Synthesizing Physical Character-Scene Interactions

MOHAMED HASSAN, Electronic Arts, USA
YUNRONG GUO, NVIDIA, Canada
TINGWU WANG, NVIDIA, Canada
MICHAEL BLACK, Max-Planck-Institute for Intelligent Systems, Germany
SANJA FIDLER, University of Toronto, Canada and NVIDIA, Canada
XUE BIN PENG, NVIDIA, Canada and Simon Fraser University, Canada

Fig. 1. Our framework enables physically simulated characters to perform scene interaction tasks in a natural and life-like manner. We demonstrate the effectiveness of our approach through three challenging scene interaction tasks: carrying, sitting, and lying down, which require coordination of a character's movements in relation to objects in the environment.

Movement is how people interact with and affect their environment. For realistic character animation, it is necessary to synthesize such interactions between virtual characters and their surroundings. Despite recent progress in character animation using machine learning, most systems focus on controlling an agent's movements in fairly simple and homogeneous environments, with limited interactions with other objects. Furthermore, many previous approaches that synthesize human-scene interactions require significant manual labeling of the training data. In contrast, we present a system that uses adversarial imitation learning and reinforcement learning to train physically-simulated characters that perform scene interaction tasks in a natural and life-like manner. Our method learns scene interaction behaviors from large unstructured motion datasets, without manual annotation of the motion data. These scene interactions are learned using an adversarial discriminator that evaluates the realism of a motion within the context of a scene. The key novelty involves conditioning both the discriminator and the policy networks on scene context. We demonstrate the effectiveness of our approach through three challenging scene interaction tasks: carrying, sitting, and lying down, which require coordination of a character's movements in relation to objects in the environment. Our policies learn to seamlessly transition between different behaviors like idling, walking, and sitting. By randomizing the properties of the objects and their placements during training, our method is able to generalize beyond the objects and scenarios depicted in the training dataset, producing natural character-scene interactions for a wide variety of object shapes and placements. The approach takes physics-based character motion generation a step closer to broad applicability. Please see our supplementary video for more results.

1 INTRODUCTION

Realistically animating virtual characters is a challenging and fundamental problem in computer graphics. Most prior work focuses on generating realistic human motions and often overlooks the fact that, when humans move, the movements are often driven by the need to interact with objects in a scene. When interacting with a

scene, characters need to "perceive" the objects in the environment and adapt their movements by taking into account environmental constraints and affordances. The objects in the environment can restrict movement, but also afford opportunities for interaction. Therefore characters need to adapt their movements according to object-specific functionality. Lying down on a bunk bed requires different movements than lying down on a sofa. Similarly, picking up objects of different sizes may require different strategies.

Existing techniques for synthesizing character-scene interactions tend to be limited in terms of motion quality, generalization, or scalability. Traditional motion blending and editing techniques [Gleicher 1997; Lee and Shin 1999] require significant manual effort to adapt existing motion clips to a new scene. Data-driven kinematic models [Hassan et al. 2021; Holden et al. 2017; Starke et al. 2019; Zhang et al. 2018] produce high-quality motion when applied in environments similar to those seen during training. However, when applied to new scenarios, such kinematic models struggle to generate realistic behaviors that respect scene constraints. Physics-based methods are better able to synthesize plausible motions in new scenarios by leveraging a physics simulation of a character's movements and interactions within a scene. Reinforcement learning (RL) has become one of the most commonly used paradigms for developing control policies for physically-simulated characters. However, it can be notoriously difficult to design RL objectives that lead to high-quality and natural motions [Heess et al. 2017]. Motion tracking [Peng et al. 2018] can improve motion quality by training control policies to imitate reference motion data. However, it can be difficult to apply tracking-based methods to complex scene-interaction tasks, where a character may need to compose, and transition between, a diverse set of skills in order to effectively interact with its surroundings.

Recently, Adversarial Motion Priors (AMP) [Peng et al. 2021] have been proposed as a means of imitating behaviors from large unstructured motion datasets, without requiring any annotation of the motion data or an explicit motion planner. This method leverages an adversarial discriminator to differentiate between motions in the dataset and motions generated by the policy. The policy is trained to satisfy a task reward while also trying to fool the discriminator by producing motions that resemble those shown in the dataset. Crucially, the policy need not explicitly track any particular motion clip, but is instead trained to produce motions that are within the distribution of the dataset. This allows the policy to deviate, interpolate, and transition between different behaviors as needed to adapt to new scenarios. This versatility is crucial for character-scene interaction, which requires fine-grain adjustments to a character's behaviors in order to adapt to different object configurations within a scene.

In this work, we present a framework for training physically simulated characters to perform scene interaction tasks. Our method builds on AMP and extends it to character-scene interaction tasks. Unlike the AMP discriminator, which only considers the character's motion, our discriminator jointly examines the character and the object in the scene. This allows our discriminator to evaluate the realism of the character's movements within the context of a scene (e.g., a sitting motion is realistic only when a chair is present). In addition, given a small dataset of human-object interactions, our policy discovers how to adapt these behaviors to new scenes. For example, from about five minutes of motion capture data of a human carrying a single box, we

are able to train a policy to carry hundreds of boxes with different sizes and weights. We achieve this by populating our simulated environments with a wide range of object instances and randomizing their configuration and physical properties. By interacting with these rich simulated environments, our policies learn how to realistically interact with a wide range of object instances and environment configurations. We demonstrate the effectiveness of our method with three challenging scene-interaction tasks: sit, lie down, and carry. As we show in our experiments, our policies are able to effectively perform all of these tasks and achieve superior performance compared to prior state-of-the-art kinematic and physics-based methods.

In summary, our main contributions are: (1) A framework for training physically simulated characters to perform scene interaction tasks without manual annotation. (2) We leverage a scene-conditioned discriminator that takes into account a character's movements in the context of objects in the environment. (3) We introduce a randomization approach for physical properties of objects in the scene that enables generalization beyond the objects shown in the demonstration. While our framework consists of individual components that have been introduced in prior work, the particular choice and combination of these components in the context of physics-based scene interaction tasks is novel, and we demonstrate state-of-the-art results for accomplishing these tasks with physically simulated characters.

2 RELATED WORK

Traditional animation methods generally edit, retarget, or replay motion clips from a database in order to synthesize motions for a given task. The seminal work of Gleicher [1997] adapts a reference motion to new characters with different morphologies. The method is used to adapt scene-interaction motions like box carrying and climbing a ladder. Lee and Shin [1999] introduce an interactive motion editing technique that allows motions to be adapted to new characters and new environments. Such editing and retargetting methods are limited to new scenarios that are similar to the original source motion clip. In the interest of brevity, the following discussion focuses on full body animation. However, there is a long line of related research on dexterous manipulation. See Sueda et al. [2008]; Wheatland et al. [2015]; Ye and Liu [2012]; Zhang et al. [2021] for more details.

