HabiCrowd: A High Performance Simulator for Crowd-Aware Visual Navigation

An Vuong[1], Toan Nguyen[1], Minh Nhat Vu[2,3,*], Baoru Huang[4], H.T.T Binh[5], Thieu Vo[6], Anh Nguyen[7]

***Abstract*— Visual navigation, a foundational aspect of Embodied AI (E-AI) and robotics has been extensively studied in the past few years. While many 3D simulators have been introduced for the visual navigation tasks, scarcely works have combined human dynamics, creating the gap between simulation and real-world applications. Furthermore, current 3D simulators incorporating human dynamics have several limitations, particularly in terms of computational efficiency, which is a promise of modern simulators. To overcome these issues, we introduce HabiCrowd, the new standard benchmark for crowd-aware visual navigation that includes a crowd dynamics model with diverse human settings into photorealistic environments. Empirical evaluations demonstrate that our proposed human dynamics model achieves state-of-the-art performance in collision avoidance while exhibiting superior computational efficiency compared to its counterparts. We leverage HabiCrowd to conduct several comprehensive studies on crowd-aware visual navigation tasks and human-robot interactions. The source code and data can be found at https://habicrowd.github.io/.**

I. Introduction

Embodied AI (E-AI), the intersection between computer vision, machine learning, and robotics, has gained interest amongst scientists in the past few years [1]. Training E-AI agents that can perceive, reason, and interact with the environment offers significant potential for sim2real applications [2]. The progress in E-AI research has been rapidly accelerated [3] thanks to the developments of fast and high-fidelity 3D simulators [4] that enable large-scale computational parallelization [5]. Among all E-AI studies, one of the most fundamental problems is visual navigation [6], which involves training an agent to navigate to a given goal using perception from the sensory inputs [7]. While many benchmarks have been proposed to address the visual navigation tasks [5], [8], most of them assume that navigating environments are static and do not consider human dynamics. Conversely, agents typically interact within dynamic environments, particularly those featuring humans whose positions constantly change [9]. If robots are to efficiently assist humans in routine duties, they must possess the capability to navigate effectively in *crowded and dynamic environments* [10]. Therefore, human dynamics is an essential factor that needs to be thoroughly examined when developing E-AI benchmarks [11]. This paper examines visual navigation in the presence of human dynamics to bridge the gap between simulation and real-life scenarios.

The visual navigation task with the inclusion of human dynamics is widely referred to as the crowd-aware visual navigation task in the robotics community [12]. This task is challenging because the agent needs to learn the patterns of human actions [13], which are complex and unstructured [14]. Furthermore, it is infeasible to predict the motion dynamics in crowded habitats as the motion of each person is typically affected by their neighbours [15]. Most prior works train agents using 2D simulators [9], [16], which is impractical since robots perceive human activities via visual sensors rather than mathematical social force models [14].

Recent attempts to add dynamic humans to 3D photorealistic simulators [17], [18] have shown limitations in human dynamics models and world scale. For instance, the limited number of humans employed in [17] can hinder generalization in human recognition [15]. Additionally, one of the most critical drawbacks of recent 3D simulators integrating human dynamics lies in their computational efficiency [19], even though E-AI simulators strive to facilitate rapid rendering and parallelization [4]. In general, the lack of 3D simulators with well-designed human dynamics reduces the applicability of developed methods, highlighting the need for a standardized 3D benchmark that integrates human behaviors.

Motivated by these shortcomings, we introduce **HabiCrowd**, a new benchmark for the crowd-aware visual navigation task. HabiCrowd includes comprehensive social force-based human dynamics with diverse density settings operating across numerous high-fidelity scenes from the Habitat-Matterport 3D dataset (HM3D) [7]. To assess the performance of HabiCrowd, several experiments are conducted, including benchmarking the human dynamics model, rendering speed, and memory utilization. Experimental results reveal that HabiCrowd achieves collision-free while running significantly faster than the state-of-the-art human dynamics model. Additionally, our HabiCrowd also exhibits a remarkable advantage in rendering speed and memory utilization compared to other related benchmarks. Upon HabiCrowd, we propose a new metric for crowd-aware navigation and benchmark two visual navigation tasks. The outcomes highlight the importance of human dynamics in navigating procedures. We also demonstrate that our new simulator enables various studies regarding human density and human-robot interactions. Our contributions are summarized as follows:

- We present HabiCrowd, a new dataset and benchmark for crowd-aware visual navigation. Experiments show that our simulator excels in terms of performance,

[1] FPT Software AI Center, Vietnam anvd2@fpt.com
[2] Automation & Control Institute, TU Wien, Austria
[3] Austrian Institute of Technology (AIT), Austria
[4] Imperial College London, UK
[5] Hanoi University of Science and Technology, Vietnam
[6] Faculty of Mathematics and Statistics, TDTU, Vietnam
[7] Department of Computer Science, University of Liverpool, UK
[*] Corresponding author minh.vu@ait.ac.at

