Decoupled Generative Modeling for Human-Object Interaction Synthesis

Supplementary Material

Overview

In this supplementary material, we provide additional details and analyses to complement the main paper. Section A summarizes the experimental setup, including key implementation details and training and inference configurations. Section B defines the evaluation protocols and metrics, while Section C introduces our reconstruction guidance for improving hand-object contact and foot grounding. Section D describes DynaPlan for collision-aware dynamic planning, and Section E reports additional experiments, including an ablation on the generator loss weight, comparisons under privileged waypoint supervision, and an oracle trajectory study. Sections F to J detail the loss landscape visualization, the implementation of the OMOMO variants, the user study design, extended qualitative results, and limitations and future directions.

A. Experimental Setup

A.1. Implementation Details

Our models are implemented in the PyTorch deep learning framework. All experiments are conducted on a single NVIDIA RTX 3090 GPU with 24 GB of memory, and training requires approximately 25 GPU hours. We optimize the generators with the Adam optimizer [5] using a learning rate of 1×10^{-4} and a batch size of 32, where each training sample is a sequence with $T = 120$ frames. During training, input sequences shorter than T are zero-padded to match the length. The transformer-based denoising network consists of 4 attention heads and 4 layers. We train the model for 580k steps and report results using the checkpoint that achieves the best validation performance. The diffusion process uses 1,000 noising steps during training, and we employ the standard DDPM [4] sampling procedure at inference time.

For adversarial training of the discriminator, we use the same optimizer configuration as for the generators, but set the learning rate to 2×10^{-5}. The discriminator is updated once for every generator update. The loss balancing weights for the trajectory generator, action generator, and forward kinematics objectives are fixed to $\lambda_{\text{TG}} = 0.1$, $\lambda_{\text{AG}} = 1.0$, and $\lambda_{\text{FK}} = 1.0$, respectively. The adversarial loss weight λ_{G} is adjusted within the range $[0.01, 0.05]$ during training.

In addition, to prevent error accumulation from the trajectory generator (TG) propagating into the action generator (AG), we provide clean trajectories for both the human and the object as part of the noisy input to AG during training. In other words, AG is conditioned on ground-truth trajectories while the remaining components are corrupted with noise, which is inspired by the teacher forcing strategy [12] commonly used to mitigate error accumulation. Note that during training TG receives noisy inputs for all frames except the start and goal conditions. At inference time the trajectories predicted by TG are then used as inputs to AG.

A.2. Inference Runtime and Resource Consumption

Methods	Inference time (s)	Inference VRAM (GB)
CHOIS [8]	2.00	1.9
HOIFHLI [13]	162.71	5.2
Ours (DecHOI)	3.41	2.1

Table 1. Inference runtime and GPU memory usage for CHOIS [8], HOIFHLI [13], and our DecHOI.

In this section, we report the computational cost of DecHOI and the baselines used in our comparison. DecHOI employs two lightweight denoising networks as specialized experts for the decoupled generative modeling of trajectories and interactions. This design incurs a modest increase in inference time compared to CHOIS [8], while remaining significantly more efficient than HOIFHLI [13], which relies on a generation pipeline with multi-stage for grasp generation and refinement and therefore has substantially higher inference time.

We further observe that DecHOI requires comparable or even lower VRAM usage than the prior models. Considering the qualitative and quantitative improvements, the additional inference cost of DecHOI is modest relative to the overall computational budget. Moreover, when taking into account the manual effort required to annotate intermediate waypoints for the baselines, DecHOI offers a more practical solution in realistic deployment scenarios.

B. Evaluation Details

For an accurate and fair comparison, all trajectories are expressed in a common global world frame, and all protocols and thresholds used for metric computation follow the same settings as CHOIS [8].

B.1. Condition Matching

We measure T_s and T_e as the Euclidean distances between the predicted and ground-truth object centroids at the start frame and at the final target position, respectively. Since our generator uses only the start and goal points, we evaluate condition matching solely with T_s and T_e.

B.2. Human Motion Quality

Foot height. H_{feet} is the mean distance from the feet to the floor. To obtain the floor level z_{floor}, we identify frames where the toe joints move below a small speed threshold (quasi-static stance), cluster their toe heights, and take the lowest cluster as z_{floor}. All foot z-coordinates are then measured relative to this floor level when computing H_{feet} and foot sliding.

