NIFTY: Neural Object Interaction Fields for Guided Human Motion Synthesis

Nilesh Kulkarni[1,2] Davis Rempe*[3,4] Kyle Genova[2] Abhijit Kundu[2]
Justin Johnson[2] David Fouhey[2] Leonidas Guibas[2,4]

[1]University of Michigan [2]Google [3]NIVIDA [4]Stanford University

Abstract

We address the problem of generating realistic 3D motions of humans interacting with objects in a scene. Our key idea is to create a neural interaction field attached to a specific object, which outputs the distance to the valid interaction manifold given a human pose as input. This interaction field guides the sampling of an object-conditioned human motion diffusion model, so as to encourage plausible contacts and affordance semantics. To support interactions with scarcely available data, we propose an automated synthetic data pipeline. For this, we seed a pre-trained motion model, which has priors for the basics of human movement, with interaction-specific anchor poses extracted from limited motion capture data. Using our guided diffusion model trained on generated synthetic data, we synthesize realistic motions for sitting and lifting with several objects, outperforming alternative approaches in terms of motion quality and successful action completion. We call our framework **NIFTY**: **N**eural **I**nteraction **F**ields for **T**rajectory s**Y**nthesis. Project Page: https://nileshkulkarni.github.io/nifty

1 Introduction

Animating a character to sit in a chair or pick up a box is useful in gaming, character animation, and populating digital twins. Yet, generating realistic 3D human motion trajectories with objects is challenging for two main reasons. One challenge is creating *effective models* that capture the nuance of human movements, particularly during the final phase of object interaction called the "last mile." Unlike navigation that is primarily collision avoidance, the last mile involves intricate contacts and object affordances, which influence the motion. The second challenge is *acquiring paired data* that includes high-quality human motions and diverse object shapes, which is essential for training.

Recent approaches to motion modeling can synthesize realistic human movements using state-of-the-art generative models [13, 33, 42]. They are, however, scene-agnostic and cannot produce interactions with specific objects. To address this, some approaches condition motion synthesis on scene geometry (*e.g.*, a scanned point cloud) [19, 47, 48, 50]. This enables learning object interactions, but motion quality is hindered by the lack of paired full scene-motion data. Other approaches [11, 38, 56] instead focus on a small set of interactions with a single type of object (*e.g.*, sitting in a chair), and can produce high-quality motions in their domain. However, these methods require a high-quality motion capture (mocap) dataset for each action/object separately and may make action-specific modeling assumptions (*e.g.*, affecting the human approach and/or contact with object).

In this work, we tackle both the *modeling* and *data* aspects of interaction synthesis to enable generating realistic interactions with a variety of objects, such as sitting on a chair, table, or stool and lifting a suitcase, chair, *etc.*. We extend a human motion diffusion model [42] to condition on object geometry,

*Work done while at Stanford University

Preprint. Under review.

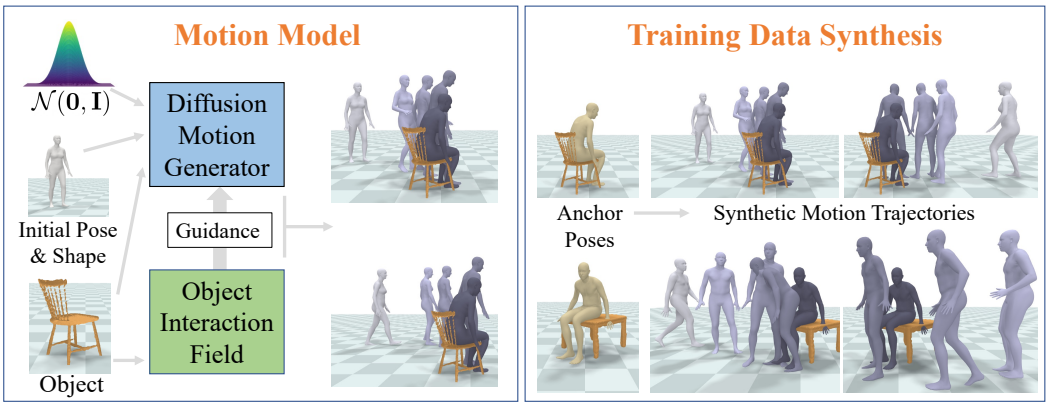

Figure 1: **NIFTY Overview.** (Left) Our learned **object interaction field** guides an object-conditioned **diffusion model** during sampling to generate plausible human-object interactions like sitting. (Right) Our automated *training data synthesis pipeline* generates data for this model by combining a scene-unaware motion model with small quantities of annotated interaction anchor pose data.

and pair it with a *learned object interaction field* to encourage realistic movements at test time in the last mile of interaction. To train this model and overcome the lack of available mocap data, we develop an *automated data pipeline* that leverages a powerful pre-trained and scene-agnostic human motion model. As shown in Fig. 1, our interaction field, diffusion model, and motion data pipeline make up a general framework to synthesize human-object interactions for a desired character that is flexible to multiple actions, even when dense mocap data is *unavailable*. We refer to this framework as **NIFTY**: **N**eural **I**nteraction **F**ields for **T**rajectory s**Y**nthesis.

To ensure realistic motions in the last mile of interaction, we propose an object-centric interaction field that takes in a human pose and learns to regress the distance to a valid interaction pose (*e.g.*, the final sitting pose). At test time, our object-conditioned diffusion model is *guided* by this interaction field to encourage high-quality motions. Unlike manually designed guidance objectives that encourage contact and discourage penetration [19], our interaction field is data-driven and implicitly captures notions of contact, object penetration, and any other factors learned from data.

We propose using synthetic data generation to enable learning interactions from limited mocap data. In particular, we leverage a pre-trained motion model [33] that produces high-quality motions but is unaware of object geometry. Starting from an anchor pose that captures the core of a desired interaction (*e.g.*, the final sitting pose in Fig. 1, right), the pre-trained model is used to sample a large variety of motions that *end* in the anchor pose. This approach generates a diverse set of plausible interactions from only a handful of anchor poses, which are readily available from existing small datasets [3] or are relatively easy to capture.

We evaluate NIFTY on sitting and lifting interactions for a variety of objects, demonstrating the superior quality of synthesized human motions compared to alternative approaches. Overall, this work contributes **(1)** a novel object interaction field approach that guides an object-conditioned human motion diffusion model to synthesize realistic interactions, **(2)** an automated synthetic data generation technique to produce large numbers of plausible interactions from limited pose data, and **(3)** high-quality motion synthesis results for human interactions with several objects.

2 Related Work

Synthesizing Human Motion and Interactions. While various methods have been successful in generating human motion in isolation [13, 17, 30, 33, 42, 57], our work is primarily focused on incorporating environmental context [4, 5]. Some approaches condition motion generation on scanned scene geometry that encompasses multiple objects [19, 47, 48, 49], but these methods typically offer limited control over the specific objects for interaction. Object-centric models are trained to generate motions for a single character [11, 38] and limited actions [56], such as sitting on a chair. These models heavily rely on high-quality motion capture datasets but still exhibit issues like floating, skating, and penetration. Our work also focuses on individual objects but utilizes diffusion guidance and a learned interaction field to minimize undesired artifacts. In contrast to prior work, we train our

models using a novel data generation pipeline to learn interactions from limited data. Our focus is on macroscopic interactions like sitting and lifting with objects, distinguishing us from other works that generate full-body motions for grasping and manipulation [8, 39, 40].

Motion Modeling with Diffusion. Following success for image [14, 28] and video [16] generation, diffusion models [37] have shown promise in modeling motion for robots [20] and pedestrians [34]. Recently, diffusion models have been successful in generating full-body 3D human motion [6, 42, 44, 55]. SceneDiffuser [19] generates human motion conditioned on a point cloud from a scanned scene. It employs gradient-based guidance and analytic objectives to ensure collision-free, contact-driven, and smooth motion during the denoising process. On the contrary, our approach is object-centric and does not rely on noisy motions [12] for training. Our *data-driven* interaction field guides denoising by implicitly capturing plausible interactions and obviating the need for hand-designed objectives.

