NIFTY: Neural Object Interaction Fields for Guided Human Motion Synthesis

Nilesh Kulkarni[1] Davis Rempe[2] Kyle Genova[3] Abhijit Kundu[3]
Justin Johnson[1] David Fouhey[4] Leonidas Guibas[3,5]
University of Michigan[1] NVIDIA[2] Google[3] New York University[4] Stanford University[5]

Contents

1. Overview

This supplementary material provides additional context on the details of the paper along with supplemental results that were omitted due to space constraints. In addition to this document (`supplementary.pdf`), we also provide an HTML webpage of video results (`supplementary.html`) and `supp_video_source` contains all the videos in the webpage. **We encourage the reader to view the webpage** of qualitative results and data examples, which are best to judge the quality of motion and compare results.

In §3, we discuss additional details of our data collection algorithm and evaluate data quality with a user study, while §5 provides details of our NIFTY model. §6 discusses additional qualitative results. In §2, we provide additional details on experiments from the main paper, including our baseline comparison user study in §2.1 and the behavior of different metrics. §4 provides supplemental results to further analyze the performance of our diffusion model and interaction field. Finally, §7 discusses limitations.

2. Experimental Details

This section provides additional details on the implementation of our user study and metrics from the main paper in §4.

2.1. A/B Test User Study

We conduct a user study to qualitatively evaluate the performance of two methods. We design a study such that, given a pair of motions, a user must choose one that is the most realistic. Specifically, we ask the user "Which motion among the both is more realistic?" when we show them two videos (each containing a motion generated by a different method) "LEFT VIDEO" & "RIGHT VIDEO". Fig. 1 shows the instructions and user interface from the study. We conduct 3 such studies using `hive.ai` [1], the results of which are in Fig 5 of the main paper.

Filtering Unreliable Users. We require users to understand instructions given in English. User selection for the study is conditioned on the performance of a qualification test. Users with an accuracy of $\geq 80\%$ on this test are allowed to take the study. To ensure continued reliability during the labeling process we randomly mix the real task data with "obvious" honeypot data where the labels are objective. We require users to have a performance of $\geq 89\%$ on these honeypot tasks. A drop in performance below this results in the user being disqualified from taking the study further.

2.2. Metrics

Apart from performing the user study described in §2.1 we also evaluate all our models and baselines on several quantitative metrics. We detail these metrics below (apart

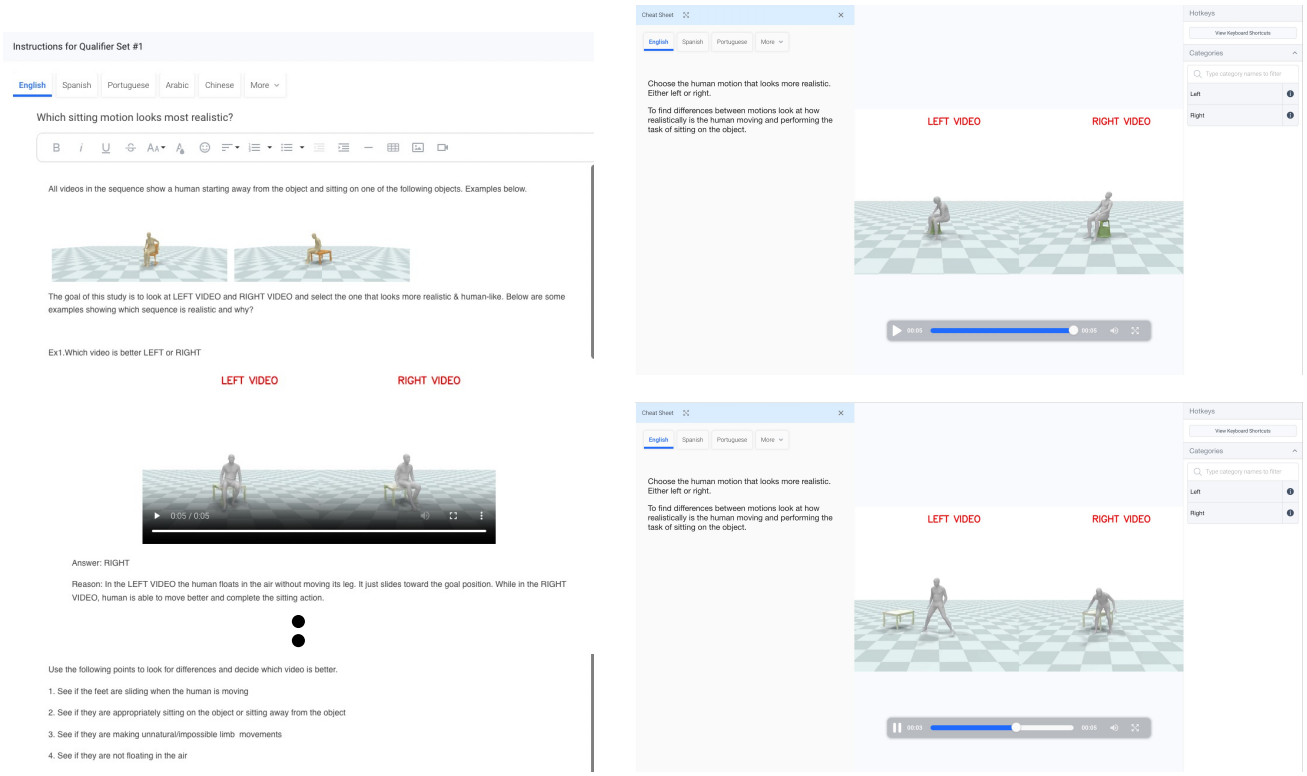

Figure 1. **A/B Testing User Study** We use this study to compare the quality of motions generated by different methods by requiring them to generate human-object interaction motions. On the left, we show the instruction set following which all users are required to pass a qualification exam to participate in the study. On the right, we show the user interface as visible to users. The users answer the question "Which motion is more realistic" and are required to choose one between “LEFT VIDEO" or “RIGHT VIDEO".

from the details already described in Sec 4.2 of the main paper).

