free RL which does not require prior knowledge of the transition model. The RL agent leverages proximal policy optimization (PPO), a state-of-the-art deep policy gradient algorithm which has demonstrated great potential in a variety of benchmarks and applications [4, 35, 39]. In the following sections, we highlight our designs of three core components: (1) observations (what information the agent collects from the environment), (2) actions (what actions the agent produces to control the placement of 3D UI in MR), and (3) reward (how the agent is rewarded or penalized based on its action).

**3.2.1 Observations.** In each time step  $t$ , the agent receives the following observations: (1)  $\{Ray_1, Ray_2 \dots Ray_{22}\}$ : twenty-two rays evenly cast around the virtual content to collect information about its surrounding in the current time step, including whether any ray hits the physical environment mesh, and if so, the start/end point of each ray hit, the hit object type, and the length of the ray. As such, the RL agent observes the content’s relative position to its surrounding environment; (2)  $\{Pos_c, Rot_c, Dir_c\} \in \mathbb{R}^3$ : the virtual content’s position, rotation, and facing direction in the local space of the environment; and (3)  $\{V_c, Dist\} \in \mathbb{R}$ : the virtual content’s local velocity and its euclidean distance in relation to the user; (4)  $\{Pos_u, Dir_u\} \in \mathbb{R}^3$ : the user’s current position and facing direction in space (see Figure 1 left). The observations were normalized before feeding into the network.

**3.2.2 Action Space.** Based on the observations, the RL agent produces a three-dimensional continuous action  $\{f_x, f_y, f_z\} \in [-1, 1]$ , symbolizing a 3D force vector to apply to the virtual UI content (see Figure 1 right). With physic simulation, the virtual content could be freely maneuvered, accelerated, and decelerated in 3D based on the applied force vector. The RL agent only controls the content’s position in space. The content was always oriented in a billboard fashion to the users, ensuring maximum visibility. The action was produced every five time steps.

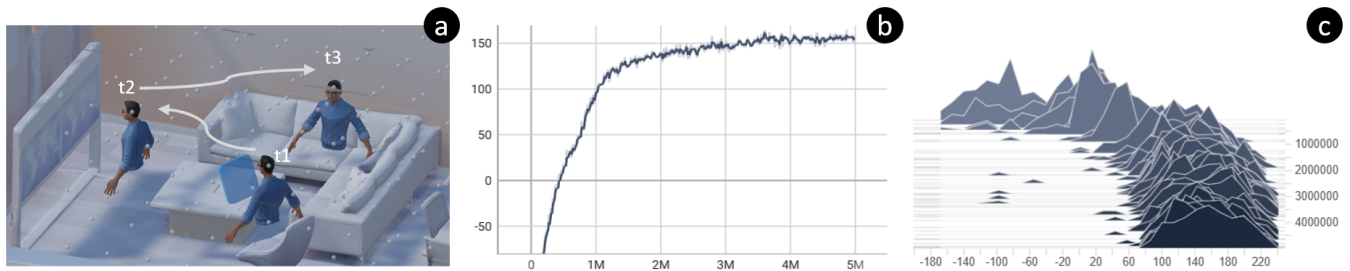
**3.2.3 Reward Design.** Design of the reward functions plays critical roles in enabling the RL agent to actively query to evaluate the quality of a given action or state. In line with previous work on MR layout optimizations [10, 17, 26, 32], we aim for 3D UI placements that are continuously visible, reachable, demonstrate realistic physical behaviors, and stable. We designed the final reward to be the aggregation of four rewards  $\{R_{\text{visibility}}, R_{\text{reachability}}, R_{\text{physicality}}, R_{\text{stability}}\}$ :

$$R_{\text{visibility}} = \begin{cases} 0.1 \cdot P_{\text{percent}}, & \text{if } P_{\text{percent}} > 0 \\ -0.1, & \text{otherwise} \end{cases} \quad (1)$$

$$R_{\text{reachability}} = \begin{cases} 0.1 \cdot e^{\frac{(Dist-0.5)^4}{-0.05}}, & \text{if UI is in front of the user} \\ -0.1, & \text{otherwise} \end{cases} \quad (2)$$

$$R_{\text{physicality}} = \begin{cases} -0.01, & \text{if UI overlaps with physical objects} \\ 0.01, & \text{otherwise} \end{cases} \quad (3)$$

$$R_{\text{stability}} = \begin{cases} 0, & \text{if the user is moving} \\ 0.01, & \text{if the user is stationary and } V_c < 0.3 \\ -0.01, & \text{if the user is stationary and } V_c \geq 0.3 \end{cases} \quad (4)$$