Neural Distance Fields for Pose and Interaction. Neural networks have been used to learn a parametric function that outputs a distance given a query coordinate [52]. Grasping Fields [22] parameterize hand-object grasping through a spatial field that outputs distances to valid hand-object grasps. Pose-NDF [43] learns an object-unaware distance field in the full-body pose space for human poses. NGDF [51] and SE(3)-DiffusionFields [45] learn a field in the robot gripper pose space to define a manifold of valid object grasps. Our object interaction field extends this idea to full-body human-object interactions by learning to predict the distance between a human pose and the interaction pose manifold. Unlike prior works, we use this field to guide denoising.

Human Interaction Data. Though large-scale mocap data is available to train scene-agnostic human motion models [25], learning human-object interactions is hampered by the challenge of capturing humans in scenes. Datasets that contain full scene scans paired with human motion [10, 12, 18, 35, 58] are relatively small and often noisy due to capture difficulties. Other datasets contain single-object interactions with a small set of objects [3, 11, 21, 40, 56]. These are better quality due to simpler capture conditions, but are small with limited scope. Recent approaches circumvent the data issue through automated synthetic data generation. For example, 3D scenes can be inferred from pre-recorded human motions to get plausible paired scene-motion data [50, 53, 54]. However, motions from these methods are limited to available pre-recorded data. Our data generation pipeline requires only a small set of interaction anchor poses and generates novel motions not contained in prior datasets using tree-based rollouts [57] from a pre-trained generative model [33].

3 Method

In this section, we detail our NIFTY pipeline for learning to synthesize realistic human-object interaction motions. §3.1 introduces a conditional diffusion model to generate human motions given the geometry of an object. §3.2 details the object-centric interaction field, which guides the denoising process of the diffusion model to capture the nuances of interactions in a data-driven way. In §3.3, we discuss the synthetic data generation using a pre-trained motion model that is seeded with anchor poses from a smaller dataset. This data is used to train the diffusion model and interaction field.

3.1 Motion Generation using Diffusion Modeling

Motion Representation. Motion generation is formulated as predicting a sequence of 3D human pose states that capture a person's motion over time. The pose state representation is based on the SMPL body model [23] and is similar to prior successful human motion diffusion models [9, 42]. The human pose state X_i at frame i in a motion sequence is:

$$X_i = \{\boldsymbol{j}_i^p, \boldsymbol{j}_i^r, \boldsymbol{j}_i^v, \boldsymbol{j}_i^\omega, \boldsymbol{t}_i^p, \boldsymbol{t}_i^v\}, \tag{1}$$

which includes joint positions $\boldsymbol{j}_i^p \in \mathbb{R}^{3\times22}$, rotations $\boldsymbol{j}_i^r \in \mathbb{R}^{6\times22}$, velocities $\boldsymbol{j}_i^v \in \mathbb{R}^{3\times22}$, and angular velocities $\boldsymbol{j}_i^\omega \in \mathbb{R}^{3\times22}$ for all 22 SMPL joints including the root (pelvis). Additionally, the SMPL global translation $\boldsymbol{t}_i^p \in \mathbb{R}^3$ and velocity $\boldsymbol{t}_i^v \in \mathbb{R}^3$ are included. A motion (trajectory) is a sequence of N poses denoted as $\boldsymbol{\tau} = \{X_1, \ldots, X_N\}$ where all poses are in a canonical coordinate frame, namely, the local frame of the pose X_1 at the first timestep where the human is at the origin and its front-facing vector is aligned with the $+Y$ axis.

Model Formulation. The diffusion model simultaneously generates all human poses in a motion sequence [42] to achieve a desired interaction. Intuitively, diffusion is a noising process that converts clean data into noise. We want our motion model to learn the reverse of this process so that realistic motions can be generated from randomly sampled noise. Mathematically, forward diffusion is a

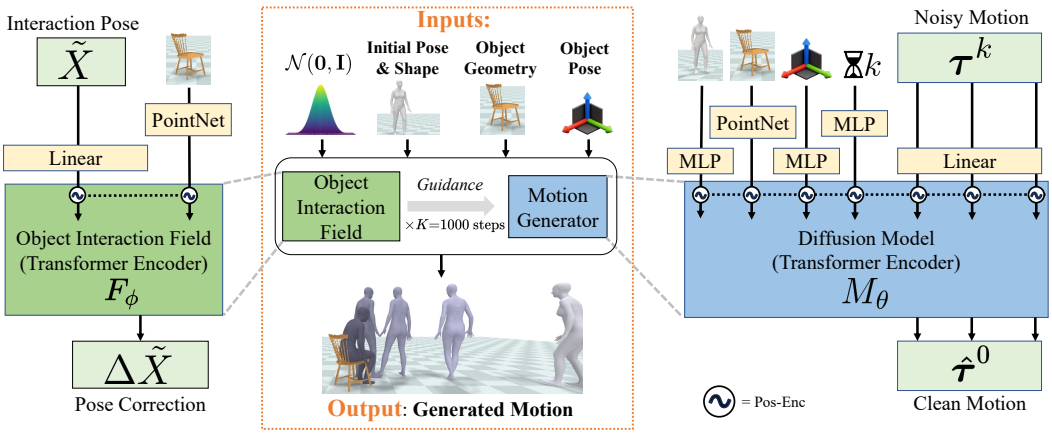

Figure 2: **Model Architecture.** Our full motion synthesis method (middle) consists of an object interaction field F_ϕ (left), which guides the diffusion model M_θ (right) at sampling time to produce plausible interaction motions. At each step $k \in [0, K = 1000]$ of denoising, the diffusion model predicts a clean motion $\hat{\boldsymbol{\tau}}^0$ from a noisy motion input $\boldsymbol{\tau}^k$ and conditioning information. The object interaction field takes the last pose from the diffusion output as input, and uses guidance to push the pose towards the valid interaction manifold using a predicted pose correction.

Markov process with a transition probability distribution:

$$q(\boldsymbol{\tau}^k|\boldsymbol{\tau}^{k-1}) := \mathcal{N}(\boldsymbol{\tau}^k; \mu = \sqrt{1-\beta^k}\boldsymbol{\tau}^{k-1}, \sigma = \beta^k\mathbf{I}), \tag{2}$$

where $\boldsymbol{\tau}^k$ denotes the motion trajectory at the k^{th} noising step, and a fixed β^k is chosen such that $q(\boldsymbol{\tau}^K) \approx \mathcal{N}(\boldsymbol{\tau}^K; \mathbf{0}, \mathbf{I})$ after K steps. Our generative model learns the reverse of this process (denoising), *i.e.*, it recovers $\boldsymbol{\tau}^{k-1}$ from a noisy input trajectory $\boldsymbol{\tau}^k$ at each step and doing this repeatedly results in a final clean motion $\boldsymbol{\tau}^0$. Because the model is generating *interaction* motions with an object, we condition denoising on interaction information $C = \{P_o, R_o, \mathbf{b}, X_0\}$, which includes the canonicalized object point cloud $P_o \in \mathbb{R}^{5000\times3}$, rigid object pose relative to the person $R_o \in \mathbb{R}^{4\times4}$, SMPL body shape parameters $\mathbf{b} \in \mathbb{R}^{10}$, and starting pose of the person X_0. Each reverse step is then:

$$p_\theta(\boldsymbol{\tau}^{k-1}|\boldsymbol{\tau}^k, C) := \mathcal{N}(\boldsymbol{\tau}^{k-1}; \mu = \mu_\theta(\boldsymbol{\tau}^k, k, C), \sigma = \beta^k\mathbf{I}), \tag{3}$$

where the diffusion step k is also given as input. Instead of predicting the noise $\boldsymbol{\epsilon}^k$ added at each step of the diffusion process [14, 20], our model directly predicts the final clean signal [34, 42]. Mathematically, the motion model M_θ with parameters θ predicts a clean trajectory $\hat{\boldsymbol{\tau}}^0 = M_\theta(\boldsymbol{\tau}^k, k, C)$ from which the mean $\mu_\theta(\tau^k, k, C)$ is easily computed [28]. This formulation has the benefit that physically grounded objectives can be easily computed on $\hat{\boldsymbol{\tau}}^0$ in the pose space, which is useful for guidance as discussed below.