Foot sliding. For the ankles and toes, we accumulate horizontal displacement between consecutive frames, but only when each joint is near the floor, as:

$$\sum_{t=0}^{T-1} \mathbb{1}(z_{j,t} < H_j) d_{j,t} (2 - 2^{z_{j,t}/H_j}), \tag{1}$$

where $\mathbb{1}(\cdot)$ is the indicator function, $z_{j,t}$ is the height of joint j at frame t, H_j is the joint specific height threshold that defines the near floor region, and $d_{j,t}$ is the horizontal displacement between frame t and $t+1$. The index j runs over the left and right ankles and toes. The accumulated displacement is then averaged over time and across the four joints.

R-Precision. For each text-motion pair, we compute feature vectors for the text and candidate motions, then rank the motions by cosine similarity to the text, following [2, 8]. R_{prec} (top-3) is the fraction of pairs for which the ground-truth motion appears in the top-3 ranked motions, averaged over all evaluation pairs.

FID. We compute motion features for all generated and real sequences, fit a Gaussian to each set, and use the Fréchet Inception Distance (FID) between the two Gaussians as the FID score [3]. Lower FID indicates that the distribution of generated motions is closer to that of real motions.

DIV. For each model, we collect motion features from all generated results, sample random feature pairs, and average their Euclidean distances to obtain DIV, following [2]. We report a single DIV value per model, where values closer to the DIV of real data indicate more realistic diversity.

B.3. Interaction Quality

Contact metrics. For each frame, we assign a binary contact label based on the minimum distance between the hand joints and the object surface: if the closest distance from either hand joint to any object vertex is below 5 cm, the frame is marked as in contact. Given ground-truth contact labels, $C_{\%}$ is the fraction of frames predicted in contact, C_{prec} and C_{rec} are precision and recall for the contact class, and C_{F1} is their F1 score.

Penetration metrics. We measure mesh interpenetration using signed distance fields (SDFs). At every frame, we map human and object vertices into the corresponding SDF frame and sample their signed distances, with negative values indicating penetration. We average the negative signed distances into four scalars: P_{hand} and P_{body} for penetration of hand and full-body vertices into the manipulated object, and $P_{o\to s}$ and $P_{h\to s}$ for penetration of the object and the human into the static scene.

Figure 1. Visualization of trade-off relationships induced by the metric definitions. The top row illustrates the trade-off between foot related metrics, and the bottom row shows the relationship between contact and penetration metrics.

B.4. Ground Truth Difference

MPJPE. We report MPJPE as the mean Euclidean distance between predicted and ground-truth joint positions [9], averaged over all joints and frames in the same world frame.

Root and object translation error. T_{root} and T_{obj} are the mean Euclidean distances between predicted and ground-truth root joint positions and object centroids, respectively, averaged over all frames.

Object orientation error. O_{obj} measures the discrepancy between predicted and ground-truth object rotations. We compute the mean Frobenius norm of the difference between the two rotation matrices per frame and average over time.

B.5. Metrics Trade-off

Foot-Ground trade-off. Foot sliding (FS) assigns larger penalties when H_{feet} is small, as defined in Eq. B.2. Consequently, as illustrated in the top row of Fig. 1, even large horizontal displacements yield little or no foot sliding when the feet remain high above the ground (hovering). In contrast, when the feet stay very close to the floor, even small displacements are amplified in the FS score. This behavior induces a trade-off between H_{feet} and FS.

Contact-Penetration trade-off. As shown in the bottom row of Fig. 1, the contact-relative score and the penetration score also exhibit a trade-off relationship. Since con-

tact is determined based on the distance between each hand joint and the nearest object vertex, severe penetration can still yield a stable contact score. Conversely, when the hand does not approach the object, the penetration score can be favorable due to the absence of intersecting regions, while the contact score becomes poor. Therefore, metrics with such trade-off behavior should not be interpreted as a single score in isolation but evaluated jointly.

C. Inference-time Reconstruction Guidance

To obtain robust and plausible motion during inference, we apply reconstruction guidances following prior work [8]. First, to strengthen hand and foot contacts, we define a guidance term that regularizes the distance between the distal human joints and their nearest object vertices. For the hand joints this term is written as:

$$\mathcal{F}_{\text{cont}} = \|\mathbf{M}_l \odot |H_l - V_l|\|_1 + \|\mathbf{M}_r \odot |H_r - V_r|\|_1, \quad (2)$$

where H_l and H_r denote the left and right hand joint positions and V_l and V_r are the corresponding closest object vertices. The binary masks $\mathbf{M}_l$ and $\mathbf{M}_r$ indicate whether a hand joint is in contact with the object and are obtained by thresholding the hand to object distance with a threshold of 5 cm.