**Figure 2: (a) Simulating user movement during training by dynamically interpolating an avatar along a 3D Cartesian grid facing a random direction every few seconds; (b) the accumulative reward per episode (y-axis) in relation to the number of total steps (x-axis); (c) the histogram of the reward distribution (x-axis) in relation to the total number of steps (y-axis).**

We acknowledge that designing quality reward functions which empower the agents to learn, explore, and exploit demands non-trivial effort in the domains of RL. The current reward design was developed after extensive empirical testing and benchmarks.  $P_{percent}$  represents the aggregation of percentages of the pixels that belong to the content within the user’s camera render texture and the proportion of the content that is visible for the current time step (i.e., how much of the content is visible and how much it occupies the user’s viewport);  $Dist$  represents the current euclidean distance from the content to the user;  $V_c$  represent the current velocity of the content measured by meters/second. The reachability reward applied a kernel function, as inspired by [10]. The scale of the rewards (i.e., 0.1, 0.01) followed prior guidelines [7, 8, 20] and our testing. The thresholds for  $V_c$  and  $P_{percent}$  were determined empirically.

### 3.3 Training Configuration & Results

The Unity3D engine was employed to simulate the 3D training environment, which consists of the simulated physical environment, the user, and the virtual content. The ML-Agents toolkit<sup>2</sup> was leveraged for training the RL agent [27]. The toolkit bridges information collected in the Unity3D engine with a PyTorch backend to run the deep RL training.

The PPO training network consists of (1) an encoder which encodes the observations and states of the simulation environment; (2) an actor network to learn the optimal policy  $\pi(s_t, a_t)$  for the agent, and (3) a critic network which estimates the accumulated reward  $R(t)$  for a given action. Both the actor and critic networks were designed with two fully-connected hidden layers with 128 units. The training was configured with six parallel training instances to speed up the process. Each episode consisted of training for around one thousand steps, concluding once the maximum step limit was reached. The entire training lasted for five million steps in total, taking around six hours on a desktop PC with a dedicated Nvidia GeForce RTX3080 GPU, an Intel i9 24-core CPU, and a 64 GB RAM. The learning rate was initialized as  $3e - 4$ , which decayed linearly until max step is reached so the training converges stably. We used a buffer size of 409,600 and a batch size of 2,048 for the agents to gather enough information from its experiences to learn the policy. Figure 2 (b) shows the average accumulated reward per episode in relation to the number of total steps, and Figure 2 (c) shows the histogram distribution of the rewards. The reward increased

effectively during training while its standard deviation decreased, indicating the successful learning of a policy by the RL agent.

## 4 PRELIMINARY EVALUATION

### 4.1 Setup

We conducted an initial evaluation on our model by placing the RL agent in each of the four Replica environments (two training and two validation environments) and running a simulation continuously for ten thousand steps, with a simulated virtual user randomly traversing the environment. The RL model was evaluated based on three considerations:

- **Heuristics:** A set of heuristics to evaluate the placement continuously derived by the RL model, including visible UI % (i.e., average percent of the UI that is visible to the user each time step), non-collision % (i.e., percentage of steps that the UI does not overlap / collide with any physical object), distance offset (i.e., average absolute distance offset from the UI to the user in meters as compared to a half-meter baseline), and speed (i.e., the average speed of the content per time step in meters per second).
- **Generalizability:** In addition to the training scenes, we picked two additional indoor scenes as validation sets. We collected the same heuristics for the different indoor scenes to see if the agent could maintain ideal results while experiencing an environment that it has no experience in.
- **Dynamics:** We introduced moving obstacles to symbolize moving people or objects that the RL-controlled UI should avoid. The obstacle was represented by a 1 by 1 by 1 meter cube which moved randomly in the environment (see Figure 3 (b)). We collected the same heuristics with and without the presence of these dynamic obstacles to gauge the ability of the model to react to unexpected dynamic changes that take place around the users.