While training the diffusion model, a ground truth clean trajectory $\boldsymbol{\tau}^0$ is noised and given as input, then the model is trained to minimize the objective $\|\hat{\boldsymbol{\tau}}^0 - \boldsymbol{\tau}^0\|_2^2$. To enable using classifier-free guidance [15] at sampling time, the conditioning C is randomly masked out with 10% probability during the training process so that the model can operate in both conditional and unconditional modes.

Sampling and Guidance. At test time, samples are generated from the model given random noise and interaction conditioning C as input. We find that leveraging classifier-free guidance [15] tends to generate higher-quality samples. This amounts to generating one conditional and one unconditional sample from the model and then combining them as $\hat{\boldsymbol{\tau}}^0 = M_\theta(\boldsymbol{\tau}^k, k) + s(M_\theta(\boldsymbol{\tau}^k, k, C) - M_\theta(\boldsymbol{\tau}^k, k))$, where the strength of the conditioning is controlled by the scalar s.

Ensuring that the sampled motions adhere to the geometric and semantic constraints of the object is key to plausible interactions. Diffusion models are well-suited for this, since *guidance* can encourage samples to meet desired objectives at test time [20]. The core of guidance is a differentiable function $G(\boldsymbol{\tau}^0)$ that evaluates how well a trajectory meets a desired objective; this could be a learned [20] or an analytic [34] function. In our case, we want $G(\boldsymbol{\tau}^0)$ to evaluate how plausible an interaction motion is w.r.t. the object, and in §3.2 we show that this can be done with a learned object interaction field. Throughout denoising during sampling, the gradient of the objective function will be used to

nudge trajectory samples in the correct direction. We use a formulation of guidance that perturbs the clean trajectory output from the model $\hat{\boldsymbol{\tau}}^0$ at every denoising step k as follows [16, 34]:

$$\tilde{\boldsymbol{\tau}}^0 = \hat{\boldsymbol{\tau}}^0 - \alpha \nabla_{\boldsymbol{\tau}^k} G(\hat{\boldsymbol{\tau}}^0) \tag{4}$$

where α controls the guidance strength. The updated trajectory $\tilde{\boldsymbol{\tau}}^0$ is then used to compute μ.

Architecture. As shown in Fig. 2 (right), the motion model M_θ is based on a transformer encoder-only architecture [42, 46]. The model takes as input the current noisy trajectory $\boldsymbol{\tau}^k$, the denoising step k, and the conditioning C. Each human pose in the trajectory is a token, while each conditioning becomes a separate token. Of note, the object point cloud P_o is encoded with a PointNet [32], the rigid pose R_o is encoded with a three-layer MLP, and k is encoded using a positional embedding [41]. Our noise levels k vary between 0 to 1000 diffusion steps. The transformer handles variable-length sequence inputs and outputs the clean motion prediction $\hat{\boldsymbol{\tau}}^0$. Full details are available in the supplementary material.

3.2 Object Interaction Fields

After training on human-object interactions, the diffusion model can generate reasonable motion sequences but fails to fully comply with constraints in the last mile of interaction [2, 7], even when conditioned on the object. This causes undesirable artifacts such as penetration with the object. To alleviate this issue, we propose to guide motion samples from the diffusion model (*i.e.*, use Eq. (4)) with a learned objective G that captures realistic interactions for a specific object.

We take inspiration from recent work that uses neural distance fields to learn valid human pose manifolds [43] and robotic grasping manifolds [51]. For our purposes, the field must take in an arbitrary human pose and output how far the query pose is from being a "valid" object interaction pose. We define an *interaction pose* to be an *anchor* frame in a motion sequence that captures the core of the interaction, *e.g.*, the moment a person settles in a chair during sitting (as in Fig. 1) or contacts an object before lifting.

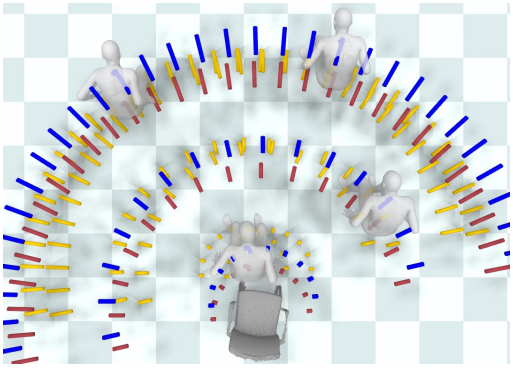

Figure 3: **Interaction Field Visualization.** We query the field in several locations with a sitting pose (a subset shown in grey) and visualize the output for **pelvis**, **feet**, and **neck** joints. All cylinders are oriented towards the chair, indicating the correction vector's magnitude and direction. This correction is due to the misalignment between the sitting pose and chair position.

We propose an *object interaction field* that operates in the local coordinate frame of a specific object. The interaction field F_ϕ takes as input a simplified pose $\tilde{X}=\{\boldsymbol{j}^p, \boldsymbol{t}^p\}$, which includes joint positions and global translation. The field outputs an offset *vector* $\Delta\tilde{X}=F_\phi(\tilde{X})$ that projects the input pose to the manifold of valid interaction poses for the object: $\tilde{X}+\Delta\tilde{X}$ is then a plausible interaction pose. Fig. 3 visualizes the output vectors of an example interaction field for a chair. Querying the field with a sitting pose away from the chair (*i.e.*, not a valid interaction) gives a correction pointing back towards the chair. For further away points, the visualized vectors are longer, indicating larger corrections are needed.

Guidance Objective. The object interaction field serves as a differentiable function that can be incorporated into the guidance objective to judge how far a motion is from the desired interaction manifold. Let $\tilde{X}_i \in \boldsymbol{\tau}$ be the simplified pose from the i^{th} frame of a motion $\boldsymbol{\tau}$. If we know that this pose *should* be a valid interaction pose, then the guidance objective is defined as $G(\boldsymbol{\tau}) = \|F_\phi(\tilde{X}_i)\|_2^2$. During denoising at test time, we feed output poses from the diffusion model into this guidance objective to encourage the generated motion to contain a valid interaction poses.

Training. Supervising F_ϕ requires a dataset of invalid poses with corresponding valid interaction poses. We collect this *after* training the diffusion model detailed in §3.1. In particular, we feed a noisy ground-truth interaction motion $\boldsymbol{\tau}^k$ at a random noise level k to the diffusion model as input. This gives an output motion $\hat{\boldsymbol{\tau}}^0$, which *should* match the ground truth $\boldsymbol{\tau}^0$ if the model is perfect. In practice, denoising back to ground truth is difficult at high noise levels (*e.g.*, k=900), so we consider $\hat{\boldsymbol{\tau}}^0$ as an invalid interaction motion with a corresponding valid motion $\boldsymbol{\tau}^0$. When the diffusion model

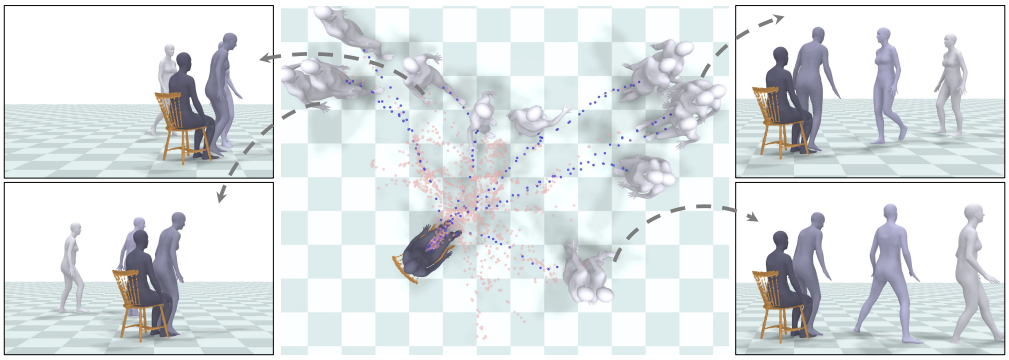

Figure 4: **Generated Synthetic Data**. We visualize motion sequences from one tree rollout for one sitting **anchor pose**. The middle shows a bird's-eye view of the *pelvis* joint trajectories in **light pink**. All trajectories end in the same sitting pose, but start at diverse locations around the chair. We highlight a few trajectories in **blue** and show full-body motions from the corresponding generations on the left and right sides. Our complete dataset contains many trees for different objects and humans.

has been trained on the dataset described in §3.3, we know that the last frame of the motion $\tilde{X}_N \in \hat{\boldsymbol{\tau}}^0$ should be the *interaction pose*, so we can throw away all other poses to arrive at a training dataset for the interaction field. We further augment this dataset by applying random rigid transformations to the invalid interaction poses.