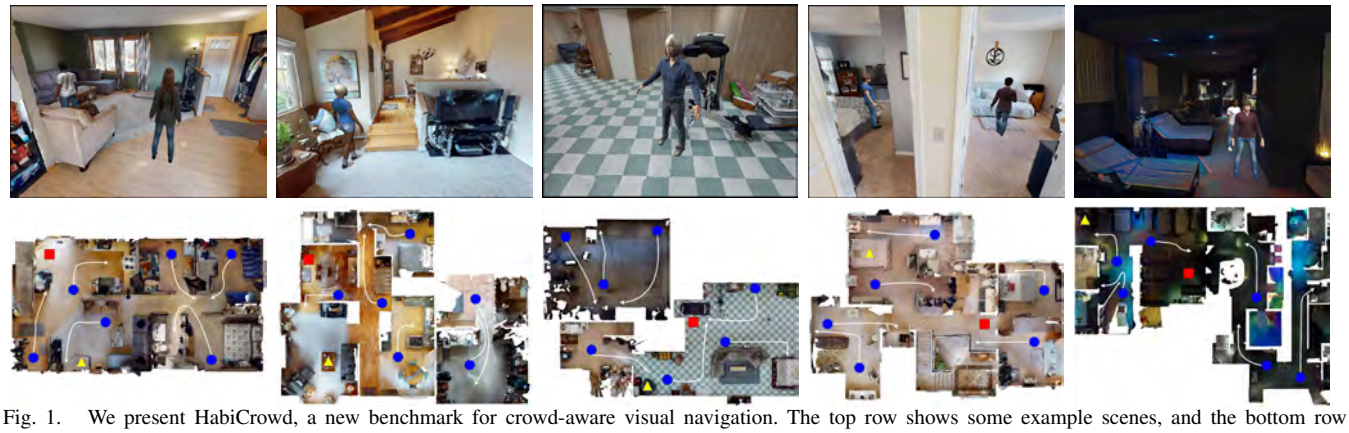

Fig. 1. We present HabiCrowd, a new benchmark for crowd-aware visual navigation. The top row shows some example scenes, and the bottom row shows the floor plan with human dynamics.

human diversity and computational efficiency.

- We comprehensively benchmark navigation tasks using HabiCrowd to showcase that our simulator offers detailed analyses, such as studies on human density and human-robot avoidance in crowd-aware environments.

II. Related Works

Visual Navigation. Traditionally, roboticists have addressed visual navigation tasks by utilizing control methods such as Model Predictive Control (MPC) [20]. Several following works have been proposed to improve control-based methods [21]. The idea of control methods has been conserved by utilizing the advantage of deep learning [20]. Nevertheless, control-based methods demand a dynamics model and are often saturated when the number of ambiguities and the prediction horizon grows [22]. Recently, achievements in reinforcement learning (RL) [23] have brought about a significant shift in visual navigation [24]. Over the years, deep RL approaches have emerged as a widely adopted solution for addressing visual navigation tasks [1], [25], and many RL-based methods have achieved state-of-the-art results in visual navigation tasks such as point-goal navigation [4], object-goal navigation [26], image-goal navigation [1].

Crowd-aware Navigation. Navigation in crowded circumstances has significantly been examined by the robotics community [12]. Earlier methods in crowd-aware navigation are primarily based on predicting the trajectories of human entities [27]. As the number of humans grows, the environment becomes exponentially denser; consequently, these trajectory-based approaches often suffer from the well-known freezing robot problem [28]. Learning-based methods have been introduced to overcome the freezing robot issue by automatically determining optimal waypoints [9]. Following this idea, RL frameworks are widely adopted to perceive human behaviors and interactions in a latent mechanism [12], [29]. However, RL methods based on 2D simulators often suppose that the dynamics of all human entities are explicit and well-determined while the real-life interactions and human dynamics are much more arbitrary and complex [29]. To address this limitation, we study crowd-aware navigation in 3D settings, wherein the agent can only perceive the environment through egocentric inputs instead of 2D floor plans with known human dynamics.

3D Simulators for Visual Navigation. There have been a large number of simulators for the tasks of visual navigation [30]. Gazebo, introduced in [31], is one of the first steps towards virtualizing the real world. Followed by Gazebo, several simulators have been established to study daily-conventional behaviors of robots, such as UnrealCV [32], AI2-THOR [33], Gibson [34], Habitat [35]. Although the mentioned simulators provide opportunities to study interactions with humans, none of them have yet included human dynamics until the establishment of iGibson-Social Navigation (iGibson-SN) [17]. More recently, Isaac Sim [18] is also established with the inclusion of virtual humans to the scenes. Nevertheless, the main drawbacks of both iGibson-SN and Isaac Sim are they have a limited number of scenes and human models. They also lack a measurement for crowdedness, which is an important factor in crowd-aware navigation problems [36]. With that inspiration, we propose a human dynamics model that is carefully designed and included in Habitat 2.0 [35] to create a standard and diverse crowd-aware visual navigation benchmark that closely reflects real-world conditions. Table I indicates the main differences between our HabiCrowd, iGibson Social Navigation (iGibson-SN), and Isaac Sim simulator. The comparisons underscore the substantial advantages of our simulator, which offers a significantly greater number of scenes and a wider range of navigation settings, environments, and human models.

TABLE I
SIMULATOR COMPARISON.

	Dynamics model	Navigation settings	Num. scenes	Environment type	Num. humans
iGibson-SN [17]	ORCA [37]	Point-goal	15	residence	3
Isaac Sim [18]	✗	Point-goal	7	office, depot traffic, hospital	7
HabiCrowd (ours)	UPL++	Point-goal Object-goal Image-goal	480	residence, gym, office, studio, club, restaurant	40

III. The HabiCrowd Simulator

In this section, we present HabiCrowd, a crowd-aware visual navigation simulator utilizing Habitat 2.0 [35] as the

foundational simulator. The motivation behind this choice comes from the diverse environment settings and photorealistic scenes offered by Habitat 2.0 [3]. By integrating human dynamics into this simulator, HabiCrowd could further bridge the gap between simulation and real-life scenarios. Furthermore, Habitat 2.0 also enables parallel computing utilization [38], which could significantly facilitate the training procedure. We specify the configurations, trajectories, and controls of human dynamics of HabiCrowd.

