CoLM: Controllable Text-Driven Humanoid Loco-Manipulation Generation

Anonymous authors
Paper under double-blind review

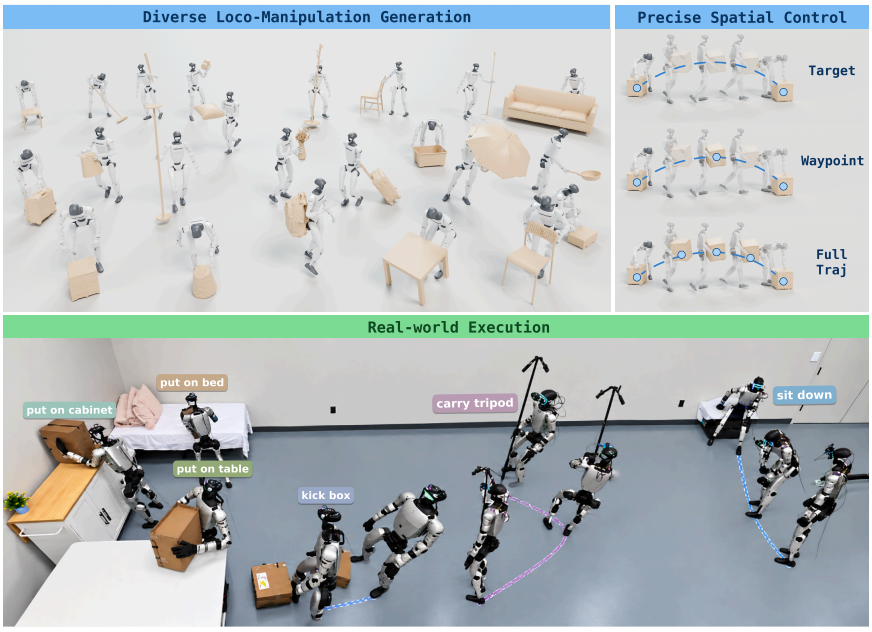


Figure 1: CoLM generates diverse loco-manipulation motions conditioned on text and spatial constraints and demonstrates its applicability to real-world tasks.

Abstract

Humanoid loco-manipulation requires coordinated whole-body motion that is semantically aligned, contact-consistent, spatially controllable, and physically executable. However, collecting such motion data through teleoperation or motion capture is costly, while existing generative approaches do not model or control the manipulated object. To this end, we present **CoLM**, a framework for generating and executing humanoid loco-manipulation from language and explicit object-level spatial constraints. Given a text description, object geometry, and the initial humanoid and object states, CoLM jointly generates humanoid and object motion while supporting spatial control ranging from a target object pose to a full trajectory. To provide training data at scale, we construct a humanoid loco-manipulation dataset by retargeting existing human-object interaction data to humanoid with *contact-guided refinement*. We further introduce the *interaction field*, a dense representation of humanoid-object spatial relations that serves as a shared interaction objective across generation and execution. Experiments show that CoLM outperforms existing baselines in motion quality, semantic fidelity, contact preservation, and dynamic executability, while satisfying specified object-level spatial constraints without sacrificing contact quality. We further demonstrate the real-world applicability of CoLM through successful execution of diverse loco-manipulation tasks on a Unitree G1 humanoid. Additional results are available on our website.

1 Introduction

Humanoid loco-manipulation is more than generating plausible whole-body motion. A humanoid must coordinate locomotion and contact to move an object to a desired pose or along a prescribed

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

path. This calls for controllability at both the semantic and geometric levels: language specifies what interaction to perform, while object-level spatial constraints specify where the object should go and how it should move. Additionally, the humanoid must preserve the robot–object contacts required by the interaction throughout the motion. However, obtaining such motion remains a major bottleneck. Teleoperation (He et al., 2024; Ben et al., 2025; Ze et al., 2025) and motion capture (Li et al., 2023; Bhatnagar et al., 2022) provide high-quality references but are difficult to scale due to the need for specialized setups and manual operation. Generative model, however, offers a scalable alternative by synthesizing diverse humanoid loco-manipulation motion from user-specified conditions.

Despite recent advances in humanoid motion generation, most existing approaches (Rempe et al., 2026; Mao et al., 2025; Xie et al., 2026b; Li et al., 2026) generate diverse body-only motions from text but do not model or control the manipulated object. Prior human–object interaction methods (Peng et al., 2025; Diller & Dai, 2024; Xu et al., 2023) bring objects into the generation process, but adapting their outputs to humanoid robots introduces a substantial embodiment gap. Consequently, generating humanoid loco-manipulation that follows both task descriptions and specified object motion while preserving physical interaction remains challenging.

We present **CoLM**, a framework for generating and executing humanoid loco-manipulation from language and object-level spatial constraints (Fig. 1). Given a text description, object geometry, and the initial states of the humanoid and object, CoLM jointly generates humanoid and object motion. In addition to language, it supports spatial control from an object target pose to a complete trajectory, enforced as hard spatial constraint. A goal-conditioned RL policy then tracks the generated motion for physical execution. To provide training data at scale, we retarget six public human–object interaction datasets to humanoid robot and introduce *contact-guided refinement* to recover contacts lost during body-only retargeting, producing around 10k humanoid loco-manipulation trajectories. To preserve interaction geometry during both generation and execution, we introduce an *interaction field*, a dense representation of robot–object spatial relations. During generation, the field is jointly predicted with motion and used to enforce geometric consistency. During execution, it defines a dense tracking reward that encourages the robot to preserve the reference interaction geometry. Experiments show that CoLM generates humanoid loco-manipulation motions with more accurate object contact than existing baselines, satisfies specified object spatial constraints while maintaining contact, and improves physical execution through interaction-aware tracking. We further demonstrate the successful deployment of CoLM on a real Unitree G1 robot.

In summary, our contributions are threefold:

- We introduce CoLM, a unified framework for language-conditioned humanoid loco-manipulation generation with explicit object-level spatial control and physical robot execution.
- To support scalable learning of contact-rich interactions, we build a 10K-scale humanoid loco-manipulation dataset using a retargeting pipeline with explicit contact-guided refinement.
- We introduce interaction field, which explicitly models humanoid-object spatial relations and improves both motion generation and tracking when used as a shared interaction objective.

2 RELATED WORKS

Method	Emb.	Object	Text	Object Control	Constraint
CHOIS (2024)	Human	✓	✓	✓	Soft
DreamControl (2025)	Human	✗	✓	✗	✗
Kimodo (2026)	Robot	✗	✓	✗	✗
OMG (2026)	Robot	✗	✓	✗	✗
GRAIL (2026a)	Robot	✓	✓	✗	✗
VLK (2026)	Robot	✓	✓	✗	✗
CoLM	**Robot**	✓	✓	✓	**Hard**

Table 1: Comparison of representative methods.

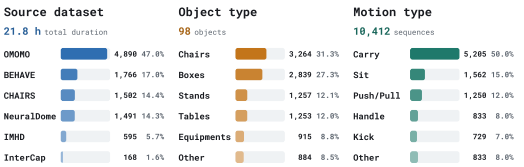


Figure 2: Distribution of our curated dataset.

Humanoid Loco-Manipulation Early humanoid loco-manipulation methods address individual tasks through reward engineering (Liu et al., 2025; Dao et al., 2024), or use specialized architectures such as skill blending and hierarchical task policies (Kuang et al., 2025; Zhang et al., 2025). More recent approaches formulate loco-manipulation as a motion-tracking problem, training policies to jointly follow reference humanoid and object trajectories and thereby reducing the need for task-specific reward design (Weng et al., 2025; Zhao et al., 2025; Xie et al., 2026a; Zhao et al., 2026). This paradigm, however, relies on suitable reference motions, which are typically acquired through teleoperation (He et al., 2024; Ze et al., 2025), motion capture (Li et al., 2023; Bhatnagar et al.,

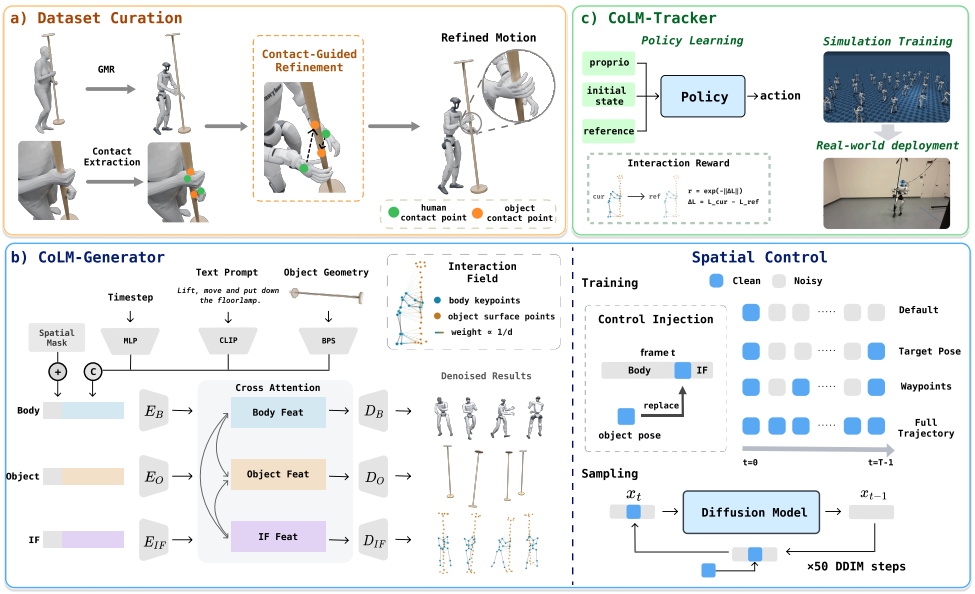


Figure 3: Overview of CoLM, including humanoid loco-manipulation dataset curation, interaction-aware controllable motion generation, and whole-body tracking.

2022), or video reconstruction (Weng et al., 2025; Xie et al., 2026a) and remain costly and difficult to obtain at scale. In contrast, CoLM directly generates humanoid loco-manipulation data from text and explicit object-level spatial constraints, providing a scalable data source for physical execution.

Motion and Interaction Generation. Text-to-motion models (Tevet et al., 2022; Chen et al., 2023; Zhang et al., 2024) synthesize human motion from language but typically omit objects and their interactions. Human-object interaction generation methods, such as CHOIS (Li et al., 2024), jointly model human and object motion but are designed for human embodiments. Recent works extend generation to humanoids: DreamControl (Kalaria et al., 2025) controls human joints before retargeting the generated motion to humanoid robot, whereas Kimodo (Rempe et al., 2026) directly generates motion on the robot embodiment. However, these methods impose spatial control on body motion without explicitly modeling or controlling the manipulated object. In contrast, CoLM jointly generates humanoid and object motion under explicit object-level spatial constraints (Table 1), supporting control through target poses, intermediate waypoints, and full trajectories.

