Generating Continual Human Motion in Diverse 3D Scenes

Aymen Mir[1, 2] Xavier Puig[3] Angjoo Kanazawa[4] Gerard Pons-Moll[1, 2]

[1] Tübingen AI Center, University of Tübingen, Germany
[2]Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
[3]FAIR at Meta
[4]University of California, Berkeley

{aymen.mir,gerard.pons-moll}@uni-tuebingen.de, xavierpuig@meta.com, kanazawa@eecs.berkeley.edu

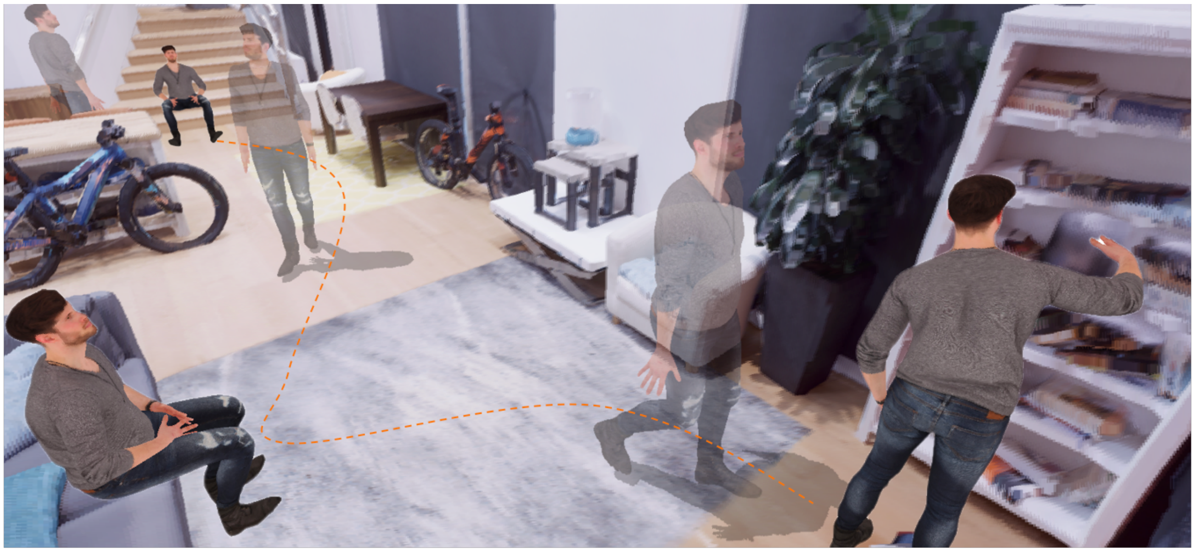

Figure 1. Our method synthesizes diverse animator guided human motion such as sitting and grabbing in diverse 3D scenes. We urge readers to watch the supplementary video for more results.

Abstract

We introduce a method to synthesize animator guided human motion across 3D scenes. Given a set of sparse (3 or 4) joint locations (such as the location of a person's hand and two feet) and a seed motion sequence in a 3D scene, our method generates a plausible motion sequence starting from the seed motion while satisfying the constraints imposed by the provided keypoints. We decompose the continual motion synthesis problem into walking along paths and transitioning in and out of the actions specified by the keypoints, which enables long generation of motions that satisfy scene constraints without explicitly incorporating scene information. Our method is trained only using scene agnostic mocap data. As a result, our approach is deployable across 3D scenes with various geometries. For achieving plausible continual motion synthesis without drift, our key contribution is to iteratively generate motion in a goal-centric canonical coordinate frame where the next immediate target is situated at the origin. Our model can generate long sequences of diverse actions such as grabbing, sitting and leaning chained together in arbitrary order, demonstrated on scenes of varying geometry: HPS, Replica, Matterport, ScanNet scenes. Several experiments demonstrate that our method outperforms existing methods that navigate paths in 3D scenes.

1. Introduction

Our goal is to generate animator guided rich long-term human behavior in arbitrary 3D scenes, including a variety of actions and transitions between them. Such a system should allow for goal-directed generation of humans moving about from one place to another, for example, walking towards the couch to sit on it, and then standing up and approaching the shelf to grab something from it, as illustrated in Figure 1. It should allow users to specify with minimal effort what kind of actions to perform, while keeping the realism and expressivity required for applications such as synthetic data generation, robotics, VR/AR, gaming, etc.

While the community has seen promising progress in animator guided motion synthesis in 3D scenes, most works are restricted to a single action and do not handle transitions [64, 70, 74], preventing them from producing long range diverse motion.

They are also not deployable in a wide variety of real scenes [27, 59, 65, 67]. The reason for this is that they synthesize motion by conditioning on scene geometry and require training on a dataset featuring 3D humans interacting in 3D scenes and objects [26, 27, 74]. Generalizing these methods to arbitrary 3D scenes would require collecting motion data registered to a myriad of possible 3D scenes and objects, which is difficult to scale.

In contrast, real humans can navigate cluttered novel scenes, pick objects from a shelf they have never seen before, and sit on novel furniture and surfaces. Most of the scene clutter is often ignored, and what matters most are avoiding obstacles and the contact points with the scene. Our hypothesis is that motion, to a large extent, is driven to avoid obstacles and focused on reaching the next goal or target contacts in the environments. Thus, it should be possible to generate human motion without accounting for all the details in the 3D scene.

Based on this insight, we propose a novel framework for animator-guided long range motion synthesis in 3D scenes without relying on human motion data registered to the scene. As such, our method can be trained on regular mocap data, which is relatively easily captured and abundantly available [47]. Since our method does not explicitly condition on the geometry of the scene, it can be deployed across 3D scenes with varied geometry.

To materialize this insight, our novel algorithm is built upon the following key ideas. First, we control motion with *action keypoints* in a 3D scene: A set of sparse desired target contacts (we use 3 or 4 contacts such as the location of the two feet and a hand). Action keypoints can be provided by an animator using an interface or generated by automated heuristics, allowing animators to trade off the speed and control. Experiments show that action keypoints are a powerful abstraction of several actions in 3D scenes, and can be used to execute instructions such as "sit there" or "grab at this height". Avoiding obstacles in 3D scenes can be achieved by path following. The challenge with auto-regressive motion models is to follow arbitrarily long paths with multiple actions in between. We need to ensure that the human can navigate long distances without drift, smoothly making the human transition into and out of the action, and then walk towards the next target. This typically requires time consuming manual action phase annotation [27, 59]. Instead, we address this with a transformer based motion synthesis model which is trained with *scene-agnostic motion data* to reach the origin of a *canonical coordinate frame (CCF)*. Instead of simply predicting future poses, we train a walking model and a transitions model such that the last pose is always at the origin of the CCF. The models are trained to converge at the origin of the CCF (walk model), or target pose located at the origin of the CCF We find that these networks successfully learn to converge at the orgin of the CCF, which allows us to control motion. To make the model walk at test time, we break the path into multiple origins at waypoints, and compute the CCF from tangents on the path. Similarly, to satisfy in and out transitions, we optimize a target pose at the CCF to satisfy the action keypoints while leveraging a pose prior [51]. Given target waypoints for walking or target poses for transitions, synthesizing motion is done by iteratively running the walk and transition model on the *CCF*, allowing for long range motion synthesis with transitions in a 3D scene.

