DOI: 10.1111/cgf.14962
Pacific Graphics 2023
R. Chaine, Z. Deng, and M. H. Kim
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 7

DAFNet: Generating Diverse Actions for Furniture Interaction by Learning Conditional Pose Distribution

Taeil Jin[1] and Sung-Hee Lee[1]

[1]Korea Advanced Institute of Science and Technology, Republic of Korea

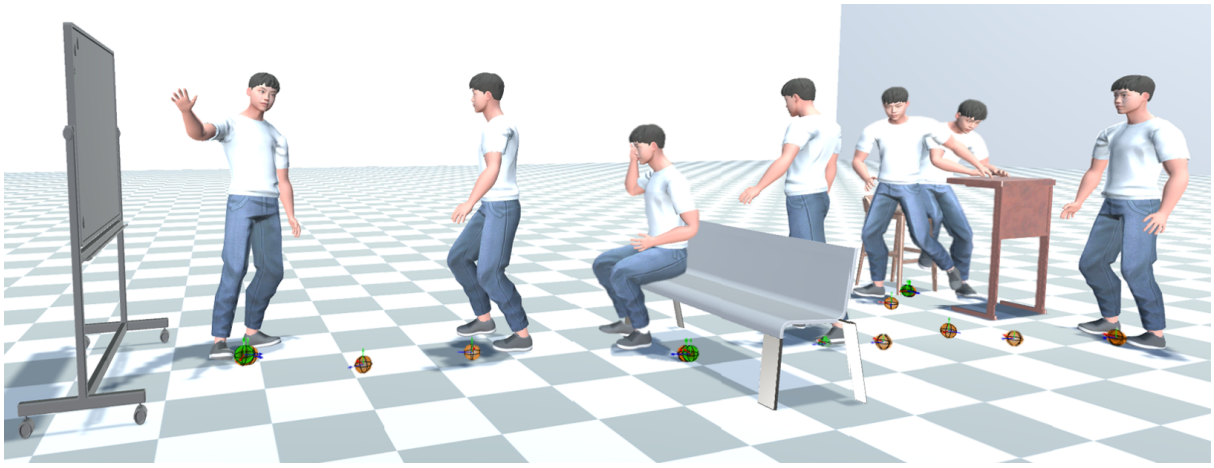

Figure 1: *We propose DAFNet, a novel data-driven framework that can generate various actions for indoor environment interactions. Given the desired root and upper-body pose as control inputs, DAFNet generates whole-body poses for a character appropriate for furniture of various shapes and combinations.*

Abstract

We present DAFNet, a novel data-driven framework capable of generating various actions for indoor environment interactions. By taking desired root and upper-body poses as control inputs, DAFNet generates whole-body poses suitable for furniture of various shapes and combinations. To enable the generation of diverse actions, we introduce an action predictor that automatically infers the probabilities of individual action types based on the control input and environment. The action predictor is learned in an unsupervised manner by training Gaussian Mixture Variational Autoencoder (GMVAE). Additionally, we propose a two-part normalizing flow-based pose generator that sequentially generates upper and lower body poses. This two-part model improves motion quality and the accuracy of satisfying conditions over a single model generating the whole body. Our experiments show that DAFNet can create continuous character motion for indoor scene scenarios, and both qualitative and quantitative evaluations demonstrate the effectiveness of our framework.

CCS Concepts

• ***Computing methodologies*** → ***Motion processing;***

1. Introduction

The generation of character motion under given condition has been a significant research topic for a long time. With the use of deep learning techniques, researchers have made remarkable progress in generating high-quality motions for various actions, such as walking [HAB20], athletic motions [XSLvdP22] with given root trajectory, and dancing with music [VPHB*21]. Compared to generating these motions in free space, creating interactive motion that involves various pieces of furniture in an indoor environment remains a challenging problem. In addition to ensuring the natural-

ness of the resulting motion, this problem imposes challenging requirements, such as avoiding collision with the environment and satisfying various control inputs that specify the character's task.

To this end, researchers have developed regression-based deep learning networks (e.g., [SZKS19, HCV*21]) that output high-quality character motions for given environmental conditions. To enable a single motion generator to create suitable actions for a variety of furniture, these methods train the networks to generate labeled actions specific to each furniture type. However, in indoor environments where multiple furniture items are often involved in simultaneous interactions (e.g., writing at a desk while sitting in a chair), it would be more advantageous if actions could be generated without requiring labeling associated with specific furniture types.

In addition, these regression-based pose generation methods typically require future and goal-oriented features, as well as past and current features of the character, as inputs. These features help to narrow down the range of feasible output poses. However, when fewer network inputs are provided, the regression-based approach may struggle to produce high-quality motion due to the indeterminacy. On the other hand, normalizing flows-based generative models, such as MoGlow [HAB20], have demonstrated remarkable performance in learning the conditional pose distribution using only past and current conditions. However, these methods have so far only been able to generate a single action type, such as locomotion [YYKB21] and dancing [VPHB*21], within a single network. Enabling a single network to produce a wider range of actions still remains a challenging goal.

To address these limitations, we propose DAFNet, a novel framework capable of generating poses for multiple actions interacting with indoor environments, such as walking, sitting, and writing on a board, using a single trained network. The control input to DAFNet is the desired position and orientation of the character's root, and may also include desired head and hand positions or the desired upper-body pose. DAFNet then generates whole-body poses appropriate for nearby pieces of furniture of various shapes while satisfying the control input.

The key component that enables the generation of multiple actions is our action predictor-based learning of multi-conditional pose distribution. To achieve this, we train a Gaussian mixture variational autoencoder (GMVAE) in an unsupervised manner, where the encoder network, dubbed the action predictor, is trained to infer the probabilities of individual action types based on the control input and environment. The action probabilities are provided to the pose generator as an additional condition.

Our pose generator is modeled with the normalizing flow approach. We modify the baseline model MoGlow [HAB20] for a more accurate pose generation under diverse conditions. Specifically, we divide the pose generation module into two: the first normalizing flow transformation for the upper body, followed by the next flow that generates the lower body. This two-step generation approach for the upper and lower body has been demonstrated effective in [HTBX23, GCO*21]. We show that our sequential structure increases the degree of control input satisfaction.

The effectiveness of our method is validated through a number of experiments. We demonstrate that DAFNet can create continuous character motion for indoor scene scenarios, and an ablation study is performed to assess the advantages of the major components of our framework.

Our work makes the following contributions:

- DAFNet generates character poses of diverse action types for a given environment.
- The two-part normalizing flow structure improves the accuracy of the generated poses under diverse conditions.
- Our framework is the first generative approach to learning the pose distribution conditioned on diverse furniture types and control inputs.

2. Related Work

Motion Generation for Control Inputs Optimization-based approaches have been extensively explored to achieve plausible human motions satisfying given control inputs [WK88, GMHP04, LWH*12]. To increase the resulting motion's naturalness, some approaches use single or multiple reference motion data, but the motion quality can be compromised if the given task requires the solution to deviate significantly from the reference motion.

Recent studies developed regression-based deep learning models to learn the relationship between the control inputs and their corresponding motions in a deterministic way [HKS17, SZKS19, BBKK17, FNM19, FLFM15, MBR17]. These models can generate high-quality motions that satisfy the control inputs encoded by neural encoder networks [ZSKS18, SZKS19, SZKZ20, SZZK21, CGM*20]. However, the regression-based approach has limited capability to generate diverse poses from the same input. For the pose generation task, it usually requires highly detailed control inputs, including future and goal-oriented features, which may not be available for real-time applications.