Given a ground truth interaction pose $\tilde{Y}_N \in \boldsymbol{\tau}^0$ and corresponding output pose from the diffusion model $\tilde{X}_N \in \hat{\boldsymbol{\tau}}^0$, the interaction field training loss is computed as $\|F_\phi(\tilde{X}_N) - (\tilde{Y}_N - \tilde{X}_N)\|_1$. Note that training on outputs from the diffusion model is important since the interaction field operates on these kinds of outputs during test-time guidance.

Architecture. As shown in Fig. 2 (left), the interaction field architecture is an encoder-only transformer that operates on the input pose as a token. In practice, it also takes in the canonical object point cloud as a conditioning token to allow training a single field for multiple objects.

3.3 Automatic Synthetic Data Generation

Training the diffusion model requires a large, realistic, and diverse dataset of motions for each human-object interaction we wish to synthesize. Unfortunately, this data exists only for specific interactions [56] and is difficult and expensive to collect from scratch. Therefore, we propose an automated pipeline to generate synthetic interaction motion data. In short, we first select *anchor* pose frames from an existing small dataset [3] that are indicative of an interaction we want to learn. Our key insight is to use a pre-trained scene-unaware motion model [33] to sample a diverse set of motions that *end* at a selected anchor pose, and therefore demonstrate the desired interaction. We provide the key details in this section and a full description appears in the supplementary material.

Anchor Pose Selection. We require a small set of anchor poses that capture the core frame of an interaction motion. As described in §3.2, for sitting on a chair this is the sitting pose when the person first becomes settled in the chair (see Fig. 1). In generating motion data, these anchor poses will be the *final* frame of each synthesized motion sequence. For the experiments in §4, these anchor frames are chosen manually from a small dataset that contains a variety of interactions [3].

Generating Motions in Reverse. The goal is to generate human motions that *end* in the chosen anchor poses and reflect realistic object interactions. We leverage HuMoR [33], which is a conditional motion VAE trained on the AMASS [25] mocap dataset. It generates realistic human motions through autoregressive rollout, but is *scene-unaware*. To force rollouts from HuMoR to match the final anchor pose, we could use online sampling or latent motion optimization, but these are expensive and not guaranteed to exactly converge. Instead, we re-train HuMoR as a time-reversed motion model that predicts the past instead of the future motion given a current input pose. Starting from a desired interaction anchor pose X_N, our reverse HuMoR will generate $X_{N-1}, X_{N-2}, \cdots, X_1$ forming a full interaction motion that, by construction, ends in the desired pose.

Tree-Based Rollout & Filtering. To ensure sufficient diversity and realism in motions from HuMoR, we devise a branching rollout strategy that is amenable to frequent filtering and results in a tree of plausible interactions. Starting from the anchor pose, we first sample 30 frames (1 sec) of motion. Then, multiple branches are instantiated and random rollout continues for another 30 frames on these

branches independently. Continuing in this branching fashion allows growing the motion dataset exponentially while also filtering to ensure branches are sufficiently diverse and do not contain undesireable motions. Filtering involves heuristically pruning branches with motions that collide with the object, float above the ground plane, result in unnatural pose configurations, and become stationary. For the experiments in §4, we rollout to a tree depth of 7 and sample many motion trees starting from each anchor pose. Individual paths are extracted from the tree to give interaction motions, and we post-filter out sequences that start within 1 meter of the object.

Generated Datasets. We use this scalable strategy to generate data for training our motion model for *sitting* and *lifting* interactions. Fig. 4 demonstrates the diversity of our generated datasets by visualizing top-down trajectories and example motions from a single tree of sitting motions. For the sitting interaction dataset, we choose 174 anchor pose frames across 7 subjects in the BEHAVE [3] dataset. This results in a dataset of 200K motion sequences that include sitting on chairs, stools, tables, and exercise balls. Each motion sequence in this dataset ends at a sitting anchor pose. For lifting interactions, 72 anchor poses from 7 subjects produces 110K motion sequences. Each sequence ends at a lifting anchor pose when the person initially contacts the object.

4 Experiments

We evaluate our NIFTY method after training on the *sitting* and *lifting* datasets introduced in §3.3. Implementation details are given in §4.1, followed by a discussion of evaluation metrics in §4.2 and baselines in §4.3. Experimental results are presented in §4.4 along with an ablation study in §4.5.

4.1 Implementation Details

We train our diffusion model M_θ for 600K iterations with a batch size of 32 using the AdamW [24] optimizer with a learning rate of 10^{-4}. A separate model is trained for sitting and lifting. We use $K{=}1000$ diffusion steps in our model and sample the diffusion step k from a uniform distribution at each training iteration. The object interaction field F_ϕ is trained on the data described in §3.2 for 300K iterations using AdamW with a maximum learning rate of 5×10^{-5} and a one cycle LR schedule [36]. When sampling from the diffusion model, 10 samples are generated in parallel and all are guided using the object interaction field; the sample with the best guidance objective score is used as the output. We apply interaction field guidance on the last frame of motion (*i.e.*, the interaction anchor pose in our datasets). Our models are trained using PyTorch [29] on NVIDIA A40 GPUs, and take about 2 days to train. Visualizations use the PyRender engine [26].

4.2 Evaluation Setting and Metrics

To ensure we properly evaluate the generalization capability of methods trained on our synthetic interaction datasets, we *do not* create a test set using the procedure described in Sec. 3.3, which may result in a very similar distribution to training data. Instead, we create a set of 500 *test scenes* for each action, where objects are randomly placed in the scene and the human starts from a random pose generated by HuMoR. All methods are tested on these same scenes during evaluation.

Evaluating human motion coupled with object interactions is challenging and has no standardized protocol. Hence, we evaluate using a diverse set of metrics including a user perceptual study. We briefly describe the metrics next and include full details in the supplementary material.

User Study. No single metric can capture all the nuances of human-object interactions, so we employ a perceptual study [27, 39, 40, 42, 50]. For each method, we create videos from generated motions on the test scenes. To compare two methods, users are presented with two videos on the same test scene and must choose which they prefer (full user directions are in the supplement). We perform independent user studies for lifting and sitting actions using `hive.ai` [1]. Responses are collected from 5 users for every comparison video, giving 2500 total responses in each comparison study.

Foot Skating. Similar to prior work [27], we define the foot-skating score for a sequence of N timesteps as $\frac{1}{N}\sum_i^N v_i(2 - 2^{h_i/H}) \cdot \mathbb{1}_{h<=H}$, where v_i is the velocity and h_i is the height above ground of the right toe vertex for the i^{th} frame. H is 2.5 cm. Intuitively, this is the mean foot velocity when it is near the ground (where it *should* be 0), with higher weight applied closer to the ground.

Distance to Object (*D2O*). Similar to prior work [50], this evaluates whether the human gets close to the object during the interaction. It measures the minimum distance from the human body in the

Table 1: **Quantitative Comparison.** Our method outperforms baselines on both sitting and lifting. Our diffusion model, guided by the learned interaction field, generates motions that reach the object (*D2O*) with few *penetrations* and realistic *contacts*. Motions approaching the object are realistic with low *foot skating* and the final interaction pose is similar to synthetic data with low *skeleton distance*.