Penetration Score. To assess the realism of human motion when interacting with an object, we calculate the penetration score during the approach phase. We define the approach phase as the initial N_A motion frames from a sequence of 150 frames (5 sec). Our rationale for selecting N_A is that during the approach phase, there should be minimal penetration of the human motion into the object geometry. However, during the interaction, there should be increasing contact with the object. These contacts typically result in zero or positive values in the signed distance function (SDF), indicating penetration of points on the object surface into the human SMPL mesh.

We compute N_A for sitting and lifting separately based on our synthetic dataset. In particular, we determine the first frame index of motion where object penetration distance continues to only *increase* thereafter. We assume that after this point, the person is actually interacting with the object and not just approaching it. For sitting, the typical onset of motion interaction occurs after the initial 117 frames of approach, based on the median N_A. Likewise, lifting has a 15th percentile N_A of 124 frames. We use the 15th percentile instead of the median (148 frames) to make this metric more meaningful as 148 frames is almost the end of the complete motion and we wish to evaluate the approach. This difference between *sit* and *lift* action is due to the difference in their inherent interaction with the object.

For completeness, we also report this performance as a function of different N_A values in Fig. 2 (sit) and Fig. 3 (lift).

Skeleton Distance. This metric uses the anchor poses from our human-object interaction data to evaluate whether generated motions faithfully reflect interactions from data. We compute a sum over the per-joint location error (22 joints in our case) between the final generated interaction pose and the nearest neighbor anchor pose from the training dataset in the joint locations space. We report the average of this metric across generated motions.

2.3. Distribution of Test Poses

We plot the distribution of distance between the initial human pose to the object center in meters (X-axis) *v.s.* the kernel density (Y-axis). This distribution is corresponding to the objects poses from the evaluation set. The distance in meters is well distributed and our method is able handle

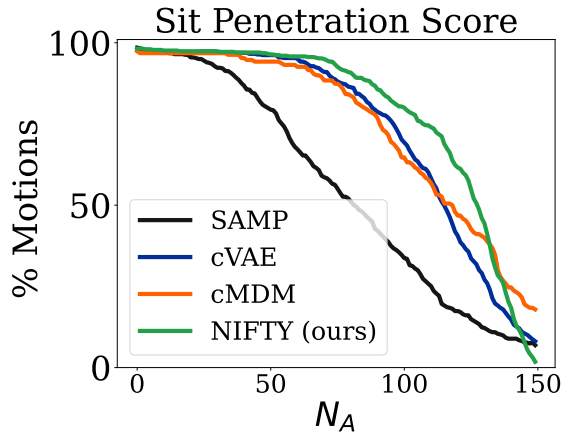

Figure 2. **Penetration Score Sitting**. We graph the percentage of motion sequences with a penetration score of less than or equal to 2cm (Y-axis), compared to the number of approach frames, denoted as N_A (X-axis). Our findings reveal that regardless of the value of N_A, NIFTY (green) consistently exhibits a greater proportion of motion sequences with low penetration scores.

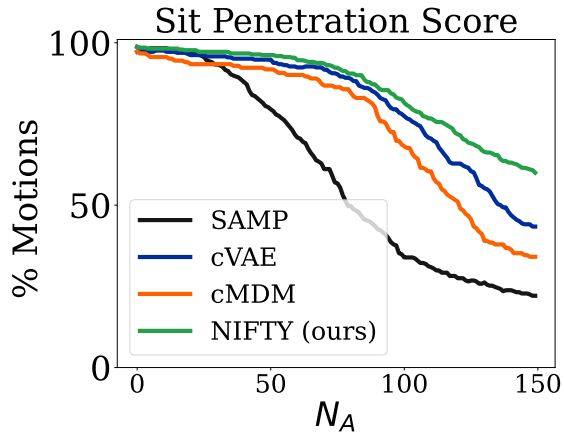

Figure 3. **Penetration Score Lifting**. We graph the percentage of motion sequences with a penetration score of less than or equal to 2cm (Y-axis), compared to the number of approach frames, denoted as N_A (X-axis). Our findings reveal that regardless of the value of N_A, NIFTY (green) consistently exhibits a greater proportion of motion sequences with low penetration scores.

distances in the range of 0m to 6m which we refer to as being in the vicinity of the object.

We highlight some examples from our supplemental page illustrating near and far objects.

Results under `supplementary.html >Addtional Qualitative Results > Sitting`
Sit Near. Wooden Chair(4/13), Arm Chair(11/11), Square Table(8/17), Yogaball(1/11), Stool(4/12)
Sit Far. Wooden Chair(5/13), Arm Chair(2/11), Square Table(6/17), Yogaball(2/11), Stool(5/12)

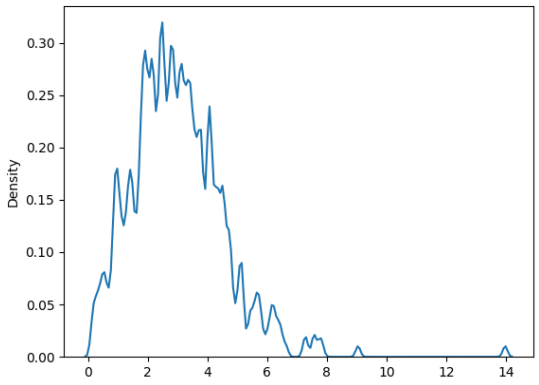

Figure 4. **Sit Action Pose distribution (Test set)** We plot the distribution distances (X-axis) between the initial human poses and the object center in meters as kernel density plot. This is a distribution of object distances for our evaluation set. We see that most of the distribution mass is concentrated between 0m to 6m which represents the operating range for our model.

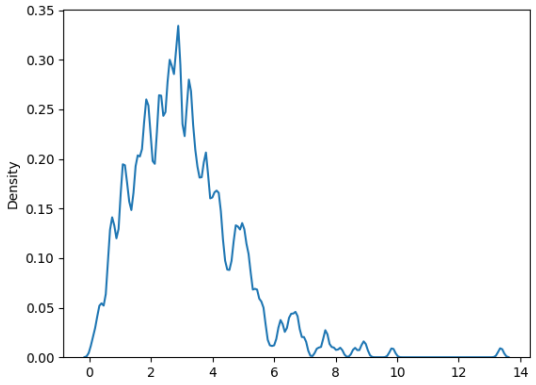

Figure 5. **Lift Action Pose distribution (Test set)** We plot the distribution distances (X-axis) between the initial human poses and the object center in meters as kernel density plot. This is a distribution of object distances for our evaluation set. We see that most of the distribution mass is concentrated between 0m to 6m which represents the operating range for our model.