2.1 Deep Learning Kinematic Methods

The applicability of deep neural networks (NN) to human motion synthesis has been studied extensively [Fragkiadaki et al. 2015; Habibie et al. 2017; Holden et al. 2016; Martinez et al. 2017; Taylor and Hinton 2009]. Unlike other regression tasks, classical architectures like CNNs, LSTMs and feed-forward networks perform poorly on motion synthesis. They tend to diverge or converge to a mean pose when generating long sequences. Thus, several novel architectures have been introduced in the literature to improve the motion quality. For instance, instead of directly training a single set of NN parameters, Phase-Functioned Neural Networks [Holden et al. 2017] compute the NN parameters at each frame as a function of the phase of a motion. This model can generate high-quality motions but is limited to cyclic behaviors that progress according to a well-defined phase variable. Starke et al. [2019] use a phase variable and mixture of

experts [Eigen et al. 2014; Jacobs et al. 1991] to synthesize object interaction behaviors, such as sitting and carrying. SAMP [Hassan et al. 2021] avoids the need for phase labels by training an auto-regressive cVAE [Diederik and Welling 2014; Sohn et al. 2015] using scheduled sampling [Bengio et al. 2015]. Instead of manually labelling a single phase for a motion, local motion phase variables can also be automatically computed for each body part using an evolutionary strategy [Starke et al. 2020]. Such data-driven kinematic scene-interaction methods typically require high-quality 3D human-scene data, which is scarce and difficult to record. Since these methods only learn from demonstrations, their performance degrades when applied to scenarios unlike those in the training dataset [Wang et al. 2022, 2021; Zhang et al. 2022; Zhang and Tang 2022].

2.2 Physics-Based Methods

Physics-based methods generate motions by leveraging the equations of motion of a system [Raibert and Hodgins 1991]. The physical plausibility of the generated motion is guaranteed, but the resulting behaviors may not be particularly life-like, since simulated character models provide only a coarse approximation of the biomechanical properties of their real-life counterparts. Heuristics, such as symmetry, stability, and power minimization [Raibert and Hodgins 1991; Wang et al. 2009] can be incorporated into controllers to improve the realism of simulated motions. Imitation learning is another popular approach to improve the realism of physically simulated characters. In this approach, a character learns to perform various behaviors by imitating reference motion data [Peng et al. 2018]. Motion tracking is one of the most commonly used techniques for motion imitation and is effective at reproducing a large array of challenging skills [Bergamin et al. 2019; Chentanez et al. 2018; Wang et al. 2020; Won et al. 2020]. However, it can be difficult to apply tracking-based methods to solve tasks that require composition of diverse behaviors, since the tracking-objective is typically only applied with respect to one reference motion at a time. Inspired by Generative Adversarial Imitation Learning (GAIL) [Ho and Ermon 2016], Peng et al. [2021] train a motion discriminator on large unstructured datasets and use it as a general motion prior for training control policies. This technique allows characters to imitate and compose behaviors from large datasets, without requiring any annotation of the motion clips, such as skill or phase labels. In this work, we leverage an adversarial imitation learning approach, but go beyond prior work to develop control policies for character-scene interaction tasks.

2.3 Character-Scene Interaction

Very little work has tackled the problem of synthesizing physical character-scene interactions. Early work simplifies the object manipulation problem by explicitly attaching an object to the hands of the character [Coros et al. 2010; Mordatch et al. 2012; Peng et al. 2019], thereby removing the need for the character to grasp and manipulate an object's movements via contact. Liu and Hodgins [2018] use a framework based on trajectory optimization to learn basket-ball dribbling. Chao et al. [2019] propose a hierarchical controller to synthesize sitting motions, by dividing the sitting task into sub-tasks and training separate controllers to imitate relevant reference motion clips for each sub-task. A meta controller is then trained to select
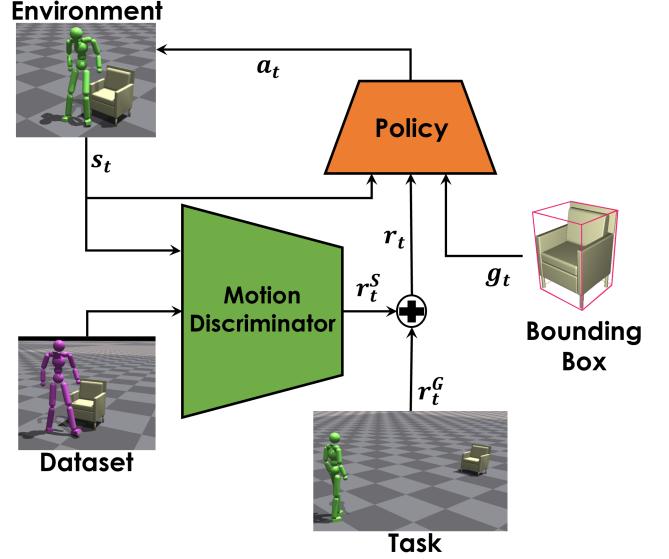

Fig. 2. Our framework has two main components: a policy and a discriminator. The discriminator differentiates between the behaviors generated by the policy and the behaviors depicted in a motion dataset. In contrast to prior work, our discriminator receives information pertaining to both the character and the environment. Specifically, the policy is trained to control the character movements to achieve a task reward r^G while producing a motion that looks like realistic human behavior within the context of a given scene.

which sub-task to execute at each time step. A similar hierarchical approach is used to train characters to play a simplified version of football [Huang et al. 2021; Liu et al. 2021]. Merel et al. [2020] train a collection of policies, each of which imitates a motion clip depicting a box-carrying or ball-catching task. The different controllers are then distilled into a single latent variable model that can then be used to construct a hierarchical controller for performing more general instances of the tasks. In contrast to the prior work, our approach is not hierarchical, generalizes to more objects and scenes, can be trained on large datasets without manual labels, and is easily applicable to multiple tasks.

3 METHOD

To train policies that enable simulated characters to interact with objects in a natural and life-like manner, we build on the Adversarial Motion Priors (AMP) framework [Peng et al. 2021]. Our approach consists of two components: a policy and a discriminator as shown in Fig. 2. The discriminator's role is to differentiate between the behaviors produced by the simulated character and the behaviors depicted in a motion dataset. The role of the policy π is to control the movements of the character in order to maximize the expected accumulative reward $J(\pi)$. The agent's reward r_t at each time step t is specified according to:

$$r_t = w^G r^G(\mathbf{s}_t, \mathbf{g}_t, \mathbf{s}_{t+1}) + w^S r^S(\mathbf{s}_t, \mathbf{s}_{t+1}). \tag{1}$$

The task reward r^G encourages the character to satisfy high-level objectives, such as sitting on a chair or moving an object to the

desired location. The style reward r^S encourages the character to imitate behaviors from a motion dataset as it performs the desired task. $\mathbf{s}_t \in \mathcal{S}$ is the state at time t. $\mathbf{a}_t \in \mathcal{A}$ are the actions sampled from the policy π at time step t. $\mathbf{g}_t \in \mathcal{G}$ denotes the task-specific goal features at time t. w^G and w^S are weights. The policy is trained to maximize the expected discount return $J(\pi)$,

$$J(\pi) = \mathbb{E}_{p(\tau|\pi)} \left[\sum_{t=0}^{T-1} \gamma^t r_t \right], \tag{2}$$

where $p(\tau|\pi)$ denotes the likelihood of a trajectory τ under the policy π. T is the time horizon, and $\gamma \in [0, 1]$ is a discount factor.