A. Human Dynamics Model

We design a continuous and force-based human dynamics model, which progresses concurrently with the agent's navigation, based on the universal power law (UPL) dynamics model proposed by [36]. Previously, UPL lacks the inclusion of torque components to control the facing direction of humans. To address this limitation, we further incorporate the torque dynamics model into the simulator. Consequently, our improved dynamics model is denoted as UPL++.

Assume the agent is navigating in a scene Λ with the set of obstacles W_Λ and n humans. Table II explains the notations in this section. We represent each human by $\mathbf{P}_i = \left(\mathbf{x}_i, \mathbf{v}_i, \hat{\mathbf{e}}_i, v_i, \psi_i, \psi_i^0, \omega_i, \omega_i^0, r_i, m_i\right)$ and consider their movements as particle kinematics on the 2D floor $\mathbf{F}$ of Λ. Each $\mathbf{P}_i$ will try to navigate to a destination $\mathbf{d}_i$. The model of $\mathbf{P}_i$ is depicted in Figure 2. We denote $\mathbf{f}_i$ and $\mathcal{M}_i$ as the applied force and torque to each $\mathbf{P}_i$, with the subscripts "adj," "soc," and "con" corresponding "adjective," "social," and "contact", respectively.

Force Model. We use a force model for human entities to guide them to their destinations while minimizing collisions. The force model of $\mathbf{P}_i$ projecting on $\mathbf{F}$ can be written as:

$$\mathbf{f}_i = \mathbf{f}_i^{\text{adj}} + \sum_{j=1, j\neq i}^{n} \left(\mathbf{f}_{i,j}^{\text{soc}} + \mathbf{f}_{i,j}^{\text{con}}\right) + \sum_{w\in W_\Lambda} \mathbf{f}_{i,w}^{\text{con}} + \zeta_i. \quad (1)$$

The fluctuation force $\zeta_i = \zeta_i' \mathbf{n}\left(\varphi_i\right)$ is included to break the symmetry of all humans, where $\zeta_i' \sim \mathcal{N}(0, \sigma_\zeta^2)$, $\varphi_i \sim \mathcal{U}(0, 2\pi)$, and $\mathbf{n}\left(\varphi_i\right)$ is the unit vector rotated by φ_i about the origin of $\mathbf{F}$.

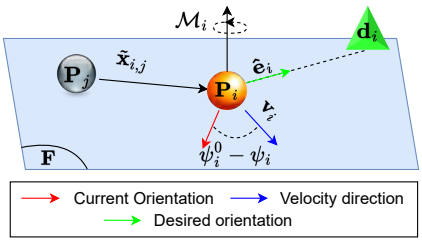

Fig. 2. **Human dynamics model.** We design a model to shape $\mathbf{P}_i$ towards $\mathbf{d}_i$ while avoiding collisions with $\mathbf{P}_j$. The human dynamics includes a force model and a torque model.

The contact force $\mathbf{f}_{i,*}^{\text{con}}$, which is a consequence of a collision between other entities like objects/humans, indicates the corresponding physical reaction. In the former notation, $*$ is a placeholder indicating either an obstacle w or another human index j. We rely on the core physic engine of Habitat 2.0, PyBullet [39] to model the contact force.

TABLE II
HUMAN DYNAMICS NOTATIONS.

Not.	Description	Not.	Description
r_i	Particle's radius	$\hat{\mathbf{e}}_i$	Desired direction
$\mathbf{x}_i$	Human coordinate	$\mathbf{v}_i$	Linear velocity
v_i	Desired velocity	m_i	Human mass
ω_i	Current angular velocity	ω_i^0	Desired angular velocity
ψ_i	Angular coordinate of the current orientation	ψ_i^0	Angular coordinate of the desired orientation

The adjust force $\mathbf{f}_i^{\text{adj}}$ shapes the agent to navigate towards its destination $\mathbf{d}_i$. Specifically, $\mathbf{f}_i^{\text{adj}} = m_i/\tau^{\text{adj}}\left(v_i\hat{\mathbf{e}}_i - \mathbf{v}_i\right)$, where τ^{adj} is the characteristic time of adjusting force.

Finally, we establish the social force model between $\mathbf{P}_i$ and $\mathbf{P}_j$ to prolong their time-to-collision based on their linearly-estimated positions. Let τ be the time-to-collision between two entities and $\tilde{\mathbf{x}}, \tilde{\mathbf{v}}$ be their relative position and velocity, respectively. Inspired by [36], we define the interaction energy between $\mathbf{P}_i$ and $\mathbf{P}_j$, as follows:

$$E(\tau) = \frac{k^{\text{soc}}}{\tau^2} \exp\left(-\frac{\tau}{\tau^{\text{soc}}}\right), \quad (2)$$

where k^{soc} is a value for adjusting social force, τ^{soc} is the interaction time horizon. The skin-to-skin distance h between two human entities after τ is given by $h(\tau) = \|\tilde{\mathbf{x}} + \tau\tilde{\mathbf{v}}\|_2 - (r_i + r_j)$.