Method	Sitting						Lifting					
	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}}\tilde{\%}$ ↓	Skel. Dist. ↓	Contact IoU ↑	% Pen. $\leq 2cm$ ↑	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}}\tilde{\%}$ ↓	Skel. Dist. ↓	Contact IoU ↑	% Pen. $\leq 2cm$ ↑
Cond. VAE [50]	0.77	88.8	0.13	1.07	0.21	46.6	0.66	57.4	0.66	1.70	0.03	60.1
Cond. MDM [42]	**0.36**	37.9	1.06	2.81	0.04	50.5	**0.28**	36.3	1.14	2.58	0.02	46.2
NIFTY (ours)	0.47	**99.6**	**0.00**	**0.54**	**0.54**	**65.0**	0.34	**77.7**	**0.05**	**0.42**	**0.17**	**68.5**

last frame of the motion sequence to any point on the object's surface. We report the % of sequences within 2 cm distance to avoid sensitivity to outliers, along with the 95^{th} percentile ($\tilde{\%}$) of this distance.

Penetration Score (*% Pen*). To evaluate realism as the human approaches an object for interaction, we measure how much they penetrate the object. Based on our synthetic data, we define the first N_A frames of motion to be the approach for each action type (see supplement). Then the penetration distance for a trajectory is $\frac{1}{N_A}\sum_v \sum_i^{N_A} \texttt{sdf}_i(v) \cdot \mathbb{1}_{\texttt{sdf}_i(v)>0}$, where $\texttt{sdf}_i$ is the signed distance function of the human in the i^{th} frame and v is one of 2K points on the object's surface. We report the percentage of trajectories with penetration distance ≤ 2 cm (% Pen. $\leq 2cm$) ignoring trajectories with *D2O* > 2 cm, since trajectories that do not approach the object will trivially avoid penetration.

Skeleton Distance & Contact IoU. These evaluate how well generated interaction poses align with ground truth poses and their human-object contacts. We start by finding the minimum distance between the final pose of a generated sequence and the anchor poses in the synthetic training data. The distance to this nearest neighbor pose is reported as the skeleton distance. To measure how well contacts from the generated motion match the data, we compute the IoU between contacting vertices (those that penetrate the object) on the predicted body mesh and those on the nearest neighbor mesh.

4.3 Baselines

Cond. VAE [50]. Closest to our problem definition, this model comes from recent work HUMANISE [50], which learns plausible human motions conditioned on scene and language for four actions (lie, sit, stand, walk). This state-of-the-art model is a conditional VAE with a GRU motion encoder and sequence-level transformer decoder. Since we evaluate on sitting and lifting actions separately, we modify their approach to remove language conditioning. The model is trained on our synthetic data for 600K iterations with the recommended hyperparameters and learning rate of 10^{-4}.

Cond. MDM [42]. This baseline is the motion diffusion model (MDM) [42] with added conditioning C, *i.e.*, our object-conditioned diffusion model without interaction field guidance.

4.4 Experimental Results

User Study. Fig. 5 shows how often users prefer our method (NIFTY) over baselines and Synthetic Data (Syn. Data) for both sitting and lifting. We perform separate studies for each comparison. Users prefer NIFTY over baselines a vast majority of the time. Averaged over both actions, NIFTY is preferred over the Cond. VAE [50] baseline 89.4% of the time. Similarly, NIFTY is preferred over Cond. MDM [42] 86.3% of the time, highlighting the importance of using guidance with our interaction field during sampling. Compared to held out motions from synthetic data, NIFTY is preferred 47.2% of the time, which indicates that the motions are nearly indistinguishable from those of our data generation pipeline.

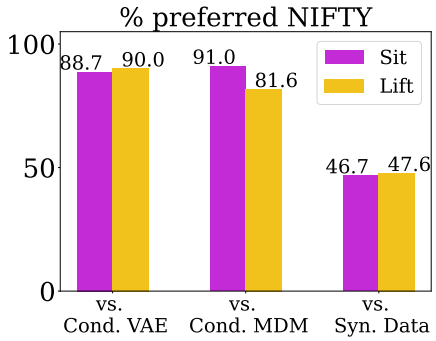

Figure 5: **User Study.** NIFTY is preferred $\geq$ 88.7% of the time for sitting and $\geq$81.6% for lifting compared to baselines. Our motions are also nearly indistinguishable from synthetic data trajectories.

We further extract robust consensus across users by majority vote over the 5 responses for each video. In this case, motions generated by our method are preferred 94.4% (sitting) and 97.8% (lifting) of the time over Cond. VAE [50] motions, making the improvement gap even more apparent.

Table 2: **Ablation Study.** Comparison between using an interaction field trained to predict a full offset vector (NIFTY) or a single scalar distance (Distance OIF).

Method	Sitting						Lifting					
	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}}\%$ ↓	Skel. Dist. ↓	Contact IoU ↑	% Pen. $\leq 2cm$ ↑	Foot Skating ↓	% *D2O* $\leq 2cm$ ↑	*D2O* $95^{\text{th}}\%$ ↓	Skel. Dist. ↓	Contact IoU ↑	% Pen. $\leq 2cm$ ↑
Distance OIF	**0.41**	80.9	0.47	1.25	0.24	**66.8**	**0.30**	57.4	0.74	1.31	0.07	**70.1**
NIFTY	0.47	**99.6**	**0.00**	**0.54**	**0.54**	65.0	0.34	**77.7**	**0.05**	**0.42**	**0.17**	68.5

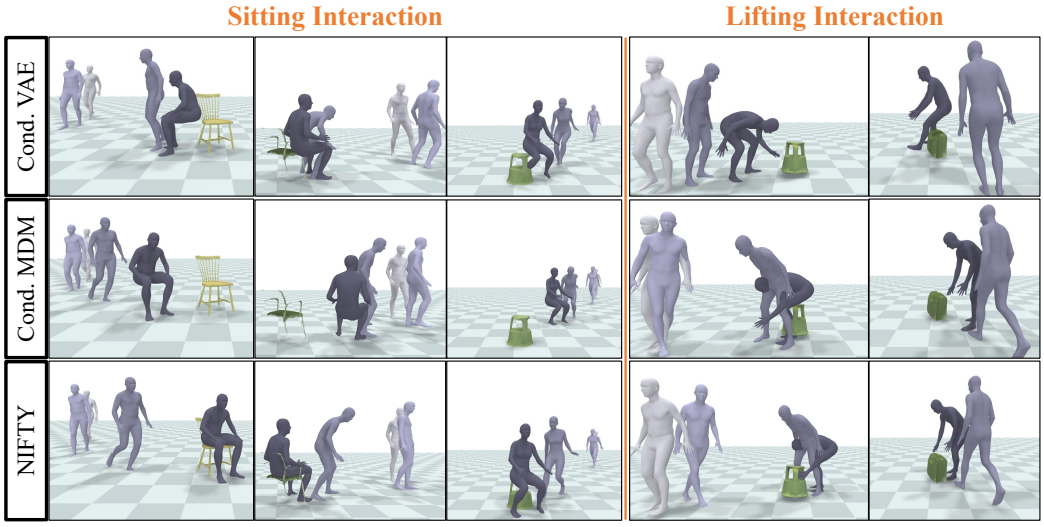

Figure 6: **Qualitative Results.** Our method (bottom row) generates realistic interaction motions that reach the desired object with plausible contacts (*e.g.* col 1 & 4) while avoiding penetrations, unlike baselines. The mesh color gets darker as time progresses. Cond. VAE [50] motions have the **final interaction pose** away from the object (col 1,3,4), incorrect (col 2 & 5), or intersecting (col 5). Cond. MDM [42] generates sitting poses far away from the object (col 1 & 3).

Additionally, we also conduct an user study on a Likert scale of scores between 1 (unrealistic) to 5 (very realistic). We report that motions from our synthetic dataset achieve a score of 4.39 *vs.* 4.87 for motions in the AMASS [25]. Further details are available in Supp. § A.1.

Quantiative Results. In Tab. 1, NIFTY is compared to baselines for both sitting and lifting interactions. NIFTY generates motions that reach the target object and approach realistically, as indicated by distance-to-object (*D2O*) and *penetration* metrics. Although Cond. MDM [42] produces realistic motion with low *foot skating*, it struggles to properly approach the object since it does not use guidance from the learned interaction field. We see that interaction poses and the resulting object contacts generated by our method do reflect the synthetic dataset, resulting in low *skeleton distance* and high *contact IoU*, unlike Cond. VAE [50] which is worse across all metrics.