The style reward r^S is modeled using an adversarial discriminator that evaluates the similarity between the motions produced by the physically simulated character and the motions depicted in a dataset of motion clips. The discriminator is trained according to the objective proposed by Peng et al. [2021]:

$$\arg\min_D \ - \mathbb{E}_{d^{\mathcal{M}}(\mathbf{s},\mathbf{s}_{t+1})} \left[\log\left(D(\mathbf{s},\mathbf{s}_{t+1})\right) \right] \tag{3}$$

$$- \mathbb{E}_{d^{\pi}(\mathbf{s},\mathbf{s}_{t+1})} \left[\log\left(1 - D(\mathbf{s},\mathbf{s}_{t+1})\right) \right] \tag{4}$$

$$+ w^{\mathrm{gp}} \mathbb{E}_{d^{\mathcal{M}}(\mathbf{s},\mathbf{s}_{t+1})} \left[\left\| \nabla_\phi D(\phi) \big|_{\phi=(\mathbf{s},\mathbf{s}_{t+1})} \right\|^2 \right], \tag{5}$$

where $d^{\mathcal{M}}(\mathbf{s},\mathbf{s}_{t+1})$ and $d^{\pi}(\mathbf{s},\mathbf{s}_{t+1})$ represent the likelihoods of the state transition from $\mathbf{s}$ to $\mathbf{s}_{t+1}$ under the dataset distribution $\mathcal{M}$ and the policy π respectively. w^{gp} is a manually specified coefficient for a gradient penalty regularizer [Mescheder et al. 2018]. The style reward r^S for the policy is then specified according to:

$$r^S(\mathbf{s}_t, \mathbf{s}_{t+1}) = -\log(1 - D(\mathbf{s}_t, \mathbf{s}_{t+1})). \tag{6}$$

4 STATE AND ACTION REPRESENTATION

The state $\mathbf{s}$ is represented by a set of features that describes the configuration of the character's body, as well as the configuration of the objects in the scene relative to the character. These features include:

- Root height
- Root rotation
- Root linear and angular velocity
- Local joints rotations
- Local joints velocities
- Positions of four key joints: right hand, left hand, right foot, and left foot
- Object position
- Object orientation

The height and rotation of the root are recorded in the world coordinate frame while velocities of the root are recorded in the character's local coordinate frame. Rotations are presented using a 6D normal-tangent encoding [Peng et al. 2021]. The positions of four key joints, object position, and object orientation are recorded in the character's local coordinate frame. A key difference from prior work is the inclusion of object features in the state. These object features enable the discriminator to not only judge the realism of the motion but also how realistic the motion is w.r.t. to the object. Note that the object can move during the action and the agent must react appropriately. Combined, these features result in a 114D state space. The actions

a generated by the policy specify joint target rotations for PD controllers. Each target is represented as an exponential map $\mathbf{a} \in \mathbb{R}^3$ [Grassia 1998], resulting in a 28D action space.

We demonstrate the effectiveness of our framework on three challenging interactive tasks: sit, lie down, and carry. Separate policies are trained for each task. The style reward r^S is the same for all tasks. Please refer to the supplementary material for a detailed definition of the task-specific reward r^G.

5 MOTION DATASET

In order to train the character to interact with objects in a life-like manner, we train our method using a motion dataset of human-scene interactions. For the sit and lie down tasks; we use the SAMP dataset [Hassan et al. 2021], which contains 100 minutes of MoCap clips of sitting and lying down behaviors. Furthermore, the dataset also records the positions and orientations of objects in the scene, along with CAD models for seven different objects. For the carry task; we captured 15 MoCap clips of a subject carrying a single box. In each clip, the subject walks towards the box, picks it up, and carries it to a target location. The initial and target box locations are varied in each clip. In addition to full-body MoCap, the motion of the box is also tracked using optical markers.

The SAMP dataset provides examples of interactions with only seven objects, similarly our object-carry dataset only contains demonstration of carrying a single box. Nonetheless we show that our reinforcement learning framework allows the agent to generalize from these limited demonstrations to interact with a much wider array of objects in a natural manner. This is achieved by exposing the policy to new objects in the training phase. Our policy is trained using multiple environments simulated in parallel in IsaacGym [Makoviychuk et al. 2021]. We populate each environment with different object instances to encourage our policy to learn how to interact with objects exhibiting natural class variation. For the sit and lie down tasks we replace the original objects with different objects of the same class from ShapeNet [Chang et al. 2015]. The categories are: regular chairs, armchairs, tables, low stools, high stools, sofas, and beds. In total, we used $\sim$ 350 unique objects from ShapeNet [Chang et al. 2015]. To further increase the diversity of the objects, we randomly scale the objects in each training episode by a scale factor between 0.8 and 1.2. For the carry task; the size of the object is randomly scaled by a factor between 0.5 and 1.5.

6 TRAINING

At the start of each episode, the character and objects are initialized to states sampled randomly from the dataset. This leads to the character sometimes being initialized far from the target, requiring it to learn to walk towards the target and execute the desired action. At other times, it is initialized close to the completion state of the task, i.e. sitting on the object or holding a box. In contrast to always initializing the policy to a fixed starting state, this Reference State Initialization approach [Peng et al. 2018] has been shown to significantly speed up training progress and produce more realistic motions.

Since the reference motions depict only a limited set of scenarios, initialization from this alone is not sufficient to cover all possible configurations of the scene. In order to train general policies that

are able to execute the desired task from a wide range of initial configurations, we randomize the object position w.r.t. the character at the beginning of each episode. The object is placed anywhere between one and ten meters away from the character on the horizontal plane. The object orientation is sampled uniformly between $[0, 2\pi]$. The episode length is set to 10 seconds for the sit and lie down tasks, and 15 seconds for the carry task. In addition, we terminate the policy early if any joint, except the feet and hands, is within 20cm of the ground, or if the box is within 30cm of the ground.

The policy π is modeled using a neural network that takes as input the current state $\mathbf{s}_t$ and goal $\mathbf{g}_t$, then predicts the mean $\mu(\mathbf{s}_t, \mathbf{g}_t)$ of a Gaussian action distribution $\pi(\mathbf{a}_t|\mathbf{s}_t, \mathbf{g}_t) = \mathcal{N}(\mu(\mathbf{s}_t, \mathbf{g}_t), \Sigma)$. The covariance matrix Σ is manually specified and kept fixed during training. The policy, value function and the discriminator are modeled by separate fully-connected networks with the following dimensions $\{1024, 512, 28\}$, $\{1024, 512, 1\}$, $\{1024, 512, 1\}$ respectively. ReLU activations are used for all hidden units. We follow the training strategy of Peng et al. [2021] to jointly train the policy and the discriminator.

7 RESULTS

In this section, we show results of our method on different scene-interaction tasks. In Fig. 3 we show examples of our character executing sit, lie down, and carry tasks. In each task the character is initialized far from the object with a random orientation. The character first approaches the object, using locomotion skills like walking and running, and then seamlessly transitions to task-specific behavior, such as sitting, lying down, or picking up the object. The character is able to smoothly transition from idling to walking, and from walking to the various task-specific behaviors. For the carry task, note that the object is not attached to the character's hand, and is instead simulated as a rigid body and moved by forces applied by the character.