Human-Object Interaction in Indoor Scene The relationship between humans, scenes, and objects has been a recurring topic of research in computer graphics and vision. Traditional studies have focused on techniques for detecting 3D objects [GD07, GSEH11] and predicting affordances based on human poses [DFL*12, GGVG11, FDG*12]. In more recent works, there has been an emphasis on generating realistic static poses within the context of a 3D scene [LLK*19, ZBT21, ZZM*20, HGT*21, ZWZ*22, HWL*23, GDG*23], leveraging newly collected datasets on human interactions [HCTB19, GMSPM21, BXP*22, TGBT20]. Compared to the studies focused on creating static poses, our work tackles a task of generating coherent motions that align with the scene.

Regarding the generation of human-object interaction motions, Sebastian et al. [SZKS19] and Hassan et al. [HCV*21] have developed methods to generate motions interacting with furniture. COUCH [ZBS*22] can generate character motion that achieves inferred hand positions for the target chair. These methods require labeling motion data into discrete action types, where one action is associated with one furniture type (e.g., chair-sitting, bed-lying). On the other hand, our framework learns the probabilities of different action types in a given environment in an unsupervised manner, which naturally allows for performing multiple actions simultaneously. By not associating furniture type with particular action types,

our method can generate many plausible actions appropriate for furniture shape and input upper body pose (e.g., sitting on a bed).

Conditional Probabilistic Models. Probabilistic models form another branch of deep learning-based human motion generation. The probabilistic models use generative models such as Variational Autoencoder (VAE) (e.g., [SMK22, LVC*21]) and Generative Adversarial Network (e.g. [CODB21, KML18]) to learn the distribution of human pose features x. The generative models learn the relationship between a complex distribution $p(x)$ and a simple prior distribution $p(z)$ using neural networks. Then sampling data from $p(x)$ is performed by sampling $z \sim p(z)$ followed by obtaining $x = f(z)$ from the learned networks f. In addition, by adding a condition vector c into the network inputs, they can learn the conditional probability $p(x|c)$. VAE and GAN are popular generative models that have shown remarkable achievements in various tasks. However, VAE may produce blurry results as it optimizes a variational lower bound on model likelihood rather than learning actual maximum likelihood. GAN may suffer from the mode collapse phenomenon.

Our framework is based on the normalizing flows [KD18, HCS*19, MMR*19], a generative model that learns the mapping network f with a series of invertible network modules. The function invertibility allows for computing the exact probability $p(x)$, making it possible to learn the distribution $p(x)$ by training the networks to directly maximize the likelihood. The model can also learn the conditional probability $p(x|c)$ by adding a condition vector c into the network. Using the normalizing flow approach, Henter et al. [HAB20] developed MoGlow, which generates locomotion satisfying a target 2D root trajectory. MoGlow has also been found useful for reconstructing motions with missing markers [YYKB21] and generating dancing motions with music input [VPHB*21]. To more accurately satisfy input conditions, FLAG [ACB*22] introduced an additional network to infer noise input for the pose generative model. However, these models have only learned single action type with a single framework. Our work is the first normalized flow-based glow model to learn conditional pose distributions under various conditions for indoor environment interaction. To address this, we propose several modifications to MoGlow that increase the accuracy of satisfying control inputs.

3. Normalizing Flows

We first explain the basics of normalizing flows, and introduce MoGlow, a method that learns the conditional pose distribution for generating the continuous motion based on Glow structure.

A normalizing flow models data $x \in \mathbb{R}^D$ as an output of an invertible, differentiable function f of noise $z \in \mathbb{R}^D$:

$$x = f(z) \text{ where } z \sim \pi(z). \tag{1}$$

The probability density of x under the function f is obtained by the change of variables formula:

$$p(x) = p(f^{-1}(x)) \left| \det \frac{\partial (f^{-1})}{\partial x} \right|. \tag{2}$$

Intuitively, the model f compresses and expands the density of the noise distribution $p(z)$ and the amount of change is quantified by the determinant of the Jacobian of the transformation by f. The noise distribution $p(z)$ is assumed to be simple, typically a standard normal distribution. The model $f = f_0 \circ f_1 \circ \cdots \circ f_K$ consists of a K-series of invertible neural network module blocks, each of which is called a *flow step*. A well-designed flow step should be easy to calculate its inverse and the determinant of Jacobian. Given a dataset $T = \{x^{(n)}\}_{n=1}^N$, the parameters of f are trained by maximizing the total log-likelihood $\mathbb{L}_{NF} = \sum_n \log p(x^{(n)})$.

The flow step of our baseline model MoGlow [HAB20] adopts the coupling layer scheme of Glow [KD18]. The coupling layer splits input x into two halves, $x = [x_A, x_B]$, and then transforms only x_A using an affine transformation, of which parameters are determined by a deep-learning network according to x_B as well as other conditions. Glow introduced a trainable permutation layer to enhance the generation capability. MoGlow modified Glow with LSTM modules to generate a continuous pose sequence.

4. Method

4.1. Overview

DAFNet is a normalizing flows-based pose generation framework that learns the conditional distribution of human pose. Figure 2 shows the overall architecture of our framework. Denoting a pose $x \in \mathbb{R}^{3j}$ by the positions of j (= 22) number of joints with respect to the root which is a transformation matrix denoted as $R \in \mathbb{R}^{4\times4}$. The root's position is determined as the ground-projection of the pelvis joint, and the root's rotation is computed using the up-vector and the forward direction of the pose. Our framework generates x^τ at time τ according to given desired conditions. The total desired conditions c^τ for our framework consist of

$$c^\tau = \{c^\tau_{vel}, c^\tau_{env}, c^\tau_{ob}\}, \tag{3}$$

where c_{vel} denotes the target velocity of the root and c_{env} is the environment condition. An optional condition c_{ob} represents the desired condition for the upper body.

Specifically, when a new target root R^τ is given for time τ, c^τ_{env} and c^τ_{ob} are expressed with respect to R^τ, while the root velocity c_{vel} from $R^{\tau-1}$ to R^τ is expressed with respect to $R^{\tau-1}$. The environment condition c_{env} is a encoded feature vector that encodes the occupancy ($\in [0,1]$) of the furniture in 2640 sphere shape colliders inside a cylindrical volume, as used by [SZKS19]. As explained in Sec. 4.3, the upper-body observation vector $c_{ob} \in \mathbb{R}^{3j}$ may specify the desired upper-body pose in terms of full joints or only head and hand joints.

Human motion interacting with indoor environments includes many actions. Our experiment (Section 6.1) shows that the baseline model, MoGlow [HAB20], struggles to generate diverse, high-quality actions with the conditions c^τ given as input.

To address this issue, our framework consists of two main components: action predictor and two-part pose generative model. The action predictor automatically disentangles the pose data by predicting action probabilities. The multi-conditional pose generator, designed as a two-part Glow model, generates the upper-body pose and the lower-body pose in sequence to satisfy the desired conditions more accurately.