Second, to encourage stable foot grounding and a realistic stance, we introduce a guidance term that regularizes the distance between the feet and the floor plane. Given the positions of the left and right feet F_l and F_r, this term is defined as:

$$\mathcal{F}_{\text{feet}} = \|\min(F_l, F_r) - h\|_2, \quad (3)$$

where $h = 0.02$ m is a threshold for foot height estimated from the ground-truth motion. The error is computed only along the vertical coordinate.

D. DynaPlan

Inspired by [8], we project each 3D indoor scene from Replica [11] into a 2D grid layout, marking walkable regions as traversable and expanding non-walkable regions (e.g., walls and furniture) with a distance transform to enforce a human scale safety margin. *DynaPlan* then performs path planning for the agent and obstacle on this grid using A*. After the paths are optimized, the resulting trajectories are lifted back into the original 3D scene using the known metric scale [8, 11], and resampled for the agent and obstacles so they provide global plans and motion priors for full-body HOI generation in a consistent 3D coordinate frame. Fig. 2 visualizes the overall execution process of *DynaPlan*.

D.1. Obstacle Trajectory Modeling

Given obstacle start and goal points, the moving obstacle follows a deterministic A*-based trajectory, advancing by one grid cell at each time step t. *DynaPlan* uses a pre-trained Social-STGCNN [10] to forecast future obstacle positions. Given the past T_{obs} obstacle positions, the model predicts T_{pred} future positions. These predicted positions do not control the obstacle. Instead, they are used only to construct predictive risk regions that *DynaPlan* uses for planning.

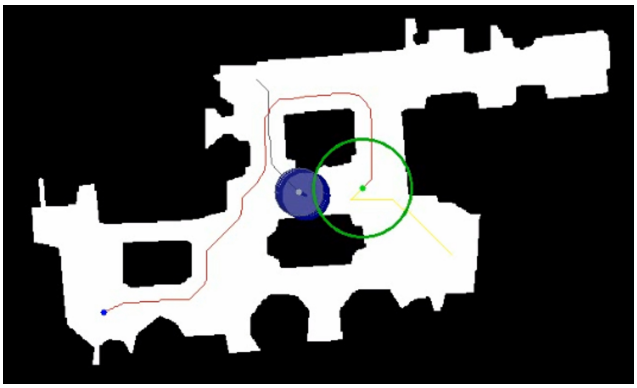

Figure 2. The green dot and circle denote the agent and R_{agent}, and the red curve shows the re-planned path. When the obstacle (gray) and its influence region (blue) enter R_{agent}, that area is given high cost in the risk map and dynamic re-planning is triggered.

D.2. Collision Detection

Given an agent start and goal points, *DynaPlan* first computes an initial path using A*, minimizing the accumulated Euclidean step length ($\sum_{t=1}^{T} \|x_t - x_{t-1}\|_2$). In addition, we assign an influence radius to the agent and obstacle, and declare a collision whenever the obstacle's radius along its path overlaps with that of the agent. For collision detection we use only the predicted obstacle trajectories. Once a collision is detected, the agent updates a risk map and calls A* again to re-plan its path.

D.3. Dynamic Re-planning

When a potential collision is detected, *DynaPlan* performs dynamic re-planning by updating the risk map based on the current and predicted obstacle positions. For a grid location x, the risk map $R(x)$ is:

$$R(x) = \exp\left(-\frac{\|x - o_t\|_2^2}{2\sigma^2}\right) + \sum_{k=1}^{T_{\text{pred}}} \gamma_k \exp\left(-\frac{\|x - \hat{o}_{t+k}\|_2^2}{2\sigma^2}\right). \quad (4)$$

We use a Gaussian weight with $\sigma = 1$ to assign smaller values to locations farther from the obstacle, and set $\gamma \in (0, 1)$ so that predicted future positions contribute less than the current position.

Methods	Condition Matching		Human Motion Quality		Interaction Quality		
	$T_s\downarrow$	$T_e\downarrow$	FS$\downarrow$	$R_{prec}\uparrow$	$C_{F1}\uparrow$	$P_{\text{hand}}\downarrow$	$P_{\text{body}}\downarrow$
$\lambda_G = 0.01$	1.69	7.60	0.45	0.70	**0.69**	0.55	0.56
$\lambda_G = 0.03$	1.69	7.83	0.40	0.69	0.67	**0.53**	**0.54**
$\lambda_G = 0.05$	1.98	7.87	0.42	0.69	0.66	**0.53**	0.56
Ours (DecHOI)	**1.59**	**6.91**	**0.38**	**0.72**	0.67	**0.53**	**0.54**

Table 2. Ablation comparing fixed and adaptive λ_G settings shows that the adaptive strategy yields a better overall balance.