From human demonstrations of interacting with seven objects only, we teach our policy to sit and lie down on ~ 350 training objects. We demonstrate the generalization capabilities of our model by testing on objects that were not seen during training as shown in Fig. 4. Our method successfully sits and lies down on a wide range of objects and is able to adapt the character's behaviors accordingly to a given object. The character jumps to sit on a high chair, leans back on a sofa, and puts its arms on the armrests of a chair when present. We used ~ 350 training objects and tested on 21 new objects. Similarly, our policy learns to carry boxes of different sizes as shown in Fig. 5. We tested our policy on box sizes sampled uniformly between $25 \times 17.5 \times 15$cm and $75 \times 52.5 \times 45$cm. Our method generalizes beyond what is shown in the original human demonstrations. For example, the character can carry very small boxes as shown in Fig. 5, although no such objects were depicted in the human demonstration dataset. We further test our policy on different scales of the same object as shown in Fig. 6. We observe that the policy learns to adapt to the different sized objects in order to successfully sit or lie down on the support surface. More examples are available in the supplementary video.

Humans have the ability to interact with the same object in a myriad of different styles. As shown in Fig. 7, our character also demonstrates diversity in its interactions with a given object. The character exhibits different styles while sitting, including regular sitting, leaning backwards, or sitting with different arms movements.

Table 1. Success rate, average execution time, and average precision for all tasks. All metrics are averaged over 4096 trails per task.

Task	Success Rate (%)	Execution Time (Seconds)	Precision (cm)
Sit	90.4	5.0	6.7
Lie down	90.2	6.3	13.4
Carry	94.3	9.1	8.3
Carry (weights)	97.2	8.7	10.3

7.1 Evaluation

We quantitatively evaluate our method by measuring the success rate for each task. Table 1 summarizes the performance statistics on the various tasks. Success rate records the percentage of trials where the character successfully completes the task objectives. We consider sitting to be successful if the character's hip is within 20 cm of the target location. Similarly, we declare lying down to be successful if the hip and the head of the character are both within 30 cm from a target location. The carry task is successful if the box is within 20 cm of the target location. All tasks are considered unsuccessful if their success criterion is not met within 20 seconds. We evaluate the sit and lie down tasks on 16 and 5 unseen objects respectively. To increase the variability between the objects, we randomly scale the objects at each trial with a scale factor between 0.8 and 1.2. For the carry task, we randomly scale the original box shown in the human demonstration by a scale factor between 0.5 and 1.5 in each trial. The default box has a size of $50 \times 35 \times 30$ cm. The character is randomly initialized anywhere between 1 m and 10 m away from the object and with a random orientation. In addition to the success rate, we also measure the average execution time and precision for all successful trials. Execution time is the average time until the character succeeds in executing the task, according to the success definitions above. Precision is the average distance between the hip, head, box and their target locations for sit, lie down, and carry respectively. All metrics are evaluated over 4096 trials per task. Similarly, we evaluate our carry policy, which is trained to carry boxes of the same size but different weights, in Table 1 using the same metrics. Please refer to the supplementary material for more details. Despite the diversity of test objects and configurations, our policies succeed in executing all task with a higher than 90% success rate.

Moreover, the character is able to generalize beyond the limited reference clips and succeeds in executing the tasks from initial configurations not shown in the reference motion as can be seen in Fig. 8. In the reference clips, the character starts up to three meters away from the object, nonetheless the character learns to execute the tasks even when initialized up to ten meters away from the object. This is partly due to the scene randomization approach used during training as described in Sec. 6.

Next we study the robustness of our policy to external perturbations. We pelt the character with 20 projectiles of weight 1.2 kg at random time steps of the trial. We found that our policy is very robust to these perturbations, and is able to recover and resume the task upon being hit by a projectile. Examples of these recovery behaviors are shown in the supplementary video. We also randomly move the object during the execution of a task (e.g. move the chair away

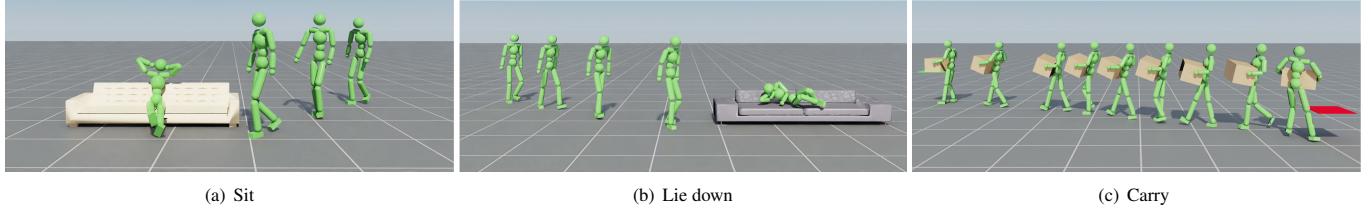(a) Sit · (b) Lie down · (c) Carry

Fig. 3. Our method successfully executes three challenging scene-interaction tasks in a life-like manner.
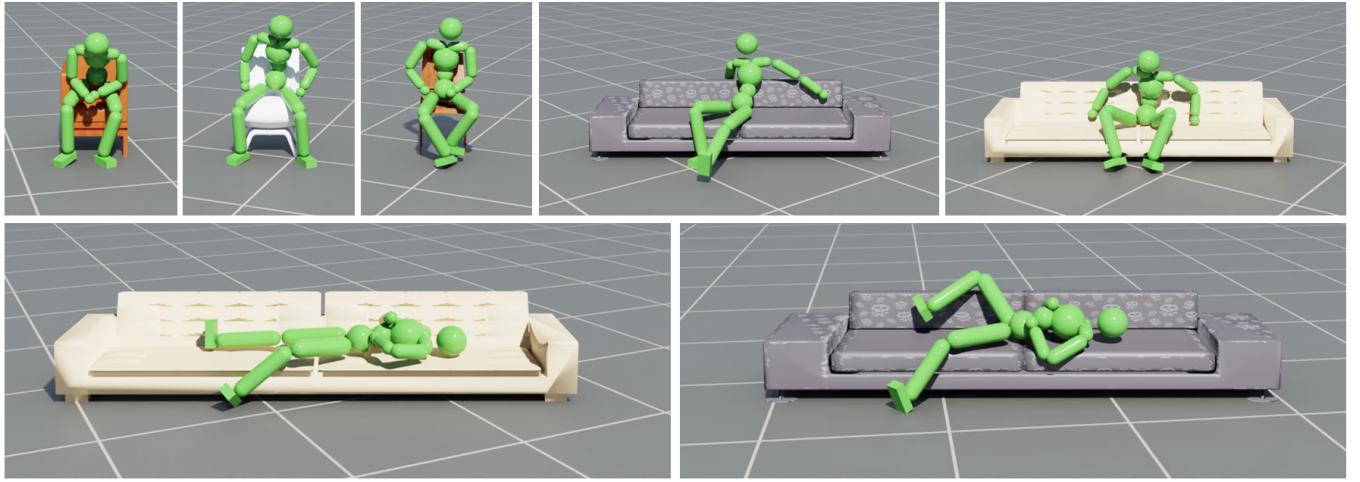

Fig. 4. Our method successfully sits and lies down on a wide range of objects and is able to adapt the character's behaviors to new objects.
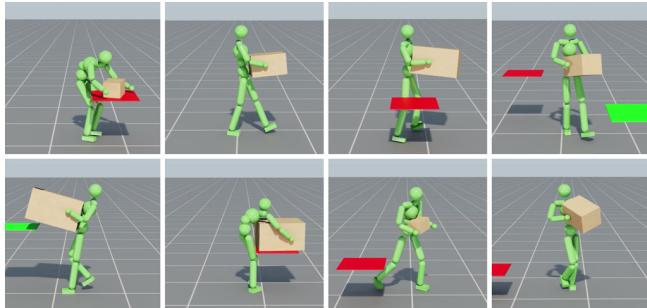

Fig. 5. From a human demonstration of carrying a single box, our method generalizes to carrying boxes of different sizes.