For the first time, we demonstrate long-range human motion synthesis on a wide range of scene datasets: Replica [62], Matterport [8], HPS [22], Scannet [10]. Furthermore, we show that our model can perform actions at different places, such as grabbing from any shelf, table or cabinet at any height or sitting on any surface that affords sitting. We will make our code and models publicly available which can be used to synthesize goal directed human motion across 3D scenes.

To summarize, our contributions are as follows:

- We present a method that departs from existing methods for motion synthesis in 3D scenes by only using regular motion capture data and that is deployable across varied 3D scenes.
- We introduce a novel idea of iteratively converging motion at the origin of a canonical coordinate frame, which allows to synthesize long-range motion in 3D scenes.

2. Related Work

Human Motion Prediction without the 3D scene. Predicting human motion is a long studied problem in vision and graphics. Classic works explored using Hidden Markov Chains [5] and Gaussian Processes [68], physics based models [45] for predicting future motion. Recently, recurrent neural networks [19, 32] have been used for motion prediction [3, 17, 49] also in combination with Graph Neural Networks [11, 37, 41, 48], and variational Autoencoders [36] to add diversity [23, 71, 76]. An intrinsic problem of recurrent methods is that they drift over time [2].

More recent approaches employ transformers to generate unconditional or text and music conditioned motion sequences [2, 40, 42, 52, 53]. We also build on transformer

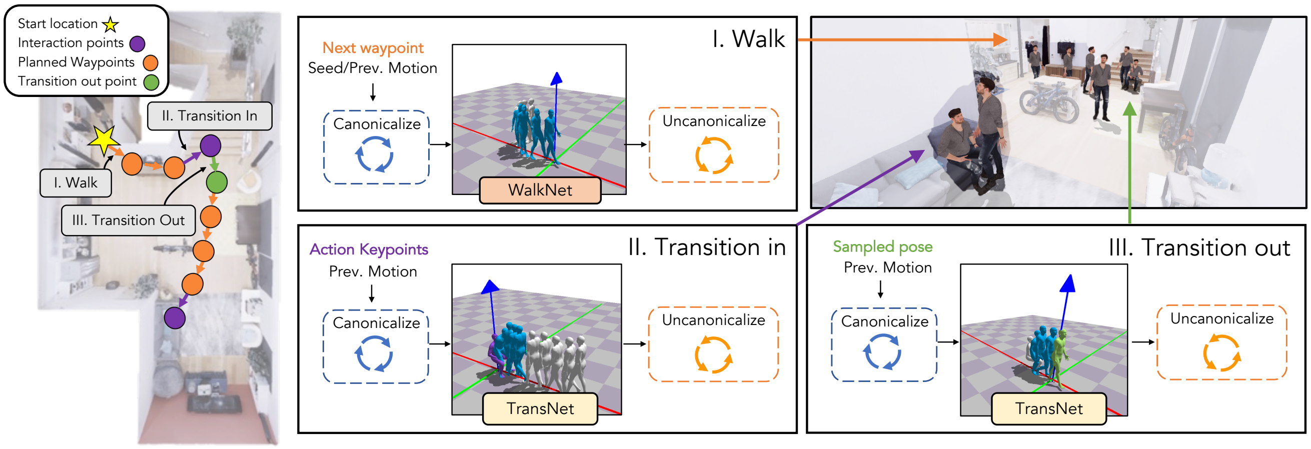

Figure 2. Overview of our method. We generate human motion satisfying action keypoint constraints by diving it into 3 stages: a *Walk Motion*, which animates the human as it walks between actions, a *Transition-In*, which blends the walking motion with the target pose computed from keypoints and a *Transition-Out*, which animates the human back to the walking pose. We use an autoregressive transformer, *WalkNet*, to synthesize the walking motion, and a masked-autoencoder transformer to generate the blending motion *TransNet*.

architectures but aim to generate motion in real 3D scenes.

Motion Inbetweening [1, 14, 25, 35, 50, 72] is another classic paradigm for motion synthesis where the task is to fill in frames between animator provided keyframes.

Our approach is based on transformer architectures [42], and classical ideas such as motion inbetweening, combined with a novel iterative canonicalization and action keypoint to generate continual motion.

Character Control in Video Games. Motion matching [55], its learnt-variant [9, 34] and motion graphs [15, 38, 39, 56, 57] are classical methods often employed in the industry for generating kinematic motion sequences, controlled by environment and user specified constraints. Similar to our goal, some works [6, 54] use a combination of these approaches and IK to generate human behaviors in synthetic scenes. However, these approaches require significant human effort to author realistic animations, and IK approaches easily produce non-realistic animations.

Deep learning variants such as Holden *et al*. [33] introduce phase-conditioning in a RNN to model the periodic nature of walking motion. In several works by Starke *et al*. [59–61] the idea of local phases is extended to synthesize scene aware motion, basketball motion and martial arts motion. All these methods generate convincing motion but phases are non-intuitive for non-periodic motion and often require manual labelling.

Static Human Pose Conditioned on Scenes. The relationship between humans, scenes, and objects is another recurrent subject in computer vision and graphics. Classical works include methods for 3D object detection [20, 21] and affordance prediction using human poses [12, 16, 18]. Several recent works, generate plausible static poses conditioned on the a 3D scene [28, 43, 69, 73, 76, 77] using recently captured human interaction datasets [4, 7, 22, 26, 58, 63]. Instead of static poses, we generate *motion* coherent with the scene which is notoriously harder.