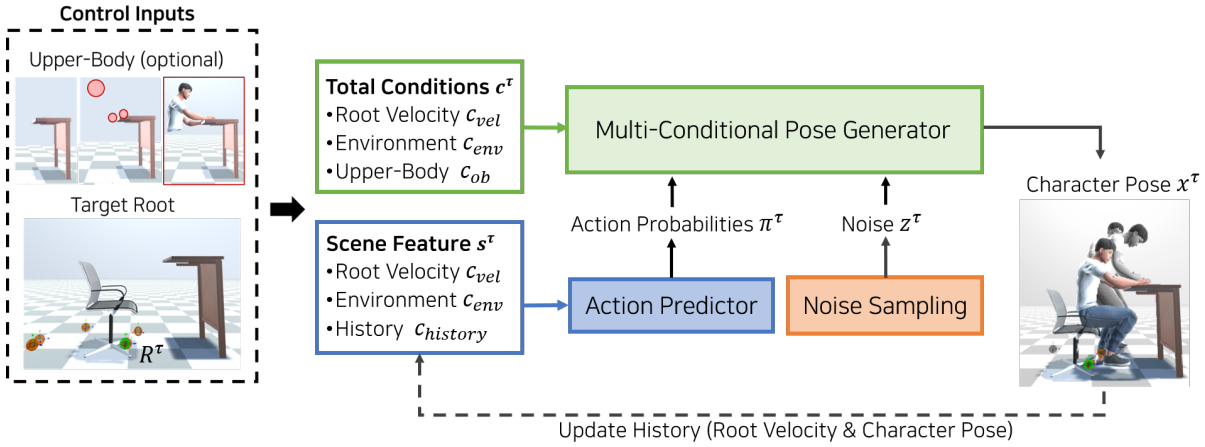

Figure 2: *The overview of DAFNet framework. Given target root velocity and optionally upper-body pose as control inputs, DAFNet generates poses for diverse actions appropriate for the surrounding environment.*

4.2. Action Predictor

The action predictor is the core component of our framework that enables a single pose generative model to generate various interaction motions by automatically disentangling the pose space into action types. Specifically, the action predictor learns to infer the probabilities π of individual action types according to the scene feature defined as:

$$s^\tau = \{c^\tau_{vel}, c^\tau_{env}, c^\tau_{history}\}, \tag{4}$$

where $c_{history}$ represents the past 10 frames of the root velocities c_{vel} and poses x. Figure 3 shows that our action predictor disentangles the data properly.

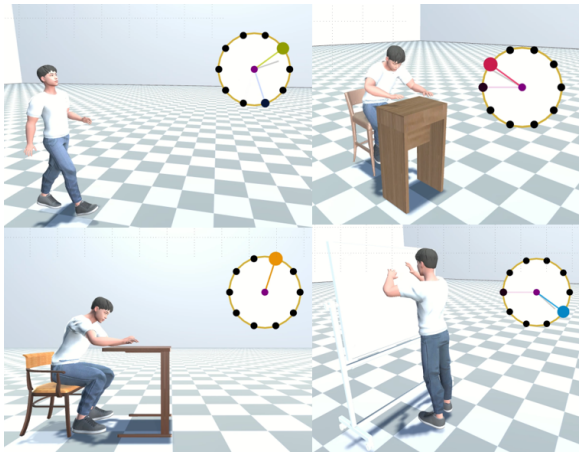

Figure 3: *Examples of the predicted action types for given environments. The action probabilities of the current frame is shown in a circle with the transparency of the line indicating the probability.*

To obtain the action predictor, we modify Gaussian Mixture VAE (GMVAE) [DMG*16] for our purpose as shown in Figure 4. The lower stream constructs the manifold of s. The upper stream is trained such that q_α maps s into K-dimensional probabilities, which are then projected into the manifold of s. From the learned GMVAE model, we only use q_α as our action predictor.

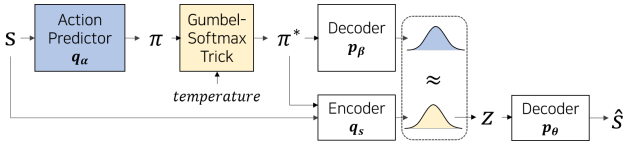

Figure 4: *Training of the action predictor within GMVAE framework.*

The entire networks are trained end-to-end in an unsupervised manner by minimizing the loss terms below:

$$\mathbb{L}_{total} = \mathbb{L}_{recon} + \mathbb{L}_{KL} + \mathbb{L}_{cat}. \tag{5}$$

The reconstruction term $\mathbb{L}_{recon}$ is modeled as a mean squared error between s and $\hat{s}$ to construct the latent space of s. The prior loss term $\mathbb{L}_{KL}$ ensures consistency between the Gaussian distribution from the decoder $p_\beta(z|\pi)$ and the manifold of s obtained by $q_s(z|s)$:

$$\mathbb{L}_{KL} = KL(q_s(z|s)||p_\beta(z|\pi)). \tag{6}$$

The original work [DMG*16] also used the prior and reconstruction loss terms for training. However, as their objective was to sample noise z from a K-Gaussian mixture, the decoder network p_β was modeled to output K means and variances. Additionally, they performed a marginalization over all K labels to calculate $p_\beta(z)$. In contrast, our goal is to develop an action predictor based on scene feature. Therefore, our decoder $p_\beta(z)$ produces a single Gaussian from the estimated probabilities π.

In addition, we include the category term $\mathbb{L}_{cat}$ to encourage the output logit of the action predictor to generate probability. We compute the entropy loss with the logit and the probability value that

is obtained by applying a soft-max function to the logit. The estimated K action probabilities are used as one of the conditions for the pose generator. We empirically set K to 10. A smaller value of K made it difficult to distinguish interactions with different furniture items, such as distinguishing the sitting poses of low and high chairs. Conversely, a higher value of K could represent similar motions with multiple action types. We determined that K=10 provides a balanced division of action types.

The decoder network p_β's performance depends on the accuracy of the predicted action probabilities π. Instead of directly inputting the predicted probability π to the decoder network from the beginning, scheduled sampling for π may help stabilize training and prevent the network from falling into a sub-optimum. For this, we use the Gumbel-softmax reparameterization trick [JGP16] to sample π^*. It starts with a high *temperature* value, which tends to make π^* a uniform probability ($\pi_i^* = 1/K$), and gradually decreases the temperature to allow the predicted probabilities π to be fed as input to the decoder network. We take this annealing process to give a high temperature of 1.0 at the first epoch and to continuously reduce it to 0.5. After training, we set the temperature as 0.7.

4.3. Multi-Conditional Pose Generator

We modeled our multi-conditional pose generator by modifying MoGlow to satisfy the conditions, such as the upper-body pose, more accurately. The pose generator generates a pose for the current time step from the noise sampled from the prior distribution along with the total conditions c^τ and the action probabilities π^τ.

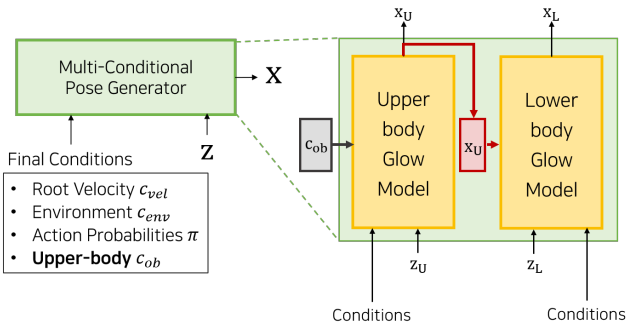

Figure 5: *The structure of our two-part Glow model. The observation vector for the lower-body Glow model is updated to match the generated upper-body pose in the upstream.*

Two-Part Glow Model Figure 5 shows the detailed structure of our two-part pose generator that first generates the upper-body pose followed by generating the lower-body pose. To achieve coherence between the upper and the lower-body poses, we design the model to receive different pose observation vector for each Glow model. We first generate the upper-body pose $x_U \in \mathbb{R}^{u=42}$ with the observation vector in the total conditions. Then for the lower-body Glow model, the upper-body part of the observation vector is updated with the generated upper-body pose x_U.