3 Method

3.1 Dataset Curation

Our dataset is constructed from six public human–object interaction datasets (Li et al., 2023; Bhatnagar et al., 2022; Jiang et al., 2023; Zhang et al., 2023; Zhao et al., 2024; Huang et al., 2022). Each clip consists of paired human and object motion trajectories $\{(\mathbf{h}_t, \mathbf{m}_t)\}_{t=1}^{T}$, together with the corresponding object mesh $\mathcal{M}$. To convert these clips into humanoid loco-manipulation trajectories $\{(\mathbf{r}_t, \mathbf{o}_t)\}_{t=1}^{T}$, we first retarget the human motion $\mathbf{h}_t$ to an initial robot motion $\mathbf{b}_t$ (Araujo et al., 2025), as shown in Fig. 3. Since body-only retargeting does not accurately preserve physical contacts, we subsequently apply **contact-guided refinement** using contacts extracted from the source data. Specifically, we first adjust the object motion $\mathbf{m}_t$ to align with the original contact locations, yielding $\mathbf{o}_t$. We then refine $\mathbf{b}_t$ to recover these contacts while preserving motion fidelity. The optimization further encourages temporal smoothness and stable foot placement while enforcing joint-limit and nonpenetration constraints, ultimately producing $\mathbf{r}_t$. Fig. 2 shows the dataset distribution across source datasets, object types, and motion types. More details are provided in Appendix A.2.

3.2 CoLM-Generator

Using the curated trajectories, CoLM-Generator jointly models robot motion, object motion, and their interaction geometry. Conditioned on a text prompt and object geometry, it generates humanoid loco-manipulation trajectories while optionally satisfying user-specified spatial constraints.

3.2.1 MOTION REPRESENTATION

At each sequence timestep t, we represent the motion as $\mathbf{x}_t = [\mathbf{r}_t \mid \mathbf{o}_t \mid \mathbf{L}_t]$. Here, $\mathbf{r}_t$ encodes the robot motion, including the root pose, joint poses, joint velocities, and binary foot-contact. $\mathbf{o}_t$ encodes the object pose and velocity and $\mathbf{L}_t$ encodes the interaction field.

Interaction Field. Although interaction geometry is implicit in the motion trajectories, we encode it explicitly as a dense interaction field $\mathbf{L}_t$ to provide direct supervision for their spatial coupling, and facilitate the learning of coordinated robot–object motion. Let $\mathbf{p}_t \in \mathbb{R}^{K\times 3}$ denote robot body keypoints in the object-centric frame, and let $\mathbf{q} \in \mathbb{R}^{M\times 3}$ denote farthest-point samples of the object mesh $\mathcal{M}$. We stack $\mathbf{p}_t$ and $\mathbf{q}$ as $\mathbf{V}_t \in \mathbb{R}^{(K+M)\times 3}$, whose i-th row is $\mathbf{v}_{t,i}$, and treat the K+M points as nodes of a fully connected graph and assign normalized inverse-distance weights:

$$d_{t,ij} = \|\mathbf{v}_{t,i} - \mathbf{v}_{t,j}\|_2, \qquad W_{t,ij} = \frac{d_{t,ij}^{-1}}{\sum_{\ell\neq i} d_{t,i\ell}^{-1}} \quad \text{for } i \neq j, \qquad W_{t,ii} = 0. \tag{1}$$

The interaction field is then defined as:

$$\mathbf{L}_t = (\mathbf{I} - \mathbf{W}_t)\mathbf{V}_t = \mathbf{V}_t - \mathbf{W}_t\mathbf{V}_t \in \mathbb{R}^{(K+M)\times 3}. \tag{2}$$

Each row of $\mathbf{L}_t$ represents the offset of a robot keypoint or object surface point from its surrounding spatial configuration. The inverse-distance weighting emphasizes nearby geometry, causing robot–object relations to change smoothly as the robot approaches and contacts the object. This provides a dense interaction signal beyond binary contact states. We then flatten $\mathbf{L}_t$ into a feature vector.

3.2.2 MODEL ARCHITECTURE

Multi-Stream Transformer. CoLM-Generator adopts a multi-stream Transformer (Vaswani et al., 2017) with separate streams for robot motion $\mathbf{r}_t$, object motion $\mathbf{o}_t$, and interaction field $\mathbf{L}_t$. We prepend three conditioning tokens to each stream: the diffusion step, object geometry, and text prompt. The object geometry is encoded using a Basis Point Set representation (Prokudin et al., 2019), while the text prompt is encoded using CLIP (Radford et al., 2021). Each stream first processes its modality independently. A three-way cross-attention module then enables information exchange among the robot, object, and interaction streams. By explicitly encoding the relative geometry between the robot and object, the interaction stream provides interaction-aware features that guide the two motion streams toward geometrically consistent robot-object interactions.

Training Objective. We train CoLM-Generator with the $\mathbf{x}_0$-prediction formulation of DDPM (Ho et al., 2020). We first apply a component-wise reconstruction loss $\mathcal{L}_{\text{rec}}$ to supervise the predicted robot motion, object motion, and interaction field. However, component-wise reconstruction alone doesn't ensure consistent interaction geometry. We thus apply forward kinematics to the predicted robot motion, recompute the interaction field from the predicted robot and object motions, and align its body-keypoint features with the ground truth:

$$\mathcal{L}_{\text{cons}} = \sum_{t=1}^{T}\sum_{k=1}^{K} w_{t,k} \left\| \mathbf{L}_{t,k}^{\text{fk}}(\hat{\mathbf{x}}_0) - \mathbf{L}_{t,k}^{\text{gt}} \right\|_2^2, \qquad w_{t,k} = \frac{\exp\left(-d_{t,k}^{\text{gt}}/\tau\right)}{\sum_{j=1}^{K}\exp\left(-d_{t,j}^{\text{gt}}/\tau\right)}. \tag{3}$$

Here, $\mathbf{L}_{t,k}^{\text{fk}}$ and $\mathbf{L}_{t,k}^{\text{gt}}$ denote the recomputed and ground-truth interaction field for body keypoint k at timestep t. $w_{t,k}$ is a softmax over the distance $d_{t,k}^{\text{gt}}$ from keypoint k to its nearest object-surface sample, so the loss concentrates on keypoints near contact. The final objective is $\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{cons}}$.

3.2.3 SPATIAL CONTROL

Beyond text-conditioned generation, practical tasks may specify how the object should move. CoLM-Generator supports three forms of spatial control: a full object trajectory, a target object pose, and a set of intermediate waypoints. We unify them through masked conditioning. The full-trajectory mode conditions the object pose at every frame, whereas the target and waypoint modes condition it only at the specified frames. During training, entries of the noisy sequence $\mathbf{x}_t$ selected by a binary mask $\mathbf{m}$ are replaced with their clean values from $\mathbf{x}_0$. In all modes, the complete first frame is conditioned to specify the initial state. The model then predicts the remaining robot motion, object motion, and interaction field. At inference time, we reimpose the constraints before each network evaluation and after each denoising update:

$$\mathbf{x}_t \leftarrow \mathbf{m} \odot \mathbf{x}^{\text{cond}} + (1 - \mathbf{m}) \odot \mathbf{x}_t, \tag{4}$$

where $\mathbf{x}^{\text{cond}}$ contains the clean first-frame features and specified object poses. For the target and waypoint modes, we additionally apply endpoint gradient guidance to shape the approach toward the final pose. Hard conditioning ensures that the generated kinematic reference exactly satisfies the specified poses, while guidance shapes the motion between them. More details are in Appendix B.5.

3.3 CoLM-Tracker

While CoLM-Generator produces reference data, physical execution requires a policy that tracks them under robot and object dynamics. We therefore train an interaction-aware whole-body tracking policy with an asymmetric actor-critic architecture optimized using PPO (Schulman et al., 2017). Standard robot and object tracking rewards supervise the two trajectories separately but do not explicitly preserve their relative geometry. We address this with an additional interaction reward.

Interaction Reward. We define the interaction reward using the same interaction field employed by CoLM-Generator (Eq. 2). At each timestep t, we compute $\mathbf{L}_t^{\text{cur}}$ from the simulated state and $\mathbf{L}_t^{\text{ref}}$ from the reference trajectory:

$$r_t^{\text{IF}} = \exp\left(-\frac{1}{\sigma}\sum_{i=1}^{K+M}\alpha_i \left\|\mathbf{L}_{t,i}^{\text{cur}} - \mathbf{L}_{t,i}^{\text{ref}}\right\|_2\right), \tag{5}$$

where α_i applies the same distance-based softmax weighting as in $\mathcal{L}_{\text{cons}}$, computed from the reference keypoint-to-anchor distances, and σ controls the reward sensitivity. This reward encourages the tracker to preserve the reference robot–object geometry throughout execution. Additional policy and training details are provided in Appendix C.

4 Experiments

We evaluate CoLM from data construction to physical execution. We first assess motion quality in the retargeted dataset (Sec. 4.1), then evaluate the generator's semantic fidelity and executability (Sec. 4.2.1) and its ability to satisfy object-level spatial constraints (Sec. 4.2.2). We further evaluate loco-manipulation tracking in simulation (Sec. 4.3) and demonstrate the complete framework on a real humanoid (Sec. 4.4). Finally, ablations examine the contributions of interaction-field prediction, geometric consistency, and the interaction reward (Sec. 4.5).

4.1 Retargeting Quality Evaluation

Method	Jitter (10^2 m/s^3) ↓	Contact F1 (%) ↑	Gap (cm) ↓
OmniRetarget	2.45	43.1	6.60
GMR	0.85	28.8	10.26
Ours	**0.47**	**82.5**	**4.42**

Table 2: **Quantitative results** of retargeting. Best results are in **bold**. Our method best preserves source contacts, with the highest Contact F1, smallest contact gap, and lowest jitter.

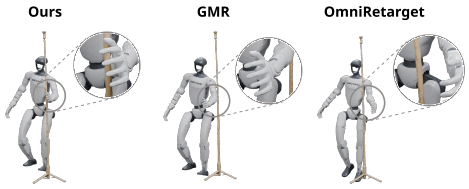


Figure 4: **Qualitative results** of retargeting. Our method preserves accurate contact.

We evaluate whether contact-guided refinement recovers source contacts while preserving motion smoothness. All methods are compared on the 9,130 clips for which OmniRetarget returns feasible solutions (Appendix E.2).

Baselines. We compare our method against existing methods. GMR (Araujo et al., 2025) is a general retargeting method that optimizes body motion without explicitly modeling the object. OmniRetarget (Yang et al., 2025) explicitly preserves human-object spatial relations during retargeting.

Metrics. Jitter measures the third-order smoothness of joint positions. Contact F1 compares binary contact labels for each frame and body region, using human-object proximity for source labels and robot-link–object proximity for retargeted labels. We report the harmonic mean of precision and recall. Gap measures link-to-object surface distance where source contacts are lost.

Results. Table 2 shows that contact-guided refinement recovers accurate contact while improving motion smoothness. Contact F1 reaches 82.5%, exceeding the interaction-aware OmniRetarget by 39.4 percentage points and GMR by 53.7 points. The contact gap decreases to 4.42 cm from 6.60 cm and 10.26 cm, respectively, while jitter is lowest at 0.47. These gains provide more faithful contact references for downstream training without sacrificing motion smoothness. Fig. 4 illustrates that while GMR and OmniRetarget lose contact, our method preserves it.