Scene Aware Motion Synthesis. Some works leverage reinforcement learning to synthesize navigation in *synthetic* 3D scenes [29, 44, 75]. Other works focus on a single action, such as grabbing [64, 70] but do not demonstrate transitions to new motions. These methods are not demonstrated in real 3D scenes with multiple objects and clutter. Recent real interaction datasets [4, 7, 22, 26, 58, 63] have powered methods to synthesize 3D scene aware motion [7, 65–67]. These datasets are crucial to drive progress, but do not capture the richness and variety of real world scenes. Hence, these methods are often demonstrated only on small scenes from PROX [26] and Matterport [8]. We draw inspiration from Hassan et al. [27] which combine path planning with neural motion synthesis, and from Zhang et al. [74] which synthesize contact controlled human chair interaction. These methods require the geometry of the isolated interacting object as input, which make them hard to generalize to real 3D scenes. Unlike these methods, we demonstrate *long chained sequences of actions* in *generic real 3D scenes*, which is enabled with our origin canonicalization and action keypoints.

3. Method

Our method takes as input a seed motion sequence and a list of action keypoints $\{\mathbf{a}_1, \ldots, \mathbf{a}_N\}$ specifying N interactions at different locations in the scene. Action keypoints can be specified by users or generated using language commands and scene segmentations (Sec. 3.2). Our goal is to synthesize motion that starts at the seed motion and transitions in and out each of the action keypoints in the input list.

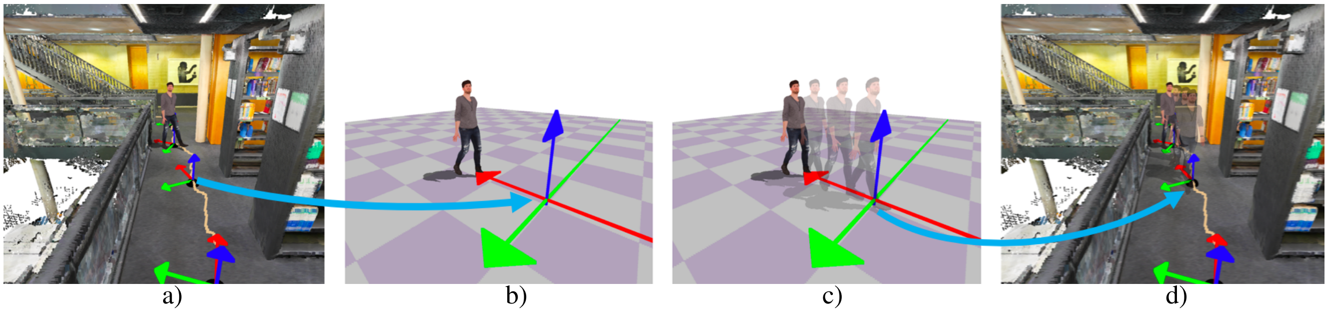

Figure 3. a) Using keypoints and tangents along a path, we move motion from the scene coordinate frame into b) the goal-centric canonical coordinate frame, where c) *WalkNet* synthesizes motion that converges at the origin of the coordinate frame. d) Once the synthesized motion reaches the origin, we move it back to the scene coordinate frame, and iterate the process to reach the next waypoint.

The first step is to optimize for a pose that fits the action keypoints at target locations using Inverse Kinematics and a pose-prior (Sec. 3.3). These poses along with the starting seed motion act as anchors to guide the motion synthesis.

Using scene-agnostic motion capture data placed in a goal-centric CCF (Sec. 3.4), we train *Walknet* (Sec. 3.5) to synthesize walking motion that converges at the origin of a CCF, and *TransNet* (Sec. 3.6) that synthesizes motion inbetween a seed motion sequence and a target pose also at the origin. At test time (Fig. 2), *WalkNet* is used to reach canonicalized intermediate goals along a path computed with a path planning algorithm, thus creating long motion by successively reaching the origin. Once the walking motion reaches the vicinity of an anchor pose, *TransNet* synthesizes transition from walking motion to the anchor pose and vice versa. This allows to synthesize motion in 3D scenes without the need for motion data coupled with 3D scenes. Our framework is general and highly modular, which allows it to be updated with novel methods for motion synthesis.

3.1. SMPL Body Model

We use the SMPL body model [46] to represent the human subject. SMPL is a differentiable function $M(\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{t}, \boldsymbol{\beta})$ that maps global body orientation $\boldsymbol{\phi}$, pose $\boldsymbol{\theta}$, translation $\boldsymbol{t}$ and shape $\boldsymbol{\beta}$ parameters to the vertices of a human mesh along with the 3D joint locations of the SMPL skeleton. We assume that $\boldsymbol{\beta}$ remains static throughout our method. We denote motion sequences as an ordered list of SMPL parameter tuples. For example $\mathcal{C} = [(\boldsymbol{r}, \boldsymbol{\phi}, \boldsymbol{\theta})_j]_{j=1:D}$ denotes a motion sequence of D frames.

3.2. Generating Keypoints in a Scene

Keypoints can be efficiently collected using a 3D user interface, or inferred from the geometry of the scene, and can be therefore generated via action labels or language. An example of automatic KP generation can be seen in Fig. 4. Given a point cloud of the scene with semantic labels and a language description of a task, we can use simple heuristics to generate keypoints that can synthesize the described motion. More details can be found in the supp. mat.

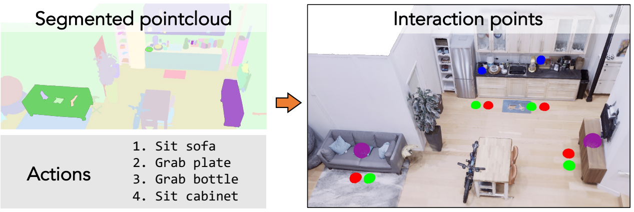

Figure 4. Using language instruction and semantic segmentation, keypoints can be automatically placed in a 3D scene.

3.3. From Action Keypoints to an Anchor Pose

The first step is to infer a pose from the action keypoints in a target location $\mathbf{a} = \{\boldsymbol{k}_i\}_{i=1}^{P}$, where $\boldsymbol{k}_i \in \mathbb{R}^3$ indicates the desired locations for corresponding SMPL joints denoted as $m_i(\cdot)$. We find as few as three to four joints ($P = 3, 4$) are usually sufficient. Since the problem is heavily under-constrained we optimize the latent space of VPOSER [51] dennoted as $\mathbf{z}$. Denoting $f(\mathbf{z}) \mapsto (\phi, \theta)$ as the mapping from the latent space $\mathbf{z}$ to the SMPL pose parameters, we minimize the following objective

$$\boldsymbol{z}, \boldsymbol{t} = \arg\min_{\boldsymbol{z},\boldsymbol{t}} \sum_{i=1}^{P} ||m_i(f(\boldsymbol{z}), \boldsymbol{t}) - \mathbf{k}_i||_2 \tag{1}$$

Please see the supplementary material for further details to make the optimization well behaved. We repeat this step for each target action $\mathbf{a}_1 \dots \mathbf{a}_N$, obtaining N pose-anchors $\mathcal{A} = \{\boldsymbol{t}_i^A, \boldsymbol{\phi}_i^A, \boldsymbol{\theta}_i^A\}_{i=1:N}$.