The rationale behind our two-part model is as follows: The trained permutation layer enhances the overall performance of the coupling layer-based normalizing flow approach (Section 3). However, from the perspective of pose data, the permutation steps break the dimensional mapping between x_i and z_i. Therefore, we cannot isolate the sub-vector of z that is exclusively mapped to x_U with a monolithic Glow model. This makes it extremely difficult to sample various lower-body poses while fixing the upper-body. In contrast, our two-part model generates lower-body pose from the lower-body noise while satisfying the upper-body pose given as the observation.

An additional benefit of the two-part model is the reduced complexity of the model. We found that 3 coupling layers are enough for learning each upper-body and lower-body model, which takes around 0.02 seconds to sample a pose. In contrast, MoGlow needs 16 coupling layers for similar motion quality and takes around 0.05 seconds for inference.

Upper-Body Conditions Our objective is to allow various modes for the upper-body condition to be specified with c_{ob}: FREE (no constraints on the upper-body), EE (only head and hand joints are constrained), and FULL (all upper-body joints are constrained). In FULL mode, all elements belonging to the upper-body (denoted as $x_U \in \mathbb{R}^{u=42}$) within c_{ob} are assigned target values. In EE mode, only the elements corresponding to the end-effector joints (denoted as $c_{ee} \in \mathbb{R}^{3*3}$) are assigned target values, while the remaining elements are set to zero. In FREE mode, all elements are set to zero. During training, we randomly choose one of the three modes, with the initial mode set to FREE. Figure 6 shows the sampling procedures for three different upper-body pose observation modes.

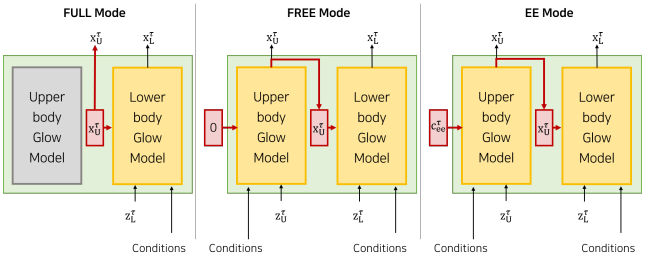

Figure 6: *The various upper-body pose observations.*

5. Implementation Details

5.1. Network Training

Figure 7 shows the process for training and inference of our framework. The two-part Glow model is trained by maximizing the log-likelihood $\mathbb{L}_{NF}$ (Sec. 3) with a pre-trained action predictor. The likelihood is computed from data x as $z = f_\theta^{-1}(x)$ (Figure 7, left) where θ denotes the network parameters.

We standardized the data of the character pose and the root velocity. Our Glow model includes LSTM networks to deal with the pose sequence data. For training, pose sequences of 70 frames are used. As shown in Figure 8, these 70 frames are used to generate 70 likelihoods, from which the mean likelihood for each mini-batch was calculated.

Training took approximately 6 hours for the GMVAE and 24 hours for the pose generator with a GeForce Titan XP. The batch size was 50. All components were trained with a learning rate of 1×10^{-4}, and the Noam learning rate scheduler [VSP*17] with a warm-up of 1k step and a peak learning rate of 1×10^{-3} was employed.

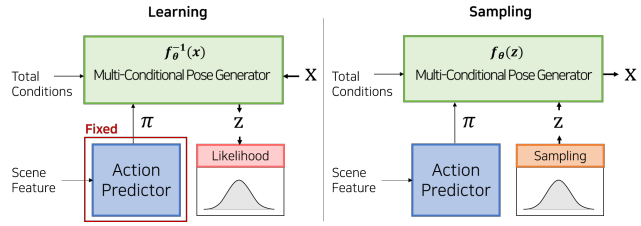

Figure 7: *Left: training of the pose generator with fixed action predictor. Right: The inference process to sample a character pose.*

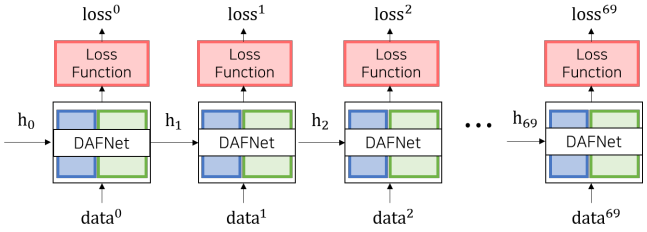

Figure 8: *A pose sequence of 70 frames ($data^{[0\cdots69]}$) is processed by LSTM modules for training.*

5.2. Data Augmentation

For locomotion data, we used publicly available locomotion datasets [HHS*17, MBS09]. In addition, we captured multi-contact motions with respect to chairs and desks with varying heights (high/medium/low) and widths (wide/normal), and a whiteboard. In the capture stage, we asked the actors to approach and interact with furniture at different speeds from different directions for each piece of furniture. Specifically, we positioned the actor in three starting positions: front, right-side, and back. During the interaction capture with the chair-desk combination, we specified which furniture the actor's hands would contact temporarily: either both hands on the desk, both hands on the chair, or one hand on the desk and the other on the chair. We imported all motions into a game engine and matched virtual furniture to be consistent with the interaction motion.

Locomotion data was augmented by mirroring as well as reverse-playing for backward-moving motions [HAB20]. For augmenting multi-contact motion, we randomly switched 3D geometry of furniture as done by [SZKS19, HCV*21] and retargeted the source motion to different heights of furniture. Our contact-engaging motion has diverse contact configurations with the hand, hip, and foot as shown in Figure 9. To retarget such non-trivial interaction motions, we adopted the method of [TAAP*16]. We represented the spatial relationship between a character and a source object using

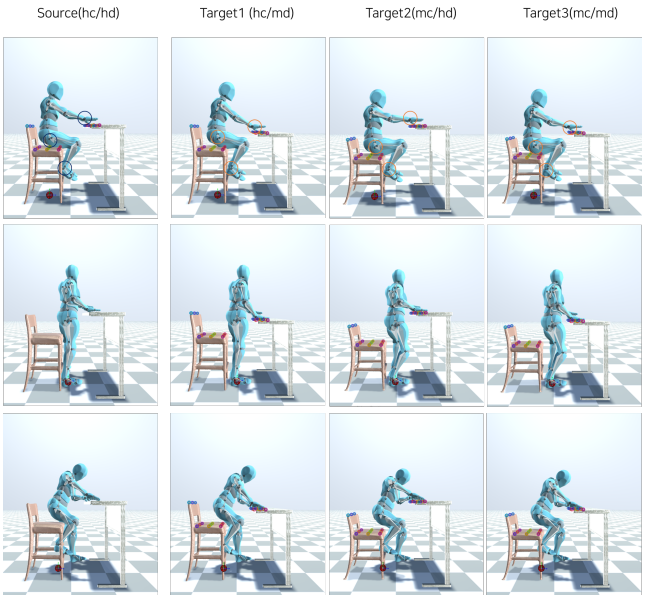

Figure 9: *Data augmentation examples. Target furniture is set to be lower than the source furniture to ensure naturalness of the retargeted motion (hc: high chair, hd: high desk, mc: medium chair, md: medium desk).*

sample points on the object surface and determined the target position of the character joints with respect to the sample points on the target object so as to preserve the original spatial relationship.