### 4.2 Results

As shown in Table 1, our results demonstrated the potential of RL-based approach for adaptive 3D UI placement. In more static environments, for the environments that the agent was trained upon, more than 91% of the UI was continuously visible. It did not collide with any environmental mesh more than 94% of the time. It also maintained an arm-reachable distance to the users. In the two validation environments that the agents have not experienced, our results show that UI could still stay in reasonable locations with around 85% of itself being continuously visible to

<sup>2</sup><https://github.com/Unity-Technologies/ml-agents>



Figure 3: (a) A snapshot of the training in which a UI driven by RL gauges its distances to surroundings and simulated user poses; (b) a snapshot of the dynamic obstacle test, in which the content (blue square) maintains visibility, reachability and avoiding collision with a presence of a randomly moving obstacle (red cube); (c) a snapshot of the placement result running our trained RL model.

Table 1: The statistics showing the average value of each heuristic in the simulation testing (T for training, and V for validation).

	Static Environments				Dynamic Obstacles			
	Env1 (T)	Env2 (T)	Env3 (V)	Env4 (V)	Env1 (T)	Env2 (T)	Env3 (V)	Env4 (V)
Visible UI %	93.28%	91.74%	86.50%	84.05%	85.27%	88.05%	83.38%	82.92%
Non-Collision %	94.09%	94.80 %	87.06%	82.18%	82.37%	90.32%	84.48%	85.86%
Distance Offset	0.06	0.04	0.11	0.21	0.11	0.17	0.22	0.31
Speed	0.68	0.72	0.69	0.74	0.72	0.75	0.77	0.84

the users each time step. It maintained a distance slightly further away from the user, but still within a reachable distance. Similarly, in dynamic settings when moving obstacles are introduced in the environments, our model still achieved good placements in both the training and validation environments with negligible performance drops especially on the validation environments, which the drops of visibility/physicality were capped at 3.12%. The UI could maintain its reachability level even when the obstacle slides in view (see Figure 3 (b)), and avoid most upcoming collisions to keep its visibility to the users. As a result, the average distance from the UI to the user slightly increased to compensate for such trade-offs. This demonstrates the RL agent’s ability to adapt and generalize to environments and certain degrees of unexpected changes that it has not experienced before. However, we are aware that the higher performance drops on the two training environments with dynamic obstacles might be a sign of overfitting. Future research is required to further validate our model in more diverse environments. Figure 3 (c) showcases the output of the model in real time, in which RL places the UI while avoiding collision with the whiteboard, in the mean time ensuring visibility and reachability to the users.

One limitation we observed is that the UI still moved at a relatively high speed, with an average more than the level we set in the reward function. One reason for this could be that our simulation symbolized a busy setting in which users relocate frequently, so there were less opportunities for the RL agent to experience the static reward. Future work is needed in terms of how to further increase the stability of the UI.

## 5 FUTURE OPPORTUNITIES

We are excited about the positive outcome of our initial exploration. Here, we would like to highlight future opportunities and challenges for leveraging RL for content placement in dynamic 3D MR environments:

- *More considerations on reward formulations.* As an initial exploration, our current RL setup did not consider other aspects while formulating the reward, such as semantics [10] and affordances [9, 45] of physical objects, content type (e.g., 2D vs 3D, text-heavy vs. images), spatial consistency [15], and user ergonomics [16]. Recent work has highlighted the benefits of such considerations. Future research should take into account these aspects and explore how to incorporate them into current reward designs, as well as how to further improve stability of the UI content.
- *VR simulation vs. actual AR/MR settings.* Our training assumes good understanding of environmental and user status by simulating AR/MR environments in VR. Training in actual AR/MR spaces may surface more challenges due to the extensive trials and errors RL needs and the imperfect understandings of the physical world. Future work is needed in order to bridge the gaps between simulated and actual user behaviors. For example, utilizing generative AI models, synthetic data could be generated from user’s actual movement trajectories for training the RL agent [12]. As such, the agent could learn from common user behavioral patterns to further optimize downstream performance.
- *Multiple users and UIs.* Our exploration demonstrated the initial feasibility of RL controlling the placements of one piece of 3D UI content. However, users frequently need access to multiple applications with different relevancy to their tasks. To support multiple content elements, multi-agent RL frameworks could be leveraged for training cooperative behaviors among multiple agents so they make the best trade-offs [13, 33]. Same strategies may also apply to multi-user collaborative settings. Future research could investigate the feasibility of these directions.
- *RL with human feedback (RLHF) for preference learning.* Research has demonstrated that AI models with better performance may not be considered more useful by the users [38]. Similarly, the quality of 3D UI placements in MR from the user’s perspective