3.4. Canonical Coordinate Frame (CCF)

One of our key ideas is to make transformers synthesize motion that converges at the origin of a CCF. This way at test time long motion is composed by consecutively going to the next goal placed at the origin. Thus, we canonicalize the training sequence clips by using the planar translation $\boldsymbol{t}_C$, and rotation $\mathbf{R_C}$ of the last frame in a sequence clip as follows

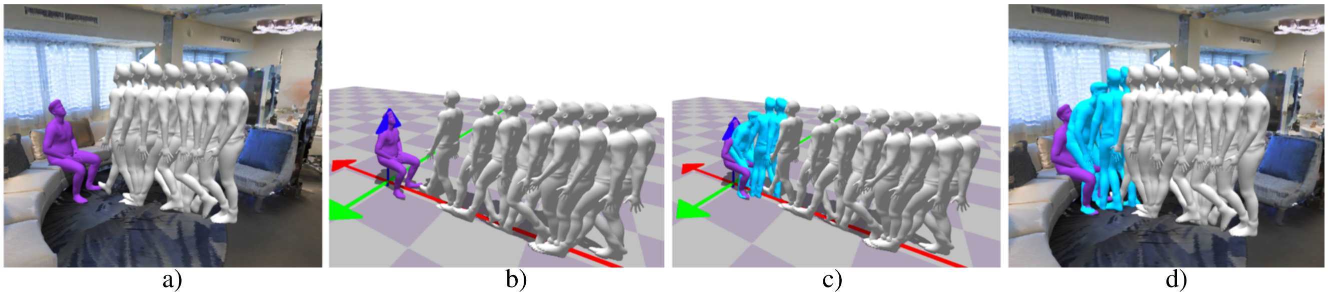

Figure 5. Using a) the motion-anchor pose in the 3D scene (purple), b) we move the motion sequence into the CCF. c) There *TransNet* synthesizes transitions (blue) between the input motion and the pose placed at the origin (purple). d) Once the motion is synthesized, we move it back to the scene coordinate frame.

$$\boldsymbol{\phi}_j^C = \mathbf{R}_C^{-1}\boldsymbol{\phi}_j \ \ , \boldsymbol{r}_j^C = \mathbf{R}_C^{-1}(\boldsymbol{r}_j - \boldsymbol{t}_C) \ . \tag{2}$$

By construction, this transformation outputs a new set of L frames $[(\boldsymbol{r}^C, \boldsymbol{\phi}^C, \boldsymbol{\theta})_j]_{j=1:L}$, where the last pose is at the origin and oriented towards a canonical axis of orientation γ (arbitrary fixed axis). Let $\mathbf{X}$ denote a matrix whose columns are vectorized motion parameters (pose and translation combined) We will use the following notation to denote the canonicalization in Eq. (2) for a full sequence as

$$\mathbf{X}^C = C(\mathbf{X}; \mathbf{R}_C, \mathbf{t}_C) \tag{3}$$

Synthesizing motion in the goal-centric CCF, allows us to synthesize walking motion along paths in a 3D scene (Sec. 3.5) and transitions in and out of actions (Sec. 3.6) without the need for scene registered data.

3.5. WalkNet

Training. Using walking sequence clips of variable lenght L canonicalized (last pose at origin), we train *WalkNet*. *WalkNet* takes K motion frames as input $\mathcal{W}_{inp} = [(\boldsymbol{r}^W, \boldsymbol{\phi}^W, \boldsymbol{\theta}^W)_j]_{j=1:K}$ and predicts the next K frames in the sequence $\mathcal{W}_{out} = [(\boldsymbol{r}^W, \boldsymbol{\phi}^W, \boldsymbol{\theta}^W)_j]_{j=K:2K}$. The training sub-clips of size $2K < L$ are randomly sampled from the training walking sequences.

Expressing sequences as matrices (columns are translations and poses) as explained in the previous section, the transformer takes as input a matrix $\mathbf{X}_{in} \in \mathbb{R}^{K\times 219}$ and outputs a matrix $\mathbf{X}_{out} \in \mathbb{R}^{K\times 219}$. We denote the learned mapping as $T : \mathbf{X}_{in} \mapsto \mathbf{X}_{out}$. Note that we input the pose as vectorized joint rotation matrices, which make learning more stable compared to using joint angles.

Test time. We use the *WalkNet* to follow long paths, by breaking the path into intermediate goals that are canonicalized to the origin (Fig. 3). To traverse scenes avoiding obstacles, we compute the path between the seed motion $\mathcal{I}$ and the first anchor pose $\mathcal{A}_1$ using A-star[24]. Along the path, we sample P goals and compute tangents to the path: $\{\boldsymbol{q}_p, \boldsymbol{l}_p \in \mathbb{R}^3\}_{p=1...P}$. Then we recursively canonicalize such that tangents $\mathbf{l}_p$ align with the canonical axis γ. Hence, canonical translation and rotation are computed as follows

$$\mathbf{t}_C = \boldsymbol{q}_p, \qquad \mathbf{R}_C = \exp(\widehat{\boldsymbol{l}_p \times \boldsymbol{\gamma}}) \tag{4}$$

where $\exp(\cdot)$ is the exponential map recovering the rotation from the screw-symmetric matrix $\widehat{\boldsymbol{l}_p \times \boldsymbol{\gamma}}$. With this, the motion sequence from goal $p-1$ to goal p is obtained by canonicalizing, predicting future motion with the learned mapping T and uncanonicalizing

$$\mathbf{X}_{in} \xrightarrow{C(\cdot, \mathbf{R}_C, \mathbf{t}_C)} \mathbf{X}_{in}^C \xrightarrow{T} \mathbf{X}_{out}^C \xrightarrow{C(\cdot, \mathbf{R}_C^T, -\mathbf{t}_C)} \mathbf{X}_{out}. \tag{5}$$

Although the transformer outputs K future frames, at test time, we use it recursively with a stride of 1 for better performance. That means we effectively predict one pose at a time, and we discard the $K+1:2K$ frames. In this manner, the motion always goes to the origin, we never have to explicitly send the goal coordinates as input to the network, and we do not drift. When we are sufficiently close to an anchor pose, we predict the transition with TransNet.