We found that setting the target furniture height lower than that of the source furniture secured naturalness of the retargeted motion. For example, a motion for a high chair-high desk combination is retargeted to a high chair-medium desk, a medium chair-high desk, and a medium chair-medium desk as shown in Figure 9. We collected approximately 34k motion clips of 70 frames as our dataset.

6. Experiments

In this section, we qualitatively and quantitatively evaluate the plausibility and flexibility of the motions generated by our framework, and compare with SAMP and MoGlow. Please refer to the accompanying video for clear qualitative evaluations. Code and data will be released, upon paper publication.

We used two datasets for comparison. SAMP dataset was used to train our model and compare with the SAMP model. The training took 24 hours. In addition, our own dataset was used to train our model and MoGlow for comparison, and to conduct ablation studies. It took 36 hours to train MoGlow and our model each. After training, we used a Ryzen 5 CPU with 6 cores for all experiments.

6.1. Qualitative Evaluation

Generating Diverse Actions. Our model is capable of generating diverse actions suitable for input furniture. Figure 10 shows that different approaching motions are created for different furniture types when only the target root trajectory are given. Figure

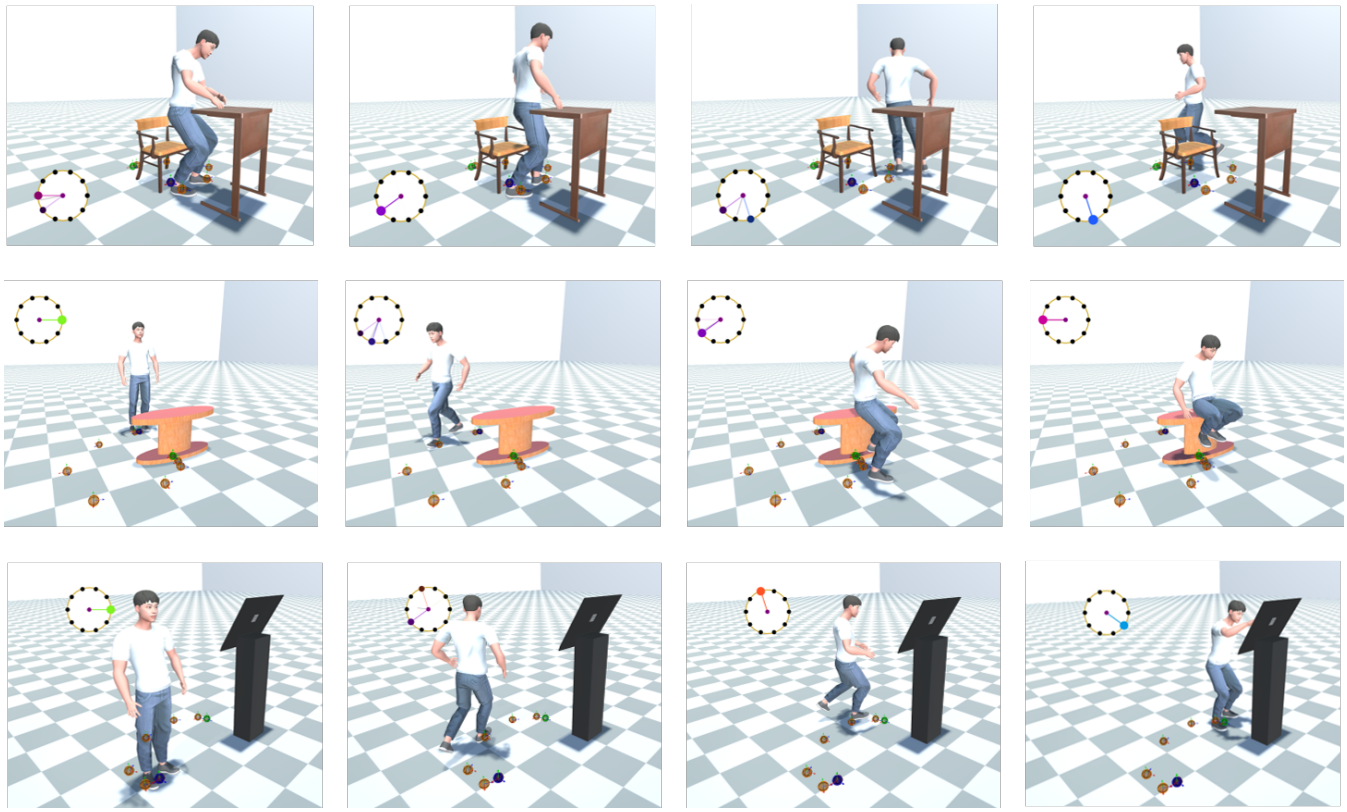

Figure 10: *Diverse actions can be generated for different furniture types.*

11 shows that different motions suitable for furniture of varying heights can be generated from the same root trajectory input.

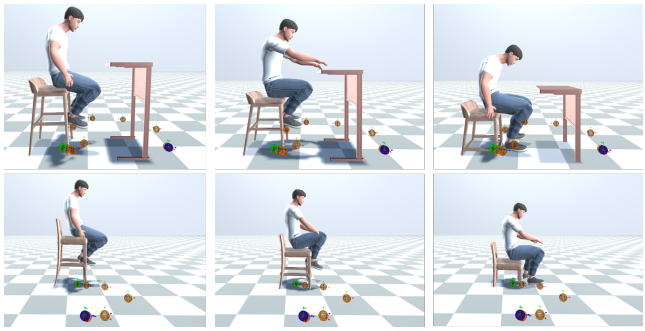

Figure 11: *Final poses for various heights of furniture with the same root trajectory.*

Accommodating Varying Upper-body Input Types. Our framework runs with different types for the upper-body conditions. Figure 12(left) shows the results when the whole upper-body pose is given as input. The whole-body pose satisfying the input upper-body pose is created adaptively to the given chair-desk variations. Figure 12(right) shows that our framework fulfills given upper body tasks, greeting and writing, by generating proper lower body motions (sitting and standing) according to given configuration of a desk.

Figure 13 shows an example of generating motions with the target positions of the two hands and head given as input. While the COUCH focuses on generating motions to realize target hand-chair contact, our model can also generate non-contact interactions, such as greetings, and continuous interactions, such as writing, by sampling poses that satisfy the target position of a given end-effector.

Figure 12: *Left: various lower body poses are generated according to the environment while satisfying the input upper-body pose. Right: different actions can be generated for the same furniture type.*

Figure 14 and Figure 1 show a long sequence of motion generated by our method with varying upper-body input types. When the upper-body input type changes during motion generation, discontinuity in motion may occur. To enhance motion quality, we inter-

polate the joint angles between consecutive poses when input type changes.

Figure 13: *Motion generated with the desired end-effector (head and two hands) trajectories.*

Figure 14: *A continuous motion created in an indoor scene with various furniture pieces.*

Comparison with SAMP and MoGlow. In contrast to previous regression-based pose generators, our framework can learn pose distribution with sparser conditions, without future or goal-directed conditions. When trained with the same input as ours, SAMP failed to generate plausible motions. A vector concatenating the observation vector c_{ob} and scene feature s was used state for SAMP, and it took 5 days for learning with our dataset.