Results under `supplementary.html >Addtional Qualitative Results > Lifting`

Lift Near. Wooden Chair(7/14), Stool (10/15), Square Table(14/14), Suitcase (2/14)

Lift Far. Wooden Chair(10/14), Stool (2/15), Square Table(2/14), Suitcase (5/14)

2.4. Robustness and Performance

NIFTY uses a fixed N frames for generation. When motions start *far away*, motion guidance can fail since such motions are OOD of our training set. Fig. 6 shows the *Skel. Distance* and *Foot Skate* for different starting distances to the object. Skel. distance is similar at different distances (between 0 to 6m), but at farther distances the foot skate score increases. For large distances, a navigation model would help.

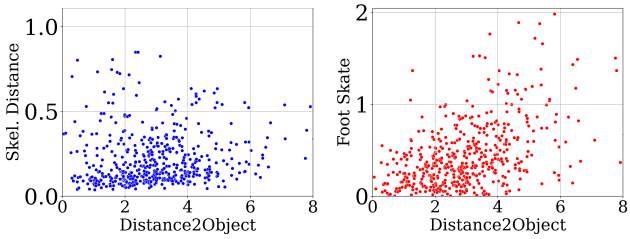

Figure 6. We plot errors in Skel. Distance and Foot Skate for start positions at various distances from the object. We see at far distances ($> 4m$) the Foot Skate gets worse due the fixed, $N = 150$, motion length. Skel. Distance does show any such trends on distances in 1m-6m as it depends on final interaction.

3. Automated Synthetic Training Data Generation

All models in the paper train on synthetic human-object interaction motion data generated using this pipeline. To evaluate the quality of generated data compared to other data, in § 3.1 we perform a large scale user-study with 10K user responses. In § 3.2 we describe the complete details of data generation including pseudo-code for the algorithm.

3.1. Data Quality User Study

Our synthetic data generation pipeline helps us collect high-quality motion data corresponding to different interaction anchor poses. We show that this generated data is high-quality by conducting a user study on a five-point Likert scale as in prior work [10, 11]. Our results show that the generated synthetic training data is on par with data collected using a real mocap setup.

User Study Setup. We created a user-study dataset of 2000 videos, consisting of 500 motions from the AMASS subset of HUMANISE sitting data [13] (*i.e.* real-world *motion captured* data), 500 motions from our data generation pipeline, 500 predicted motions from our NIFTY sitting model, and 500 from cVAE [13] predictions. For each motion, we rendered a video without an object present in the scene to make the source of the video indistinguishable. All motions had a random number of frames uniformly sampled from 60 to 120, where the last motion frame always corresponded to the sitting interaction pose. We only show results on sitting as the HUMNISE [13] does not have lifting interaction AMASS subset in their data.

We ask the users to rate the video on its realism. Users are asked to rate on a scale of 1 to 5 corresponding to "Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree". We set up the study on `hive.ai` [1]. Instructions to the user are shown in Fig. 7.

User Study Results. Results are shown in Fig. 8. As expected, AMASS has a high realism score of 4.87 since it is actual mocap data. Training data generated using our algorithm has an average user rating of 4.39, implying the quality is comparable motion collected using an expensive mocap setup. We also report the performance of NIFTY and cVAE [13] methods on the same study for completeness. NIFTY achieves a strong score of 4.11 (between "agree" and "strongly agree"), which is close to score of the Syn. Data. The cVAE [13] performance remains low at 2.33 (between "disagree" and "neutral").

Filtering Unreliable Users. Note that every user is required to pass a qualification test containing easy examples to label. User accuracy is computed and users with accuracy > 60% are admitted. To ensure that we collect valid responses and that users completely understand the task during the actual study, they are occasionally tested on "obvious" data called "honey pots" during the labeling process. To this end, we add motions with objective "Strongly Agree" labels (motions from AMASS) and some with "Strongly Disagree" labels (low-quality motions generated by cVAE). This is common practice while conducting such a study, and we also do this for the user study in the main paper as detailed in §2.1. The honeypot accuracy for this task is set at 82%: drops in performance below this thresholds removes a user from continuing the study any further.

3.2. Training Data Synthesis Algorithm

Our generation process revolves around utilizing a pre-trained motion model, specifically the HuMoR generative model [8], to produce motion trajectories that *end* in a specific anchor pose. However, we train this model on reverse-time sequences, enabling us to generate reverse-time sequences that *start* from the provided anchor seed pose. Then, when we convert these rollouts into forward motions (*i.e.* play them backwards), the final generated pose in the rollout aligns with the anchor pose by design.

Our full algorithm for generating a single motion tree is shown in Algorithm 1. This algorithm constructs a tree of a specified depth, where each node corresponds to a 1 sec motion clip. Each node is connected to several possible branches to continue the motion (based on a branching factor B). The algorithm begins by creating a root node starting at an input anchor pose. It then repeatedly constructs the tree by generating motion sequences using the `RollOut`