Intuitively, the collision occurs when the skin-to-skin distance zeros; therefore, by solving the quadratic equation $h(\tau) = 0$, we obtain τ as:

$$\tau\left(\tilde{\mathbf{x}}\right) = \frac{b - \sqrt{b^2 - ac}}{a}, \quad (3)$$

where $a = \|\tilde{\mathbf{v}}\|_2^2; b = -\tilde{\mathbf{x}}^{\mathsf{T}}\tilde{\mathbf{v}}; c = \|\tilde{\mathbf{x}}\|_2^2 - (r_i + r_j)^2$. Following [36], we calculate the desired social force as $\mathbf{f}_{i,j}^{\text{soc}}(\tilde{\mathbf{x}}) = -\nabla_{\tilde{\mathbf{x}}} E(\tau)$.

Torque Model. We design a torque model to keep the orientation of $\mathbf{P}_i$ coinciding with the direction of $\mathbf{v}_i$. We denote $I_i = m_i r_i^2$ as the $\mathbf{P}_i$'s moment of inertia. Then, the torque is formulated as follows:

$$\mathcal{M}_i = \mathcal{M}_i^{\text{adj}} + \sum_{j=1, j\neq i}^{n} \mathcal{M}_{i,j}^{\text{con}} + \sum_{w\in W_\Lambda} \mathcal{M}_{i,w}^{\text{con}} + \eta_i. \quad (4)$$

The fluctuation torque $\eta_i \sim \mathcal{N}\left(0, \sigma_\eta^2\right)$ aims to break the crowd symmetry by randomly altering $\mathbf{P}_i$'s orientation.

The contact torque $\mathcal{M}_{i,*}^{\text{con}}$ is also based on the foundation simulator, which is similar to $\mathbf{f}_{i,*}^{\text{con}}$.

The adjusting torque shapes the orientation of the human entity towards its desired orientation, which is the current linear velocity $\mathbf{v}_i$ (see Figure 2). Additionally, we want to keep the angular velocity at a reasonable rate ω_i^0. As a consequence, we design the adjusting torque as follows:

$$\mathcal{M}_i^{\text{adj}} = \frac{I_i}{\tau^{\text{rot}}}\left[\left(\frac{\left(\psi_i^0 - \psi_i\right) \mod 2\pi}{\pi} - 1\right)\omega_i^0 - \omega_i\right], \quad (5)$$

where τ^{rot} is the characteristic time of adjusting torque.

Computational Complexity. It can be implied from Eqs. (1) and (4) that the computation complexity for $\mathbf{f}_i$

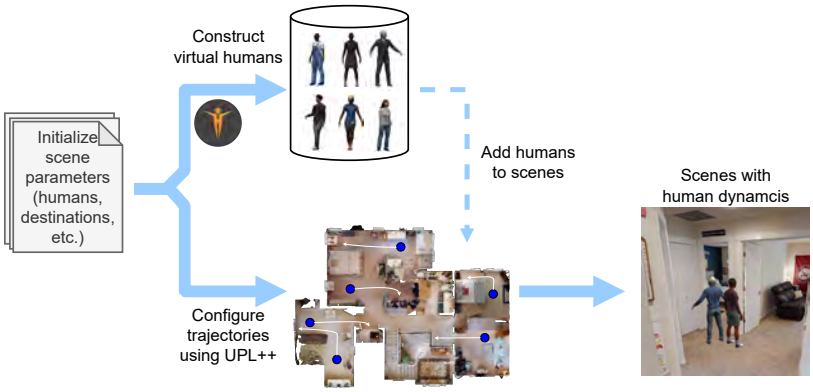

Fig. 3. **Dataset construction pipeline.**

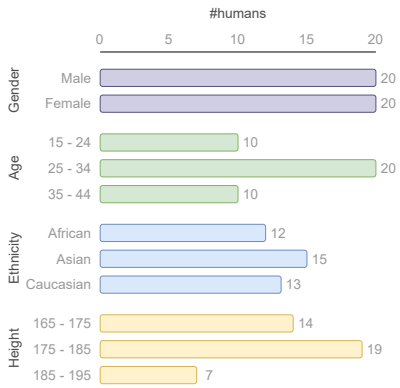

Fig. 4. **Human statistics.**

and $\mathcal{M}_i$ is both $O(n)$. This is because, apart from $\mathbf{f}_{i,j}^{\text{soc}}$ and $\mathbf{f}_{i,j}^{\text{con}}$, which require $O(n)$ computations, the remaining components only need constant computations.

We note that our UPL++ dynamics model and ORCA [37] that is used in iGibson-SN [17] share the same computational complexity of $O(n)$. However, the major difference between our UPL++ and ORCA is the approach used to determine the dynamic terms, $\mathbf{f}_i$ and $\mathcal{M}_i$. While UPL++ directly computes these terms, ORCA employs an indirect approach by solving an LP problem. Empirical results in Section IV-A confirm that the constant factor associated with UPL++ is considerably smaller than ORCA's, indicating a computational efficiency advantage in favor of UPL++.

B. Dataset Construction

Figure 3 illustrates the pipeline to build HabiCrowd scenes. We begin with initializing parameters for each virtual human $\mathbf{P}_i$ and their destination $\mathbf{d}_i$. Next, we construct virtual humans using the MakeHuman [40]. Then the trajectories of humans are configured by using the dynamics model in Section III-A and are added to the scenes of HM3D dataset [7] to create the final scenes. Note that when a human entity reaches its destination, we navigate it back to its original position and repeat this loop to maintain human dynamics during the agent's navigation.