Table 3: **Quantitative results and ablations of loco-manipulation generation**. IF denotes the Interaction Field generation branch and $\mathcal{L}_{\text{cons}}$ the geometric consistency loss. FID is multiplied by 100 for readability. Foot Skate is in cm/s and Jitter is in 10^2 m/s^3. Contact F1, Joint Limit and Tracking SR are in %. Best and second best results are highlighted.

Method	Semantic Fidelity				Kinematic Quality			Dynamic Executability	
	FID ↓	R@3 ↑	MM-Dist ↓	Diversity →	Contact F1 ↑	Foot Skate ↓	Jitter ↓	Joint Limit ↓	Tracking SR ↑
Ground Truth	0.10	0.709	0.690	1.40	100.0	3.45	0.45	0.70	91.4
DreamControl	4.89	0.545	0.760	1.27	28.5	8.98	1.58	1.69	35.8
Kimodo	3.79	0.661	0.756	1.33	55.9	5.81	0.43	2.59	68.5
CHOIS + Retarget	3.56	0.583	0.775	1.34	51.2	7.52	0.29	**0.28**	64.3
CHOIS (G1)	3.14	0.659	0.734	1.37	49.3	8.21	0.57	0.95	59.3
Ours (Human) + Retarget	2.83	0.667	0.703	1.37	68.7	**4.13**	**0.17**	0.33	84.3
Ours w/o IF	2.92	0.676	0.715	1.38	63.2	4.42	0.40	0.84	75.3
Ours w/o $\mathcal{L}_{\text{cons}}$	2.76	0.680	0.705	1.36	62.0	4.31	0.39	0.81	76.4
Ours w/o IF & $\mathcal{L}_{\text{cons}}$	3.24	0.665	0.726	1.37	56.0	4.49	0.42	0.91	71.1
Ours	**1.80**	**0.687**	**0.697**	**1.39**	**71.2**	4.34	0.41	0.82	**89.9**

4.2 CoLM-Generator Evaluation

4.2.1 Text-Conditioned Generation

We evaluate on 597 of the 600 test sequences after filtering short clips, covering diverse object and motion categories. All methods share the train/test split (Appendix E.3).

Baselines. DreamControl (Kalaria et al., 2025) generates human motion under joint-level control before retargeting to G1; Kimodo (Rempe et al., 2026) directly generates robot motion with end-effector control. For both methods, we extract end-effector poses from ground-truth motions as control conditions and pair the generated body motions with ground-truth object trajectories for evaluation. CHOIS (Li et al., 2024) jointly generates human and object motion from object waypoints. CHOIS + Retarget applies our contact-guided retargeting pipeline to the generated human motion; CHOIS (G1) directly generates robot and object motion on the same embodiment as our model. Ours (Human) + Retarget uses our generator on a human skeleton before retargeting, testing the embodiment gap against direct robot-space generation.

Metrics. We evaluate all the methods from three aspects. Details are provided in Appendix E.1.

Semantic fidelity. We evaluate whether the generated motion matches the text description and follows the distribution of real motions. We use a TMR (Petrovich et al., 2023) evaluator trained on our training split and report FID between generated and reference motion embeddings, R@3 for text-motion retrieval, MM-Dist between motion and text embeddings, and Diversity.

Kinematic quality. We evaluate whether the generated motion preserves meaningful humanoid-object interaction geometry. We report Contact F1, Foot Skate, and Jitter. Contact F1 compares generated contacts with robot reference contacts; Jitter measures third-order positional variation. Foot Skate measures foot speed over frames labelled as foot contact, quantifying unintended sliding.

Dynamic executability. We evaluate whether generated motions can be executed in a physical environment. Joint Limit measures the fraction of joint-angle samples outside the URDF limits, while Tracking SR measures whether a physics-based tracker successfully follows each generated sequence. Tracking SR uses the same 100 clips and success criterion as Sec. 4.3.

Results. CoLM-Generator improves semantic fidelity, contact preservation, and executability together (Table 3). Compared with CHOIS (G1) on the same embodiment, it reduces FID by 43% (3.14 to 1.80) and improves Contact F1 and Tracking SR by 21.9 and 30.6 percentage points, reaching 71.2% and 89.9%. It also achieves the best R@3 and MM-Dist among generated methods, with near-ground-truth diversity (1.39 vs. 1.40). Direct robot-space generation improves Contact F1 by 2.5 points and SR by 5.6 points over Ours (Human) + Retarget, bringing it within 1.5 points of ground truth. Foot skating remains low, although retargeting-based methods achieve fewer joint-limit violations. The higher SR despite this trade-off suggests that kinematic validity alone does not determine interaction executability. Compared with Kimodo, Contact F1 and SR improve by 15.3 and 21.4 points despite its ground-truth end-effector conditions. This gap highlights the value of jointly modeling body-object interaction beyond specifying body-level control signals. Fig. 5 illustrates improved contact preservation. In *Pull Chair*, CHOIS + Retarget, CHOIS (G1), and Kimodo lose hand contact; in *Carry Trashcan*, Kimodo places the object against the torso instead of maintaining hand contact. DreamControl totally fails to preserve interaction across all the tasks. Our method maintains the intended hand-object contacts in these examples.

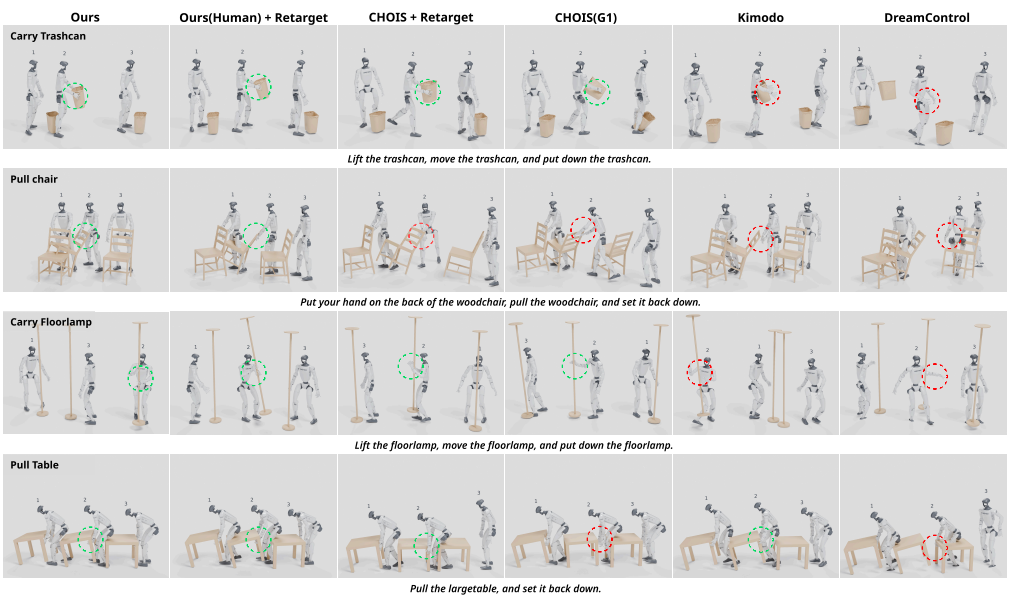


Figure 5: **Qualitative results** of CoLM-Generator and other methods. Green dashed circles denote correct contacts, while red dashed circles denote lost contacts or wrong contacts.

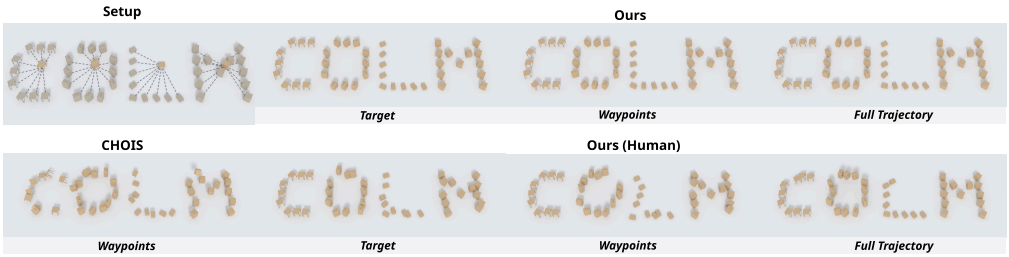


Figure 6: **Qualitative results** of spatial control ability. The setup illustrates the initial object positions and the target pattern formed by moving the objects to desired locations. Our method accurately fulfills these spatial constraints, forming exactly the target pattern in all three control modes.

4.2.2 Spatial Control

We evaluate whether CoLM-Generator can satisfy spatial constraints while preserving interaction.

Baselines. Since DreamControl (Kalaria et al., 2025) and Kimodo (Rempe et al., 2026) do not support object-level spatial control, we include CHOIS + Retarget, CHOIS (G1), and the human-skeleton variant of our method. Ours (Human) + Retarget tests the effect of imposing spatial constraints before retargeting, while CHOIS (G1) compares object-level control on the same robot embodiment without subsequent retargeting.

Metrics. *Control accuracy* is measured by translation and orientation errors. We report both final-frame errors, $T_{\text{obj}}^{\text{end}}$ and $O_{\text{obj}}^{\text{end}}$, and sequence-level errors, T_{obj} and O_{obj}, averaged over all frames. Translation error is measured by Euclidean distance, while orientation error is measured by geodesic rotation distance. *Interaction quality* is evaluated using Contact F1, Foot Skate, and Jitter, following the same definitions as in Sec. 4.2.1.

Results. CoLM-Generator satisfies hard object-level spatial constraints while maintaining contact (Fig. 6 and Table 4). Contact F1 remains at 70.1–72.3% across control modes, close to the default 71.2% and above all compared methods. Conditioned poses are fixed during denoising, giving zero final-pose errors in all modes and zero sequence-level errors under full-trajectory control; these guarantees apply to generated references. In contrast, the human-skeleton variant incurs 13.23 cm final translation error after retargeting even with full-trajectory control. Direct robot-space generation also improves Contact F1 over this variant by 4.8, 1.0, and 3.0 points under target, waypoint, and full-trajectory control, respectively, combining accurate spatial control with better interaction qual-

Table 4: **Quantitative results of motion generation with spatial control**. $T_{\text{obj}}^{\text{end}}$ and T_{obj} are in cm, $O_{\text{obj}}^{\text{end}}$ and O_{obj} are in degrees. Best and second best results are highlighted.