may often be highly subjective. Thus, it is critical to derive placements that align well with user expectations, preferences and agency [34]. Recent work has revealed the potential of RL to incorporate human-in-the-loop preference learning with a handful of queries, which generalizes well to new tasks without having to retrain the models [11, 23]. This empowers preference elicitation on-the-go, which could be more flexible as compared to constraints-based methods in optimizations. Future research could further investigate the potential of RLHF for deriving personalized and preferable UI placements in MR spaces.

- **Model-based RL.** Our exploration leveraged model-free RL, which could be a powerful technique for tackling real-world situations when the environment is noisy, complex, and unpredictable. The downside is model-free RL requires an extensive amount of trials for the agent to learn a good policy. In the field of HCI, a wide variety of models have been developed to predict human motor and cognitive performances [5, 19, 24]. This makes *model-based RL* viable as an alternative, in which the RL agent simulates consequences of its actions without executing them to plan ahead accordingly. As such, the training could be done more efficiently. Recent work by Todi et al. demonstrated the feasibility of *model-based RL* for adapting 2D menus leveraging models such as Fitts' law [43]. Future research could further explore such directions for MR use cases, in which user performance models are leveraged.
- **Comparisons with optimization-based methods.** Due to the scope of this work, we did not compare RL with optimization-based methods which demonstrated great potential in recent MR adaptive UI work, such as linear/integer programming [10, 22, 32], simulated annealing [17], and genetic algorithms [25, 26]. Future research is needed to explore the trade-offs among these method in both experimentally-controlled and ecologically-valid settings of MR UI scenarios.
- **More explorations on generalizability.** Though our results provide initial evidence on the generalizability of our RL-based approach, the training was conducted in less diverse environments. Introducing a higher degree of variability and intricacy may potentially facilitate a more robust learning process (e.g., multi-floor environments with moving bystanders), allowing the model to adapt more effectively to diverse UI scenarios. We aim to simulate more accurate real-world scenarios, thereby enabling the model to acquire a broader range of behaviors.

## 6 CONCLUSION

In this work, we explored the potential of reinforcement learning (RL) for 3D user interface (UI) placement in mixed reality (MR) environments. By having a RL agent interactively engaging with a simulated physical environment with trials and errors, it could learn a policy that assist the placements of 3D UIs in order to maximize the utility for the users. We trained a RL agent that is capable of assisting the placements of 3D UI continuously, demonstrated the potential of our approach through a preliminary simulated evaluations that explore the generalizability of the model to new and dynamic environments, and called attention to challenges such as UI stability and overfitting. Based on our results, we further highlight opportunities for future research that explore RL for personalized and adaptive information displays in MR environments.