Dataset Statistics. We utilize 480 scenes from HM3D [7] and 40 human entities. The train/val/test splits are 400/40/40. HabiCrowd's virtual humans have a gender ratio of 1/1 and ages range from 15 to 44 (Figure 4). The human density and navigation area distributions are depicted in Figure 5. Our simulator provides a broad spectrum of navigation settings including different levels of human density, spanning from 0.1 to 0.5, as well as navigation areas ranging from $10\,\text{m}^2$ to $300\,\text{m}^2$. The average human density of HabiCrowd is 0.226 humans per m^2, which is significantly greater than 0.143, the real crowd density in the real world [41]. Our simulator also has scenes with higher density (for example, 0.45) to make the problem more challenging.

IV. Experiments

A. HabiCrowd Evaluation

Performance and Memory Benchmarks. Figure 8 compares the average rendering speed (frame per second - FPS) and memory (RAM) utilization for each scene among three simulators. The process is conducted over a set of 6 scenes of each simulator on a cluster of 4 RTX 3090 GPUs. We observe a downward trend in FPS and an upward trend in memory usage for all simulators as the number of virtual humans within the scenes increases. These phenomena are primarily due to the saturation of computational complexity when more humans are added to the scenes. Nevertheless, our HabiCrowd consistently exhibits an average FPS that is nearly three times higher than iGibson-SN's and two times higher than Isaac Sim's. Furthermore, in terms of memory efficiency, both our simulator and iGibson-SN stand out by requiring less than 400 MB of RAM to render each scene. In contrast, Isaac Sim necessitates over 2560 MB of RAM to render scenes with high fidelity. To conclude, our proposed simulator achieves the most efficient computational utilization compared to other baselines.

Human Dynamics Evaluation. We leverage UMANS engine [42] to compare our dynamic model UPL++ with ORCA dynamic model in iGibson-SN on the following metrics: *i)* Mean computational time (mCT), the average time each model takes to update states for virtual humans during navigation. *ii)* Collision avoidance rate (CAR), measuring how often collisions are prevented between virtual humans. *iii)* Goal-reaching rate (GR), indicating the frequency of virtual humans reaching their destinations Table III reports the evaluation results. This table shows that both iGibson-SN and our simulator achieve collision-free navigation. Although HabiCrowd demonstrates a goal-reaching rate comparable to iGibson-SN's (with a slight margin of 0.78%), our model shows remarkable improvement in terms of computational time, which is *nearly twice faster* than iGibson-SN.

Recall from Section III-A that the computational complexity of iGibson-SN and HabiCrowd is $O(n)$. However, from Figure 7, we observe that the slope of iGibson-SN is considerably steeper than HabiCrowd's, indicating that the corresponding constant factor of HabiCrowd is significantly smaller than its counterpart. As a result, the computational efficiency advantage of HabiCrowd over iGibson-SN is preserved even when the number of humans scales up.

User Study. We provide a user study with 52 participants from diverse professions, such as designers, animators, de-

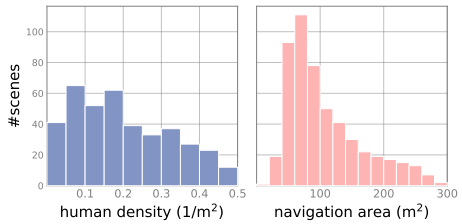

Fig. 5. **Navigating statistics.**

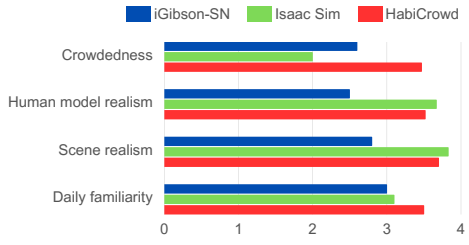

Fig. 6. **User evaluation.**

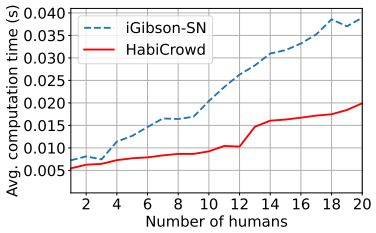

Fig. 7. **Computational time.**

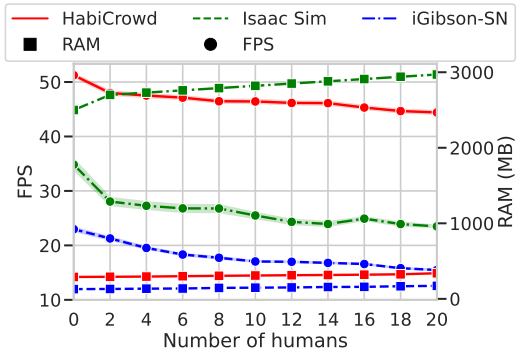

Fig. 8. **Performance and memory benchmarks.**

(a) Human pose (b) iGibson-SN (c) Isaac Sim (d) HabiCrowd

Fig. 9. **Human poses comparison.**

TABLE III

HUMAN DYNAMICS EVALUATION.

	CAR ↑	GR ↑	mCT ↓
iGibson-SN [17]	**100%**	**94.1%**	0.023
HabiCrowd (our)	**100%**	93.3%	**0.012**

TABLE IV

HUMAN DENSITY ANALYSIS.

Baseline \ Density	<0.2	0.2-0.3	>0.3
LB-WPN [43]	16.37	12.64	6.28
DOA [19]	17.78	13.25	7.36
DD-PPO [4]	15.83	11.79	7.07
DS-RNN [9]	14.95	7.51	4.24
OVRL-v2 [1]	**22.38**	**16.49**	**11.27**

velopers, and marketers. We ask them to rate HabiCrowd, Isaac Sim, and iGibson-SN on four criteria: crowdedness, human model realism, scene realism, and daily familiarity. Figure 6 shows that in all aspects, our simulator is preferred over iGibson-SN. In addition, HabiCrowd shows comparable levels of human model realism and scene realism, with a higher familiarity compared to Isaac Sim.