Method	Mode	Control Accuracy				Kinematic Quality		
		$T_{\text{obj}}^{\text{end}} \downarrow$	$O_{\text{obj}}^{\text{end}} \downarrow$	$T_{\text{obj}} \downarrow$	$O_{\text{obj}} \downarrow$	Contact F1 ↑	Foot Skate ↓	Jitter ↓
Ground Truth	–	0	0	0	0	100.0	3.45	0.45
CHOIS + Retarget	waypoints	17.37	46.70	20.80	41.23	51.2	7.52	0.29
CHOIS (G1)	waypoints	8.12	12.15	19.45	21.57	46.3	9.93	0.56
Ours (Human) + Retarget	target	20.90	1.59	28.88	35.49	67.5	**3.29**	**0.19**
	waypoints	20.70	1.59	25.34	28.83	69.1	3.32	0.20
	full trajectory	13.23	1.56	11.02	1.56	68.8	3.43	0.20
Ours	target	**0.00**	**0.00**	24.07	32.00	**72.3**	4.21	0.45
	waypoints	**0.00**	**0.00**	19.39	24.98	70.1	4.28	0.41
	full trajectory	**0.00**	**0.00**	**0.00**	**0.00**	71.8	4.41	0.48

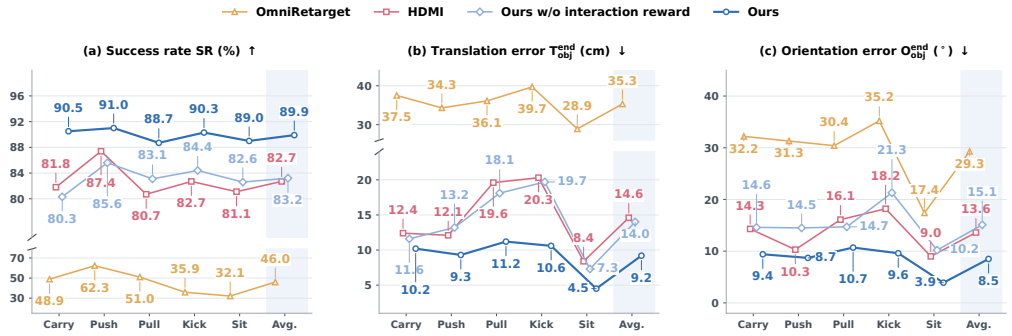


Figure 7: **Tracker comparison and interaction reward ablation.** Results are evaluated on the same 100 generated reference clips across five interaction categories with 1024 environments.

ity. Adding waypoints reduces our sequence-level translation and orientation errors from 24.07 cm and 32.00° to 19.39 cm and 24.98°. CHOIS (G1) has similar sequence-level errors but misses the endpoint by 8.12 cm and 12.15°, with 46.3% Contact F1.

4.3 CoLM-Tracker Evaluation

We evaluate 100 CoLM-Generator references from the test set: 20 each for carry, push, pull, kick, and sit. They cover 48 objects from all six datasets and two-hand, one-hand, foot, and pelvis contacts. Each tracker trains one policy per clip; we report category-wise and overall averages.

Baselines. We compare with HDMI (Weng et al., 2025) and OmniRetarget (Yang et al., 2025). OmniRetarget follows reference body and object trajectories through standard motion tracking rewards. HDMI jointly introduces a reward to encourage contacts.

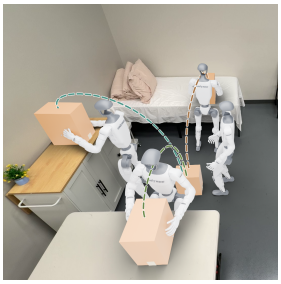
Metrics. We report success rate (SR) and final object pose errors. SR is the percentage of rollouts completed without termination and with mean object position error at most 20 cm. Final translation and orientation errors, $T_{\text{obj}}^{\text{end}}$ and $O_{\text{obj}}^{\text{end}}$, compare the rollout-end object pose with the intended final pose (Appendix E.1).

Results. CoLM-Tracker improves both execution success and final object error (Fig. 7), leading on all metrics across all categories. Overall SR reaches 89.9%, exceeding HDMI and OmniRetarget by 7.2 and 43.9 points, with gains of 3.6–8.7 points over HDMI across categories. Average final translation and orientation errors decrease to 9.2 cm and 8.5°, from HDMI's 14.6 cm and 13.6° and OmniRetarget's 35.3 cm and 29.3°, reducing both errors by about 37% relative to HDMI.

4.4 Real-World Deployment

We deploy CoLM on a Unitree G1 to evaluate whether the framework can generate and execute diverse loco-manipulation tasks from language and spatial constraints in the real world.

Setup. We design four tasks: *Carry Box*, *Carry Tripod*, *Kick Box*, and *Sit Down*, covering hand, foot, and pelvis contact. We estimate the initial object pose using FoundationPose (Wen et al., 2024),



Task	SR
Carry Box To Desk	100%
Carry Box To Bed	90%
Carry Box To Cabinet	90%
Carry Tripod	100%
Kick Box	90%
Sit Down	85%

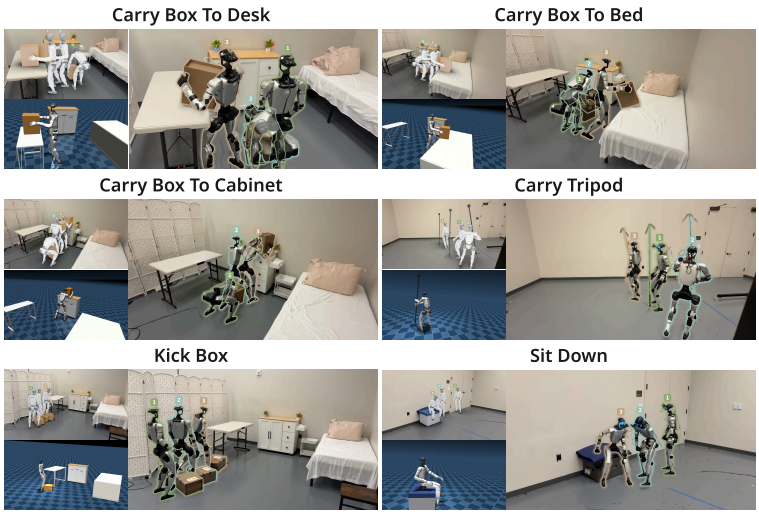


Figure 8: **Real-world deployment of CoLM.** Left: the three commanded target box poses, and the success rate over 20 trials per task. Right: the four tasks executed on the real humanoid.

generate a reference, and train a tracker for deployment. In *Carry Box*, we examine the spatial control ability by commanding different target box destinations. All physical object instances are **unseen in our motion dataset** to test the generalization ability of CoLM.

Results. Fig. 8 demonstrates the generated motion and hardware execution across four tasks. 111 of 120 trials succeed (92.5%), with 20 trials per setting and at least 85% SR in each, demonstrating the real-world applicability of CoLM. Deployment details are provided in Appendix D.

4.5 Ablation Study

We test the shared interaction objective by ablating field prediction and geometric consistency during generation, and the interaction reward during physical execution.

Generator interaction modeling. We separately remove the IF prediction stream with its reconstruction term, the geometric consistency loss $\mathcal{L}_{\text{cons}}$, or both. Without IF prediction, $\mathcal{L}_{\text{cons}}$ still supervises the field recomputed from body and object trajectories through forward kinematics. Metrics follow Sec. 4.2.1. Table 3 shows that both components contribute to interaction quality: either removal reduces Contact F1 and Tracking SR; removing both is worst (56.0% and 71.1% vs. 71.2% and 89.9%). Without $\mathcal{L}_{\text{cons}}$, foot skating, jitter, and joint-limit violations slightly improve, but Contact F1 and SR fall to 62.0% and 76.4%. Thus, smoother motions with fewer joint-limit violations do not necessarily yield successful interaction; both forms of interaction supervision improve execution by explicitly modeling robot-object geometry.

Tracker interaction reward. Removing r_t^{IF} while retaining body and object tracking rewards reduces SR from 89.9% to 83.2% on the same 100 clips (Fig. 7). Translation and orientation errors rise from 9.2 cm and 8.5° to 14.0 cm and 15.1°. The reward benefits all five categories, especially carry (10.2-point SR gain). By rewarding reference body-object relative geometry, it complements separate trajectory tracking and encourages coordinated motion.

5 Conclusion and Limitations

We presented CoLM for generating and executing humanoid loco-manipulation from language and spatial constraints. Contact-guided retargeting supplies training data, while an interaction field supports generation and tracking. Experiments demonstrate improved semantic fidelity, contact preservation, and execution success. Real-world demonstrations on the Unitree G1 further validate execution across diverse tasks.

Our framework has three main limitations. First, the tracker cannot reliably correct inaccurate contacts or physical infeasibility in generated reference motions. Second, the generator generalizes mainly within seen object and motion categories and degrades on entirely unseen ones; addressing both requires integrating execution feedback into generation and broadening training coverage. Third, the framework does not support dexterous manipulation requiring fine-grained finger control.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing our method, primarily by developing supporting scripts for tasks such as data processing. We have not used generative AI tools for generating synthetic datasets, developing theoretical models or conceptual frameworks, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, translation, cleaning or reformatting datasets, or interpreting results. The remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for modifying figures, language polishing and sourcing for information.

All AI-assisted work was verified for correctness by the authors.We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, and no new data were collected from human participants. All human motion data used in this work come from publicly available datasets and are used in accordance with their respective licenses. The data are already de-identified by the original providers and consist of skeletal or parametric body representations containing no personally identifiable information. We do not redistribute any data whose license prohibits redistribution.

All real-world experiments were conducted in a controlled laboratory environment under continuous human supervision, with an emergency stop available at all times. No humans were present within the robot workspace during policy execution, and all policies were validated in simulation before deployment on hardware.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide code covering the complete pipeline, from data processing and retargeting to motion generation, policy training, evaluation, and deployment, at `https://anonymous.4open.science/r/CoLM-Code/`. The appendices complement this code with detailed descriptions of each stage and the experimental settings. Appendix A documents the data sources, annotations, train/test split, and contact-guided retargeting procedure. Appendices B and C detail the generator and tracker, including motion preprocessing, model architectures, training objectives, hyperparameters, and sampling procedures. Appendix E specifies the evaluation metrics, baseline adaptations, spatial control conditions, and tracking protocols used in our experiments.

REFERENCES

Joao Pedro Araujo, Yanjie Ze, Pei Xu, Jiajun Wu, and C Karen Liu. Retargeting matters: General motion retargeting for humanoid motion tracking. *arXiv preprint arXiv:2510.02252*, 2025.

Qingwei Ben, Feiyu Jia, Jia Zeng, Junting Dong, Dahua Lin, and Jiangmiao Pang. Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit. *arXiv preprint arXiv:2502.13013*, 2025.

Bharat Lal Bhatnagar, Xianghui Xie, Ilya A Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Behave: Dataset and method for tracking human object interactions. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15914–15925. IEEE, 2022.

Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18000–18010. IEEE, 2023.

Jeremy Dao, Helei Duan, and Alan Fern. Sim-to-real learning for humanoid box loco-manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16930–16936. IEEE, 2024.

Christian Diller and Angela Dai. Cg-hoi: Contact-guided 3d human-object interaction generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19888–19901. IEEE, 2024.

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 *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5142–5151. IEEE, 2022.

Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

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.

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

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

Siqiao Huang, Kun-Ying Lee, Dongming Qiao, Guanqi He, Zhenyu Wang, Yitang Li, Shaoting Zhu, and Hang Zhao. Omg: Omni-modal motion generation for generalist humanoid control, 2026. URL https://arxiv.org/abs/2606.10340.

Yinghao Huang, Omid Taheri, Michael J Black, and Dimitrios Tzionas. Intercap: Joint markerless 3d tracking of humans and objects in interaction. In *DAGM German Conference on Pattern Recognition*, pp. 281–299. Springer, 2022.