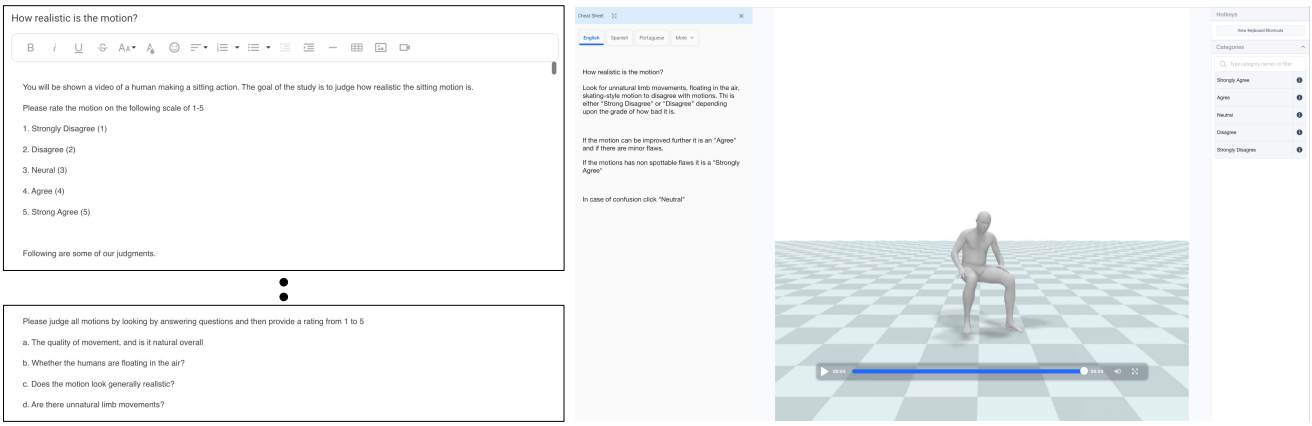

Figure 7. **Likert User Study.** We conduct a user study to assess the motion quality in our Synthetic Dataset. On the left, we present the qualification instructions for participants, allowing only those who perform well to proceed to the actual study. On the right, we display the user interface used for labeling motions, where users select from five options: “Strongly Agree", “Agree", “Neutral", “Disagree", or “Strongly Disagree". The results of this study can be found in Fig. 8

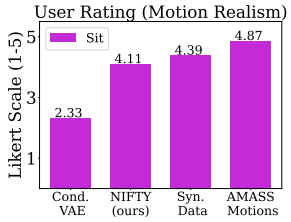

Figure 8. **Likert User Study Results.** We conduct a study to judge the realism of sitting motions on a scale of 1-5. Instructions for this study are available in §3.1. We show that synthetic training data (Syn. Data) generated using our algorithm Algorithm 1 has an average rating of **4.39**. This is comparable to AMASS motions which represent quality of real captured data (using a mocap setup).

function and checking their validity using the `PruneCheck` function. If a valid motion sequence is obtained, a child node is created and added to the tree. The process continues until the desired depth is reached or the tree is fully explored (no more branches left to explore)

The algorithm maintains a queue of nodes to be processed, allowing for breadth-first construction of the tree. If a node reaches the maximum depth, it is skipped to ensure the tree is constructed as per the specified depth. The algorithm outputs the resulting tree, which contains valid motion sequences as paths from the root to the leaf nodes.

RollOut Function. The `RollOut` function takes an start pose and utilizes the pre-trained motion model to generate a short 1 sec (30 frame) motion sequence. It iteratively runs the motion model until a valid sequence is obtained or a specified maximum number of attempts is reached. If a valid sequence is found, it is returned as the generated motion.

PruneCheck Function. The `PruneCheck` function examines a given motion sequence to determine its validity. It algorithmically checks if the motion collides with the object, has unnatural human poses, if the human is floating in the air, or intersecting with the floor *etc*. It returns a boolean value indicating whether the motion sequence is valid or not.

Implementation. In our implementation, we set B as 6 for the nodes at depths 1 and 2, while $B = 2$ for nodes at higher depths. We also set `NTries` as 20 to secure a good rollout sequence. We then convert all the motion nodes in these trees into individual motion sequences for a particular interaction.

4. Supplemental Results

In this section, we include supplemental analyses to support the evaluations in the main paper that were not included due to space constraints. First, we evaluate the effect of having a parametric *vs* a non-parametric guidance field in §4.1. In §4.2, 4.3, and 4.4 we evaluate the impact of hyperparameters like the number of samples at inference, number of anchor poses at training, and a variant of our *Object Interaction Field* that guides a motion *sequence* instead of just the final *interaction frame*. We also evaluate the difference in performance across different objects.

4.1. Non-Parametric Object Interaction Field

We conducted a comparison between our method and a variant where we replaced the object interaction field with a non-parametric field implemented using the nearest neighbor measure. Specifically, during the guidance phase, we identified the nearest anchor pose of the object from the training set and used the difference between this pose and the predicted final pose as the correction. This correction was then utilized to define our distance field and guide the diffusion model accordingly.

Tab. 1 presents the comparison between this baseline and our method. The skeleton distance metric can be sensitive to

Algorithm 1 Tree Generation. Our proposed tree-roll out algorithm using a pre-trained motion-model

```
function RollOut(startPose, N)                          ▷ Input: start pose, N defining number of rollout attempts
    validSequence ← False
    count ← 0
    while not validSequence and count < N do
        motion ← pretrained motion model generate motion using startPose
        validSequence ← PruneCheck(motion)
        count ← count + 1
    end while
    if validSequence then
        return motion
    else
        return null
    end if
end function

function PruneCheck(motionSequence)                     ▷ Input: motion sequence
    valid ← check if motionSequence is valid
    return valid
end function

queue ← empty queue
rootAnchorPose ← input anchor pose
root ← create root node NULL motion                     ▷ For the root node there is no past motion (NULL).
root.lastPose ← root.anchorPose                         ▷ The anchor pose is the seed for future roll-outs
queue.push(root)
while queue is not empty do
    currentNode ← queue.pop()
    if currentNode.depth = MaxDepth then
        continue
    end if
    for child ← 1 to B do
        GMotion ← RollOut(currentNode.lastPose, NTries)     ▷ Create a RollOut
        if GMotion ≠ null then                              ▷ Check if Good RollOut?
            childNode ← create child node with GMotion
            childNode.lastPose ← GMotion last frame         ▷ Set the last motion frame for childNode
            currentNode.children.push(childNode)
            queue.push(childNode)                           ▷ Add childNode to queue
        end if
    end for
end while
```

outliers (*e.g.*, a few generations that are far from the object), so we additionally report % Skel. Dist. $\leq 25cm$ to get a more robust metric. The results demonstrate that our learning approach offers a significant improvement of at least 20% in terms of *Skeleton Distance* $\leq$ 25 cm, as well as an additional 10% in terms of *Contact IoU*. The main paper reports results on the Parametric approach as our primary model.