3.6. TransNet

We synthesize transitions between walks and actions again in a canonicalized frame. To do so, we train *TransNet* - a transformer based motion inbetweener - using AMASS sequences placed in the CCF. The task of *TransNet* is to fill in the motion from a seed sequence $\mathbf{X}_{in}$ to a target *anchor pose*.

Training. We train *TransNet* by asking it to recover training clips from masked out ones. We observe that directly asking to infill many frames does not work reliably. Inspired by training of language models [13], we progressively grow the mask during training until the desired length. Formally, let $\mathbf{X}$ be a training clip of length M, let $\mathbf{V} \in [0,1]^{M\times 219}$ be a matrix mask with zero-column vectors for the frames that need to be infilled. The network is tasked to recover $\mathbf{X}$ from the masked out matrix $\mathbf{X} \odot \mathbf{V}$. The mask $\mathbf{V}$ is progressively grown to mask all the motion frames between $\frac{M}{2}$ to $M-1$ frames – everything except the seed motion and the last anchor pose. For more details, please see supp. mat.

Test time. We use *TransNet* to synthesize transitions in 3D scenes by moving $\frac{M}{2}$ frames of a motion sequence into the CCF by using the orientation and position of the motion-anchor pose - the motion-anchor pose is thus placed at the origin of the CCF. *TransNet* is then tasked to infill the missing frames (Fig. 5).

3.7. Chained actions

With our models and representations we can chain actions trivially. At run time, we have to satisfy an arbitrary number of actions keypoints $\{\boldsymbol{a}_1, \ldots \boldsymbol{a}_N\}$ at different locations. First we compute anchor poses as explained in Sec. 3.3. Obstacle free paths connecting the locations of actions are computed with A*. We rely on *WalkNet* to follow paths until we are sufficiently close to the first anchor pose. Feeding TransNet with the last $M/2$ predicted frames of *WalkNet* and the anchor pose, we predict the transition into the first anchor pose. To transition out we also use TransNet with no modification. We sample a location along the path from $\boldsymbol{a}_1$ to $\boldsymbol{a}_2$ at a fixed distance δ and place a walking pose from our database. TransNet then can transition into this walking pose (Fig. 5). Then we activate *WalkNet* and the process is repeated until all actions are executed. In addition, we can repeatedly use TransNet to execute several actions at the same location, like grabing at different heights.

4. Experiments

In this section we present implementation details of our method. Next, we compare our approach with existing methods. Our experiments show that we clearly outperform existing baselines. Next, we ablate our design choices and finally present qualitative results of our method.

4.1. Implementation Details

WalkNet and *TransNet* are BERT [13] style full-attention transformers. Both consist of 3 attention layers - each composed of 8 attention heads. We use an embedding size of 512 for both transformers. For more details please see the supplementary material. For training both transformers, we set the learning rate to $1e^{-5}$. Both networks are trained using an $L2$ loss. We set $M = 120$ and $K = 30$. We experimented with three different values of M and found that $M = 120$ produces the least foot-skating. Please see the supplementary material for these experiments.

4.2. Datasets

Motion Data: To train *TransNet* and *Walknet* we use the large mocap dataset **AMASS** [47]. For exact details how this is done, please see the supplementary material.

Scene Datasets: We demonstrate that our method is able to generate realistic human motion in scenes from **Matterport3D**, **HPS**, **Replica** and **ScanNet** datasets. All these datasets have been reconstructed using RGB-D scanners or LIDAR scanners and contain scans with sizes ranging from 20 m^2 to 1000 m^2. While Replica, Matterport scenes contain perfect geometry, ScanNet scenes do not. Our method is able to generalize across all these scenes.

4.3. Evaluation Metrics:

We compare our method with existing baselines using perceptual studies and a foot skate metric. Additionally, we ablate various components of our method.

Perceptual Study: We synthesize two motion sequences - one using our method and another using a baseline method and show the two synthesized sequences to participants in our perceptual study. The participant is asked to answer "Which motion looks most realistic?" and "Which motion satisfies scene constraints best?".

Foot Skating (FS): The foot-skate metric measures how much foot-skate occurs during a synthesized motion measured in cm/frame. For N frames, it is defined as:

$$s = \sum_{p=1}^{N} [v_p(2 - 2^{\frac{h_p}{H}})\mathbb{1}_{h_p <= H}]$$

where h_p is the height of the vertex and v_p is the velocity of a foot vertex on the right toe in frame p and $H = 2.5$ cm

4.4. Comparison with Baselines

As aforementioned, no method addresses the task of continual motion synthesis in arbitrary 3D scenes. For completeness we do our best to compare our approach with three existing methods: SAMP [27], GAMMA [75], Wang et al. [67] which all generate animator guided motion by navigating A* paths in 3D scenes. Though, these methods use different forms of animator guidance - such as action labels, we modify them by incorporating the KP information used by our method. Note that except GAMMA, none of these baselines can be deployed in arbitrary 3D scenes without significant modifications, as described below.

SAMP: SAMP is written entirely in Unity and can only synthesize sitting and lying actions in synthetic scenes. It cannot synthesize chained actions or be deployed in arbitrary 3D scenes. For instance it cannot sit on stairs nor perform a grabbing action near a bookshelf. Hence, we compare with SAMP for walking along paths and the actions it was trained on. SAMP synthesizes motion by explicitly conditioning on the geometry of the object of interaction, and by navigating A* paths. For a fair comparison, we replace the real object of interaction in one of our test scenes with a synthetic object in Unity, we port and orient the A* paths from our test 3D scenes into Unity; and use the publicly available code of SAMP. For exact details, please see the supp. mat.

	Ours	SAMP	Ours	GAMMA	Ours	Wang et al.
Which motion is most realistic (%) ↑	**71.8**	28.2	**95.6**	4.4	**100**	0
Which motion satisfies scene constraints best (%) ↑	**76.8**	23.2	**100**	0	**100**	0

Table 1. Comparisons between our method and existing baselines using a perceptual study.

	Language	Manual	*WalkNet*	MoGlow	*TransNet*	NeMF
Foot Skate (cm/f) ↓	0.93	**0.92**	**0.91**	1.88	**1.1**	1.54
User Study (%) ↑	**53.8**	46.2	**75.7**	24.3	**66.8**	33.2

Table 2. Analysis of different components in our method. We compare our method with different baselines across three design components: using language based or manually specified keypoints, the walking motion and the transition motion.

	Ours	SAMP	GAMMA	Wang et al.
Foot-skate ↓	**0.91**	1.34	0.94	4.53

Table 3. Comparisons between our method and existing baselines using the foot-skate metric.