Given a risk map R, the cost of a candidate path P is:

$$\mathcal{C}(P \mid R) = \sum_{t=1}^{T} \left(\|x_t - x_{t-1}\|_2 + \lambda_R R(x_t) \right). \quad (5)$$

We set $\lambda_R = 4.0$ to balance A* cost and risk cost.

To decide between detouring and waiting, *DynaPlan* evaluates candidate waiting time steps ranging from zero (detouring) to T_{pred}. For each waiting candidate, the agent holds its position for the corresponding duration while the obstacle progresses, and the risk map is updated accordingly. A waiting time penalty is applied to discourage long waits, by adding to the risk map cost an amount proportional to the number of waiting time steps.

DynaPlan selects the candidate with the lowest score, providing a lightweight, prediction-driven re-planning mechanism that enables collision-aware navigation.

E. Additional Experiments

E.1. Ablation Study on Generator Weight

In this section, we investigate the effect of the generator loss weight λ_G on the overall objective (Eq. 9) defined in the main paper. Tab. 2 compares fixed settings where λ_G is set to 0.01, 0.03, or 0.05 with an adaptive strategy that dynamically adjusts λ_G between 0.01 and 0.05 according to the performance of the discriminator $\mathcal{D}$. When $\mathcal{D}$ becomes too strong, the adaptive scheme increases λ_G so that the generator places more emphasis on fooling the discriminator. When $\mathcal{D}$ influence weakens, the scheme decreases λ_G to avoid over-regularization by the adversarial term.

With fixed values, training tends to drift toward an unbalanced regime where either the generator or the discriminator dominates, which often prevents the loss from converging. This lack of convergence not only degrades interaction quality but also harms the overall synthesis performance, leading to weaker condition matching and motion quality. In contrast, the adaptive weighting maintains a better balance in the adversarial training, stabilizes convergence, and yields the best performance across metrics. These results demonstrate that adaptive generator loss weighting effectively mitigates the inherent instability and bias of adversarial training and enables more reliable optimization.

E.2. Effect of Privileged Waypoint

In the main paper, we evaluate DecHOI and prior waypoint-based methods [8, 13] under a fair protocol where all models receive the same inputs. For completeness, we additionally report an experiment on the *FullBodyManipulation* [7] in which prior methods are evaluated under their original waypoint supervised configuration. In this experiment, DecHOI still operates without any intermediate waypoints, while the prior methods are given waypoints and are thus evaluated in a privileged information setting. The quantitative results of this comparison are summarized in Tab. 3. Note that for all baselines we report scores reproduced using the official released implementations, since several metrics are not reported in the CHOIS and HOIFHLI papers (e.g., DIV and P_{body}).

Both CHOIS [8] and HOIFHLI [13] benefit noticeably from waypoint supervision. In particular, human motion quality and GT difference metrics improve significantly. The GT difference metrics are highly sensitive to waypoints, and for T_{obj} we observe improvements of up to a factor of five, since waypoints directly specify object target positions. For this reason, GT difference metrics are not suitable for a direct comparison with DecHOI, which does not receive any waypoint input.

In contrast, DecHOI achieves comparable or better performance than the prior approaches on most metrics, even without privileged conditions. Considering the trade-off structure among metrics discussed in Sec. B.5, DecHOI shows slightly better overall performance in terms of H_{feet} and FS and also provides improved text alignment, realism, and diversity, enabled by our decoupled design. Moreover, given the inherent trade-off between contact score and penetration, DecHOI attains superior interaction quality than prior work, despite operating without intermediate waypoints while the baselines use them. These results suggest that DecHOI enables efficient and realistic interaction synthesis under weaker input priors and that the proposed adversarial training makes a clear contribution to improving interaction quality.

E.3. Oracle Trajectory Ablation

We evaluate an oracle setting that bypasses the trajectory generator (TG) and feeds the action generator (AG) with ground-truth object and human trajectories $\{\mathcal{T}_o^{\text{GT}}, \mathcal{T}_h^{\text{GT}}\}$ for all T frames. The AG architecture, checkpoint, and inference-time guidance are kept unchanged, and the conditioning interface follows Sec. 3.2 (see Fig. 2) in the main paper. This design disentangles the contributions of the TG and AG, allowing us to assess the standalone performance of the TG and to estimate the theoretical upper bound of AG performance within the decoupled pipeline. We conduct this experiment on the *FullBodyManipulation* [7], and report the results in Tab. 4.