Fig. 6. Our policy is able to adapt to different sized objects.

Fig. 7. Different styles of sitting on the same object.

Table 2. Success rate under physical perturbations.

Task	Success Rate (%)
Sit	87.5
Lie down	82.0
Carry	89.4

the robustness of the policy to such sudden changes to the environment. Our policies maintain a high success rate under these physical perturbations for all three tasks, as reported in Table 2 .

7.2 Comparisons

There have only been a few previous attempts in the area of synthesizing character-scene interactions. We compare our physics-based model to NSM [Starke et al. 2019] and SAMP [Hassan et al. 2021], which are both kinematic models. We also compare to Chao et al. [2019], which is a hierarchical-based physical approach. All three methods are trained on the sitting task. Kinematic models (NSM

as the character is about to sit). The supplementary video shows
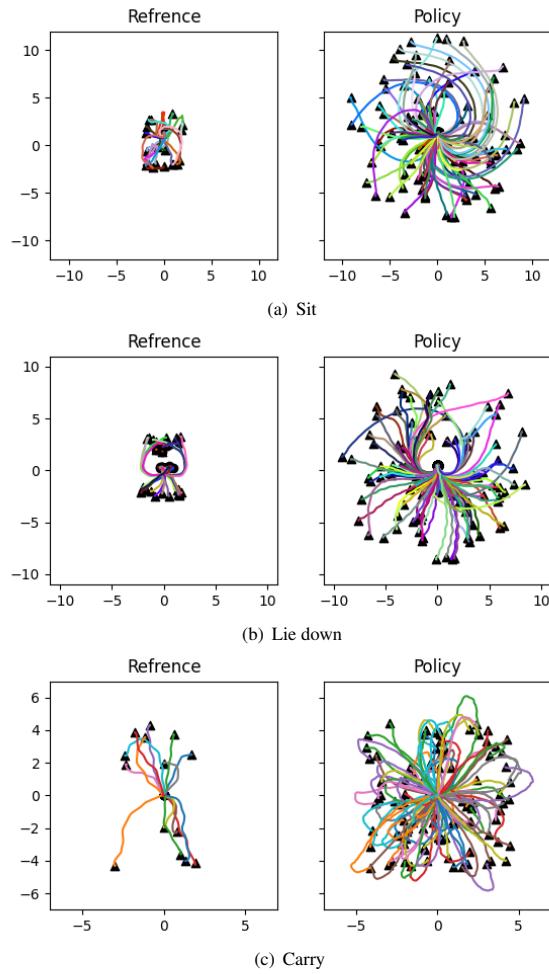
(a) Sit

(b) Lie down

(c) Carry

Fig. 8. Reference motion trajectories and the trajectories generated by our policies when initialized randomly. Triangles indicate starting positions and the target position is indicated with a circle. From limited reference clips covering limited configurations, our policy learns to successfully execute the actions in a wide range of configurations.

Table 3. Performance comparision to NSM [Starke et al. 2019], SAMP [Hassan et al. 2021], Chao et al. [2019]

Metric	Sit				Lie down	
	NSM	SAMP	Chao et al.	Ours	SAMP	Ours
Success Rate(%)	75.0	75.0	17	**93.7**	50	**80**
Execution Time(seconds)	7.5	7.2	-	**3.7**	9.5	**6.9**
Precision (meters)	0.19	**0.06**	-	0.09	**0.05**	0.3

character to autonomously walk towards a box, lift the box, and *carry* it to the destination. We use the pre-trained open-source models of NSM [Starke et al. 2019], and SAMP [Hassan et al. 2021], and evaluate them on the same test objects as our method. Note that our method and SAMP are trained on the same dataset. Retraining NSM is infeasible due to the missing phase labels. For Chao et al. [2019], we report the numbers provided in the paper. Table 3 shows that our method significantly outperforms these prior systems on the sit and lie down tasks.

8 DISCUSSION

Throughout our experiments, we train a separate policy for each task. Multi-task RL remains a difficult and open problem [Ruder 2017] and should be investigated in future work. Unlike previous attempts to synthesize carry motions [Coros et al. 2010; Mordatch et al. 2012; Peng et al. 2019], our box is not welded to the character's hand. The box is simulated as a rigid object and is moved by forces applied by the character. In a few cases, the character approaches the object but fails to complete the task successfully within the duration of an episode. For example, the character might stand next to the object until the end of the episode. In other cases, the character might not reach the target object in time because it follows a suboptimal path; some examples are shown in Fig. 8. We focus on environments of one objects only. Nonetheless, our state representation could be augmented to contain other objects. In addition, it would be exciting to explore adding virtual eyes to our character. This would allow for interaction with more complex scenes. We show quantitatively and qualitatively that our randomization approach enables the character to interact with a wide range of test objects. These objects are not used during training and are randomly selected from ShapeNet. We also show that our method can adapt to different object sizes (Fig. 5 and Fig. 6) and weights (Table. 1). Nonetheless, if the test size or weight is far from the training distribution, we expect the success rate to drop. We focus on generalization to different objects, future work should explore generalization to different skills, such as jumping.

9 CONCLUSION

We presented a method that realistically synthesizes physical and realistic character-scene interaction. We introduced a scene-conditioned policy and discriminator that take into account a character's movements in the context of objects in the environment. We applied our method to three challenging scene interaction tasks: sit, lie down, and carry. Our method learns when and where to transition from one behavior to another to execute the desired task. We introduced an efficient randomization approach for the training objects, their placements, sizes, and physical properties. This randomization approach allows our policies to generalize to a wide range of objects

and SAMP) tend to produce non-physical behaviors, such as footskating/floating and object penetrations. Some examples are shown in the supplementary video. Since, kinematic models learn from human demonstration only, without interaction with the environment, these models can struggle to generalize to new scenarios. The work of Chao et al. [2019] synthesizes motions using a physics simulation, however it often fails to sit on the target object. Most of time the character falls when approaching the object.

A quantitative comparison to previous methods is available in Table 3. A trial is considered successful, only if character does not penetrate the object while approaching it. None of the baselines are capable of consistently completing the full carry task. NSM [Starke et al. 2019] trains a character to walk towards a box and lift it up. However, the character needs to be manually controlled to carry the box to a destination. Our policy, on the other hand, enables the

and scenarios not shown in the human demonstration. We showed that our policies are robust to different physical perturbations and sudden changes in the environment. We qualitatively and quantitatively showed that our method significantly outperforms previous systems. We hope our system provides a step towards creating more capable physically simulated characters that can interact with complex environments in a more intelligent and life-like manner.

Disclosure: The work was done while Mohamed Hassan was an intern at Nvidia. MJB has received research funds from Adobe, Intel, Nvidia, Facebook, and Amazon. While MJB is a part-time employee of Amazon, his research was performed solely at, and funded solely by, Max Planck. MJB has financial interests in Amazon, Datagen Technologies, and Meshcapade GmbH.

REFERENCES

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1* (Montreal, Canada) (*NIPS'15*). MIT Press, Cambridge, MA, USA, 1171–1179.