Wang et al.: We run the pre-trained code of Wang et al. on scenes from HPS, Replica and Matterport Datasets. Instead of using action labels to generate anchor poses as done in the original paper, we replace this step with the motion anchors generated using our inverse kinematics step. Since Wang et al. [67] is trained using the PROX dataset and synthesizes navigational motion across A* paths by explicitly conditioning on scene geometry, it does not generalize to 3D scenes beyond these datasets.

GAMMA: GAMMA can navigate 3D scenes but cannot synthesize human-scene interaction. Similiar to the navigation part of our method, it uses the start and end of a path as animator guidance. For the purpose of this comparison, we generate a set of paths in 3D scenes using A* and synthesize walking motion along this path using GAMMA and our method. GAMMA is unable to follow the exact waypoints of the path and as such produces significant interpenetrations with the 3D scene.

For visualizations of motion synthesized by these baselines, please see the supplementary video. We synthesize 5 motion sequences of a total duration of 300 seconds using each method in 5 different scenes for our perceptual study. In Tab. 1, we report the results of our perceptual study with 50 participants (see Sec. 4.3). Each column corresponds to the percentage of users who choose the method corresponding to the column heading. Our results are preferred by the majority of the participants. In Tab. 3, we report the numbers corresponding to foot skate metric.

4.5. Ablation Studies

Can *TransNet* be replaced with other inbetweeners? We compare *TransNet* with the SoTA inbetweening method NeMF [30] for the task of transitioning in and out of actions. For our task of infilling $\frac{M}{2} - 1$ frames in the CCF, *TransNet* produces more natural motion and less foot skating. We hypothesize that this occurs as NeMF is a general purpose inbetweener that can infill an arbitrary number of frames, whereas *TransNet* is a motion inbetweener custom designed for the purpose of infilling $\frac{M}{2} - 1$ motion frames in the CCF. We conduct a new user study with 36 participants, asking users to rate the naturalness of 20 motion sequences by NeMF and *TransNet*. Results are reported in Tab 2.

Can *WalkNet* be replaced with other path following methods? We provide comparisons with SAMP, Wang et al, and GAMMA which all navigate A* paths. As our experiments illustrate, our method outperforms these existing methods for navigation. For further completeness, we trained the SoTA walking method, MoGlow [31], on our walking data. When deployed on 150-200 meter long A* paths, it produces significant foot-skating after about 30 seconds. We hypothesize that this occurs because MoGlow synthesizes motion in an egocentric coordinate frame, and hence the control signal provided by A* changes rapidly, leading MoGlow to synthesize motion with significant drift. We compare our method to MoGlow on these paths using a user study with 36 participants in Tab. 2, where our approach outperforms MoGlow.

How well does language based keypoint placement work? In this experiment, we compare motion synthesized using manual keypoint placement with language based keypoint placement. We synthesize 5 motion sequences using keypoints generated by these two approaches and compare the sythesized sequences using a user study with 36 participants. When used for motion synthesis, these KPs produce similar quality as manual KP placement (Tab. 2).

How long does it take for a user to provide keypoints manually? We develop a user interface which allows users to navigate 3D scenes and to click on locations of interaction. We instruct 7 participants how to navigate 3D scenes with our user interface. On average it takes 245 seconds for users to learn the interface. We then ask each user to provide 5 sets of 3 action keypoints (the location of the root and the two feet or the location of one hand and two feet) for a total of 15 keypoints per scene in 5 different 3D scenes. On average it takes 125 seconds to select these points per scene.

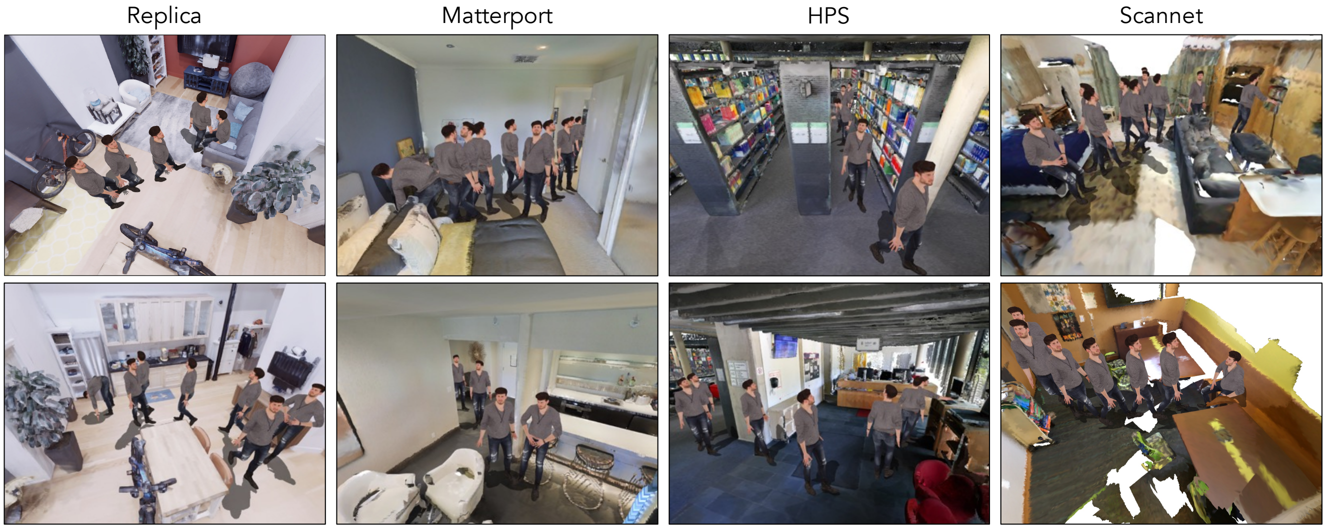

Figure 6. Our method allows to generate motion that generalizes across different scenes. Here we show motion generation in scenes from 4 different datasets: Replica [62], Matterport [8], HPS [22] and Scannet [10].