Methods	Condition Matching		Human Motion Quality					Interaction Quality					GT Difference			
	$T_s\downarrow$	$T_e\downarrow$	$H_{\text{feet}}\downarrow$	$FS\downarrow$	$R_{prec}\uparrow$	$FID\downarrow$	$DIV\rightarrow$	$C_{\text{prec}}\uparrow$	$C_{\text{rec}}\uparrow$	$C_{F1}\uparrow$	$P_{\text{hand}}\downarrow$	$P_{\text{body}}\downarrow$	MPJPE$\downarrow$	$T_{\text{root}}\downarrow$	$T_{\text{obj}}\downarrow$	$O_{\text{obj}}\downarrow$
CHOIS [8]	1.72	6.95	4.49	**0.35**	0.66	0.69	8.37	**0.80**	0.64	**0.68**	0.59	0.60	15.30	24.43	13.35	0.98
HOIFHLI [13]	1.64	8.05	4.91	0.36	0.62	1.51	8.85	0.79	**0.65**	0.67	0.60	0.58	15.93	**23.31**	**10.98**	1.10
Ours (DecHOI)	**1.59**	**6.91**	**4.42**	0.38	**0.72**	**0.33**	**8.86**	**0.80**	0.64	0.67	**0.53**	**0.54**	**15.27**	25.47	22.96	**0.86**

Table 3. Quantitative comparison on the *FullBodyManipulation* [7] in the privileged waypoint supervised setting, where DecHOI operates without intermediate waypoints and CHOIS [8], HOIFHLI [13] receive sparse intermediate waypoints.

Methods	Human Motion			Interaction				
	$R_{\text{prec}}\uparrow$	$FID\downarrow$	$DIV\rightarrow$	$C_{\text{prec}}\uparrow$	$C_{\text{rec}}\uparrow$	$C_{F1}\uparrow$	$P_{\text{hand}}\downarrow$	$P_{\text{body}}\downarrow$
DecHOI	**0.72**	0.33	8.86	0.80	0.64	0.67	**0.53**	**0.54**
DecHOI (oracle)	0.66	**0.15**	**8.97**	**0.82**	**0.65**	**0.69**	0.56	0.57

Table 4. Oracle trajectory ablation study on the FullBodyManipulation [7]. DecHOI (oracle) denotes a variant in which the trajectory generator is replaced by ground truth trajectories.

Results and discussion. In this oracle configuration, metrics that quantify condition matching or deviation from the conditioning trajectories are less informative, since the AG is driven directly by ground-truth object and human paths. We therefore focus our analysis on human motion quality and interaction quality. The oracle setting generally achieves higher scores on most metrics, yet the gap relative to the standard DecHOI that uses predicted trajectories remains modest. On several metrics (R_{prec}, P_{hand}, and P_{body}), the standard DecHOI even slightly outperforms the oracle, and its overall performance is very strong when the typical trade-offs between contact and penetration are taken into account. This small performance difference indicates that the TG generates realistic trajectories that closely follow the distribution of ground-truth paths, and suggests that the AG is not tightly constrained by the TG performance but is capable of producing high quality interaction synthesis independently.

F. Loss Landscape Visualization

We visualize the loss landscape around our converged model by evaluating the training objective on a two-dimensional subspace of the parameter space, following [6]. We fix a batch $\mathcal{B}$ drawn from the same distribution as the training data, so that variations in the loss reflect only changes in the model parameters. Let w_0 be the vector obtained by flattening and concatenating all trainable tensors of the final trained model.

To explore the neighborhood of w_0, we construct two random perturbation directions. For each trainable tensor, we sample Gaussian noise with the same shape, normalize it to obtain a comparable scale across layers, and concatenate these tensors to form the first direction d_x. We repeat this procedure to obtain a second direction d'_y and orthogonalize it with respect to d_x using Gram-Schmidt to obtain the final direction d_y. This keeps the perturbations balanced across layers and avoids a single layer dominating the change.

We then form a grid of coefficients $\alpha, \beta \in [-r, r]$ and, for each pair (α, β), construct a perturbed parameter vector $w(\alpha, \beta) = w_0 + \alpha d_x + \beta d_y$, reshape it back to the original tensor shapes, and evaluate the total loss on $\mathcal{B}$. The guidance term is omitted since it is used only at inference time and is not part of the training objective. Plotting the resulting loss values over the (α, β) grid as contour and surface plots provides a qualitative view of the local loss landscape around w_0 and indicates whether the optimization problem near the solution is relatively simple with a smooth basin or more complex with sharper curvature and narrower valleys.

G. OMOMO Implementation Details

We implement Lin-OMOMO and Pred-OMOMO within the original OMOMO framework [7], following the variant definitions introduced in CHOIS [8]. To match the conditioning used in our experiments, our OMOMO variants are provided only with the object start and goal states.