4.2. Effect of Number of Samples

In the main paper, we generate 10 guided samples from the diffusion model and use the one with the best guidance score. We investigate the impact of varying these number of samples in Tab. 2. We observe that increasing the number of samples leads to improved performance. Particular improvements occur when transitioning from 1 sample to 5 samples. Since guidance does not always result in perfect samples, drawing a diverse set gives better chance for a high-quality

Table 1. **Nearest Neighbor Comparison.** We investigate the effect of learning a parametric function for the Interaction field compared to using the nearest neighbor approach (explained in § 4.1). Our results demonstrate that guiding the diffusion model with our learned field outperforms using a non-parametric field. Specifically, for the sitting action dataset, our Parametric method surpasses the Non-Parametric method by 0.09 points in Contact IoU and achieves an 18% improvement in Skel. Dist $\leq 25cm$. Similar trends are observed in the lift action dataset.

Sitting							
Guidance Objective	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}\%}$ ↓	Skel. Dist. ↓	% Skel. Dist. $\leq 25cm$ ↑	Contact IoU ↑	% Pen. $\leq 2cm$ ↑
Non-Parametric	0.44	99.80	0.00	0.31	47.01	0.45	64.67
Parametric	0.47	99.60	0.00	0.54	65.94	0.54	65.40
Lifting							
Guidance Objective	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}\%}$ ↓	Skel. Dist. ↓	% Skel. Dist. $\leq 25cm$ ↑	Contact IoU ↑	% Pen. $\leq 2cm$ ↑
Non-Parametric	0.32	71.12	0.07	0.52	29.88	0.11	63.02
Parametric	0.34	77.69	0.05	0.42	61.55	0.17	69.49

output. Note that drawing additional samples can be done efficiently in parallel.

Table 2. **Number of Samples Analysis.** We study the impact of drawing multiple samples and guiding them. Drawing more samples helps generate better-quality motions.

Sitting							
# Samples	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}\%}$ ↓	Skel. Dist. ↓	% Skel. Dist. $\leq 25cm$ ↑	Contact IoU ↑	% Pen. $\leq 2cm$ ↑
1	0.66	86.25	7.36	5.72	41.83	0.40	62.59
2	0.56	94.62	4.29	2.36	51.20	0.47	65.47
5	0.47	98.81	0.00	0.67	62.55	0.51	64.72
10	0.47	99.60	0.00	0.54	65.94	0.54	65.40
Lifting							
# Samples	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}\%}$ ↓	Skel. Dist. ↓	% Skel. Dist. $\leq 25cm$ ↑	Contact IoU ↑	% Pen. $\leq 2cm$ ↑
1	0.36	73.11	4.84	2.21	42.03	0.14	64.58
2	0.35	75.70	0.08	1.17	48.80	0.14	67.37
5	0.34	77.29	0.06	0.59	57.57	0.17	67.53
10	0.34	77.69	0.05	0.42	61.55	0.17	69.49

4.3. Effect of Number of Anchors Poses

We also train our Interaction Field (IF) using subsets of motion that yield a limited number of anchor poses. Specifically, we train the IF using 10%, 25%, and 50% of the available seed anchor poses and report results in Tab. 3. It is worth noting that *Contact IoU* and *Skeleton Dist* metrics are calculated using all anchor poses in the training set. However, methods trained with only $X\%$ of the anchor data will not be able to generate the complete range of seed poses. Therefore, when comparing methods trained with different percentages of seed anchor poses, we primarily assess them based on other metrics, but *Contact IoU* and *Skeleton Dist* are still included for completeness.

NIFTY's performance remains stable even with the limited availability of anchor poses. Looking at *Foot Skating*, *D2O*, and *Penetration* metrics, there is not a significant decline in performance. The main paper reports results on 100% data for NIFTY.

Table 3. **Number of Anchors at Training.** We vary the number anchor poses available for training the Interaction Field. We see metrics like Foot Skating, D20, and Pen. are relatively stable as compared to a number of anchors. The evaluation using Skel.Distance and Contact IoU uses all the anchor poses in the training dataset and this evaluation hence hurts the methods that have access to the less anchor poses during training. For this particular ablation we consider Foot Skating, D2O, and Pen. are primary metrics for this ablation.

Sitting							
% Anchors	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}\%}$ ↓	% Pen. $\leq 2cm$ ↑	Skel. Dist. ↓	% Skel. Dist. $\leq 25cm$ ↑	Contact IoU ↑
10%	0.55	95.82	0.00	53.02	1.90	12.35	0.27
25 %	0.54	98.01	0.00	53.86	1.28	28.88	0.34
50 %	0.49	98.21	0.00	59.23	0.96	34.86	0.40
100%	0.47	99.60	0.00	65.40	0.54	65.94	0.54
Lifting							
% Anchors	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}\%}$ ↓	% Pen. $\leq 2cm$ ↑	Skel. Dist. ↓	% Skel. Dist. $\leq 25cm$ ↑	Contact IoU ↑
10%	0.37	83.27	0.07	50.72	0.98	14.54	0.06
25%	0.37	84.86	0.05	46.24	1.32	22.11	0.07
50%	0.36	78.49	0.06	56.34	1.01	24.90	0.08
100%	0.34	77.69	0.05	69.49	0.42	61.55	0.17

4.4. Effect of Number of Input Frames on Interaction Field

In the main paper, our interaction field only considers the last interaction pose, denoted as $\tilde{X}$. However, we want to investigate the impact of extending the interaction field to operate on a sequence of frames rather than just the final interaction frame. To achieve this, we modify our Object Interaction Field to process a sequence of frames from $N-m$ to N, represented as $\{\tilde{X}_{N-m} \dots \tilde{X}_N\}$. Using a transformer encoder, we encode this sequence and obtain a correction vector, denoted as $\Delta\{\tilde{X_{N-m}} \dots \tilde{X_N}\}$. In Tab. 4, we present preliminary results using this spatiotemporal configuration. The results indicate that training such an interaction field is feasible but requires a more careful tuning of different hyperparameters, *e.g.*, the guidance weights. Further investigation into this matter is left for future research.