Human Poses Comparison. We extract human poses from all datasets by using the PCT approach [44] and provide the qualitative analysis in Figure 9. The findings reveal that our simulator showcases a greater degree of diversity in human poses when compared to the other simulators. Consequently, HabiCrowd captures a wider range of human pose variations, therefore, reflecting a more comprehensive representation of real-life human settings.

B. Crowd-aware Visual Navigation

Experimental Settings. We examine two crowd-aware visual navigation tasks upon HabiCrowd, namely, *point navigation* [4] and *object navigation* [5]. The agent is initialized at a pre-defined location in a human-crowded scene and asked to navigate to an objective. At each timestep, the agent receives RGB and Depth observations. We introduce a new objective that requires the agent to minimize the number of collisions with virtual humans.

TABLE V

VISUAL NAVIGATION RESULTS.

	Point Navigation				Object Navigation			
	CPD ↓	SPL ↑	SR ↑	DTG ↓	CPD ↓	SPL ↑	SR ↑	DTG ↓
LB-WPN [43]	0.0521	78.13	91.84	0.6048	0.0462	7.989	15.31	5.801
DOA [19]	0.0626	80.44	92.96	0.5809	0.0441	8.278	16.57	5.742
DD-PPO [4]	0.0617	79.34	92.55	0.5991	0.0415	7.424	14.93	5.956
DS-RNN [9]	0.0476	76.82	90.73	0.6674	0.0305	6.078	12.34	5.972
OVRL-v2 [1]	**0.0322**	**82.65**	**94.61**	**0.4895**	**0.0229**	**10.267**	**20.79**	**5.413**

Denote d_t as the distance from the agent to the goal at timestep t, we design the reward model as: $r_t = R^{\text{c}}\mathbb{I}_t^{\text{c}} + R^{\text{s}}\mathbb{I}_t^{\text{s}} + R^{\text{f}} \times (d_t - d_{t-1})$, where R^{c} is the collision penalty, R^{s} is the reward for completing the task, R^{f} is the shaping reward for moving closer to the objective, and $\mathbb{I}_t^{\text{c}}$, $\mathbb{I}_t^{\text{s}}$ are the indicators of collision and success of the current step, respectively. This reward model helps balance between navigating towards the objective and avoiding collisions. We train all methods using the above reward model. Followed by [5], we set $R^{\text{s}} = 2.5, R^{\text{f}} = 1.0$. The default value of R^{c} is set to -10^{-4}. For the human dynamics, we set $\tau^{\text{adj}} = 0.5, \tau^{\text{soc}} = 3.0, \tau^{\text{rot}} = 0.2$ and $k^{\text{soc}} = 1.5$ in all setups. Each method is trained in an end-to-end fashion using a total of 30M frames. We run each experiment on a computing cluster utilizing a node of four 48GB-RAM V100 GPUs.

Baselines. We compare the following baselines: *i)* DS-RNN [9]: A spatial and temporal learning-based agent. *ii)* DD-PPO [4]: A widely-used state-of-the-art optimization algorithm in E-AI tasks. *iii)* LB-WPN [43]: A method that combines a learning-based perception module and a model-based planning module for autonomous navigation among humans. *iv)* DOA [19]: The winning entry of the 2021 iGibson-SN challenge [17] that applies the dynamic obstacle augmentation method. *v)* OVRL-v2 [1]: A recent baseline utilizing the vision transformer approach (ViT).

Metrics. We employ three metrics in traditional visual navigation tasks [5], namely, *i)* Distance to goal (DTG), *ii)* Success rate (SR - in %), *iii)* Success path length [45] (SPL - in %). Additionally, we introduce *iv)* ***Collisions per distance (CPD)***, a new metric to quantify collisions occurring between the navigating agent and virtual humans. This metric is computed by dividing the number of collisions between the agent and humans by the distance traveled by the agent.

TABLE VI

IMPACT OF COLLISION PENALTY.

	Without our reward model				With our reward model											
	$R^c = 0.0$				$R^c = -10^{-4}$				$R^c = -10^{-3}$				$R^c = -10^{-2}$			
	CPD↓	SPL↑	SR↑	DTG↓	CPD↓	SPL↑	SR↑	DTG↓	CPD↓	SPL↑	SR↑	DTG↓	CPD↓	SPL↑	SR↑	DTG↓
LB-WPN [43]	0.076	5.90	10.62	6.19	0.046	7.99	15.31	5.80	0.036	4.78	8.99	6.38	0.021	3.44	6.04	6.78
DOA [19]	0.085	6.89	12.68	5.95	0.044	8.28	16.57	5.74	0.028	5.22	9.35	6.36	0.031	3.74	6.29	6.80
DD-PPO [4]	0.059	6.23	11.74	6.03	0.042	7.42	14.93	5.96	0.032	4.54	8.26	6.41	0.027	3.13	5.83	6.89
DS-RNN [9]	0.055	5.28	9.26	6.14	0.031	6.08	12.34	5.97	0.026	3.48	7.42	6.67	0.018	2.15	4.73	6.94
OVRL-v2 [1]	**0.041**	**7.57**	**14.09**	**5.86**	**0.023**	**10.27**	**20.79**	**5.41**	**0.020**	**6.15**	**11.73**	**6.27**	**0.014**	**4.22**	**7.79**	**6.68**

Point Navigation Results. We present the point navigation results in Table V. Overall, every baseline achieves decent performance, OVRL-v2 achieves the most comprehensive performance in terms of all metrics. Since HabiCrowd does not provide dynamics information regarding virtual humans to the agent due to our aim of enabling more realistic navigation settings, baselines such as DS-RNN [9], LB-WPN [43] are unable utilize human dynamics models to address the task effectively. On the other hand, OVRL-v2 leverages ViT, a robust vision backbone capable of extracting entities (such as virtual humans) from visual inputs as additional clues, thereby proficiently handling the task.