G.1. Lin-OMOMO

Lin-OMOMO uses the original OMOMO generator and training setup. The object trajectory is defined by linearly interpolating the object centroid between its start and goal positions and is used as the object translation input in the OMOMO conditioning stream. All centroid-related values are updated accordingly. Since Lin-OMOMO only interpolates the object centroids and keeps the orientation identical to the ground-truth, we omit O_{obj} for Lin-OMOMO from our reported results.

G.2. Pred-OMOMO

In Pred-OMOMO, the original OMOMO generator and training configuration are kept intact, while the object motion is obtained from CHOIS. Given the object's start and goal conditions as input, CHOIS predicts the full trajectory of object centroids and rotations. We use this predicted object motion as the input in the OMOMO conditioning stream and update all related pose values to stay in sync with these predictions. Because both the trajectory of the object centroid and its rotations are supplied directly by CHOIS, the object-level metrics T_{obj} and O_{obj} for Pred-OMOMO are identical to those of CHOIS in our experiments.

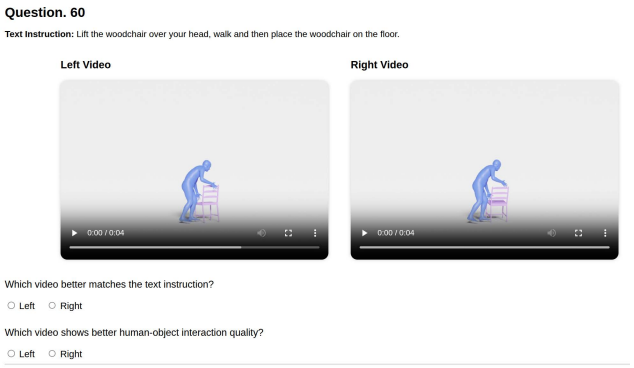

Figure 3. Example 2AFC interface in which participants read a text instruction and compare two anonymized clips to judge text alignment and interaction quality.

H. User Study Details

We conducted a two-alternative forced-choice (2AFC) perceptual study to evaluate two criteria of synthesized human-object interactions: Text Alignment (how faithfully a clip follows the natural-language instruction) and Interaction Quality (the perceived realism and plausibility of contact, including stable hand and foot contacts, few penetrations, and minimal temporal jitter).

For each comparison round (DecHOI-CHOIS [8] and DecHOI-HOIFHLI [13]), we randomly sampled 30 text-scene pairs from the full test set without scene duplication, yielding 60 clips in total. To ensure fairness, we shuffled both the order of videos and the left-right placement of each method in every trial, and fixed all video durations to 4 seconds. As shown in Fig. 3, the interface presents the Text Instruction at the top and two anonymized videos. Participants could replay videos and answer two questions. They selected which clip better matched the instruction (Text Alignment) and which clip exhibited better human-object interaction quality (Interaction Quality). To avoid presentation bias, no method identifiers or cues were provided.

Using the results in Fig. 8 of the main paper, participants preferred DecHOI in 71.5% of trials versus CHOIS and 67.5% versus HOIFHLI for Text Alignment, and in 69.0% of trials versus CHOIS and 63.5% versus HOIFHLI for Interaction Quality. These trends are consistent with our quantitative metrics and support the claim that decoupling trajectory and action generation improves both semantic faithfulness and perceptual interaction quality.

I. Additional Qualitative Results

We provide additional qualitative results that complement the examples in the main paper.

Fig. 4 extends Fig. 4 in the main paper by including additional scenes from *FullBodyManipulation* [7], comparing DecHOI against CHOIS [8] and HOIFHLI [13]. Across diverse instructions (e.g., lift-move-place, push and pull), DecHOI maintains stable hand-object contacts and grounded object trajectories, exhibiting less penetration and hovering. In contrast, the baselines show drift, temporal desynchronization between human and object, or incomplete contacts, especially during lift-place transitions and when objects change orientation.

Fig. 5 provides additional qualitative results that complement Fig.5 in the main paper, showcasing *3D-FUTURE* [1] objects that are unseen during training. DecHOI maintains precise hand-object contact locations, more coherent placement motion, and consistent text to motion alignment, whereas CHOIS often produces mesh intersections or unstable object poses. The qualitative patterns align with the quantitative improvements reported for condition matching, motion stability, and contact reliability on unseen shapes.

Fig. 6 complements Fig. 7 in the main paper with additional long-horizon dynamic scenes from *DynaPlan*. We visualize collision events together with the corresponding re-planned direction. The dashed arrows indicate the original direction of motion, which would be the optimal path in free space, yet the agent re-routes around obstacles instead of strictly following this direction. These behaviors demonstrate that obstacle detection and dynamic planning are applied effectively and suggest that the framework can scale to more complex scenarios.