Figure 15 compares our framework and MoGlow in terms of learning actions for furniture. A vector concatenating the observation vector c_{ob} and scene feature s was used as the input condition for MoGlow. When trained with both interaction and locomotion data in our dataset, MoGlow shows significant artifacts, such as severe foot sliding during walking and floating above chair. In contrast, our framework generates much improved motion qualities across the actions.

6.2. Quantitative Experiment

Experiments were conducted for two upper-body condition modes: EE and FREE. We assessed the models' performance using three metrics. For control satisfaction accuracy, we measured the mean of absolute error of the end-effector (head and hands) positions (EED). The error values were normalized to the character's height of 1.7 meters. For motion fidelity, we measured the foot sliding counts (FS) and collision counts. A foot was considered sliding if the toe-base joint is below 7cm from the ground and had a velocity higher than 0.4cm/s. If both feet were sliding in a frame, FS was increased by two. To detect collisions, we attached sphere colliders of 10cm radius to every joint and performed collision detection between the character and furniture. The collision of joints except the hip, hands, and feet were counted as bad collision (BCC).

Comparison with SAMP We compared the original decoder network of SAMP and our model, both trained on the SAMP dataset. The test dataset consists of sitting motions for four different types of chairs and a lying motion for a bed within a total of 5359 frames.

As the autoregressive structure of SAMP receives an action input and a target furniture, predicts a state that includes various features including the root trajectory, and utilizes the predicted state for estimating the next state. Therefore, it is difficult to give a current condition, such as target hand positions, as input to these autoregression-based motion generation approach. As such we compared the SAMP with our model in FREE mode for fairness. In addition, we compare our model with a variation of the SAMP, called SAMP-A, in which only the features of the predicted pose are autoregressed while other features, including the root path, are set from the ground truth data. We chose SAMP-A to examine the SAMP framework under a similar condition as our model, where the target root is provided.

Table 1 shows the comparison results. Our model outperformed SAMP and SAMP-A in terms of motion quality. In case of SAMP, the motion quality degraded when input action changed, such as sitting to walking. In case of SAMP-A, foot sliding artifact was significant for turning motions. In our framework, the action predictor outputs different action probabilities according to different furniture type and root velocity, such as turn, which helps to improve the motion quality.

Table 1: *Comparison with SAMP.*

	FS ↓	BCC ↓
SAMP	5701	9617
SAMP-A	5427	9846
Ours	**5170**	**8129**

Ablation Study For the ablation study, five types of furniture were used as a test dataset, consisting of low chair, high/wide chair, chair-desk combinations of low and high heights, and a whiteboard, totaling 4135 frames. Each test data includes motion that approaches furniture from different directions, sits or stands, and begins interacting motions.

We compare DAFNet with MoGlow and an ablated variation of our model for quantitative evaluation. AP-P is modeled by replacing our two-part Glow model with a single Glow model. Therefore, comparing MoGlow and AP-P demonstrates the effect of the action

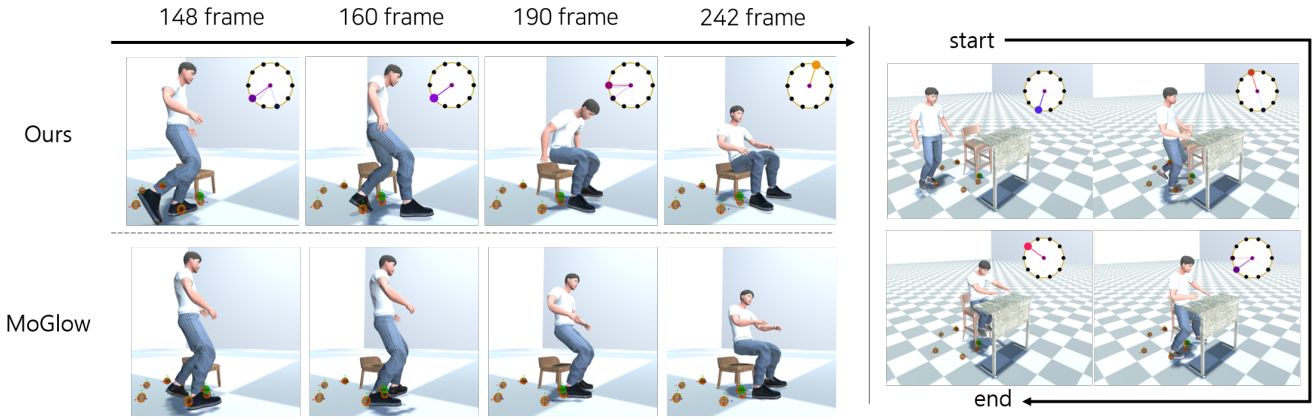

Figure 15: *Motions are generated with only target root trajectory, without specifying the target upper body pose. Left: our method (top) generates higher-quality motions than MoGlow (bottom) when trained with a multi-action dataset. Right: Our method can generate plausible motions even in challenging environments with complex furniture arrangements. The probability of action types is shown in a circle with the transparency of the line indicating the probability.*

predictor, while comparing AP-P and Ours shows the effect of the two-part model. We used 6 coupling layers for every Glow model except for MoGlow, which includes 16 coupling layers.

Table 2 shows the comparison results. Our model outperforms the compared models for most criteria, which suggests that the major components of our framework are all crucial for achieving high performance. MoGlow shows poor performance for all criteria, which suggests that simple concatenation of conditions is not effective for generating diverse action types for human-object interaction. Comparing AP-P and MoGlow shows that the action predictor helps our model learn multiple action types within a single framework. In EE mode, the AP-P has a lower BCC score than Ours. However, when comparing AP-P and Ours in terms of EED, the two-part model significantly increases the accuracy of control input satisfaction. Figure 16 compares the quality of satisfying target end-effector positions. Our model (right), which comprises a two-part Glow model, can satisfy the desired hand positions better than MoGlow and AP-P.

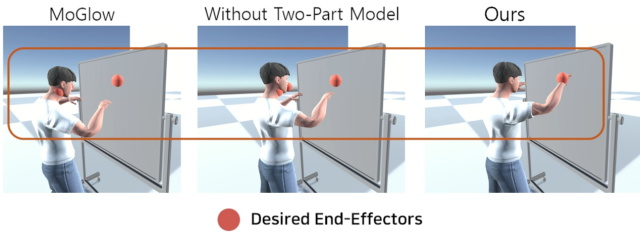

Figure 16: *Motion generated with the desired end-effector (head and two hands). Our model (right) follows the control inputs well, unlike MoGlow (left), and AP-P (middle) which consists of a monotonic Glow model.*

Table 2: *Comparison Results.*

	FS ↓		EED ↓	BCC ↓	
	FREE	EE	EE	FREE	EE
MoGlow	5495	5361	0.230	1328	609
AP-P	2623	2344	0.106	561	**289**
Ours	**2528**	**2050**	**0.026**	**289**	385

7. Limitation and Future Works

Our framework has several limitations that need to be overcome by future research. First, our model is designed to determine the upper body pose first and then the lower body pose, making it suitable for most upper body-oriented daily movements. However, it may not be applicable when the desired lower body pose needs to be specified, such as kicking. In such cases, a two-part structure with the opposite hierarchy to ours should be used. An integrated framework that can flexibly specify the upper or lower body pose as control input is an intriguing research topic.