Kevin Bergamin, Simon Clavet, Daniel Holden, and James Richard Forbes. 2019. DReCon: Data-Driven Responsive Control of Physics-Based Characters. *ACM Trans. Graph.* 38, 6, Article 206 (Nov. 2019), 11 pages. https://doi.org/10.1145/3355089.3356536

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. 2015. *ShapeNet: An Information-Rich 3D Model Repository*. Technical Report arXiv:1512.03012 [cs.GR]. Stanford University — Princeton University — Toyota Technological Institute at Chicago.

Yu-Wei Chao, Jimei Yang, Weifeng Chen, and Jia Deng. 2019. Learning to sit: Synthesizing human-chair interactions via hierarchical control. *Proceedings of the AAAI Conference on Artificial Intelligence* (2019).

Nuttapong Chentanez, Matthias Müller, Miles Macklin, Viktor Makoviychuk, and Stefan Jeschke. 2018. Physics-Based Motion Capture Imitation with Deep Reinforcement Learning. In *Proceedings of the 11th Annual International Conference on Motion, Interaction, and Games* (Limassol, Cyprus) (*MIG '18*). Association for Computing Machinery, New York, NY, USA, Article 1, 10 pages. https://doi.org/10.1145/3274247.3274506

Stelian Coros, Philippe Beaudoin, and Michiel van de Panne. 2010. Generalized Biped Walking Control. *ACM Trans. Graph.* 29, 4, Article 130 (jul 2010), 9 pages. https://doi.org/10.1145/1778765.1781156

P Kingma Diederik and Max Welling. 2014. Auto-encoding variational bayes. In *International Conference on Learning Representations ICLR*.

David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. 2014. Learning factored representations in a deep mixture of experts. In *International Conference on Learning Representations ICLR*.

Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. 2015. Recurrent Network Models for Human Dynamics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) (ICCV '15*). IEEE Computer Society, USA, 4346–4354.

Michael Gleicher. 1997. Motion Editing with Spacetime Constraints. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics* (Providence, Rhode Island, USA) (*I3D '97*). Association for Computing Machinery, New York, NY, USA, 139–ff. https://doi.org/10.1145/253284.253321

F. Sebastin Grassia. 1998. Practical Parameterization of Rotations Using the Exponential Map. *J. Graph. Tools* 3, 3 (March 1998), 29–48. https://doi.org/10.1080/10867651.1998.10487493

I. Habibie, Daniel Holden, Jonathan Schwarz, Joe Yearsley, and T. Komura. 2017. A Recurrent Variational Autoencoder for Human Motion Synthesis. In *BMVC*.

Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael Black. 2021. Stochastic Scene-Aware Motion Prediction. In *Proceedings of the International Conference on Computer Vision 2021*.

Nicolas Heess, Dhruva TB, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, SM Eslami, et al. 2017. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286* (2017).

Jonathan Ho and Stefano Ermon. 2016. Generative Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Eds.), Vol. 29. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2016/file/cc7e2b878868cbae992d1fb743995d8f-Paper.pdf

Daniel Holden, Taku Komura, and Jun Saito. 2017. Phase-Functioned Neural Networks for Character Control. *ACM Trans. Graph.* 36, 4, Article 42 (July 2017), 13 pages. https://doi.org/10.1145/3072959.3073663

Daniel Holden, Jun Saito, and Taku Komura. 2016. A deep learning framework for character motion synthesis and editing. *ACM Trans. Graph.* 35, 4 (2016), 1–11.

Shiyu Huang, Wenze Chen, Longfei Zhang, Ziyang Li, Fengming Zhu, Deheng Ye, Ting Chen, and Jun Zhu. 2021. TiKick: Towards Playing Multi-agent Football Full Games from Single-agent Demonstrations. *arXiv preprint arXiv:2110.04507* (2021).

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. 1991. Adaptive Mixtures of Local Experts. *Neural Computation* 3, 1 (1991), 79–87. https://doi.org/10.1162/neco.1991.3.1.79

Jehee Lee and Sung Yong Shin. 1999. A Hierarchical Approach to Interactive Motion Editing for Human-like Figures. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99*). ACM Press/Addison-Wesley Publishing Co., USA, 39–48. https://doi.org/10.1145/311535.311539

Libin Liu and Jessica Hodgins. 2018. Learning Basketball Dribbling Skills Using Trajectory Optimization and Deep Reinforcement Learning. *ACM Trans. Graph.* 37, 4, Article 142 (jul 2018), 14 pages. https://doi.org/10.1145/3197517.3201315

Siqi Liu, Guy Lever, Zhe Wang, Josh Merel, SM Eslami, Daniel Hennes, Wojciech M Czarnecki, Yuval Tassa, Shayegan Omidshafiei, Abbas Abdolmaleki, et al. 2021. From motor control to team play in simulated humanoid football. *arXiv preprint arXiv:2105.12196* (2021).

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. 2021. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470* (2021).

Julieta Martinez, Michael J Black, and Javier Romero. 2017. On human motion prediction using recurrent neural networks. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2891–2900.

Josh Merel, Saran Tunyasuvunakool, Arun Ahuja, Yuval Tassa, Leonard Hasenclever, Vu Pham, Tom Erez, Greg Wayne, and Nicolas Heess. 2020. Catch & Carry: Reusable Neural Controllers for Vision-Guided Whole-Body Tasks. *ACM Trans. Graph.* 39, 4, Article 39 (July 2020), 14 pages. https://doi.org/10.1145/3386569.3392474

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. 2018. Which Training Methods for GANs do actually Converge?. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80*), Jennifer Dy and Andreas Krause (Eds.). PMLR, Stockholmsmässan, Stockholm Sweden, 3481–3490. http://proceedings.mlr.press/v80/mescheder18a.html

Igor Mordatch, Emanuel Todorov, and Zoran Popović. 2012. Discovery of Complex Behaviors through Contact-Invariant Optimization. *ACM Trans. Graph.* 31, 4, Article 43 (jul 2012), 8 pages. https://doi.org/10.1145/2185520.2185539

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018. DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills. *ACM Trans. Graph.* 37, 4, Article 143 (jul 2018), 14 pages. https://doi.org/10.1145/3197517.3201311

Xue Bin Peng, Michael Chang, Grace Zhang, Pieter Abbeel, and Sergey Levine. 2019. *MCP: Learning Composable Hierarchical Control with Multiplicative Compositional Policies*. Curran Associates Inc., Red Hook, NY, USA.