J. Limitations & Future Work

Our framework focuses on synthesizing human-object interactions that involve manipulating rigid objects. In real environments, however, many objects are deformable or articulated. Handling such cases requires contact regions that adapt over time to object parts whose positions change, and in some scenarios it is necessary to model underlying physical effects such as friction, inertia, and gravity. Addressing these challenges would require more sophisticated representations of articulated structure and dynamics. Incorporating recent articulation estimation networks as object priors is a promising direction, and systematically extending our approach to deformable and articulated objects constitutes an important avenue for future work.

In addition, the proposed *DynaPlan* module currently supports dynamic planning and interaction synthesis for a single manipulated object and a single moving obstacle. Real-world scenarios often involve more complex environments, such as crowds of agents and multiple objects being manipulated simultaneously. Scaling to such settings may require path planners that go beyond classical A* and can reason about many interacting agents. Moreover, supporting multiple manipulated objects would likely depend on constructing richer datasets, for example by combining motion capture with procedural composition of multi object interactions. Exploring these extensions would be highly valuable for broadening the applicability of our framework.

Figure 4. Additional qualitative comparison of DecHOI with CHOIS [8] and HOIFHLI [13] on the *FullBodyManipulation* [7]. Also DecHOI produces stable contacts, smooth motion, and accurate object trajectories, while prior methods show drift, penetration, or inconsistent coordination between human and object motions.

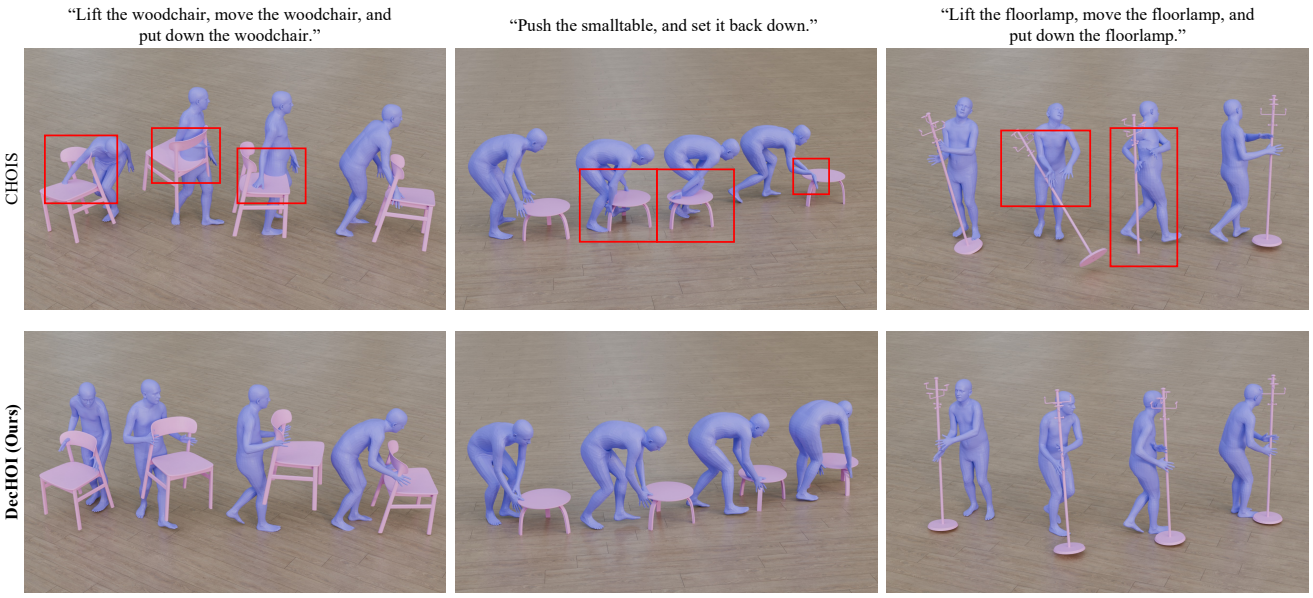

Figure 5. Additional qualitative comparison of DecHOI and CHOIS [8] on the *3D-FUTURE* [1], demonstrating generalization to unseen object categories such as woodchair, smalltable, and floorlamp.