Nan Jiang, Tengyu Liu, Zhexuan Cao, Jieming Cui, Zhiyuan Zhang, Yixin Chen, He Wang, Yixin Zhu, and Siyuan Huang. Full-body articulated human-object interaction. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9331–9342. IEEE, 2023.

Dvij Kalaria, Sudarshan S Harithas, Pushkal Katara, Sangkyung Kwak, Sarthak Bhagat, Shankar Sastry, Srinath Sridhar, Sai Vemprala, Ashish Kapoor, and Jonathan Chung-Kuan Huang. Dream-control: Human-inspired whole-body humanoid control for scene interaction via guided diffusion. *arXiv preprint arXiv:2509.14353*, 2025.

Yuxuan Kuang, Haoran Geng, Amine Elhafsi, Tan-Dzung Do, Pieter Abbeel, Jitendra Malik, Marco Pavone, and Yue Wang. Skillblender: Towards versatile humanoid whole-body loco-manipulation via skill blending. *arXiv preprint arXiv:2506.09366*, 2025.

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

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*, pp. 54–72. Springer, 2024.

Peng Li, Zihan Zhuang, Yangfan Gao, Yi Dong, Sixian Li, Changhao Jiang, Shihan Dou, Zhiheng Xi, Enyu Zhou, Jixuan Huang, et al. From-w1: Towards general humanoid whole-body control with language instructions. *arXiv preprint arXiv:2601.12799*, 2026.

Fukang Liu, Zhaoyuan Gu, Yilin Cai, Ziyi Zhou, Hyunyoung Jung, Jaehwi Jang, Shijie Zhao, Sehoon Ha, Yue Chen, Danfei Xu, et al. Opt2skill: Imitating dynamically-feasible whole-body trajectories for versatile humanoid loco-manipulation. *IEEE Robotics and Automation Letters*, 2025.

Jiageng Mao, Siheng Zhao, Siqi Song, Chuye Hong, Tianheng Shi, Junjie Ye, Mingtong Zhang, Haoran Geng, Jitendra Malik, Vitor Guizilini, et al. Universal humanoid robot pose learning from internet human videos. In *2025 IEEE-RAS 24th International Conference on Humanoid Robots (Humanoids)*, pp. 1–8. IEEE, 2025.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PmLR, 2021.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10975–10985, 2019.

Xiaogang Peng, Yiming Xie, Zizhao Wu, Varun Jampani, Deqing Sun, and Huaizu Jiang. Hoi-diff: Text-driven synthesis of 3d human-object interactions using diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2869–2879. IEEE, 2025.

Mathis Petrovich, Michael J Black, and Gül Varol. Tmr: Text-to-motion retrieval using contrastive 3d human motion synthesis. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9454–9463. IEEE, 2023.

Sergey Prokudin, Christoph Lassner, and Javier Romero. Efficient learning on point clouds with basis point sets. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 3072–3081. IEEE, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Davis Rempe, Mathis Petrovich, Ye Yuan, Haotian Zhang, Xue Bin Peng, Yifeng Jiang, Tingwu Wang, Umar Iqbal, David Minor, Michael de Ruyter, et al. Kimodo: Scaling controllable human motion generation. *arXiv preprint arXiv:2603.15546*, 2026.

Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad, and Stephen Boyd. Osqp: an operator splitting solver for quadratic programs: B. stellato et al. *Mathematical Programming Computation*, 12(4):637–672, 2020.

Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

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.

Yen-Jen Wang, Jiaman Li, Sirui Chen, Takara E Truong, Pei Xu, Pieter Abbeel, Rocky Duan, Koushil Sreenath, Angjoo Kanazawa, Carmelo Sferrazza, et al. Vlk: Learning humanoid loco-manipulation from synthetic interactions in reconstructed scenes. *arXiv preprint arXiv:2606.30645*, 2026.

Xinyue Wei, Minghua Liu, Zhan Ling, and Hao Su. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics (TOG)*, 41(4):1–18, 2022.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17868–17879. IEEE, 2024.

Haoyang Weng, Yitang Li, Nikhil Sobanbabu, Zihan Wang, Zhengyi Luo, Tairan He, Deva Ramanan, and Guanya Shi. Hdmi: Learning interactive humanoid whole-body control from human videos. *arXiv preprint arXiv:2509.16757*, 2025.

Tianyi Xie, Haotian Zhang, Jinhyung Park, Zi Wang, Bowen Wen, Jiefeng Li, Xueting Li, Qingwei Ben, Haoyang Weng, Yufei Ye, et al. Grail: Generating humanoid loco-manipulation from 3d assets and video priors. *arXiv preprint arXiv:2606.05160*, 2026a.

Weiji Xie, Jiakun Zheng, Jinrui Han, Jiyuan Shi, Weinan Zhang, Chenjia Bai, and Xuelong Li. Textop: Real-time interactive text-driven humanoid robot motion generation and control. *arXiv preprint arXiv:2602.07439*, 2026b.

Sirui Xu, Zhengyuan Li, Yu-Xiong Wang, and Liang-Yan Gui. Interdiff: Generating 3d human-object interactions with physics-informed diffusion. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14882–14894. IEEE, 2023.

Sirui Xu, Dongting Li, Yucheng Zhang, Xiyan Xu, Qi Long, Ziyin Wang, Yunzhi Lu, Shuchang Dong, Hezi Jiang, Akshat Gupta, et al. Interact: Advancing large-scale versatile 3d human-object interaction generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7048–7060. IEEE, 2025.

Lujie Yang, Xiaoyu Huang, Zhen Wu, Angjoo Kanazawa, Pieter Abbeel, Carmelo Sferrazza, C Karen Liu, Rocky Duan, and Guanya Shi. Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction. *arXiv preprint arXiv:2509.26633*, 2025.

Kevin Zakka, Qiayuan Liao, Brent Yi, Louis Le Lay, Koushil Sreenath, and Pieter Abbeel. mjlab: A lightweight framework for gpu-accelerated robot learning. *arXiv preprint arXiv:2601.22074*, 2026.

Yanjie Ze, Siheng Zhao, Weizhuo Wang, Angjoo Kanazawa, Rocky Duan, Pieter Abbeel, Guanya Shi, Jiajun Wu, and C Karen Liu. Twist2: Scalable, portable, and holistic humanoid data collection system. *arXiv preprint arXiv:2511.02832*, 2025.

Juze Zhang, Haimin Luo, Hongdi Yang, Xinru Xu, Qianyang Wu, Ye Shi, Jingyi Yu, Lan Xu, and Jingya Wang. Neuraldome: A neural modeling pipeline on multi-view human-object interactions. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8834–8845. IEEE, 2023.

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *IEEE transactions on pattern analysis and machine intelligence*, 46(6):4115–4128, 2024.

Yuanhang Zhang, Yifu Yuan, Prajwal Gurunath, Ishita Gupta, Shayegan Omidshafiei, Ali-akbar Agha-mohammadi, Marcell Vazquez-Chanlatte, Liam Pedersen, Tairan He, and Guanya Shi. Falcon: Learning force-adaptive humanoid loco-manipulation. *arXiv preprint arXiv:2505.06776*, 2025.

Chengfeng Zhao, Juze Zhang, Jiashen Du, Ziwei Shan, Junye Wang, Jingyi Yu, Jingya Wang, and Lan Xu. I'm hoi: Inertia-aware monocular capture of 3d human-object interactions. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 729–741. IEEE, 2024.

Siheng Zhao, Yanjie Ze, Yue Wang, C Karen Liu, Pieter Abbeel, Guanya Shi, and Rocky Duan. Resmimic: From general motion tracking to humanoid whole-body loco-manipulation via residual learning. *arXiv preprint arXiv:2510.05070*, 2025.

Siheng Zhao, Yuanhang Zhang, Ziqi Lu, Pieter Abbeel, Rocky Duan, Koushil Sreenath, Yue Wang, C. Karen Liu, and Guanya Shi. Ladderman: Learning humanoid perceptive ladder climbing, 2026. URL https://arxiv.org/abs/2606.05873.

APPENDIX

A DATASET CONSTRUCTION AND RETARGETING

A.1 DATA SOURCES, ANNOTATIONS, AND STATISTICS

Sources. We curate human-object interaction clips from six public datasets, unified by InterAct (Xu et al., 2025). The body model follows each source: SMPL-X (Pavlakos et al., 2019) for OMOMO, CHAIRS, and InterCap, and SMPL-H (Romero et al., 2022) for BEHAVE, NeuralDome, and IMHD. All clips are resampled to 30 fps and retargeted to the Unitree G1 using the procedure in Appendices A.2 and A.3. The corpus contains 10,412 sequences, 2,350,676 frames, and approximately 21.8 hours of interaction with 98 object instances.

Table 5: Per-source statistics of our curated corpus. Objects are source-specific instances; shares are based on sequence counts. Durations and mean lengths are rounded.

Source	Sequences	Share	Objects	Hours	Mean length (s)
OMOMO (Li et al., 2023)	4,890	47.0%	13	8.4	6.2
BEHAVE (Bhatnagar et al., 2022)	1,766	17.0%	18	4.1	8.3
CHAIRS (Jiang et al., 2023)	1,502	14.4%	32	3.7	8.9
NeuralDome (Zhang et al., 2023)	1,491	14.3%	20	3.8	9.1
IMHD (Zhao et al., 2024)	595	5.7%	9	1.5	9.1
InterCap (Huang et al., 2022)	168	1.6%	6	0.3	7.1
Total	10,412	100%	98	21.8	7.5

Composition. Table 5 summarizes the corpus composition. Object instances are counted separately within each source, yielding 98 instances with 86 unique object names; similarly named objects from different sources have distinct meshes.

Sequence lengths. Sequence duration averages 7.5 s, with a median of 6.7 s and a 95th percentile of 14.1 s. Generator training and evaluation exclude the 43 clips shorter than 40 frames, leaving 10,369 clips (Appendix B.1).

Text annotations. Every clip carries the text descriptions released with InterAct. For BEHAVE, CHAIRS, NeuralDome, IMHD, and InterCap, InterAct segments each capture into clips according to its annotations and provides three descriptions per clip: the original description, a paraphrase, and a shortened version. OMOMO clips follow the original segmentation and carry a single description from the OMOMO release, such as "Push the largebox, and set it back down." Each time a clip is drawn during generator training, one of its descriptions is selected uniformly at random.

Categories. The rings in Fig. 2 show sequence shares by source, object category, and motion category; their centers report corpus totals. Object categories are assigned by keywords in the object name: Chairs (chairs, stools, and sofas), Boxes (boxes, cases, bags, and trash cans), Tables (tables and desks), Stands (tripods, clothes stands, and floor lamps), Equipments (sports gear and tools such as bats, balls, pans, and dumbbells), Monitors (monitors and keyboards), and Other. Each clip receives one motion category from the main verb of its description, defined as for the train/test split below: Carry (e.g., lift, hold, walk, put, and pick), Sit (e.g., sit, lie, and lean), Handle (e.g., grab, use, swing, and rotate), Push (e.g., push and shift), Pull (pull and drag), Kick, and Other for the remaining main verbs, such as stand and bend.