Our model generates suitable motion for diverse furniture items. Nevertheless, we need to enhance further motion quality. Incorporating predicted foot velocity loss into the training process would improve the generated motion's proper foot velocity. Additionally, considering the character's skin-level shape and integrating finger joints into our framework would effectively reduce inappropriate skin-level hand penetration through furniture.

We considered foot sliding and inappropriate penetrations of rigid bodies as key indicators for evaluating motion quality. Conducting a perceptual user study would enable a more comprehensive assessment of motion quality.

Employing more sophisticated data augmentation techniques can enhance the ability of our framework to handle a broader range of furniture geometries. Recent studies have demonstrated that diffu-

sion models can generate diverse furniture shapes using only partial furniture shapes as input [ZDW21]. Developing a diffusion model that can produce a range of furniture shapes based on a given human pose would be valuable in augmenting human-object interaction data.

Our framework's control input consists of the root and upper body pose. An intriguing area of future research is to develop a method that generates these control inputs based on a given high-level task. By integrating such a method with our framework, we can create an intelligent virtual agent capable of performing high-level tasks while interacting with the environment.

8. Conclusion

This paper presented DAFNet, a novel data-driven character pose generation framework for indoor environment interactions. Given the desired root and upper-body pose, our framework can generate the whole-body poses appropriate for furniture of various shapes and combinations while satisfying the desired upper-body pose. The strengths of our framework are achieved by the action predictor and a two-part normalizing flow structure. The action predictor helps generate diverse action types appropriate for the given environment. The two-part normalizing flow structure enhances the accuracy of the generated poses with both the environment and the upper-body control inputs. Overall, our framework can create continuous character motion for indoor scene scenarios.

Acknowledgement

This work was supported by IITP, MSIT, Korea (2022-0-00566) and NRF, Korea (2022R1A4A5033689).

References

[ACB*22] ALIAKBARIAN S., CAMERON P., BOGO F., FITZGIBBON A., CASHMAN T. J.: Flag: Flow-based 3d avatar generation from sparse observations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 13253–13262. 3

[BBKK17] BUTEPAGE J., BLACK M. J., KRAGIC D., KJELLSTROM H.: Deep representation learning for human motion prediction and classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 6158–6166. 2

[BXP*22] BHATNAGAR B. L., XIE X., PETROV I. A., SMINCHISESCU C., THEOBALT C., PONS-MOLL G.: Behave: Dataset and method for tracking human object interactions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 15935–15946. 2

[CGM*20] CAO Z., GAO H., MANGALAM K., CAI Q.-Z., VO M., MALIK J.: Long-term human motion prediction with scene context. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16* (2020), Springer, pp. 387–404. 2

[CODB21] CHOPIN B., OTBERDOUT N., DAOUDI M., BARTOLO A.: Human motion prediction using manifold-aware wasserstein gan. *arXiv preprint arXiv:2105.08715* (2021). 3

[DFL*12] DELAITRE V., FOUHEY D. F., LAPTEV I., SIVIC J., GUPTA A., EFROS A. A.: Scene semantics from long-term observation of people. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VI 12* (2012), Springer, pp. 284–298. 2

[DMG*16] DILOKTHANAKUL N., MEDIANO P. A., GARNELO M., LEE M. C., SALIMBENI H., ARULKUMARAN K., SHANAHAN M.: Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648* (2016). 4

[FDG*12] FOUHEY D. F., DELAITRE V., GUPTA A., EFROS A. A., LAPTEV I., SIVIC J.: People watching: Human actions as a cue for single view geometry. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12* (2012), Springer, pp. 732–745. 2

[FLFM15] FRAGKIADAKI K., LEVINE S., FELSEN P., MALIK J.: Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 4346–4354. 2

[FNM19] FERSTL Y., NEFF M., MCDONNELL R.: Multi-objective adversarial gesture generation. In *Motion, Interaction and Games*. 2019, pp. 1–10. 2

[GCO*21] GHOSH A., CHEEMA N., OGUZ C., THEOBALT C., SLUSALLEK P.: Synthesis of compositional animations from textual descriptions. In *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 1396–1406. 2

[GD07] GUPTA A., DAVIS L. S.: Objects in action: An approach for combining action understanding and object perception. In *2007 IEEE Conference on computer vision and pattern recognition* (2007), IEEE, pp. 1–8. 2

[GDG*23] GHOSH A., DABRAL R., GOLYANIK V., THEOBALT C., SLUSALLEK P.: Imos: Intent-driven full-body motion synthesis for human-object interactions. In *Computer Graphics Forum* (2023), vol. 42, Wiley Online Library, pp. 1–12. 2

[GGVG11] GRABNER H., GALL J., VAN GOOL L.: What makes a chair a chair? In *CVPR 2011* (2011), IEEE, pp. 1529–1536. 2

[GMHP04] GROCHOW K., MARTIN S. L., HERTZMANN A., POPOVIĆ Z.: Style-based inverse kinematics. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 522–531. URL: https://doi.org/10.1145/1015706.1015755, doi:10.1145/1015706.1015755. 2

[GMSPM21] GUZOV V., MIR A., SATTLER T., PONS-MOLL G.: Human poseitioning system (hps): 3d human pose estimation and self-localization in large scenes from body-mounted sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 4318–4329. 2

[GSEH11] GUPTA A., SATKIN S., EFROS A. A., HEBERT M.: From 3d scene geometry to human workspace. In *CVPR 2011* (2011), IEEE, pp. 1961–1968. 2

[HAB20] HENTER G. E., ALEXANDERSON S., BESKOW J.: Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Transactions on Graphics (TOG) 39*, 6 (2020), 1–14. 1, 2, 3, 6

[HCS*19] HO J., CHEN X., SRINIVAS A., DUAN Y., ABBEEL P.: Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning* (2019), PMLR, pp. 2722–2730. 3

[HCTB19] HASSAN M., CHOUTAS V., TZIONAS D., BLACK M. J.: Resolving 3d human pose ambiguities with 3d scene constraints. In *Proceedings of the IEEE/CVF international conference on computer vision* (2019), pp. 2282–2292. 2

[HCV*21] HASSAN M., CEYLAN D., VILLEGAS R., SAITO J., YANG J., ZHOU Y., BLACK M. J.: Stochastic scene-aware motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 11374–11384. 2, 6

[HGT*21] HASSAN M., GHOSH P., TESCH J., TZIONAS D., BLACK M. J.: Populating 3d scenes by learning human-scene interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 14708–14718. 2

[HHS*17] HABIBIE I., HOLDEN D., SCHWARZ J., YEARSLEY J., KOMURA T.: A recurrent variational autoencoder for human motion synthesis. In *28th British Machine Vision Conference* (2017). 6

[HKS17] HOLDEN D., KOMURA T., SAITO J.: Phase-functioned neural networks for character control. *ACM Trans. Graph. 36*, 4 (July 2017). URL: https://doi.org/10.1145/3072959.3073663, doi:10.1145/3072959.3073663. 2

[HTBX23] HOU S., TAO H., BAO H., XU W.: A two-part transformer network for controllable motion synthesis. *arXiv preprint arXiv:2304.12571* (2023). 2

[HWL*23] HUANG S., WANG Z., LI P., JIA B., LIU T., ZHU Y., LIANG W., ZHU S.-C.: Diffusion-based generation, optimization, and planning in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 16750–16761. 2

[JGP16] JANG E., GU S., POOLE B.: Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016). 5