Qualitative Results. Fig. 6 shows a qualitative comparison between motions generated by our method and baselines. NIFTY synthesizes realistic sitting and lifting with a variety of objects. Examples show that the baselines struggle to generalize to unseen object poses, and have no mechanism to correct for this at test time. Our learned interactions field helps to avoid this through diffusion guidance. Please see the videos provided in the supplement to best appreciate the results.

4.5 Ablation Study

As detailed in §3.2, our object interaction field (OIF) is formulated to predict an offset vector $\Delta\tilde{X}$ that captures both distance and direction for each component of the pose state, rather than a single full-body distance like prior work [43]. We ablate this design decision in Tab. 2, which compares our interaction field formulation to a version that predicts only a scalar distance to the interaction pose manifold (Distance OIF). We observe that learning a single distance is a harder task compared to predicting an offset vector, which provides a stronger learning signal for training. As a result, the ablated interaction field results in worse scores across most metrics.

5. Conclusion and Limitations

We introduced NIFTY, a framework for learning to synthesize realistic human motions involving 3D object interactions. Results demonstrate that our object-conditioned diffusion model gives improved motions over prior work when guided by a learned object interaction field and trained on automatically synthesized motion data. Our current approach is limited to the body shapes present in the training data (*e.g.*, the 7 subjects from BEHAVE [3]), so future work should explore data augmentation strategies to generalize to novel humans. Moreover, we have shown results on sitting and lifting, but we would like to widen the scope to handle additional interactions by collecting new anchor poses, synthesizing data, and training our diffusion model and interaction field.

Acknowledgements.. We express our gratitude to our colleagues for the fantastic project discussions and feedback provided at different stages. We have organized them by institution (in alphabetical order)
– *Google*: Matthew Brown, Frank Dellaert, Thomas A. Funkhouser, Varun Jampani
– *Google (co-interns)*: Songyou Peng, Mikaela Uy, Guandao Yang, Xiaoshuai Zhang
– *University of Michigan*: Mohamed El Banani, Ang Cao, Karan Desai, Richard Higgins, Sarah Jabbour, Linyi Jin, Jeongsoo Park, Chris Rockwell, Dandan Shan

This work was partly done when NK was interning at Google Research. DR was supported by the NVIDIA Graduate Fellowship.

References

[1] Hive.ai. https://thehive.ai/. Accessed: 2023-05-15. 7, 14, 17

[2] Randall Balestriero and Yann LeCun. Police: Provably optimal linear constraint enforcement for deep neural networks. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023. 5

[3] Bharat Lal Bhatnagar, Xianghui Xie, Ilya Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Behave: Dataset and method for tracking human object interactions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2022. 2, 3, 6, 7, 10

[4] Yu-Wei Chao, Jimei Yang, Weifeng Chen, and Jia Deng. Learning to sit: Synthesizing human-chair interactions via hierarchical control. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5887–5895, 2021. 2

[5] Enric Corona, Albert Pumarola, Guillem Alenya, and Francesc Moreno-Noguer. Context-aware human motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6992–7001, 2020. 2

[6] Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. Mofusion: A framework for denoising-diffusion-based motion synthesis. In *Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[7] Priya L Donti, David Rolnick, and J Zico Kolter. Dc3: A learning method for optimization with hard constraints. *arXiv preprint arXiv:2104.12225*, 2021. 5

[8] Anindita Ghosh, Rishabh Dabral, Vladislav Golyanik, Christian Theobalt, and Philipp Slusallek. Imos: Intent-driven full-body motion synthesis for human-object interactions. In *Eurographics*, 2023. 3

[9] Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5152–5161, June 2022. 3

[10] Vladimir Guzov, Aymen Mir, Torsten Sattler, and Gerard Pons-Moll. Human poseitioning system (hps): 3d human pose estimation and self-localization in large scenes from body-mounted sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4318–4329, 2021. 3

[11] Mohamed Hassan, Duygu Ceylan, Ruben Villegas, Jun Saito, Jimei Yang, Yi Zhou, and Michael J Black. Stochastic scene-aware motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11374–11384, 2021. 1, 2, 3

[12] Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J Black. Resolving 3d human pose ambiguities with 3d scene constraints. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2282–2292, 2019. 3

[13] Gustav Eje Henter, Simon Alexanderson, and Jonas Beskow. Moglow: Probabilistic and controllable motion synthesis using normalising flows. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020. 1, 2

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 3, 4

[15] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 4

[16] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022. 3, 5

[17] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)*, 36(4):1–13, 2017. 2

[18] Chun-Hao P Huang, Hongwei Yi, Markus Höschle, Matvey Safroshkin, Tsvetelina Alexiadis, Senya Polikovsky, Daniel Scharstein, and Michael J Black. Capturing and inferring dense full-body human-scene contact. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13274–13285, 2022. 3

[19] Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2, 3

[20] Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022. 3, 4

[21] Nan Jiang, Tengyu Liu, Zhexuan Cao, Jieming Cui, Yixin Chen, He Wang, Yixin Zhu, and Siyuan Huang. Chairs: Towards full-body articulated human-object interaction. *arXiv preprint arXiv:2212.10621*, 2022. 3

[22] Korrawe Karunratanakul, Jinlong Yang, Yan Zhang, Michael J Black, Krikamol Muandet, and Siyu Tang. Grasping field: Learning implicit representations for human grasps. In *2020 International Conference on 3D Vision (3DV)*, pages 333–344. IEEE, 2020. 3

[23] 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, October 2015. 3, 17

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7

[25] 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. 3, 6, 9, 23

[26] Matthew Matl. Pyrender. https://github.com/mmatl/pyrender, 2019. 7

[27] Aymen Mir, Xavier Puig, Angjoo Kanazawa, and Gerard Pons-Moll. Generating continual human motion in diverse 3d scenes. *arXiv preprint arXiv:2304.02061*, 2023. 7

[28] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 3, 4

[29] 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. 7, 17

[30] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021. 2

[31] 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. 17

[32] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 5

[33] 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. 1, 2, 3, 6, 15, 17, 23

[34] Davis Rempe, Zhengyi Luo, Xue Bin Peng, Ye Yuan, Kris Kitani, Karsten Kreis, Sanja Fidler, and Or Litany. Trace and pace: Controllable pedestrian animation via guided trajectory diffusion. *CVPR*, 2023. 3, 4, 5

[35] Manolis Savva, Angel X Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. Pigraphs: learning interaction snapshots from observations. *ACM Transactions on Graphics (TOG)*, 35(4):1–12, 2016. 3

[36] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019. 7

[37] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 3

[38] Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Trans. Graph.*, 38(6):209–1, 2019. 1, 2

[39] 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. 3, 7, 14

[40] 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. 3, 7, 14

[41] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020. 5

[42] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Amit H Bermano, and Daniel Cohen-Or. Human motion diffusion model. *ICLR*, 2023. 1, 2, 3, 4, 5, 7, 8, 9

[43] Garvita Tiwari, Dimitrije Antić, Jan Eric Lenssen, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Pose-ndf: Modeling human pose manifolds with neural distance fields. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*, pages 572–589. Springer, 2022. 3, 5, 9

[44] Jonathan Tseng, Rodrigo Castellon, and C Karen Liu. Edge: Editable dance generation from music. *arXiv preprint arXiv:2211.10658*, 2022. 3

[45] Julen Urain, Niklas Funk, Georgia Chalvatzaki, and Jan Peters. Se (3)-diffusionfields: Learning cost functions for joint grasp and motion optimization through diffusion. *arXiv preprint arXiv:2209.03855*, 2022. 3

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 5

[47] Jiashun Wang, Huazhe Xu, Jingwei Xu, Sifei Liu, and Xiaolong Wang. Synthesizing long-term 3d human motion and interaction in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9401–9411, 2021. 1, 2

[48] Jingbo Wang, Yu Rong, Jingyuan Liu, Sijie Yan, Dahua Lin, and Bo Dai. Towards diverse and natural scene-aware 3d human motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20460–20469, 2022. 1, 2

[49] Jingbo Wang, Sijie Yan, Bo Dai, and Dahua Lin. Scene-aware generative network for human motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12206–12215, 2021. 2