Train/test split. We hold out 600 clips as a test set shared by all stages. Sampling is stratified by source, targeting a 5% share while prioritizing coverage of every object and main verb within each source. The main verb is the first verb in the part-of-speech-tagged description. We adjust source allocations to reach 600 clips while preserving this coverage. The final test set contains 244 OMOMO, 111 BEHAVE, 97 NeuralDome, 75 CHAIRS, 43 IMHD, and 30 InterCap clips.

A.2 BODY RETARGETING AND CONTACT EXTRACTION

Body retargeting. We first retarget the human motion to G1 with GMR (Araujo et al., 2025), which gives a natural body pose while keeping the original object trajectory unchanged. Let $\bar{q}_t \in \mathbb{R}^{29}$ denote the GMR joint angles at frame $t = 1, \ldots, T$, and let q_t denote the joint angles to be refined. Before optimization, we smooth the floating-base translation and orientation with a

zero-phase second-order Butterworth filter at 5 Hz, aligning quaternion signs across frames and renormalizing after filtering. We also correct the root height using the smoothed maximum foot penetration. The corrected floating-base trajectory is held fixed during joint refinement.

Contact extraction. For each frame, we compute contacts between the human and object meshes and assign them to body parts. Each human vertex is labelled once, at the first frame, by its nearest body joint, which partitions the surface into the two hands, arms, shoulders, knees, feet, and the pelvis. We then transform the human vertices into the object's local frame and query the object mesh with a KD-tree: a vertex is in contact when it lies within 3 cm of the nearest object vertex, and a body part is in contact at a frame when any of its vertices is. Let $\mathcal{C}_t$ denote the set of contacting body parts at frame t. Refinement uses the two hands, the two feet, and the pelvis, retaining contact episodes lasting at least 5 frames. For each contacting body part p, we record a per-frame object-local contact point g_t^p as the mean of the nearest object-surface vertices. We low-pass these points within each contact episode at 2.5 Hz, using the episode median when the episode is too short to filter. This preserves sliding contact while suppressing high-frequency jitter. Pelvis contacts are used for object alignment only. We denote by $x_t^p(q_t)$ the world-space position of the corresponding robot contact point under pose q_t.

A.3 Contact-Guided Refinement

Using the extracted contacts, we refine the object trajectory and robot motion in two successive steps to preserve the original interactions.

Object alignment. We first align the object with the retargeted body, keeping the recorded object orientation R_t fixed and optimizing only its translation o_t. On contact frames, the target translation aligns the object contact point with the corresponding robot body part:

$$\min_{o_t} \sum_t w_t \left\| o_t - \hat{o}_t \right\|_2^2 + w_o \sum_t \left\| o_{t-1} - 2o_t + o_{t+1} \right\|_2^2, \qquad \hat{o}_t = \frac{\sum_{p \in \mathcal{H}_t} n_p \left(x_t^p(\bar{q}_t) - R_t g_t^p \right)}{\sum_{p \in \mathcal{H}_t} n_p}. \tag{6}$$

Here $\mathcal{H}_t \subseteq \mathcal{C}_t$ is the set of hands holding the object at frame t and n_p is the total number of contact frames of hand p, so a two-handed grasp is aligned to the weighted average of its two targets and a hand that touches the object only briefly counts less than one that carries it throughout. On noncontact frames the target holds the value of the nearest contact frame in time, so frames before the first contact take the first contact's target and frames after the last contact keep the object where the hand left it. We set $w_t = 1$ on contact frames and $w_t = 0.1$ elsewhere, and $w_o = 8$. The translation objective is quadratic and is solved as a linear system independently along each coordinate axis.

For sequences with contact but no hand holding the object, such as sitting or a foot push, we instead apply a constant translation offset to the recorded object trajectory o_t^{src}:

$$o_t = o_t^{\mathrm{src}} + \Delta o, \qquad \Delta o = \frac{\sum_t \sum_{p \in \mathcal{C}_t} n_p \left[x_t^p(\bar{q}_t) - R_t g_t^p - o_t^{\mathrm{src}} \right]}{\sum_t \sum_{p \in \mathcal{C}_t} n_p}, \tag{7}$$

where n_p counts the contact frames of part p. This offset preserves the recorded object displacements while aligning the trajectory to the robot contacts. After either alignment procedure, the corresponding world-space object contact point is $c_t^p = o_t + R_t g_t^p$.

Joint refinement. With the aligned object and corrected floating-base trajectories fixed, we jointly optimize the 29 joint angles across the sequence and the nonpenetration slack variables to satisfy these contact targets through a whole-sequence optimization:

$$\min_{\{q_t\}, \{s_{t,g}\}} E_{\mathrm{con}} + E_{\mathrm{reg}} + E_{\mathrm{smo}} + E_{\mathrm{foot}} + w_s \sum_{t,g} s_{t,g}^2 \quad \text{s.t.} \quad d_{t,g} \geq -\tau_g - s_{t,g}, \qquad s_{t,g} \geq 0. \tag{8}$$

$$E_{\mathrm{con}} = w_c \sum_t \sum_{p \in \mathcal{C}_t^{\mathrm{limb}}} \left\| x_t^p(q_t) - c_t^p \right\|^2, \quad E_{\mathrm{smo}} = w_a \sum_t \left\| q_{t-1} - 2q_t + q_{t+1} \right\|^2,$$

$$E_{\mathrm{reg}} = \sum_t \left\| q_t - \bar{q}_t \right\|_{\Lambda_t}^2, \qquad E_{\mathrm{foot}} = w_f \sum_f \sum_e \sum_{t \in \mathcal{S}_{f,e}} \left\| y_t^f - a^{f,e} \right\|^2.$$

Table 6: Weights used for contact-guided retargeting. The diagonal entries of Λ_t specify joint-dependent reference penalties.

Stage	Term or joint group	Weight
Object alignment	Contact / noncontact target (w_t)	$1/0.1$
	Temporal smoothness (w_o)	8
Joint refinement	Contact (w_c)	1
	Temporal smoothness (w_a)	40
	Stance-foot stabilization (w_f)	200
	Nonpenetration slack (w_s)	10^3
Reference (Λ_t)	Waist / torso	50
	Noncontacting joints and grasping-hand wrists	2.0
	Contacting-arm shoulder and elbow joints	0.02
	Stance-foot leg joints	0.1

Objective weights and regularization. E_{con} penalizes deviations from the target contact points of the hands and feet, indexed by $\mathcal{C}_t^{\text{limb}}$; pelvis contacts are excluded from this joint-refinement term. E_{smo} penalizes second-order joint differences, and E_{foot} stabilizes stance feet. We use $w_c = 1$, $w_a = 40$, and $w_f = 200$, with slack penalty $w_s = 10^3$. The reference term uses $\|v\|^2_{\Lambda_t} = v^\top \Lambda_t v$, where Λ_t is diagonal and assigns joint-dependent penalties for departing from the GMR reference. Table 6 lists the objective and regularization weights. To soften contact transitions, the binary contact mask is convolved with $[1, 2, 3, 2, 1]/9$ and clipped to $[0, 1]$; the resulting envelope ramps the contact weight and the regularization of the freed arm joints.

Foot stabilization. For each foot f, a frame is classified as stance when its GMR horizontal displacement per frame is below 0.01 m and its height is within 0.05 m of that foot's minimum height. Stance runs separated by more than 4 frames are treated separately, and runs shorter than 3 frames are discarded. Let $\mathcal{S}_{f,e}$ denote stance episode e and $a^{f,e} \in \mathbb{R}^2$ its median horizontal foot position. In E_{foot}, $y_t^f \in \mathbb{R}^2$ is the current horizontal foot position, and the Jacobian acts only on the corresponding leg joints. Each episode therefore has its own anchor rather than sharing one fixed position across the sequence.

Nonpenetration constraints. Here $d_{t,g}$ denotes the signed distance for a robot–object or robot–floor geometry pair, evaluated on convex-decomposed geometry using `mj_geomDistance` from MuJoCo (Todorov et al., 2012). We use a tolerance of $\tau_g = 0.01$ m for robot–object pairs, including the forearm of a grasping arm, and $\tau_g = 0.003$ m for foot–floor pairs. Nonnegative slacks allow violations at a quadratic cost, balancing contact preservation, collision avoidance, and temporal smoothness. Robot geometries are prefiltered to those within 0.2 m of the object along the GMR trajectory; object collision constraints are activated for pairs within a 0.05 m margin at each iteration.

Solver. We solve the sequence-level problem using 8 sequential quadratic programming iterations, re-linearizing the signed distances at the current pose. Each sparse quadratic subproblem is solved with OSQP (Stellato et al., 2020) at tolerance 10^{-4}. Updates satisfy the trust-region bound $|\Delta q_{t,j}| \leq 0.3$ rad and the joint limits $q_{\text{lo},j} \leq q_{t,j} + \Delta q_{t,j} \leq q_{\text{hi},j}$. Temporal smoothness is enforced within the objective, and no post-processing filter is applied to the optimized joint trajectory.

B CoLM-Generator Details

B.1 Motion Representation and Preprocessing

Canonical frame. All sequences use a Z-up canonical frame defined from the initial root pose: we remove the frame-0 root yaw and translate its horizontal position to the origin, without shifting height. The same transform is applied to the body and object, preserving their relative geometry. Joint rotations and velocities are expressed in this canonical frame; joint positions are root-relative in XY and absolute in Z. Object poses are canonical-frame quantities, whereas the interaction field uses object-local coordinates.

Features. Table 7 lists the per-frame features. For G1, the body uses $J = 30$ link frames, namely the pelvis and the 29 links driven by the actuated joints, giving $12J + 9 = 369$ body features and 504 features in total. The human-skeleton variant uses $J = 22$ SMPL-H joints, giving 273 body features

and 408 features in total; its object and interaction-field features are unchanged. Body motion is decoded from the root translation and link rotations only; joint positions, velocities, heading, and foot contacts serve as auxiliary channels.

Decoding to joint angles. The $6J$ rotation channels are global link rotations, so they are converted to the 29 actuated joint angles link by link rather than by inverse kinematics. Let $\mathbf{R}_t^l$ be the decoded global rotation of link l, let $\mathrm{pa}(l)$ be its parent in the URDF tree, and let $\mathbf{R}_l^{\mathrm{org}}$ and $\mathbf{a}_l$ be the fixed joint-origin rotation and hinge axis of the joint that drives l. The residual rotation about that hinge is $\mathbf{R}_{t,l}^{\mathrm{ax}} = (\mathbf{R}_l^{\mathrm{org}})^\top (\mathbf{R}_t^{\mathrm{pa}(l)})^\top \mathbf{R}_t^l$, and the joint angle is its rotation vector projected on the axis, $q_t^l = \log(\mathbf{R}_{t,l}^{\mathrm{ax}})^\vee \cdot \mathbf{a}_l$. The root translation is read from the root-position channels and the root orientation from the global rotation of the pelvis link. The predicted rotations are not constrained to be exactly compatible with the hinge, so this projection discards any off-axis residual; nothing is clipped to the joint limits and no smoothing or re-grounding is applied, which is what Joint Limit in Appendix E.1 measures. The result is a 43-dimensional trajectory holding the root pose, the 29 joint angles, and the object pose. Evaluation and CoLM-Tracker consume this same trajectory, so the generator tables and the tracker's reference clips are produced by one decoding procedure.