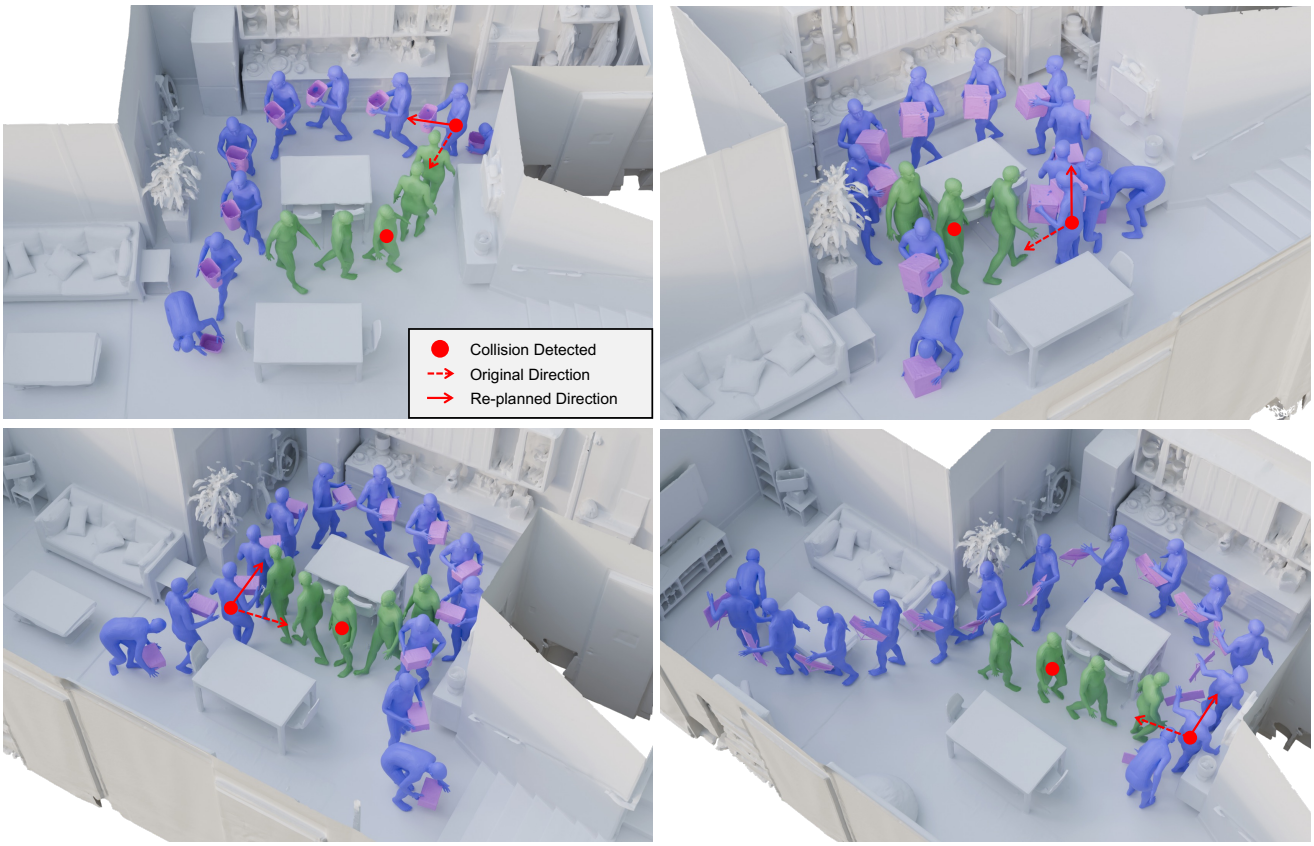

Figure 6. Visualization of DecHOI in long-sequence dynamic environments. The human agent (blue) adaptively re-plans its path when encountering a moving obstacle (green), often waiting briefly before detouring, and thereby maintaining goal-directed and collision-free motion.

References

[1] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129(12):3313–3337, 2021. 6, 8

[2] 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*, pages 5152–5161, 2022. 2

[3] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 2

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 1

[6] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018. 5

[7] Jiaman Li, Jiajun Wu, and C Karen Liu. Object motion guided human motion synthesis. *ACM Transactions on Graphics (TOG)*, 42(6):1–11, 2023. 4, 5, 6, 7

[8] Jiaman Li, Alexander Clegg, Roozbeh Mottaghi, Jiajun Wu, Xavier Puig, and C Karen Liu. Controllable human-object interaction synthesis. In *European Conference on Computer Vision*, pages 54–72. Springer, 2024. 1, 2, 3, 4, 5, 6, 7, 8

[9] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2640–2649, 2017. 2

[10] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14424–14432, 2020. 3

[11] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 3

[12] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989. 1

[13] Zhen Wu, Jiaman Li, Pei Xu, and C Karen Liu. Human-object interaction from human-level instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11176–11186, 2025. 1, 4, 5, 6, 7