[50] 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. 1, 3, 7, 8, 9, 14

[51] Thomas Weng, David Held, Franziska Meier, and Mustafa Mukadam. Neural grasp distance fields for robot manipulation. *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 3, 5

[52] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, volume 41, pages 641–676. Wiley Online Library, 2022. 3

[53] Sifan Ye, Yixing Wang, Jiaman Li, Dennis Park, C Karen Liu, Huazhe Xu, and Jiajun Wu. Scene synthesis from human motion. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 3

[54] Hongwei Yi, Chun-Hao P. Huang, Shashank Tripathi, Lea Hering, Justus Thies, and Michael J. Black. MIME: Human-aware 3D scene generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023. 3

[55] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 3

[56] Xiaohan Zhang, Bharat Lal Bhatnagar, Sebastian Starke, Vladimir Guzov, and Gerard Pons-Moll. Couch: Towards controllable human-chair interactions. October 2022. 1, 2, 3, 6

[57] Yan Zhang and Siyu Tang. The wanderings of odysseus in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20481–20491, 2022. 2, 3

[58] Yang Zheng, Yanchao Yang, Kaichun Mo, Jiaman Li, Tao Yu, Yebin Liu, C Karen Liu, and Leonidas J Guibas. Gimo: Gaze-informed human motion prediction in context. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIII*, pages 676–694. Springer, 2022. 3

A 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 § A.1 we perform a large scale user-study with 10K user responses. In § A.2 we describe the complete details of data generation including pseudo-code for the algorithm.

A.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 [39, 40]. 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 [50] (*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 Cond. VAE [50] 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 [50] 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 Cond. VAE [50] 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 Cond. VAE [50] 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

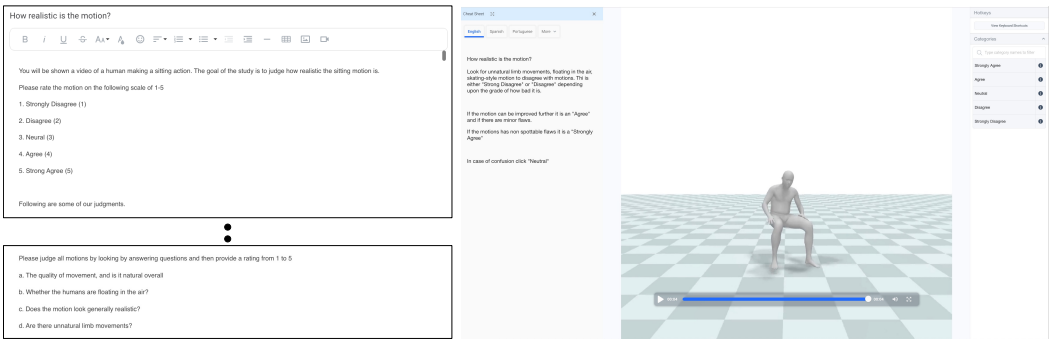

Figure 7: **Likert User Study.** We conduct a user study to assess the motion quality in our Synthetic Dataset. On the left, we present the qualification instructions for participants, allowing only those who perform well to proceed to the actual study. On the right, we display the user interface used 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

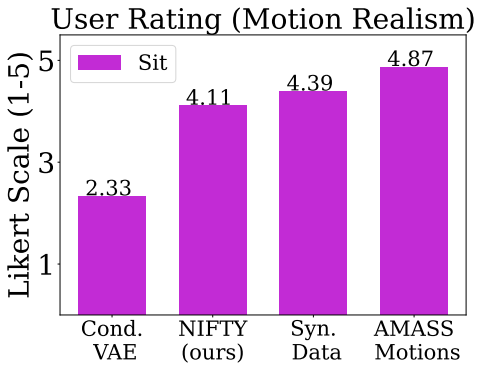

Figure 8: **Likert User Study Results.** We conduct a study to judge the realism of sitting motions on a scale of 1-5. Instructions for this study are available in §A.1. We show that synthetic training data (Syn. Data) generated using our algorithm Algorithm 1 has an average rating of **4.39**. This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

detailed in §C.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.

A.2 Training Data Synthesis Algorithm

Our generation process revolves around utilizing a pretrained motion model, specifically the HuMoR generative model [33], 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` 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.

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

B 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 [23] 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 [31]. We use the HuMoR-Qual [33] 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 [29]. Our each transformer layer consists of 4 heads and a latent dim on 512. We have 8 such layers in our transformer.

C Experimental Details

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

C.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. 9 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.

C.2 Metrics

Apart from performing the user study described in §C.1 we also evaluate all our models and baselines on several quantitative metrics. We detail these metrics below (apart 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.

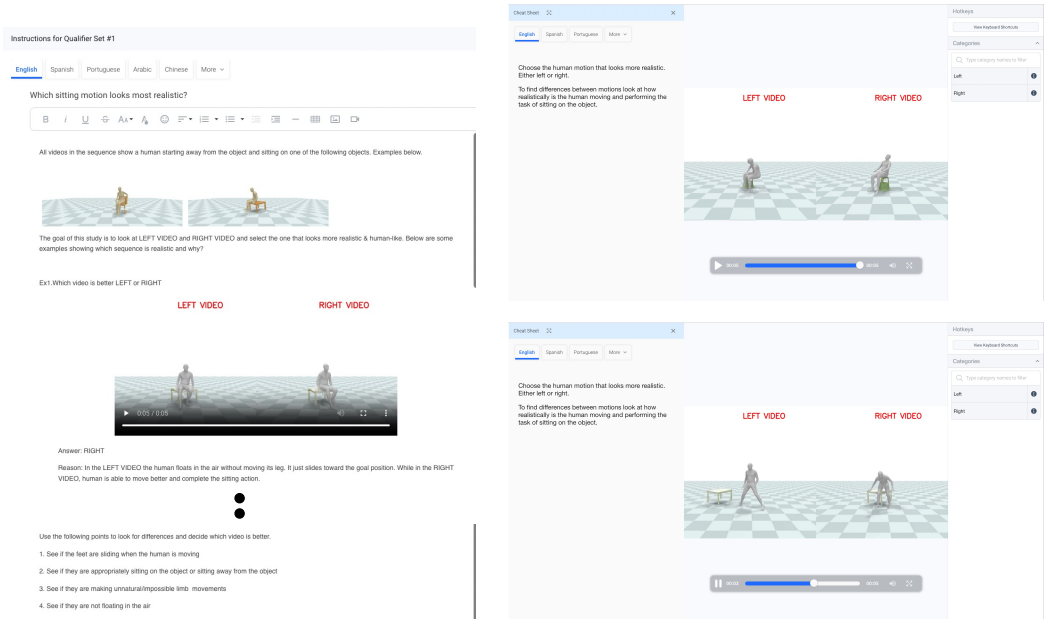

Figure 9: **A/B Testing User Study** We use this study to compare the quality of motions generated by different methods by requiring them to generate human-object interaction motions. On the left, we show the instruction set following which all users are required to pass a qualification exam to participate 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".

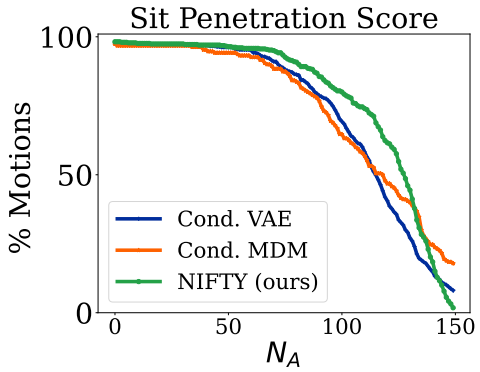

Figure 10: **Penetration Score Sitting**. We graph the percentage of motion sequences with a penetration score of less than or equal to 2cm (Y-axis), compared to the number of approach frames, denoted as N_A (X-axis). Our findings reveal that regardless of the value of N_A, NIFTY (green) consistently exhibits a greater proportion of motion sequences with low penetration scores.

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. 10 (sit) and Fig. 11 (lift).