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. 2021. AMP: Adversarial Motion Priors for Stylized Physics-Based Character Control. *ACM Trans. Graph.* 40, 4, Article 144 (jul 2021), 20 pages. https://doi.org/10.1145/3450626.3459670

Marc H. Raibert and Jessica K. Hodgins. 1991. Animation of Dynamic Legged Locomotion. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '91*). Association for Computing Machinery, New York, NY, USA, 349–358. https://doi.org/10.1145/122718.122755

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 3483–3491. http://papers.nips.cc/paper/5775-learning-structured-output-representation-using-deep-conditional-generative-models.pdf

Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. 2019. Neural State Machine for Character-Scene Interactions. *ACM Trans. Graph.* 38, 6, Article 209 (Nov. 2019), 14 pages. https://doi.org/10.1145/3355089.3356505

Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. 2020. Local Motion Phases for Learning Multi-Contact Character Movements. *ACM Trans. Graph.* 39, 4, Article 54 (July 2020), 14 pages. https://doi.org/10.1145/3386569.3392450

Shinjiro Sueda, Andrew Kaufman, and Dinesh K. Pai. 2008. Musculotendon Simulation for Hand Animation. In *ACM SIGGRAPH 2008 Papers* (Los Angeles, California) (*SIGGRAPH '08*). Association for Computing Machinery, New York, NY, USA, Article 83, 8 pages. https://doi.org/10.1145/1399504.1360682

Graham W. Taylor and Geoffrey E. Hinton. 2009. Factored Conditional Restricted Boltzmann Machines for Modeling Motion Style. In *Proceedings of the 26th Annual International Conference on Machine Learning* (Montreal, Quebec, Canada) (*ICML '09*). Association for Computing Machinery, New York, NY, USA, 1025–1032. https://doi.org/10.1145/1553374.1553505

Jingbo Wang, Yu Rong, Jingyuan Liu, Sijie Yan, Dahua Lin, and Bo Dai. 2022. Towards Diverse and Natural Scene-aware 3D Human Motion Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20460–20469.

Jiashun Wang, Huazhe Xu, Jingwei Xu, Sifei Liu, and Xiaolong Wang. 2021. Synthesizing Long-Term 3D Human Motion and Interaction in 3D Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9401–9411.

Jack M. Wang, David J. Fleet, and Aaron Hertzmann. 2009. Optimizing Walking Controllers. *ACM Trans. Graph.* 28, 5 (dec 2009), 1–8. https://doi.org/10.1145/1618452.1618514

Tingwu Wang, Yunrong Guo, Maria Shugrina, and Sanja Fidler. 2020. UniCon: Universal Neural Controller For Physics-based Character Motion. arXiv:2011.15119 [cs.GR]

Nkenge Wheatland, Yingying Wang, Huaguang Song, Michael Neff, Victor Zordan, and Sophie Jörg. 2015. State of the art in hand and finger modeling and animation. In *Computer Graphics Forum*, Vol. 34. Wiley Online Library, 735–760.

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. 2020. A Scalable Approach to Control Diverse Behaviors for Physically Simulated Characters. *ACM Trans. Graph.* 39, 4, Article 33 (jul 2020), 12 pages. https://doi.org/10.1145/3386569.3392381

Yuting Ye and C. Karen Liu. 2012. Synthesis of Detailed Hand Manipulations Using Contact Sampling. *ACM Trans. Graph.* 31, 4, Article 41 (jul 2012), 10 pages. https://doi.org/10.1145/2185520.2185537

He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. 2018. Mode-Adaptive Neural Networks for Quadruped Motion Control. *ACM Trans. Graph.* 37, 4, Article 145 (July 2018), 11 pages. https://doi.org/10.1145/3197517.3201366

He Zhang, Yuting Ye, Takaaki Shiratori, and Taku Komura. 2021. ManipNet: Neural Manipulation Synthesis with a Hand-Object Spatial Representation. *ACM Trans. Graph.* 40, 4, Article 121 (jul 2021), 14 pages. https://doi.org/10.1145/3450626.3459830

Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Vladimir Guzov, and Gerard Pons-Moll. 2022. COUCH: Towards Controllable Human-Chair Interactions. In *Computer Vision – ECCV 2022*, Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Springer Nature Switzerland, Cham, 518–535.

Yan Zhang and Siyu Tang. 2022. The Wanderings of Odysseus in 3D Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20481–20491.

Appendices

A TASKS

Our aim is to train simulated character to solve character-scene interaction tasks. To demonstrate the effectiveness of our method; we choose three challenging interactive tasks: sit, lie down, and carry. The style reward r^S is the same for all tasks as defined. The task reward r^G is task-specific as detailed in the following subsections.

A.1 Sit

The objective of this task is for the character to move to a target object and to sit on it. The object is initialized at a random orientation anywhere between one and ten meters away from the character.

The goal $\mathbf{g}_t \in \mathbb{R}^3$ is the object bounding box. The task reward is defined as :

$$r_t^G = \begin{cases} 0.7\, r_t^{\text{near}} + 0.3\, r_t^{\text{far}}, & ||\mathbf{x}^* - \mathbf{x}_t^{\text{root}}|| > 0.5m \\ 0.7\, r_t^{\text{near}} + 0.3, & \text{otherwise} \end{cases} \quad (7)$$

where $\mathbf{x}^{\text{root}}$ is the position of the character's root, $\mathbf{x}^*$ is the object position, r^{far} encourages the character to walk towards the object, while r^{near} encourages the character to sit on the object once it is close by. r^{far} is specified according to:

$$r_t^{\text{far}} = 0.5 \exp\left(-0.5||\mathbf{x}^* - \mathbf{x}_t^{\text{root}}||^2\right)$$
$$+ 0.5 \exp\left(-2.0|| v^* - \mathbf{d}_t^* \cdot \dot{\mathbf{x}}_t^{\text{root}}||^2\right) \quad (8)$$

where $\dot{\mathbf{x}}_t^{\text{root}}$ is the linear velocity of the character's root, $\mathbf{d}^*$ is a horizontal unit vector pointing from the root $\mathbf{x}_t^{\text{root}}$ to the object's location $\mathbf{x}^*$, and $v^* = 1.5m/s$ is the target speed at which the character should walk. Once the character is close to the object, r^{near} is used to encourage the character to sit on the object:

$$r_t^{\text{near}} = \exp\left(-10.0||\mathbf{x}^{\text{root}*} - \mathbf{x}_t^{\text{root}}||^2\right), \quad (9)$$

with $\mathbf{x}^{\text{root}*}$ denoting the target sitting position on the object where the character's hip should be placed.

A.2 Lie down

The objective of the lie down task is for the character to walk towards an object and then lie down on it. The goal $\mathbf{g}_t$ and the task reward r_t^G are the same as for the sitting task (see Eq. 7). r^{far} is defined as in Eq. 8, and $r_t^{\text{near}} =$

$$\exp\left(-10.0||\mathbf{x}^{\text{root}*} - \mathbf{x}_t^{\text{root}}||^2 - 10.0||h^{\text{head}*} - h_t^{\text{head}}||^2\right) \quad (10)$$

where h^{head} is the height of the character's head, and $\mathbf{x}^{\text{head}*}$ is the target head height.