## REFERENCES

- [1] R. Azuma and C. Furmanski. 2003. Evaluating label placement for augmented reality view management. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings*. 66–75. <https://doi.org/10.1109/ISMAR.2003.1240689>
- [2] Michael A Bekos, Benjamin Niedermann, and Martin Nöllenburg. 2019. External labeling techniques: A taxonomy and survey. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 833–860.
- [3] Blaine Bell, Steven Feiner, and Tobias Höllerer. 2001. View Management for Virtual and Augmented Reality. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology (Orlando, Florida) (UIST '01)*. Association for Computing Machinery, New York, NY, USA, 101–110. <https://doi.org/10.1145/502348.502363>
- [4] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. 2019. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680* (2019).
- [5] Florent Cabric, Emmanuel Dubois, and Marcos Serrano. 2021. A Predictive Performance Model for Immersive Interactions in Mixed Reality. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 202–210. <https://doi.org/10.1109/ISMAR52148.2021.00035>
- [6] Panayiotis Charalambous, Julien Pettre, Vassilis Vassiliades, Yiorgos Chrysanthou, and Nuria Pelechano. 2023. GREIL-Crowds: Crowd Simulation with Deep Reinforcement Learning and Examples. *ACM Trans. Graph.* 42, 4, Article 137 (jul 2023), 15 pages. <https://doi.org/10.1145/3592459>
- [7] Changan Chen, Yuejiang Liu, Sven Kreiss, and Alexandre Alahi. 2019. Crowd-Robot Interaction: Crowd-Aware Robot Navigation With Attention-Based Deep Reinforcement Learning. In *2019 International Conference on Robotics and Automation (ICRA)*. 6015–6022. <https://doi.org/10.1109/ICRA.2019.8794134>
- [8] Zhu-Tian Chen, Daniele Chiappalupi, Tica Lin, Yalong Yang, Johanna Beyer, and Hanspeter Pfister. 2023. RL-LABEL: A Deep Reinforcement Learning Approach Intended for AR Label Placement in Dynamic Scenarios. *IEEE Transactions on Visualization and Computer Graphics* (2023), 1–11. <https://doi.org/10.1109/TVCG.2023.3326568>
- [9] Yi Fei Cheng, Christoph Gebhardt, and Christian Holz. 2023. InteractionAdapt: Interaction-Driven Workspace Adaptation for Situated Virtual Reality Environments. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (San Francisco, CA, USA) (UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 109, 14 pages. <https://doi.org/10.1145/3586183.3606717>
- [10] Yi Fei Cheng, Yukang Yan, Xin Yi, Yuanchun Shi, and David Lindlbauer. 2021. SemanticAdapt: Optimization-Based Adaptation of Mixed Reality Layouts Leveraging Virtual-Physical Semantic Connections. In *The 34th Annual ACM Symposium on User Interface Software and Technology (Virtual Event, USA) (UIST '21)*. Association for Computing Machinery, New York, NY, USA, 282–297. <https://doi.org/10.1145/3472749.3474750>
- [11] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems* 30 (2017).
- [12] Jeremy Chu, Dongsheng An, Yan Ma, Wenzhe Cui, Shumin Zhai, Xianfeng David Gu, and Xiaojun Bi. 2023. WordGesture-GAN: Modeling Word-Gesture Movement with Generative Adversarial Network. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (Hamburg, Germany) (CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 287, 15 pages. <https://doi.org/10.1145/3544548.3581279>
- [13] Andrew Cohen, Ervin Teng, Vincent-Pierre Berges, Ruo-Ping Dong, Hunter Henry, Marwan Mattar, Alexander Zook, and Sujoy Ganguly. 2021. On the use and misuse of absorbing states in multi-agent reinforcement learning. *arXiv preprint arXiv:2111.05992* (2021).
- [14] Niraj Ramesh Dayama, Kashyap Todi, Taru Saarelainen, and Antti Oulasvirta. 2020. GRIDS: Interactive Layout Design with Integer Programming. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (Honolulu, HI, USA) (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3313831.3376553>
- [15] Barrett Ens, Eyal Ofek, Neil Bruce, and Pourang Irani. 2015. Spatial Constancy of Surface-Embedded Layouts across Multiple Environments. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction (Los Angeles, California, USA) (SUI '15)*. Association for Computing Machinery, New York, NY, USA, 65–68. <https://doi.org/10.1145/2788940.2788954>
- [16] João Marcelo Evangelista Belo, Anna Maria Feit, Tiare Feuchtner, and Kaj Grønbaek. 2021. XRgonomics: Facilitating the Creation of Ergonomic 3D Interfaces. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (Yokohama, Japan) (CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 290, 11 pages. <https://doi.org/10.1145/3411764.3445349>
- [17] João Marcelo Evangelista Belo, Mathias N. Lystbæk, Anna Maria Feit, Ken Pfeuffer, Peter Kån, Antti Oulasvirta, and Kaj Grønbaek. 2022. AUIT – the Adaptive User Interfaces Toolkit for Designing XR Applications. In *Proceedings of the 35th*

- Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) (UIST '22). Association for Computing Machinery, New York, NY, USA, Article 48, 16 pages. <https://doi.org/10.1145/3526113.3545651>
- [18] Abdur R. Fayjie, Sabir Hossain, Doukhi Oualid, and Deok-Jin Lee. 2018. Driverless Car: Autonomous Driving Using Deep Reinforcement Learning in Urban Environment. In *2018 15th International Conference on Ubiquitous Robots (UR)*. 896–901. <https://doi.org/10.1109/URAL.2018.8441797>
- [19] Paul M Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology* 47, 6 (1954), 381.
- [20] Christoph Gebhardt, Brian Hecox, Bas van Opheusden, Daniel Wigdor, James Hillis, Otmar Hilliges, and Hrvoje Benko. 2019. Learning Cooperative Personalized Policies from Gaze Data. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) (UIST '19). Association for Computing Machinery, New York, NY, USA, 197–208. <https://doi.org/10.1145/3332165.3347933>
- [21] Jens Grubert, Tobias Langlotz, Stefanie Zollmann, and Holger Regenbrecht. 2017. Towards Pervasive Augmented Reality: Context-Awareness in Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics* 23, 6 (2017), 1706–1724. <https://doi.org/10.1109/TVCG.2016.2543720>
- [22] Violet Yinuo Han, Hyunung Cho, Kiyosu Maeda, Alexandra Ion, and David Lindlbauer. 2023. BlendMR: A Computational Method to Create Ambient Mixed Reality Interfaces. *Proc. ACM Hum.-Comput. Interact.* 7, ISS, Article 436 (nov 2023), 25 pages. <https://doi.org/10.1145/3626472>
- [23] Donald Joseph Hejna III and Dorsa Sadigh. 2023. Few-shot preference learning for human-in-the-loop rl. In *Conference on Robot Learning*. PMLR, 2014–2025.
- [24] William E Hick. 1952. On the rate of gain of information. *Quarterly Journal of experimental psychology* 4, 1 (1952), 11–26.
- [25] Christoph Albert Johns, João Marcelo Evangelista Belo, Anna Maria Feit, Clemens Nylandstedt Klokmoose, and Ken Pfeuffer. 2023. Towards Flexible and Robust User Interface Adaptations With Multiple Objectives. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) (UIST '23). Association for Computing Machinery, New York, NY, USA, Article 108, 17 pages. <https://doi.org/10.1145/3586183.3606799>
- [26] Christoph Albert Johns, João Marcelo Evangelista Belo, Clemens Nylandstedt Klokmoose, and Ken Pfeuffer. 2023. Pareto Optimal Layouts for Adaptive Mixed Reality. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (CHI EA '23). Association for Computing Machinery, New York, NY, USA, Article 223, 7 pages. <https://doi.org/10.1145/3544549.3585732>
- [27] Arthur Juliani, Vincent-Pierre Berges, Ervin Teng, Andrew Cohen, Jonathan Harper, Chris Elion, Chris Goy, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. 2020. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627* (2020). <https://arxiv.org/pdf/1809.02627.pdf>
- [28] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. 2019. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374* (2019).
- [29] Jens Kober, J. Andrew Bagnell, and Jan Peters. 2013. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* 32, 11 (2013), 1238–1274. <https://doi.org/10.1177/0278364913495721> arXiv:<https://doi.org/10.1177/0278364913495721>
- [30] Thomas Köppel, M. Eduard Gröller, and Hsiang-Yun Wu. 2021. Context-Responsive Labeling in Augmented Reality. In *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)*. 91–100. <https://doi.org/10.1109/PacificVis52677.2021.00020>
- [31] David Lindlbauer. 2022. The Future of Mixed Reality is Adaptive. *XRDS* 29, 1 (oct 2022), 26–31. <https://doi.org/10.1145/3558191>
- [32] David Lindlbauer, Anna Maria Feit, and Otmar Hilliges. 2019. Context-Aware Online Adaptation of Mixed Reality Interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) (UIST '19). Association for Computing Machinery, New York, NY, USA, 147–160. <https://doi.org/10.1145/3332165.3347945>
- [33] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems* 30 (2017).
- [34] Feiyu Lu and Yan Xu. 2022. Exploring Spatial UI Transition Mechanisms with Head-Worn Augmented Reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) (CHI '22). Association for Computing Machinery, New York, NY, USA, Article 550, 16 pages. <https://doi.org/10.1145/3491102.3517723>