Table 7: Per-frame generator features. Dimensions are given for J body links; the total is 504 for G1.

Block	Feature	Dim.	Definition
Body	Root position	3	Canonical-frame root translation.
	Heading	2	$[\cos\psi_t, \sin\psi_t]$, where ψ_t is root yaw.
	Joint positions	$3J$	Root-relative XY and absolute height.
	Joint rotations	$6J$	Global link rotations in the 6D representation.
	Joint velocities	$3J$	$30(p_{t+1} - p_t)$ from global link positions; zero at the final frame.
	Foot contacts	4	Ankle/toe indicators: $z < 0.08$ m and $v_{xy} < 0.25$ m/s.
Object	Pose	9	6D rotation followed by translation.
	Velocity	9	Unscaled forward difference of the pose features, not linear or angular velocity; zero at the final frame.
Interaction	Field	117	Eq. 2 with $K = 15$ keypoints and $M = 24$ anchors, flattened.

Interaction field. The $K = 15$ body keypoints are the pelvis and the left and right hips, knees, ankles, feet, shoulders, elbows, and hands, taken at link origins without palm offsets. The $M = 24$ object anchors are selected from 4096 surface samples in each mesh's rest frame by greedy farthest-point sampling, initialized at the point farthest from their centroid. Anchor positions and ordering are fixed across all sequences of the object. For numerical stability, the inverse-distance weights in Eq. 2 are implemented as

$$a_{ij} = \frac{1}{1.5\|v_i - v_j\| + \varepsilon}, \quad a_{ii} = 0, \quad W_{ij} = \frac{a_{ij}}{\sum_k a_{ik}}, \quad L = V - WV, \qquad \varepsilon = 10^{-6}. \tag{9}$$

The regularizer keeps coincident-point weights finite. Row normalization makes the field invariant to joint translation of all points, while the object-local frame removes dependence on global object pose. Fixed point identities and dense distance weights avoid changes in discrete neighborhood connectivity across frames.

Normalization. Each feature channel is normalized using its mean and standard deviation over training-split frames, including the zero velocities at final frames. Channels with standard deviation below 10^{-4} use scale 1.

Sequence length. Sequences shorter than 40 frames are excluded from generator training and evaluation, and sequences longer than 300 frames are truncated to their first 300 frames. Batches are zero-padded to their longest sequence. A validity mask excludes padded frames from encoder attention, cross-attention memory, and all losses; spatial conditions are applied only to valid frames.

B.2 Model Architecture

Three-stream Transformer. CoLM-Generator processes the body, object, and interaction-field features in three streams with separately parameterized projections and Transformer layers. Each

stream projects its features to width 512, applies four encoder layers, exchanges information with the other two streams, applies four further encoder layers, and projects back to its original feature dimension. The three outputs are concatenated without an additional fusion layer. Table 9 lists the layer hyperparameters, which are shared by all encoder and decoder layers.

Cross-stream exchange. Each stream queries the concatenated frame tokens of the other two streams through its own two-layer Transformer decoder:

$$\begin{aligned} \tilde{H}_b &= \mathrm{Dec}_b(H_b, [H_o; H_L]), \\ \tilde{H}_o &= \mathrm{Dec}_o(H_o, [H_b; H_L]), \\ \tilde{H}_L &= \mathrm{Dec}_L(H_L, [H_b; H_o]). \end{aligned} \tag{10}$$

Concatenation is along the token axis, and the validity masks are concatenated accordingly. Conditioning tokens are removed before this exchange and prepended again afterward.

Conditioning tokens. The diffusion timestep, object geometry, and text each occupy a separate conditioning token prepended to every stream, and sinusoidal positional encoding assigns these tokens distinct positions. Text is encoded by a frozen CLIP ViT-B/32 (Radford et al., 2021) followed by a linear projection to 512 dimensions. Object geometry is represented by a basis point set (BPS) (Prokudin et al., 2019) precomputed in the mesh rest frame: 1024 basis points are sampled uniformly inside a radius-1 m ball centered at the frame origin and shared across objects, and each feature is the displacement from a basis point to its nearest mesh vertex. The resulting 3072 values, in metres, are passed to a two-layer MLP without separate normalization, preserving object scale. Text and geometry conditions are independently dropped with probability 0.1 during training. Sampling uses the conditional prediction only, i.e., a classifier-free guidance (Ho & Salimans, 2022) scale of 1.0. The spatial-conditioning mask enters the streams as described in Appendix B.4.

B.3 Training Objectives

Reconstruction loss. The model predicts the clean sequence $\hat{\mathbf{x}}_0$. Each reconstruction term averages squared errors over valid frames and the channels of its feature segment s:

$$\mathcal{L}_s = \frac{\sum_{b,t} \nu_{b,t} \|\hat{\mathbf{x}}^s_{0,b,t} - \mathbf{x}^s_{0,b,t}\|_2^2}{D_s \sum_{b,t} \nu_{b,t}}, \tag{11}$$

where $\nu_{b,t}$ is the validity mask and D_s is the segment dimension. Conditioned elements remain included in these losses, supervising reproduction of the supplied values. The reconstruction loss is the weighted sum $\mathcal{L}_{\mathrm{rec}} = \sum_s \lambda_s \mathcal{L}_s$ over the body-pose, body-velocity, foot-contact, object-pose, object-velocity, and interaction-field segments, with $(\lambda_{\mathrm{pose}}, \lambda_{\mathrm{vel}}, \lambda_{\mathrm{feet}}, \lambda_{\mathrm{objpose}}, \lambda_{\mathrm{objvel}}, \lambda_{\mathrm{if}}) = (1, 0.1, 0.1, 2, 0.1, 0.9)$.

Geometric consistency loss. $\mathcal{L}_{\mathrm{cons}}$ recomputes the interaction field from the predicted motion and compares it with the ground-truth field rather than with the predicted field channels. After de-normalization, body keypoints are reconstructed by forward kinematics from the predicted root translation and 6D link rotations, transformed into the predicted object frame, and used to recompute the field with Eq. 9. The loss averages over valid frames and includes a factor of K:

$$\mathcal{L}_{\mathrm{cons}} = \frac{\sum_{b,t} \nu_{b,t} \sum_{k=1}^{K} K \omega_{b,t,k} \|\mathbf{L}^{\mathrm{fk}}_{b,t,k}(\hat{\mathbf{x}}_0) - \mathbf{L}^{\mathrm{gt}}_{b,t,k}\|_2^2}{\sum_{b,t} \nu_{b,t}}, \qquad \omega_{b,t,k} = \mathrm{softmax}_k(-d^{\mathrm{gt}}_{b,t,k}/\tau), \tag{12}$$

with $\tau = 0.03$. Because the softmax weights sum to one, the factor K keeps the loss on the scale of an unweighted sum over the K keypoints. The main-text form of $\mathcal{L}_{\mathrm{cons}}$ omits this factor together with the batch averaging and the validity mask for clarity. $\mathcal{L}_{\mathrm{cons}}$ is added to the reconstruction loss with unit weight. The ground-truth field, keypoints, distances, and weights are computed without gradients. The predicted object pose is not detached, so the loss back-propagates through forward kinematics and the object-frame transform to the root translation, the $6J$ link-rotation channels, and the nine object-pose channels; it has no direct gradient with respect to the heading, joint-position, velocity, foot-contact, or interaction-field channels. It therefore couples the reconstructed body and object geometry without directly supervising the predicted field branch. The w/o IF variant has no field channels, so its ground-truth field is rebuilt from the ground-truth body keypoints and object pose with Eq. 9; the rebuilt field matches the stored one to within 10^{-6} m, so both variants optimize the same $\mathcal{L}_{\mathrm{cons}}$.

B.4 SPATIAL CONDITIONING

Conditioning patterns. Conditioning is specified by a per-element binary mask $\mathbf{m}$ over the feature sequence and the corresponding normalized clean values $\mathbf{c}$. Spatial commands set entries of the nine object-pose channels; object velocity is not commanded. In every mode, frame 0 of all feature channels is also conditioned, whereas padded frames are never conditioned. During training, the pattern of each example is sampled independently from the distribution in Table 8.

Table 8: Spatial conditioning patterns during training. All modes additionally condition frame 0 of every feature channel. N denotes valid sequence length.

Mode	Additional object-pose conditions	Share
Default	None	30%
Full trajectory	All valid frames	30%
Target	Final frame $N-1$	30%
Waypoints	$n \sim \mathcal{U}\{1,\dots,5\}$ frames sampled uniformly without replacement	10%

Standing-pose prefix. To support initialization from the robot's default standing pose, we prepend a 20-frame transition to each training clip with probability 0.3. The standing state uses the tracker's default joint configuration, zero velocity, and a root height that places the lowest foot point on the ground. A quintic Hermite curve connects it to the clip's first frame, with terminal velocity set to 10% of the clip's initial velocity. The object remains at its initial pose, and the interaction field is recomputed for the standing state. The prefix is added after cropping and before normalization; the augmented sequence is truncated to 300 frames and conditioned on its new first frame. Evaluation clips retain their recorded initial states.

Mask embedding. The binary mask $\mathbf{m}$ is embedded through separate bias-free linear projections to width 512 for each stream and injection point. These embeddings are added before the first encoder layer and after the cross-stream exchange, identifying the conditioned feature channels.

Clean overwrite at sampling. During sampling, masked elements are overwritten with the clean values both before each network evaluation and after each DDIM update:

$$\mathbf{x} \leftarrow (1-\mathbf{m}) \odot \mathbf{x} + \mathbf{m} \odot \mathbf{c}. \tag{13}$$

No noise is added to these values, matching the clean replacement used during training. Target and waypoint modes additionally use gradient guidance (Appendix B.5) to shape transitions toward the commanded endpoint; exact pose satisfaction is enforced by the final hard overwrite.

B.5 ENDPOINT GUIDANCE

Hard conditioning ensures pose satisfaction at specified frames but does not explicitly constrain the smoothness of the surrounding motion. Target and waypoint modes therefore use endpoint guidance to shape the approach to the final pose and penalize excessive object acceleration. Guidance is applied to the predicted clean sequence after de-normalization. For clip b of valid length N_b, let $p_{b,f}$ and $R_{b,f}$ be the predicted object translation and rotation matrix at frame f, and let $\bar{p}_b, \bar{R}_b$ be the commanded final pose. In both target and waypoint modes, guidance acts on the last 30 valid frames, $\mathcal{F}_b = \{N_b-30,\dots,N_b-1\}$, and uses the same endpoint throughout this window, encouraging arrival and holding rather than supplying a reference trajectory. The guidance energy is

$$\begin{aligned}
E_{\mathrm{ep}} &= \mathrm{mean}_{b,\, f\in\mathcal{F}_b} \|p_{b,f} - \bar{p}_b\|_2^2, \\
E_{\mathrm{rot}} &= \mathrm{mean}_{b,\, f\in\mathcal{F}_b} \|R_{b,f} - \bar{R}_b\|_F^2, \\
E_{\mathrm{acc}} &= \mathrm{mean}_b \sum_{i=0}^{N_b-3} [\|a_{b,i}\|_2 - a_{\max}]_+^2, \\
a_{b,i} &= p_{b,i+2} - 2p_{b,i+1} + p_{b,i}, \\
\mathcal{E} &= E_{\mathrm{ep}} + 0.045 E_{\mathrm{rot}} + 8 E_{\mathrm{acc}},
\end{aligned} \tag{14}$$

where $[u]_+ = \max(u, 0)$. The rotation weight $0.045 = r^2/2$ with $r = 0.3$ m scales the rotation term to metres for a typical object size. The acceleration term penalizes only accelerations above $a_{\max} = 0.0085$ m/frame^2, close to the 99th percentile (8.48 mm/frame^2) of ground-truth acceleration for the

large box; an unthresholded penalty would also flatten genuine pick-up and put-down motions. All guidance constants are shared across objects. All windows and acceleration terms use each clip's valid length, excluding padding. All reported experiments use only the object-pose guidance energy above.