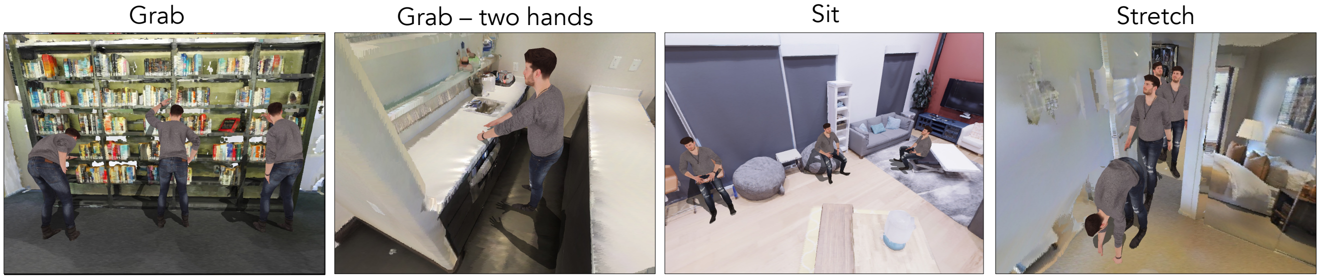

Figure 7. The keypoint representation allows us to generate diverse and highly controllable motion. We show here examples of different grabbing, sitting and newly defined motions.

4.6. Qualitative Results

Please watch the supp. video for qualitative evaluation. In Figure 6, we demonstrate examples of motion generated in scenes from 4 different datasets: Replica [62], Matterport[8], HPS[22] and Scannet[10]. Morover, representing the motion as Action Keypoints allows us to have high control and diversity over the generated motions. In Figure 7 we show how this representation allows us to sit or pick objects at different heights (left column), or generate actions such as grabbing with two hands or stretching.

5. Limitations and Conclusions

We presented the first method to synthesize continual human motion in scenes from HPS, Matterport, ScanNet, and Replica. Our core contribution is a novel method for long-range motion synthesis via iterative canonicalization and the use of action keypoints to decouple scene reasoning from motion synthesis. Experiments demonstrate that our method works better than SOTA methods that generate motion in 3D scenes. While our approach presents an important step towards long-range motion synthesis in 3D scenes, it also has limitations: It assumes a horizontal floor and valid keypoint configurations: if the keypoints do not conform to a valid pose the method will produce unnatural motion. Although we have shown that decoupling scene geometry from motion synthesis is beneficial for generalization, certain actions require more geometry information than the contacts. Future work will investigate how to incorporate geometry abstractions beyond action keypoints while still generalizing to multiple scenes like our method.

6. Acknowledgements

A.M and G.P-M are funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 409792180 (Emmy Noether Programme, project: Real Virtual Humans) and German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039A and the Carl Zeiss Foundation. G.P-M is a member of the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645.

References

[1] Emre Aksan, Manuel Kaufmann, and Otmar Hilliges. Structured prediction helps 3d human motion modelling. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. First two authors contributed equally. 3

[2] Emre Aksan, Peng Cao, Manuel Kaufmann, and Otmar Hilliges. A spatio-temporal transformer for 3d human motion prediction, 2020. 2

[3] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–971, 2016. 2

[4] Bharat Lal Bhatnagar, Xianghui Xie, Ilya A Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Behave: Dataset and method for tracking human object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15935–15946, 2022. 3

[5] Matthew Brand and Aaron Hertzmann. Style machines. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, page 183–192, USA, 2000. ACM Press/Addison-Wesley Publishing Co. 2

[6] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qizhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *ECCV*. 2020. 3

[7] Zhe Cao, Hang Gao, Karttikeya Mangalam, Qi-Zhi Cai, Minh Vo, and Jitendra Malik. Long-term human motion prediction with scene context. In *ECCV*, 2020. 3

[8] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 2, 3, 8

[9] Simon Clavet. Motion matching and the road to next-gen animation. In *Game Development Conference*, 2016. 3

[10] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 2, 8

[11] Lingwei Dang, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. Msr-gcn: Multi-scale residual graph convolution networks for human motion prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 2

[12] Vincent Delaitre, David F. Fouhey, Ivan Laptev, Josef Sivic, Abhinav Gupta, and Alexei A. Efros. Scene semantics from long-term observation of people. In *Computer Vision – ECCV 2012*, pages 284–298, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 3

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*. Association for Computational Linguistics, 2019. 5, 6

[14] Yinglin Duan, Tianyang Shi, Zhengxia Zou, Yenan Lin, Zhehui Qian, Bohan Zhang, and Yi Yuan. Single-shot motion completion with transformer, 2021. 3

[15] Anthony C Fang and Nancy S Pollard. Efficient synthesis of physically valid human motion. *ACM Transactions on Graphics (TOG)*, 22(3):417–426, 2003. 3

[16] David F Fouhey, Vincent Delaitre, Abhinav Gupta, Alexei A Efros, Ivan Laptev, and Josef Sivic. People watching: Human actions as a cue for single view geometry. *International journal of computer vision*, 110(3):259–274, 2014. 3

[17] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015. 2

[18] Helmut Grabner, Juergen Gall, and Luc Van Gool. What makes a chair a chair? In *CVPR 2011*, pages 1529–1536. IEEE, 2011. 3

[19] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013. 2

[20] Abhinav Gupta and Larry S Davis. Objects in action: An approach for combining action understanding and object perception. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 3

[21] Abhinav Gupta, Scott Satkin, Alexei A Efros, and Martial Hebert. From 3d scene geometry to human workspace. In *CVPR 2011*, pages 1961–1968. IEEE, 2011. 3

[22] Vladimir Guzov*, Aymen Mir*, Torsten Sattler, and Gerard Pons-Moll. Human poseitioning system (hps): 3d human pose estimation and self-localization in large scenes from body-mounted sensors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021. 2, 3, 8

[23] Ikhsanul Habibie, Daniel Holden, Jonathan Schwarz, Joe Yearsley, and Taku Komura. A recurrent variational autoencoder for human motion synthesis. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017. 2

[24] Peter Hart, Nils Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4 (2):100–107, 1968. 5

[25] Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. 39(4), 2020. 3

[26] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J. Black. Resolving 3D human pose ambiguities with 3D scene constraints. In *Proceedings International Conference on Computer Vision*, pages 2282–2292. IEEE, 2019. 2, 3

[27] Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael Black. Stochastic scene-aware motion prediction. In *Proceedings of the International Conference on Computer Vision 2021*, 2021. 2, 3, 6

[28] Mohamed Hassan, Partha Ghosh, Joachim Tesch, Dimitrios Tzionas, and Michael J. Black. Populating 3D scenes by learning human-scene interaction. In *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 14708–14718, 2021. 3

[29] Mohamed Hassan, Yunrong Guo, Tingwu Wang, Michael Black, Sanja Fidler, and Xue Bin Peng. Synthesizing physical character-scene interactions. In *SIGGRAPH Conf. Track*, 2023. 3

[30] Chengan He, Jun Saito, James Zachary, Holly Rushmeier, and Yi Zhou. Nemf: Neural motion fields for kinematic animation. In *NeurIPS*, 2022. 7