4.5. Effect of training Interaction Field in the Local Human Frame

Our interaction field is object-centric since it takes in a canonical object point cloud as input. To test this design choice, we implement the object interaction field in the local frame of the human requiring it to understand the spatial positioning of the object w.r.t to the human motion. As shown in Tab. 5, this leads to a subpar performance across the board on sit and lift actions.

Table 4. **Multiple Input Frames to Interaction Field** We show preliminary results on training an interaction field that considers multiple frames as input instead of a single frame like in the main paper. Our results indicate training such a field is feasible the requires further analysis to understand the effect of different hyper-parameters.

				Sitting			
# Input Frames	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}}\tilde{\%}$ ↓	Skel. Dist. ↓	% Skel. Dist.$\leq 25cm$ ↑	Contact IoU ↑	% Pen. $\leq 2cm$ ↑
1	0.47	99.60	0.00	0.54	65.94	0.54	65.40
5	0.66	86.25	7.36	5.72	41.83	0.40	62.59
10	0.56	94.62	4.29	2.36	51.20	0.47	65.47
15	0.47	98.81	0.00	0.67	62.55	0.51	64.72
				Lifting			
# Input Frames	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}}\tilde{\%}$ ↓	Skel. Dist. ↓	% Skel. Dist.$\leq 25cm$ ↑	Contact IoU ↑	% Pen. $\leq 2cm$ ↑
1	0.34	77.69	0.05	0.42	61.55	0.17	69.49
5	0.35	76.10	0.06	0.37	62.55	0.16	67.28
10	0.34	78.09	0.05	0.46	62.55	0.17	69.64
15	0.34	77.49	0.06	0.36	62.95	0.16	68.64

Table 5. **Canonical *vs.* Local Human Frame for Interaction Field Training**. We show that training an Interaction Field in the local human motion frame leads to poor performance as compared to canonical frame.

				Sitting			
Interaction Field Frame	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}}\tilde{\%}$ ↓	Skel. Dist. ↓	% Skel. Dist.$\leq 25cm$ ↑	Contact IoU ↑	% Pen. $\leq 2cm$ ↑
Local Human	0.36	40.04	0.86	2.62	0.20	0.04	53.73
Canonical	0.47	99.60	0.00	0.54	65.94	0.54	65.40
				Lifting			
Interaction Field Frame	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}}\tilde{\%}$ ↓	Skel. Dist. ↓	% Skel. Dist.$\leq 25cm$ ↑	Contact IoU ↑	% Pen. $\leq 2cm$ ↑
Local Human	0.28	41.83	1.02	2.44	0.60	0.02	43.33
Canonical	0.34	77.69	0.05	0.42	61.55	0.17	69.49

4.6. Performance Breakdown Per-Object

We analyze if the performance of our method is biased towards certain objects by computing the metrics for about 100 interaction motion samples per object instance. We show the results of this in Tab. 6. Results indicate that the performance of our method is not dependent on the kind of the object. For instance, in the case of sitting, the performance for sitting on a "Armchair" *vs* "Chair" are close. This demonstrates the flexibility of the NIFTY pipeline to a diverse set of objects.

4.7. Effect of guidance weight

We evaluate the effect of guidance weight on the performance of human object interactions. We observe that with low guidance weight (*e.g.* 10.0) the diffusion models does not satisfy contact requirements as indicated by high "Skel. Distance" (0.48) and low "Contact IoU" (0.42). When we increase the guidance weight to 100.0 it causes the (diffusion) model to break leading to a much higher skeleton distance (2.27 for sitting). This higher guidance weight also leads to more foot-skating score (0.5). We show the results of this in Table Tab. 7.

Table 6. **Performance on actions across objects**. We see that NIFTY's performance is stable across object categories and the framework handles different objects effectively. For instance, the performance on the Armchair and Chair on sitting action are close signaling the flexibility of NIFTY pipeline.

				Sitting			
Object	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}}\tilde{\%}$ ↓	Skel. Dist. ↓	% Skel. Dist.$\leq 25cm$ ↑	Contact IoU ↑	% Pen. $\leq 2cm$ ↑
Armchair	0.42	99.05	0.00	0.42	90.48	0.44	56.73
Chair	0.51	100.00	0.00	0.17	84.31	0.60	49.02
Stool	0.50	96.59	0.01	0.21	68.18	0.54	72.94
Table	0.46	100.00	0.00	0.28	55.88	0.50	68.63
Yoga Ball	0.53	100.00	0.00	0.22	73.33	0.58	52.38
				Lifting			
Object	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}}\tilde{\%}$ ↓	Skel. Dist. ↓	% Skel. Dist.$\leq 25cm$ ↑	Contact IoU ↑	% Pen. $\leq 2cm$ ↑
Chair	0.34	86.82	0.04	0.38	70.54	0.17	59.82
Stool	0.36	77.24	0.06	0.24	65.04	0.13	76.84
Suitcase	0.33	63.85	0.06	0.20	71.54	0.28	65.06
Table	0.29	90.00	0.03	0.64	51.67	0.15	57.41

Table 7. **Effect of guidance weight on performance** We evaluate the effect of guidance weight on the performance of human object interactions. We empirically find that the best performance is achieved at weight of 20.0 and use this guidance weight to report results in the main paper. Best performing numbers are in bold.

				Sitting			
Guidance Weight	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}}\tilde{\%}$ ↓	Skel. Dist. ↓	% Skel. Dist.$\leq 25cm$ ↑	Contact IoU ↑	% Pen. $\leq 2cm$ ↑
10	**0.45**	99.40	0.00	0.48	31.08	0.42	64.93
20.0	0.46	**99.80**	0.00	**0.31**	**68.92**	**0.54**	65.07
50.0	0.47	99.40	0.00	0.47	51.59	0.46	**65.73**
100.0	0.50	96.22	0.00	2.27	43.43	0.42	65.42
				Lifting			
Guidance Weight	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}}\tilde{\%}$ ↓	Skel. Dist. ↓	% Skel. Dist.$\leq 25cm$ ↑	Contact IoU ↑	% Pen. $\leq 2cm$ ↑
10.0	0.35	73.11	0.06	0.43	39.04	0.13	62.94
20.0	0.35	78.69	0.06	**0.31**	**63.15**	**0.18**	68.35
50.0	0.35	**79.88**	0.06	0.49	51.20	0.15	69.08
100.0	0.36	78.29	**0.05**	0.73	50.40	0.14	**71.76**

5. Implementation Details

Recovering Motion from τ^0. Our trajectory representation is over-parameterized and this allows using the model outputs in multiple ways. To recover the generated motion we extract the per-frame joint angles $\boldsymbol{j}_i^r$ for the SMPL model. We integrate the velocity $\boldsymbol{t}_i^v$ along the XZ plane to recover the XZ translation for the root joint and extract the corresponding Y component (upward) from $\boldsymbol{t}_i^p$. This strategy of extracting motion from the output parameterization is motivated by our use of guidance with the diffusion model, which only operates on the last frame of a motion sequence. By integrating velocity predictions over time, applying the guidance objective at the last frame will still have a strong

effect on earlier frames in the sequence.