Object Navigation Results. Table V illustrates the results on the HabiCrowd dataset by all approaches. It is observable that object navigation poses a significantly greater challenge than point navigation, as evidenced by the decline in evaluation metrics for all baseline methods. The decrease in performance is aligned with the expectations outlined in [25]. The results show that OVRL-v2 has the best success rate, outperforming the second-best algorithm (DOA) by 4.22%.

How does human density affect crowd navigation? Table IV shows the success rates when we test all methods with different human density setups. OVRL-v2 stably outperforms other algorithms by 4.60%, 3.24%, and 3.91% when the human density are from < 0.2; $0.2 - 0.3$; and > 0.3, respectively. In general, we observe that the higher the human density, the more challenging our task becomes as the success rates of all methods decrease.

Does our reward model help? Table VI reports the results with and without our reward model for all baselines. Clearly, our reward model improves CPD, SPL, and SR performance metrics across all methods. We explain as follows. The introduced quantity R^c contributes to agents' regularization towards avoiding collisions with virtual humans, consequently improving performances in unseen cases.

How does collision penalty affect crowd navigation? We study the impact of R^c on our navigation problem. We observe in Table VI that starting from $R^c = -10^{-4}$ to -10^{-2}, all baselines experience a downward tendency in success rates and an upward tendency in collision avoidance. This observation is to be expected, considering that R^c serves as a regularizing factor for the collision avoidance component. Consequently, as R^c diminishes, the focus of the baselines shifts towards prioritizing collision avoidance with virtual humans rather than navigating towards the goal.

What is the drawback of current methods? A notable weakness of all current baselines is that they do not possess an *explicit* strategy to determine human dynamics from RGB-D egocentric observations. Although OVRL-v2 leverages a robust vision backbone to extract human dynamics, the employed mechanism remains *implicit* and requires refinement.

V. DISCUSSION

From the experiments, we can see that the point-goal crowd-aware navigation task is well-addressed, while the object-goal task remains a challenge. We have highlighted the critical issue in 3D crowd-aware visual navigation tasks, emphasizing the necessity of explicitly extracting human dynamics solely from RGB-D observations. This direction should be studied as making efforts towards solving crowd-aware navigation in real-life settings, where robots can only perceive human behaviors via sensor inputs without the assumption of a known human dynamics model.

Training navigation agents in simulations like Habitat shows promise, yet transferring these learned policies to real-world environments, such as on robotic platforms, continues to be a significant challenge [46]. Early-stage sim2real initiatives, including ROS interfaces, confront issues like physics discrepancies [47]. Furthermore, the gap between simulated and practical environments is widened by differences in visual appearance, absence of real-world noise, and oversimplified physical interactions, leading to poor real-world generalization of simulation-learned navigation policies [48]. In line with these efforts, our work aims to mitigate these issues by incorporating human dynamics into the Habitat simulator to create a more realistic bridge between the simulated world and the practical environment. We believe integrating human dynamics into HabiCrowd represents an encouraging step towards developing robots capable of learning in dynamic, human-aware environments that more closely replicate the natural environment encountered daily.

VI. CONCLUSION AND FUTURE WORK

We have presented HabiCrowd, a new crowd-aware simulator built upon Habitat 2.0. HabiCrowd has a robust human dynamic model and a diverse range of virtual humans compared to related simulators. The intensive experimental results indicate that our simulator utilizes computational resources more efficiently than its counterparts. We evaluate two visual navigation tasks and show that the recognition of virtual humans depicted in egocentric inputs is essential. Nevertheless, HabiCrowd still has limitations. While humans exhibit natural movements in daily lives, the constraints imposed by Habitat 2.0 [35] only allow for rigid movements. This assumption falls considerably behind the complexity of real-life practice and thus requires further enhancements.

REFERENCES

[1] K. Yadav, A. Majumdar, R. Ramrakhya, N. Yokoyama, A. Baevski, Z. Kira, O. Maksymets, and D. Batra, "Ovrl-v2: A simple state-of-art baseline for imagenav and objectnav," *arXiv preprint arXiv:2303.07798*, 2023.

[2] S. Srivastava, C. Li, M. Lingelbach, R. Martín-Martín, F. Xia, K. E. Vainio, Z. Lian, C. Gokmen, S. Buch, K. Liu, *et al.*, "Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments," in *CoRL*, 2022.

[3] K. Yadav, R. Ramrakhya, S. K. Ramakrishnan, T. Gervet, J. Turner, A. Gokaslan, N. Maestre, A. X. Chang, D. Batra, M. Savva, *et al.*, "Habitat-matterport 3d semantics dataset," in *CVPR*, 2023.

[4] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, "DD-PPO: learning near-perfect pointgoal navigators from 2.5 billion frames," in *ICLR*, 2020.

[5] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *NeurIPS*, 2020.