[31] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020. 7

[32] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 2

[33] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017. 3

[34] Daniel Holden, Oussama Kanoun, Maksym Perepichka, and Tiberiu Popa. Learned motion matching. *ACM Transactions on Graphics (TOG)*, 39(4):53–1, 2020. 3

[35] Manuel Kaufmann, Emre Aksan, Jie Song, Fabrizio Pece, Remo Ziegler, and Otmar Hilliges. Convolutional autoencoders for human motion infilling. In *2020 International Conference on 3D Vision (3DV)*, pages 918–927, 2020. 3

[36] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014. 2

[37] Thomas N Kipf and M Welling. W. 2016. semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*. 2

[38] Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. In *ACM SIGGRAPH 2008 classes*, pages 1–10. 2008. 3

[39] Jehee Lee, Jinxiang Chai, Paul SA Reitsma, Jessica K Hodgins, and Nancy S Pollard. Interactive control of avatars animated with human motion data. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 491–500, 2002. 3

[40] Jiaman Li, Yihang Yin, Hang Chu, Yi Zhou, Tingwu Wang, Sanja Fidler, and Hao Li. Learning to generate diverse dance motions with transformer. *arXiv preprint arXiv:2008.08171*, 2020. 2

[41] Maosen Li, Siheng Chen, Yangheng Zhao, Ya Zhang, Yanfeng Wang, and Qi Tian. Dynamic multiscale graph neural networks for 3d skeleton based human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[42] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Learn to dance with aist++: Music conditioned 3d dance generation. In *ICCV*, 2021. 2, 3

[43] Xueting Li, Sifei Liu, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Putting humans in a scene: Learning affordance in 3d indoor environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12368–12376, 2019. 3

[44] Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel van de Panne. Character controllers using motion vaes. *ACM Trans. Graph.*, 39(4), 2020. 3

[45] C Karen Liu, Aaron Hertzmann, and Zoran Popović. Learning physics-based motion style with nonlinear inverse optimization. *ACM Transactions on Graphics (TOG)*, 24(3):1071–1081, 2005. 2

[46] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 2015. 4

[47] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, 2019. 2, 6

[48] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9489–9497, 2019. 2

[49] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2891–2900, 2017. 2

[50] Boris N. Oreshkin, Antonios Valkanas, Félix G. Harvey, Louis-Simon Ménard, Florent Bocquelet, and Mark J. Coates. Motion inbetweening via deep δ-interpolator, 2022. 3

[51] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 4

[52] Mathis Petrovich, Michael J. Black, and Gül Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In *International Conference on Computer Vision (ICCV)*, pages 10985–10995, 2021. 2

[53] Mathis Petrovich, Michael J. Black, and Gül Varol. TEMOS: Generating diverse human motions from textual descriptions. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[54] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8494–8502, 2018. 3

[55] Paul SA Reitsma and Nancy S Pollard. Evaluating motion graphs for character animation. *ACM Transactions on Graphics (TOG)*, 26(4):18–es, 2007. 3

[56] Alla Safonova and Jessica K. Hodgins. Construction and optimal search of interpolated motion graphs. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26(3), 2007. 3

[57] Alla Safonova, Jessica K Hodgins, and Nancy S Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Transactions on Graphics (ToG)*, 23(3):514–521, 2004. 3

[58] Manolis Savva, Angel X. Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. PiGraphs: Learning Interaction Snapshots from Observations. *ACM Transactions on Graphics (TOG)*, 35(4), 2016. 3

[59] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6), 2019. 2, 3

[60] Sebastian Starke, Yiwei Zhao, Taku Komura, and Kazi Zaman. Local motion phases for learning multi-contact character movements. *ACM Trans. Graph.*, 39(4), 2020.

[61] Sebastian Starke, Yiwei Zhao, Fabio Zinno, and Taku Komura. Neural animation layering for synthesizing martial arts movements. *ACM Trans. Graph.*, 40(4), 2021. 3

[62] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 2, 8

[63] Omid Taheri, Nima Ghorbani, Michael J. Black, and Dimitrios Tzionas. GRAB: A dataset of whole-body human grasping of objects. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[64] Omid Taheri, Vasileios Choutas, Michael J Black, and Dimitrios Tzionas. Goal: Generating 4d whole-body motion for hand-object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13263–13273, 2022. 2, 3

[65] Jiashun Wang, Huazhe Xu, Jingwei Xu, Sifei Liu, and Xiaolong Wang. Synthesizing long-term 3d human motion and interaction in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9401–9411, 2021. 2, 3

[66] Jingbo Wang, Sijie Yan, Bo Dai, and Dahua Lin. Scene-aware generative network for human motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12206–12215, 2021.

[67] Jingbo Wang, Yu Rong, Jingyuan Liu, Sijie Yan, Dahua Lin, and Bo Dai. Towards diverse and natural scene-aware 3d human motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20460–20469, 2022. 2, 3, 6, 7

[68] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):283–298, 2007. 2

[69] Xiaolong Wang, Rohit Girdhar, and Abhinav Gupta. Binge watching: Scaling affordance learning from sitcoms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2596–2605, 2017. 3

[70] Yan Wu, Jiahao Wang, Yan Zhang, Siwei Zhang, Otmar Hilliges, Fisher Yu, and Siyu Tang. Saga: Stochastic whole-body grasping with contact. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2, 3

[71] Ye Yuan and Kris Kitani. Dlow: Diversifying latent flows for diverse human motion prediction. In *European Conference on Computer Vision*, pages 346–364. Springer, 2020. 2

[72] Ye Yuan, Umar Iqbal, Pavlo Molchanov, Kris Kitani, and Jan Kautz. Glamr: Global occlusion-aware human mesh recovery with dynamic cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[73] Siwei Zhang, Yan Zhang, Qianli Ma, Michael J. Black, and Siyu Tang. PLACE: Proximity learning of articulation and contact in 3D environments. In *International Conference on 3D Vision (3DV)*, 2020. 3

[74] Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Vladimir Guzov, and Gerard Pons-Moll. Couch: Towards controllable human-chair interactions. 2022. 2, 3

[75] Yan Zhang and Siyu Tang. The wanderings of odysseus in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20481–20491, 2022. 3, 6

[76] Yan Zhang, Michael J. Black, and Siyu Tang. We are more than our joints: Predicting how 3D bodies move. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, page 3372–3382, 2021. 2, 3

[77] Kaifeng Zhao, Shaofei Wang, Yan Zhang, Thabo Beeler, , and Siyu Tang. Compositional human-scene interaction synthesis with semantic control. In *European conference on computer vision (ECCV)*, 2022. 3