Variable Length Input. Our model takes input motion trajectories with up to 150 frames. For training, we pad motion sequences of lengths shorter than this with the last interaction frame from the sequence.

SMPL model. Our SMPL [4] model does not have hand articulation, so we use the SMPL model with only 22 articulated joints.

Pre-trained Motion Model for Data Generation. We train the motion model on a subset of the AMASS dataset that does not contain extreme sporting actions like jumping, dancing, *etc*. We do this by removing sequences from AMASS based on the labels from the BABEL dataset [7]. We use the HuMoR-Qual [8] variant of the model to get high-quality motions, which uses the joint positions computed through the SMPL parametric model as input to future roll-out time steps (as opposed to using its own joint position predictions).

Transfomer Encoder.. We use a transformer encoder implemented using `torch.nn.TransformerEncoder` from PyTorch [6]. Our each transformer layer consists of 4 heads and a latent dim on 512. We have 8 such layers in our transformer.

6. Results

Motion generation results are best seen as videos on the webpage. We also include static visualizations here in Fig. 9 and Fig. 10. The webpage also shows visualizations ($\sim$10 motions) from our method for every object in our dataset. Below we highlight salient differences in qualitative results and their possible reasons.

Stocastic Scene-Aware Motions (SAMP).. We tried small variance and motion blending but did not see significant quality changes. We found it hard to adapt SAMP to our large diverse dataset (100 $\times$ size of SAMP's data). After tuning KL/recon weights and other hyper-parameters the performance was still subpar. Additionally, we had to add group norm layers (like [8]) to the SAMP model to prevent gradient explosion

The videos on webpage show SAMP motions are *not* always smooth and display flicker. There is often foot-skating during the approach motions to the object. We hypothesize our setting is challenging compared to the original SAMP [2] setting due to the diversity of characters and the final sitting poses. SAMP uses a pose representation from NSM [9] to learn an auto-regressive motion model which is specific for a single character with and trained with few sitting poses.

Research towards an improved representation could potentially lead to better results, along with modeling the full motion sequence jointly (like in the cVAE baseline). Furthermore, the independent nature of GoalNet and MotionNet in SAMP [2] sometimes leads to sampling goal poses that are not well-suited for the current starting pose, which can lead to heavy penetrations with the object. In contrast, our representation easily handles the diversity in characters and final sitting poses.

Table 8. **Quantitative Comparison** We compare performance of NIFTY and DIMOS on **All Objects(All)**, **Wooden Chair(WC)** and **Arm Chair(AC)**. In all the settings, NIFTY does better on interaction pose metrics (D2O, Skel. Dist, Cont. IoU) but not as well on foot skating(FS) and % Pen.

	Method	FS ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}}\tilde{\%}$ ↓	Skel. Dist. ↓	Cont. IoU ↑	% Pen. $\leq 2cm$ ↑
All	NIFTY	0.47	**99.6**	**0.00**	**0.54**	**0.54**	65.0
	DIMOS	**0.04**	42.63	0.23	1.39	0.01	**87.4**
WC	NIFTY	0.50	**100.0**	**0.00**	**0.16**	**0.59**	52.38
	DIMOS	**0.02**	35.71	0.22	1.39	0.01	**76.67**
AC	NIFTY	0.41	**100.0**	**0.00**	**0.14**	**0.47**	62.12
	DIMOS	**0.11**	46.97	0.33	1.80	0.01	**93.55**

cVAE [13], cMDM [12]. The videos on webpage show that cVAE is a more expressive model than SAMP as it jointly models complete motion sequences instead of single framess. Moreover, using a diffusion model MDM instead of Variational Autoencoder [3] improves the overall motion quality. cVAE and cMDM both struggle to generalize to unseen object poses, and have no mechanism to correct for this at test time.

Comparison to DIMOS [14]. We evaluate the available pre-trained DIMOS model on sitting motions. In Tab. 8 shows quantitative performance compared to NIFTY on All objects, Arm Chair(AC), and Wooden Chair(WC). In a user study, NIFTY motions are preferred by $\mathbf{77.4}\%$ of users (all objects) and by $\mathbf{82.8}\%$ of users (chairs only) compared to DIMOS. We use pre-trained models from DIMOS since both methods use the same underlying AMASS mocap dataset to learn their respective motion models. The weaker performance from DIMOS primarily stems from the RL policy missing the final goal pose, while our interaction field ensures the appropriate termination.

7. Limitations

Our proposed pipeline demonstrates the ability to achieve human-object interaction results with a diverse sets of objects while only relying on a limited number of anchor poses. One of the key factors contributing to the performance of NIFTY is the utilization of a pretrained motion model [8] trained on the AMASS repository [5]. Our data generation pipeline has the capability to generate motions and interpolate between existing data in this dataset. However, in cases where a completely novel and extreme seed anchor pose is provided, such as a headstand, HuMoR would struggle to generate reasonable and high-quality motion sequences. Developing more robust motion models which can handle such poses, would be beneficial.