[6] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *CVPR*, 2017.

[7] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, *et al.*, "Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai," in *NeurIPS Datasets and Benchmarks Track*, 2021.

[8] S. Bansal, V. Tolani, S. Gupta, J. Malik, and C. Tomlin, "Combining optimal control and learning for visual navigation in novel environments," in *CoRL*, 2020.

[9] S. Liu, P. Chang, W. Liang, N. Chakraborty, and K. Driggs-Campbell, "Decentralized structural-rnn for robot crowd navigation with deep reinforcement learning," in *ICRA*, 2021.

[10] G. Monaci, M. Aractingi, and T. Silander, "Dipcan: Distilling privileged information for crowd-aware navigation," *RSS*, 2022.

[11] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of Intelligent and Robotic Systems*, 2008.

[12] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *ICRA*, 2019.

[13] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *IJRR*, 2016.

[14] D. Dugas, O. Andersson, R. Siegwart, and J. J. Chung, "Navdreams: Towards camera-only rl navigation among humans," *arXiv preprint arXiv:2203.12299*, 2022.

[15] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *CVPR*, 2016.

[16] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *IROS*, 2017.

[17] C. Li, J. Jang, F. Xia, R. Martín-Martín, C. D'Arpino, A. Toshev, A. Francis, E. Lee, and S. Savarese, "iGibson Challenge 2022," 2022. [Online]. Available: http://svl.stanford.edu/igibson/challenge.html

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

[19] N. Yokoyama and et.al., "Benchmarking augmentation methods for learning robust navigation agents: the winning entry of the 2021 igibson challenge," in *IROS*, 2022.

[20] N. Hirose, F. Xia, R. Martín-Martín, A. Sadeghian, and S. Savarese, "Deep visual mpc-policy learning for navigation," *RA-L*, 2019.

[21] Z. Li, C. Yang, C.-Y. Su, J. Deng, and W. Zhang, "Vision-based model predictive control for steering of a nonholonomic mobile robot," *IEEE Transactions on Control Systems Technology*, 2015.

[22] S. Lucia, S. Subramanian, D. Limon, and S. Engell, "Stability properties of multi-stage nonlinear model predictive control," *Systems & Control Letters*, 2020.

[23] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, 2015.

[24] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi, "Visual semantic navigation using scene priors," in *ICLR*, 2019.

[25] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but effective: Clip embeddings for embodied ai," in *CVPR*, 2022.

[26] R. Ramrakhya, D. Batra, E. Wijmans, and A. Das, "Pirlnav: Pretraining with imitation and rl finetuning for objectnav," *arXiv preprint arXiv:2301.07302*, 2023.

[27] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *IROS*, 2013.

[28] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *IROS*, 2010.

[29] P. Trautman and K. Patel, "Real time crowd navigation from first principles of probability theory," in *ICAPS*, 2020.

[30] J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, "A survey of embodied ai: From simulators to research tasks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2022.

[31] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IROS*, 2004.

[32] W. Qiu, F. Zhong, Y. Zhang, S. Qiao, Z. Xiao, T. S. Kim, and Y. Wang, "Unrealcv: Virtual worlds for computer vision," in *ACMMM*, 2017.

[33] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv preprint arXiv:1712.05474*, 2017.

[34] F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, and S. Savarese, "Gibson env: Real-world perception for embodied agents," in *CVPR*, 2018.

[35] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets, *et al.*, "Habitat 2.0: Training home assistants to rearrange their habitat," *NeurIPS*, 2021.

[36] I. Karamouzas, B. Skinner, and S. J. Guy, "Universal power law governing pedestrian interactions," *Physical Review Letters*, 2014.

[37] J. Van Den Berg and et.al., "Reciprocal n-body collision avoidance," in *ISRR*, 2011.

[38] M. Deitke, D. Batra, Y. Bisk, T. Campari, A. X. Chang, D. S. Chaplot, C. Chen, C. P. D'Arpino, K. Ehsani, A. Farhadi, *et al.*, "Retrospectives on the embodied ai workshop," *arXiv preprint arXiv:2210.06849*, 2022.

[39] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016, accessed: April 25th 2023. [Online]. Available: https://pybullet.org/wordpress/.

[40] L. Briceno and G. Paul, "Makehuman: A review of the modelling framework," in *IEA*, 2018.

[41] D. J. Deloitte, "Employment densities guide," 2010, accessed: May 29th 2023. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/378203/employ-den.pdf.

[42] W. van Toll, F. Grzeskowiak, A. L. Gandía, J. Amirian, F. Berton, J. Bruneau, B. C. Daniel, A. Jovane, and J. Pettré, "Generalized microscropic crowd simulation using costs in velocity space," in *Symposium on Interactive 3D Graphics and Games*, 2020.

[43] V. Tolani and et.al., "Visual navigation among humans with optimal control as a supervisor," *RA-L*, 2021.

[44] Z. Geng, C. Wang, Y. Wei, Z. Liu, H. Li, and H. Hu, "Human pose as compositional tokens," in *CVPR*, 2023.

[45] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.

[46] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, "Poni: Potential functions for objectgoal navigation with interaction-free learning," in *CVPR*, 2022.

[47] G. Chen, H. Yang, and I. M. Mitchell, "Ros-x-habitat: Bridging the ros ecosystem with embodied ai," in *CRV*, 2022.

[48] M. Rosano, A. Furnari, L. Gulino, and G. M. Farinella, "On embodied visual navigation in real environments through habitat," in *ICPR*, 2021.