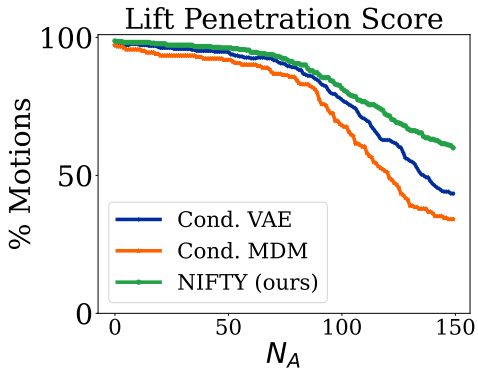

Figure 11: **Penetration Score Lifting**. We graph the percentage of motion sequences with a penetration score of less than or equal to 2cm (Y-axis), compared to the number of approach frames, denoted as N_A (X-axis). Our findings reveal that regardless of the value of N_A, NIFTY (green) consistently exhibits a greater proportion of motion sequences with low penetration scores.

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.

D 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 §D.1. In §D.2, D.3, and D.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.

D.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. 3 presents the comparison between this baseline and our method. The skeleton distance metric can be sensitive to 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* ≤ 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.

D.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. 4. 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 output. Note that drawing additional samples can be done efficiently in parallel.

Table 3: **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 § D.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 $\downarrow$	% *D2O* $\leq 2cm \uparrow$	*D2O* $95^{\text{th}}\% \downarrow$	Skel. Dist. $\downarrow$	% Skel. Dist. $\leq 25cm \uparrow$	Contact IoU $\uparrow$	% Pen. $\leq 2cm \uparrow$
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 $\downarrow$	% *D2O* $\leq 2cm \uparrow$	*D2O* $95^{\text{th}}\% \downarrow$	Skel. Dist. $\downarrow$	% Skel. Dist. $\leq 25cm \uparrow$	Contact IoU $\uparrow$	% Pen. $\leq 2cm \uparrow$
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

Table 4: **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 $\downarrow$	% *D2O* $\leq 2cm \uparrow$	*D2O* $95^{\text{th}}\% \downarrow$	Skel. Dist. $\downarrow$	% Skel. Dist. $\leq 25cm \uparrow$	Contact IoU $\uparrow$	% Pen. $\leq 2cm \uparrow$
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 $\downarrow$	% *D2O* $\leq 2cm \uparrow$	*D2O* $95^{\text{th}}\% \downarrow$	Skel. Dist. $\downarrow$	% Skel. Dist. $\leq 25cm \uparrow$	Contact IoU $\uparrow$	% Pen. $\leq 2cm \uparrow$
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

D.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. 5. 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.

D.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

Table 5: **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* $\leq 2cm \uparrow$	*D2O* $95^{\text{th}}\tilde{\%} \downarrow$	% Pen. $\leq 2cm \uparrow$	Skel. Dist. ↓	% Skel. Dist.$\leq 25cm \uparrow$	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* $\leq 2cm \uparrow$	*D2O* $95^{\text{th}}\tilde{\%} \downarrow$	% Pen. $\leq 2cm \uparrow$	Skel. Dist. ↓	% Skel. Dist.$\leq 25cm \uparrow$	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

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. 6, 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.

Table 6: **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 hyperparameters.

	Sitting						
# Input Frames	Foot Skating ↓	% *D2O* $\leq 2cm \uparrow$	*D2O* $95^{\text{th}}\tilde{\%} \downarrow$	Skel. Dist. ↓	% Skel. Dist.$\leq 25cm \uparrow$	Contact IoU ↑	% Pen. $\leq 2cm \uparrow$
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* $\leq 2cm \uparrow$	*D2O* $95^{\text{th}}\tilde{\%} \downarrow$	Skel. Dist. ↓	% Skel. Dist.$\leq 25cm \uparrow$	Contact IoU ↑	% Pen. $\leq 2cm \uparrow$
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

D.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. 7, this leads to a subpar performance across the board on sit and lift actions.

Table 7: **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 comared to

	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

D.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. 8. 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.

Table 8: **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

E Qualitative Results

Motion generation results are best seen as videos on the attached webpage. We also include static visualizations here in Fig. 12 and Fig. 13. The webpage additionally also shows visualizations (10 motions) from our method for every object in our dataset.

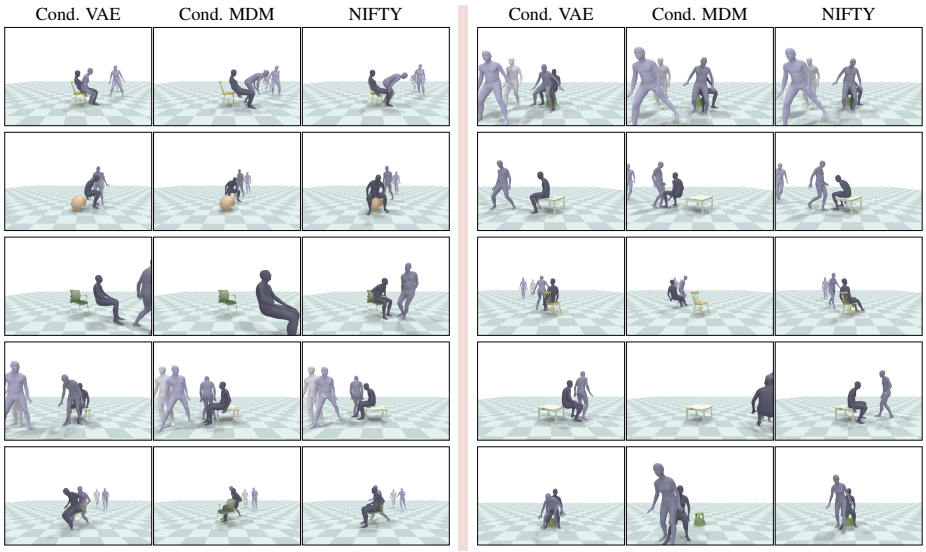

Figure 12: **Comparison Qualitative Motions Sitting**. Compared to other baselines, our method (NIFTY) produces more realistic motions. When examining the motion examples generated by the baselines, we notice that in all cases where a person approaches an object to sit, either the person completely misses 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 `supplementary.html`.

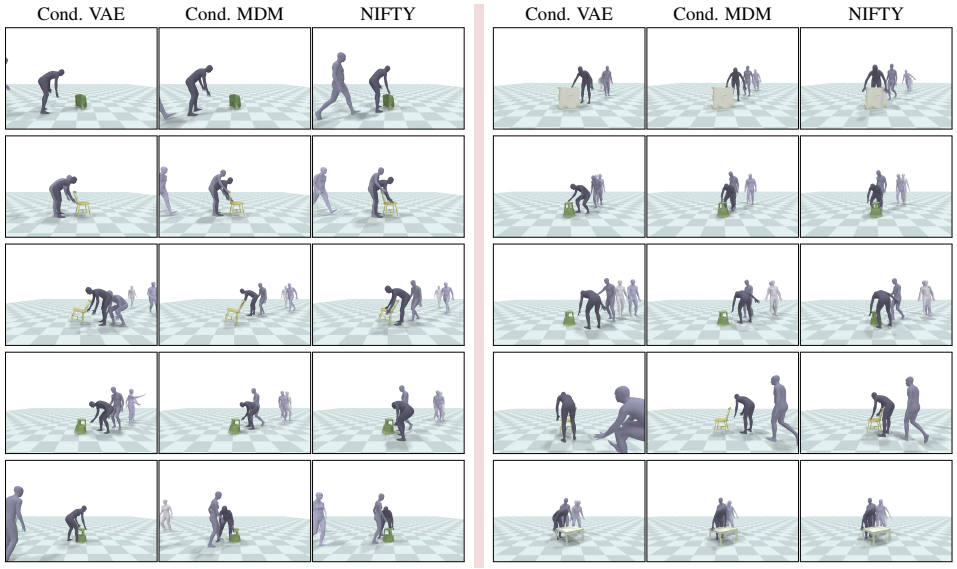

Figure 13: **Comparison Qualitative Motions Lifting**. NIFTY generates more realistic motions as compared to the baseline methods. With motions generated using the baseline methods, we see that the lifting stance is often taken far from the object. To better evaluate these results, please refer to the qualitative videos of these motions in the `supplementary.html` file.

F 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 [33] trained on the AMASS repository [25]. 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.

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.