In its current form NIFTY only works on the specific body

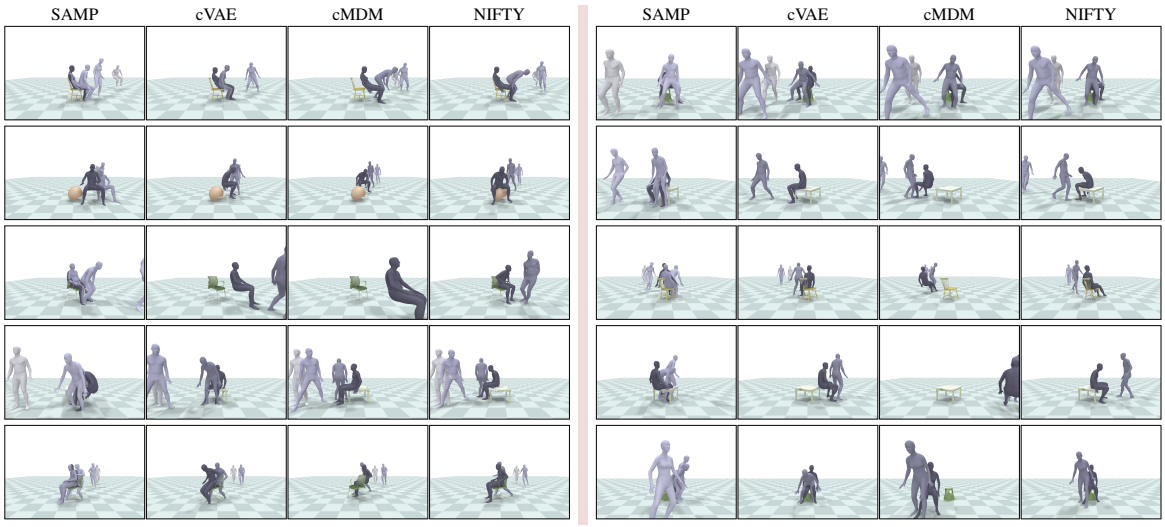

Figure 9. **Comparison Qualitative Motions Sitting**. Compared to other baselines, our method (NIFTY) produces more realistic motions. When examining the motion examples generated by the baselines, we notice that in all cases where a person approaches an object to sit, either the person completely misses the object or the sitting pose is not compatible with the object. To better evaluate these results, please refer to the qualitative videos of these motions in the `https://nileshkulkarni.github.io/nifty/supplementary.html`.

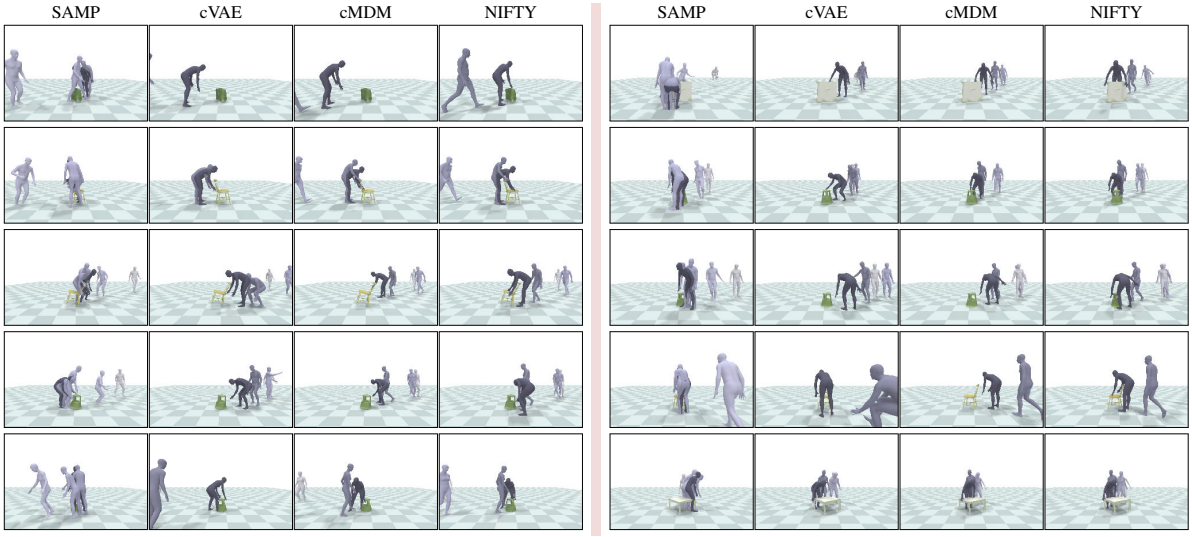

Figure 10. **Comparison Qualitative Motions Lifting**. NIFTY generates more realistic motions as compared to the baseline methods. With motions generated using the baseline methods, we see that the lifting stance is often taken far from the object. To better evaluate these results, please refer to the qualitative videos of these motions in the `supplementary.html` file.

shapes and types, to handle shape variation, the method has to be trained with additional data, requiring the synthesis of interaction poses for the new bodies. Optimization leveraging pose similarities should be possible.

Furthermore, during the inference stage, it is necessary to draw multiple samples from the diffusion model and guide them. This approach yields significantly better performance compared to guiding only a single sample. Exploring research directions that can enhance the stability of the guidance process would be valuable in consistently generating high-quality interaction motions.

References

[1] Hive.ai. `https://thehive.ai/`. Accessed: 2023-11-15. 1, 4

[2] 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. 9

[3] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 9

[4] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015. 9

[5] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 9

[6] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 9

[7] Abhinanda R Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J Black. Babel: bodies, action and behavior with english labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 722–731, 2021. 9

[8] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J. Guibas. Humor: 3d human motion model for robust pose estimation. In *International Conference on Computer Vision (ICCV)*, 2021. 4, 9

[9] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6):209–1, 2019. 9

[10] Omid Taheri, Nima Ghorbani, Michael J Black, and Dimitrios Tzionas. 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*, pages 581–600. Springer, 2020. 4

[11] 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. 4

[12] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Amit H Bermano, and Daniel Cohen-Or. Human motion diffusion model. *ICLR*, 2023. 9

[13] Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. Humanise: Language-conditioned human motion generation in 3d scenes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 4, 9

[14] Kaifeng Zhao, Yan Zhang, Shaofei Wang, Thabo Beeler, , and Siyu Tang. Synthesizing diverse human motions in 3d indoor scenes. In *International conference on computer vision (ICCV)*, 2023. 9