[KD18] KINGMA D. P., DHARIWAL P.: Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039* (2018). 3

[KML18] KIASARI M. A., MOIRANGTHEM D. S., LEE M.: Human action generation with generative adversarial networks. *arXiv preprint arXiv:1805.10416* (2018). 3

[LLK*19] LI X., LIU S., KIM K., WANG X., YANG M.-H., KAUTZ J.: Putting humans in a scene: Learning affordance in 3d indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 12368–12376. 2

[LVC*21] LI J., VILLEGAS R., CEYLAN D., YANG J., KUANG Z., LI H., ZHAO Y.: Task-generic hierarchical human motion prior using vaes. *arXiv preprint arXiv:2106.04004* (2021). 3

[LWH*12] LEVINE S., WANG J. M., HARAUX A., POPOVIĆ Z., KOLTUN V.: Continuous character control with low-dimensional embeddings. *ACM Trans. Graph. 31*, 4 (July 2012). URL: https://doi.org/10.1145/2185520.2185524, doi:10.1145/2185520.2185524. 2

[MBR17] MARTINEZ J., BLACK M. J., ROMERO J.: On human motion prediction using recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 2891–2900. 2

[MBS09] MÜLLER M., BAAK A., SEIDEL H.-P.: Efficient and robust annotation of motion capture data. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009), pp. 17–26. 6

[MMR*19] MÜLLER T., MCWILLIAMS B., ROUSSELLE F., GROSS M., NOVÁK J.: Neural importance sampling. *ACM Transactions on Graphics (TOG) 38*, 5 (2019), 1–19. 3

[SMK22] STARKE S., MASON I., KOMURA T.: Deepphase: periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics (TOG) 41*, 4 (2022), 1–13. 3

[SZKS19] STARKE S., ZHANG H., KOMURA T., SAITO J.: Neural state machine for character-scene interactions. *ACM Trans. Graph. 38*, 6 (Nov. 2019). URL: https://doi.org/10.1145/3355089.3356505, doi:10.1145/3355089.3356505. 2, 3, 6

[SZKZ20] STARKE S., ZHAO Y., KOMURA T., ZAMAN K.: Local motion phases for learning multi-contact character movements. *ACM Transactions on Graphics (TOG) 39*, 4 (2020), 54–1. 2

[SZZK21] STARKE S., ZHAO Y., ZINNO F., KOMURA T.: Neural animation layering for synthesizing martial arts movements. *ACM Transactions on Graphics (TOG) 40*, 4 (2021), 1–16. 2

[TAAP*16] TONNEAU S., AL-ASHQAR R. A., PETTRÉ J., KOMURA T., MANSARD N.: Character contact re-positioning under large environment deformation. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 127–138. 6

[TGBT20] TAHERI O., GHORBANI N., BLACK M. J., TZIONAS D.: Grab: A dataset of whole-body human grasping of objects. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16* (2020), Springer, pp. 581–600. 2

[VPHB*21] VALLE-PÉREZ G., HENTER G. E., BESKOW J., HOLZAPFEL A., OUDEYER P.-Y., ALEXANDERSON S.: Transflower: probabilistic autoregressive dance generation with multimodal attention. *ACM Transactions on Graphics (TOG) 40*, 6 (2021), 1–14. 1, 2, 3

[VSP*17] VASWANI A., SHAZEER N., PARMAR N., USZKOREIT J., JONES L., GOMEZ A. N., KAISER Ł., POLOSUKHIN I.: Attention is all you need. In *Advances in neural information processing systems* (2017), pp. 5998–6008. 6

[WK88] WITKIN A., KASS M.: Spacetime constraints. *SIGGRAPH Comput. Graph. 22*, 4 (June 1988), 159–168. URL: https://doi.org/10.1145/378456.378507, doi:10.1145/378456.378507. 2

[XSLvdP22] XIE Z., STARKE S., LING H. Y., VAN DE PANNE M.: Learning soccer juggling skills with layer-wise mixture-of-experts. In *ACM SIGGRAPH 2022 Conference Proceedings* (2022), pp. 1–9. 1

[YYKB21] YIN W., YIN H., KRAGIC D., BJÖRKMAN M.: Graph-based normalizing flow for human motion generation and reconstruction. *arXiv preprint arXiv:2104.03020* (2021). 2, 3

[ZBS*22] ZHANG X., BHATNAGAR B. L., STARKE S., GUZOV V., PONS-MOLL G.: Couch: towards controllable human-chair interactions. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V* (2022), Springer, pp. 518–535. 2

[ZBT21] ZHANG Y., BLACK M. J., TANG S.: We are more than our joints: Predicting how 3d bodies move. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 3372–3382. 2

[ZDW21] ZHOU L., DU Y., WU J.: 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 5826–5835. 10

[ZSKS18] ZHANG H., STARKE S., KOMURA T., SAITO J.: Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 1–11. 2

[ZWZ*22] ZHAO K., WANG S., ZHANG Y., BEELER T., TANG S.: Compositional human-scene interaction synthesis with semantic control. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI* (2022), Springer, pp. 311–327. 2

[ZZM*20] ZHANG S., ZHANG Y., MA Q., BLACK M. J., TANG S.: Place: Proximity learning of articulation and contact in 3d environments. In *2020 International Conference on 3D Vision (3DV)* (2020), IEEE, pp. 642–651. 2

Appendix A: Network Architecture

The network architecture of DAFNet is presented in following tables. The Table 3 shows the details of encoder and decoder networks of GMVAE and Table 4 shows the details of the coupling layer of the upper-body and lower-body flow modules.

We apply a dropout of 0.5 to non-zero environmental values during training to enhance the generalization of the environmental sensor values. The network structures of the environment encoder for both architectures are provided in their respective tables.

The final conditions of our pose generator consist of encoded environment conditions (512D), root velocity (3D), and pose observation vector (66D) along with the action probabilities (10D). In the coupling-layer of Glow's flow step, the input x (Upper: 42D, Lower: 24D) is halved into x_A and x_B, and x_B (21D) is affine transformed by scale and bias parameters. The scale and bias parameters are produced by a network that takes x_A and the final conditions as

inputs. The network of the coupling layer is designed as a LSTM module with two hidden layers.

Table 3: *GMVAE Architecture.*

Environment Encoder			
Network	Input	Output	Activation
Linear	2640	512	-
Action Predictor q_α			
Network	Input	Output	Activation
Linear	693+512	512	ReLU
Linear	512	512	ReLU
Linear(fot π)	512	10	-
Decoder p_β			
Network	Input	Output	Activation
Linear(for μ)	10	66	-
Linear(for σ)	10	66	-
Encoder q_s			
Network	Input	Output	Activation
Linear	693+512+10	512	ReLU
Linear	512	512	ReLU
Linear(for μ)	512	66	-
Linear(for σ)	512	66	-
Decoder p_θ			
Network	Input	Output	Activation
Linear	66	512	ReLU
Linear	512	512	ReLU
Linear	512	693+2640	-

Table 4: *Flow Module Architecture.*

Environment Encoder			
Network	Input	Output	Activation
Linear	2640	512	ReLU
Linear	512	512	-
Upper-body Coupling-layer			
Network	Input	Output	Layers
LSTM	512+3+66+10+21	512	2
Linear	512	21*2	-
Lower-body Coupling-layer			
Network	Input	Output	Layers
LSTM	512+3+66+10+12	512	2
Linear	512	12*2	-