- [35] Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Sungmin Bae, et al. 2020. Chip placement with deep reinforcement learning. *arXiv preprint arXiv:2004.10746* (2020).
- [36] Jason Orlosky, Kiyoshi Kiyokawa, Takumi Toyama, and Daniel Sonntag. 2015. Halo Content: Context-Aware Viewspace Management for Non-Invasive Augmented Reality. In *Proceedings of the 20th International Conference on Intelligent User Interfaces* (Atlanta, Georgia, USA) (IUI '15). Association for Computing Machinery, New York, NY, USA, 369–373. <https://doi.org/10.1145/2678025.2701375>
- [37] Antti Oulasvirta, Niraj Ramesh Dayam, Morteza Shiripour, Maximilian John, and Andreas Karrenbauer. 2020. Combinatorial Optimization of Graphical User Interface Designs. *Proc. IEEE* 108, 3 (2020), 434–464. <https://doi.org/10.1109/JPROC.2020.2969687>
- [38] Quentin Roy, Futian Zhang, and Daniel Vogel. 2019. Automation Accuracy Is Good, but High Controllability May Be Better. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–8. <https://doi.org/10.1145/3290605.3300750>
- [39] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
- [40] Michael Hunter Shumin Zhai and Barton A. Smith. 2002. Performance Optimization of Virtual Keyboards. *Human-Computer Interaction* 17, 2-3 (2002), 229–269. <https://doi.org/10.1080/07370024.2002.9667315> arXiv:<https://www.tandfonline.com/doi/pdf/10.1080/07370024.2002.9667315>
- [41] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. 2019. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797* (2019).
- [42] Markus Tatzgern, Denis Kalkofen, Raphael Grasset, and Dieter Schmalstieg. 2014. Hedgehog labeling: View management techniques for external labels in 3D space. In *2014 IEEE Virtual Reality (VR)*. 27–32. <https://doi.org/10.1109/VR.2014.6802046>
- [43] Kashyap Todi, Gilles Bailly, Luis Leiva, and Antti Oulasvirta. 2021. Adapting User Interfaces with Model-Based Reinforcement Learning. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (CHI '21). Association for Computing Machinery, New York, NY, USA, Article 573, 13 pages. <https://doi.org/10.1145/3411764.3445497>
- [44] Kashyap Todi, Daryl Weir, and Antti Oulasvirta. 2016. Sketchplore: Sketch and Explore with a Layout Optimiser. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems* (Brisbane, QLD, Australia) (DIS '16). Association for Computing Machinery, New York, NY, USA, 543–555. <https://doi.org/10.1145/2901790.2901817>
- [45] Robert Xiao, Julia Schwarz, Nick Throm, Andrew D. Wilson, and Hrvoje Benko. 2018. MRTouch: Adding Touch Input to Head-Mounted Mixed Reality. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1653–1660. <https://doi.org/10.1109/TVCG.2018.2794222>
- [46] Difeng Yu, Ruta Desai, Ting Zhang, Hrvoje Benko, Tanya R. Jonker, and Aakar Gupta. 2022. Optimizing the Timing of Intelligent Suggestion in Virtual Reality. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) (UIST '22). Association for Computing Machinery, New York, NY, USA, Article 6, 20 pages. <https://doi.org/10.1145/3526113.3545632>
- [47] Yi Zhang, Ping Sun, Yuhuan Yin, Lin Lin, and Xuesong Wang. 2018. Human-like Autonomous Vehicle Speed Control by Deep Reinforcement Learning with Double Q-Learning. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. 1251–1256. <https://doi.org/10.1109/IVS.2018.8500630>

## 7 APPENDIX - DISCLAIMER

This paper was prepared for informational purposes by the Global Technology Applied Research center of JPMorgan Chase & Co. This paper is not a product of the Research Department of JPMorgan Chase & Co. or its affiliates. Neither JPMorgan Chase & Co. nor any of its affiliates makes any explicit or implied representation or warranty and none of them accept any liability in connection with this paper, including, without limitation, with respect to the completeness, accuracy, or reliability of the information contained herein and the potential legal, compliance, tax, or accounting effects thereof. This document is not intended as investment research or investment advice, or as a recommendation, offer, or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.