A.3 Carry

The objective of the carry task is for the character to pick up a box and carry it to a destination. The goal is specified according to:

$$\mathbf{g}_t = (\tilde{\mathbf{x}}_t', b_h, b_w, b_d), \quad (11)$$

where $\tilde{\mathbf{x}}_t'$ is the target position on which the box should be placed; $\tilde{\mathbf{x}}_t'$ is represented in the character's local coordinate frame. b_h, b_w, b_d

are the box height, width, and depth respectively. The task reward is specified according to:

$$r_t^G = r_t^{\text{walk}} + r_t^{\text{carry}}, \quad (12)$$

where r^{walk} encourages the character to walk towards the box and stay close to it. More specifically, it encourages the character to move its root $\mathbf{x}_t^{\text{root}}$ towards the position of the box $\mathbf{x}_t^*$ at a target speed $\mathbf{d}^*$:

$$r_t^{\text{walk}} = \begin{cases} 0.1 \exp\left(-0.5||\mathbf{x}_t^* - \mathbf{x}_t^{\text{root}}||^2\right) + \\ 0.1 \exp\left(-2.0|| v^* - \mathbf{d}_t^* \cdot \dot{\mathbf{x}}_t^{\text{root}}||^2\right), & ||\mathbf{x}_t^* - \mathbf{x}_t^{\text{root}}|| > 0.5m \\ 0.2, & \text{otherwise} \end{cases}.$$

r^{carry} encourages the character to carry the box to a target position $\mathbf{x}_t'$:

$$r_t^{\text{carry}} = \begin{cases} r_t^{\text{carry-far}} + r_t^{\text{carry-near}}, & ||\mathbf{x}_t' - \mathbf{x}_t^*|| > 0.5m \\ 0.2 + r_t^{\text{carry-near}}, & \text{otherwise} \end{cases}. \quad (13)$$

$r_t^{\text{carry-far}}$ is defined as:

$$r_t^{\text{carry-far}} = 0.2 \exp\left(-0.5||\mathbf{x}_t' - \mathbf{x}_t^*||^2\right)$$
$$+ 0.2 \exp\left(-2.0|| v' - \mathbf{d}_t' \cdot \dot{\mathbf{x}}_t^{\text{box}}||^2\right)$$
$$+ 0.1 \exp\left(-10.0||h_t^{\text{hand}} - h_t^{\text{box}}||^2\right). \quad (14)$$

Where $\mathbf{d}_t'$ is a horizontal unit vector pointing from the box location to the target location, $\dot{\mathbf{x}}_t^{\text{box}}$ is the velocity of the box, $v' = 1.5m/s$ is the target speed. h_t^{hand} and h_t^{box} are the height of the character's hand and box height respectively. Once the box is close to the target, $r_t^{\text{carry-near}}$ encourages the character to place the box precisely on the target platform,

$$r_t^{\text{carry-near}} = 0.2 \exp\left(-10.0||\mathbf{x}_t' - \mathbf{x}_t^*||^2\right). \quad (15)$$

B RESULTS

We show how our policy deals with objects of different physical properties. We train a policy to carry boxes of the same size but different weights. The weights are sampled uniformly between 5 kg and 26 kg. For this experiment, we augment the goal $\mathbf{g}_t$ with the box density. Some examples are shown in Fig. S.1 where heavier boxes are indicated with darker colors. The character discovers how to deal with the different weights from a human demonstration of a single box.

Note that our reference data only contains demonstrations of carrying a single box in a particular style. Thus, it is expected that the character will stay close to this carrying style, even for boxes of different weights. The purpose of this experiment is to show carrying objects with weights that are different from the reference data. To learn different carrying styles, we would need such styles to be part of our dataset.

We illustrate the plausibility of the full motion trajectories generated by our policy in Fig. S.2. We initialize our policy with the first frames of the reference motion clips. We then plot the full trajectories followed by our policy alongside the reference trajectories from the reference motion clips. For the sit and lie down tasks, we plot the
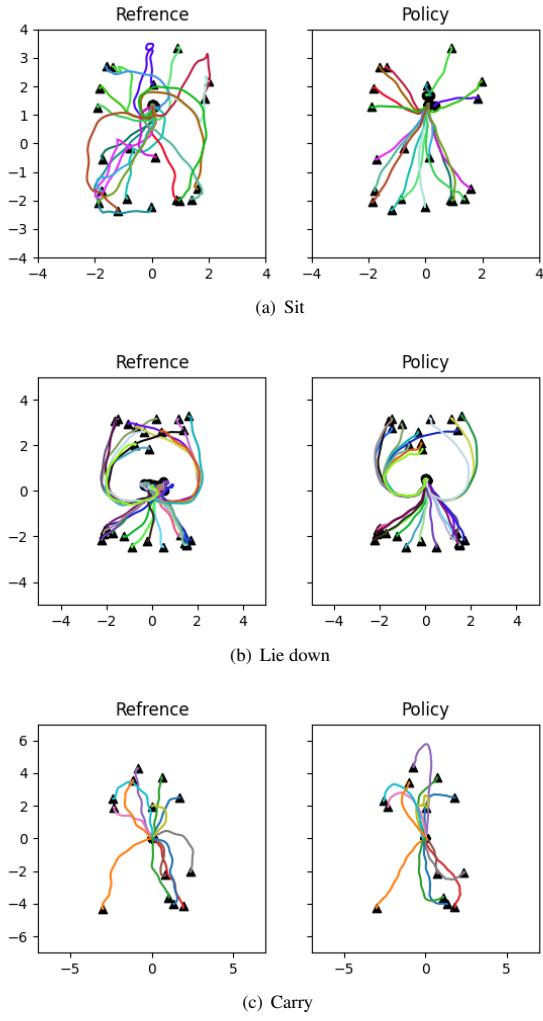
(a) Sit

(b) Lie down

(c) Carry

Fig. S.2. Reference motion trajectories and the trajectories generated by our policies when initialized with the first frame of the reference motion. Triangles indicate starting positions and the target position is indicated with a circle. Although the reference clips do not always follow the shortest trajectory to the object, our policy often does.
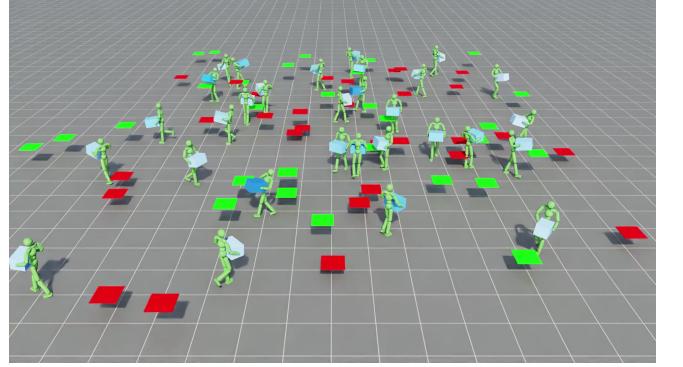

Fig. S.1. Carrying boxes of different weights. Darker colors indicate heavier weights.

Table S.1. Bounding box ablation. Success rate, average execution time, and average precision for all tasks.

Task	Success Rate (%)	Execution Time (Seconds)	Precision (cm)
Sit	88.6	5.3	6.6
Lie down	81.9	6.2	14.7
Carry	0.0	-	-

character trajectory. For the carry task, we plot the box trajectory. Although the reference clips do not always follow the shortest trajectory to the object, our policy often does as can be seen in Fig. 2(a).

Throughout our experiments, we include the bounding box of the object in our goal g_t as explained in Sec. A. To evaluate the importance of the bounding box, we retrain our policies without this information and evaluate the policies in Table S.1. We observe that the bounding box is vital especially for dynamic tasks like carry. Without this information the character fails to pick up the box from the platform. In general, excluding the bounding box information decrease the success rate for all three tasks.

C COMPARISONS

NSM [Starke et al. 2019] and SAMP [Hassan et al. 2021] have two modes of operation. Manual mode, where a user controls the character, and a goal-driven mode. In the goal-driven mode, the user only specifies the desired action(e.g. sit) and the target object. We are only comparing against the goal-driven mode of NSM and SAMP. Our method, NSM, and SAMP all receive the same input from the user which is the desired action and the target object. The full motion is generated by the algorithms and there is no manual nor random control.