Guidance is inactive in the first 4 of the 50 DDIM steps and active in the remaining 46. At each active step, we perform ten updates on the normalized clean prediction $z = \hat{\mathbf{x}}_0$. The energy in Eq. 14 is evaluated through de-normalization, so gradients are taken with respect to z:

$$\tilde{g}_b = \mathrm{Blur}_{\sigma=2}(\nabla_{z_b}\mathcal{E}) \odot \mathbb{1}_C,$$
$$\rho_b = \sqrt{\frac{\sum_{f=0}^{N_b-1}\sum_{d\in C}\tilde{g}_{b,f,d}^2}{N_b|C|}}, \qquad z_b \leftarrow z_b - 0.0015\,\frac{\tilde{g}_b}{\rho_b}. \tag{15}$$

The gradient is smoothed by a temporal Gaussian with a standard deviation of two frames, restricted to each clip's valid length with replicate padding at its ends, and masked to the channel set C, which comprises the root translation, the $6J$ link rotations, and the nine object-pose channels. Because the guidance energy depends only on object poses, body-channel gradients are zero, although these channels remain included in the RMS normalization. The RMS normalization ρ_b is computed separately for each clip over C and its valid frames, rather than over the padded batch length, and is clamped below at 10^{-12}. After the guided prediction is used in the DDIM update, the clean conditions are reimposed.

B.6 Hyperparameters

Table 9 lists the hyperparameters of CoLM-Generator; the noise schedule and timestep sampling follow Nichol & Dhariwal (2021).

C CoLM-Tracker Details

CoLM-Tracker builds on HDMI (Weng et al., 2025) and is simulated with MuJoCo Warp, the GPU implementation of MuJoCo (Todorov et al., 2012), through mjlab (Zakka et al., 2026). One policy is trained for each reference clip. All settings below are shared across clips; a clip only specifies its reference motion, object asset, and contacting hands.

C.1 Observations, Actions, and Policy Architecture

Observations. The policy runs at 50 Hz. The reference command contains the future body keypoint positions in the robot frame and the future joint positions at 1, 2, 8, 16, and 32 control steps ahead, together with the motion phase. Proprioception contains the root angular velocity, the projected gravity, the joint positions at the current step and 1, 2, 3, 4, and 8 steps earlier, and the last three actions; Gaussian noise with standard deviations 0.05, 0.05, and 0.015 is added to the first three terms. The object observation is the object position and orientation in the robot root frame, perturbed by noise with standard deviations of 2 cm and 0.05 rad. It is latched at the first step of the episode and held fixed, so the actor only ever sees the object pose measured once at the start and never a live per-frame estimate; the noise is then the error of that single reading. The policy is therefore open loop with respect to the object and needs no object tracking at deployment, which is why a single pose estimate suffices on the real robot (Appendix D). Privileged observations, available only in simulation and given only to the critic, contain noise-free nine-step proprioceptive histories, the future root pose, the differences between future reference and current body poses and velocities, the root linear velocity, ankle positions and velocities, the heights of the ankles, pelvis, and torso, the applied actions and joint torques, the live noise-free object pose, and the differences between future reference and current object poses; unlike the actor, the critic sees the object move.

Actions. The actor outputs joint position targets for the 29 body joints as offsets from the default joint configuration, scaled per joint from 0.07 rad for the wrist pitch and yaw joints to 0.55 rad for the hip pitch, hip yaw, and waist yaw joints. The targets are tracked by joint-level PD control. For tasks requiring grasping, the G1 uses Dex3 hands with an additional binary open/close command for each grasping hand. Tasks that do not require grasping use rubber hands. During training, each episode samples an actuation delay of 2 to 6 physics steps (10 to 30 ms) and a first-order action filter coefficient from $[0.8, 1.0]$.

Table 9: CoLM-Generator hyperparameters. Loss weights and guidance constants are given with their equations in Appendices B.3 and B.5.

Hyperparameter	Value
Architecture	
Transformer width	512
Encoder layers per stream	4 before and 4 after the exchange
Exchange decoder layers per stream	2
Attention heads	4
Feed-forward width	1024
Dropout	0.1
Activation	GELU
Text encoder	CLIP ViT-B/32, frozen
Geometry encoder	$\mathrm{Linear}(3072, 512)$, ReLU, $\mathrm{Linear}(512, 512)$
Trainable parameters	73.05M
Training	
Diffusion steps	1000
Noise schedule	Cosine
Prediction target	Clean sequence $\mathbf{x}_0$
Timestep sampling	Importance sampling by loss second moment
Condition dropout	0.1 for text and geometry, independently
Standing-pose prefix	Probability 0.3, 20 frames
Optimizer	AdamW
Learning rate	2×10^{-4}, constant after 1000 warm-up steps
Weight decay	0.01
Batch size	192 per GPU, 768 in total
Gradient-norm clipping	1.0
EMA decay	0.999
Precision	bfloat16 mixed precision
Training steps	30,000
Checkpoint	EMA weights at the final step
Hardware	4 NVIDIA RTX 6000 Ada Generation GPUs
Training time	About 6 hours
Sampling	
Sampler	DDIM, $\eta = 0$
Sampling steps	50, respaced from 1000
Classifier-free guidance scale	1.0 (conditional prediction only)

Policy architecture. We use an asymmetric actor-critic. The actor, an MLP with hidden layers of 512, 256, and 256 units, receives only observations available on the robot, namely the reference command, proprioception, and the object observation, and outputs a diagonal Gaussian over actions. The critic, an MLP with hidden layers of 512, 256, and 128 units followed by a scalar value output, additionally receives the privileged observations. All MLPs apply layer normalization before Mish activations. This actor is used directly for evaluation in simulation and for deployment, without distillation.

C.2 Rewards

Tracking rewards. Table 10 lists all reward terms. Each tracking term has the form $\exp(-\bar{e}/\sigma)$, where $\bar{e}$ is the mean error over the listed bodies or joints. Local keypoint terms measure errors relative to the root, and root terms measure world-frame errors. Upper-body terms use the shoulder, elbow, and wrist links, lower-body terms use the hip, knee, and ankle links, and joint terms cover the waist, hip, knee, shoulder, and elbow joints. Regularization and feet terms are penalties, and each of the two groups includes a survival bonus. The terms of all groups are summed without clipping negative totals.

Interaction reward. r_t^{IF} uses the $K = 15$ body keypoints and $M = 24$ object anchors of Appendix B.1, stacked in the object frame as $\mathbf{V}_t^{\mathrm{cur}}$ for the simulated state and $\mathbf{V}_t^{\mathrm{ref}}$ for the reference. A single operator $\mathbf{W}_t^{\mathrm{ref}}$ is built from the reference geometry with the weights of Eq. 9 and applied to both:

$$\mathbf{L}_t^{\mathrm{cur}} = (\mathbf{I} - \mathbf{W}_t^{\mathrm{ref}})\,\mathbf{V}_t^{\mathrm{cur}}, \qquad \mathbf{L}_t^{\mathrm{ref}} = (\mathbf{I} - \mathbf{W}_t^{\mathrm{ref}})\,\mathbf{V}_t^{\mathrm{ref}}. \tag{16}$$

Table 10: Reward terms of CoLM-Tracker. Tracking terms use $\exp(-\bar{e}/\sigma)$. The interaction reward is removed in the w/o r_t^{IF} ablation.

Group	Term	Weight	σ
Body tracking	Upper-body keypoint position (local)	0.5	0.3
	Upper-body keypoint orientation (local)	0.5	0.4
	Lower-body keypoint position (local)	0.5	0.3
	Lower-body keypoint orientation (local)	0.5	0.4
	Root position	0.5	0.5
	Root orientation	0.5	0.5
	Body linear velocity	0.5	0.5
	Body angular velocity	0.5	2.5
	Joint position	0.5	0.25
	Joint velocity	0.5	2.5
Object tracking	Object position	2.5	0.2
	Object orientation	1.5	0.3
Interaction	Interaction field r_t^{IF}	2.0	0.1
Regularization	Action rate	0.1	–
	Joint velocity	5×10^{-4}	–
	Joint position limit	10	–
	Joint torque limit	0.01	–
	Survival	1.0	–
Feet	Impact force	1.0	–
	Foot slip	0.5	–
	Air time (threshold 0.5 s)	5.0	–
	Survival	1.0	–

Sharing the operator is what makes the two fields comparable: recomputing $\mathbf{W}$ from the simulated points would change the weighting itself, so the difference would no longer isolate the displacement of the keypoints. The anchors occupy the same rest-frame positions in both stacks, so the two fields differ only through the body keypoints. The reward is $r_t^{\mathrm{IF}} = \exp(-e_t/\sigma)$ with $e_t = \sum_{k=1}^{K} \alpha_k \|\mathbf{L}_{t,k}^{\mathrm{cur}} - \mathbf{L}_{t,k}^{\mathrm{ref}}\|_2$ and $\alpha_k = \mathrm{softmax}_k(-d_k^{\mathrm{ref}}/\tau)$, $\tau = 0.1$, where d_k^{ref} is the distance from reference keypoint k to its nearest anchor. The M anchor rows carry weight zero, so the reward concentrates on the keypoints that are in or near contact in the reference.

C.3 Training, Simulation, and Domain Randomization

Simulation and termination. Physics runs at 200 Hz. Object collision geometry is obtained by convex decomposition of the object mesh with CoACD (Wei et al., 2022), and the object mass is randomized over 0.5–1.5 kg. An episode lasts at most the length of its reference clip. It terminates when an error remains above its threshold for 25 consecutive control steps (0.5 s): a maximum keypoint position error of 0.5 m or orientation error of 1.2 rad, measured for the torso in the world frame and for all keypoints relative to the root, an object position error of 0.3 m, or an object orientation error of 0.5 rad. Episodes whose physics state diverges are also terminated.

Initialization and curriculum. Episodes start from a reference state perturbed by up to 5 cm horizontally and 1 cm vertically in root position, 0.1 rad in roll and pitch, and 0.2 rad in yaw, with root velocity perturbations of up to 0.2 m/s and 0.5 rad/s, joint position and velocity noise of 0.1, and object position perturbations of up to 5 cm horizontally and 1 cm vertically. The start frame follows a reverse curriculum. Frames are grouped into bins of 64 frames, start frames are initially drawn from the last bin, and the lower bound moves back by one bin once the survival rate from the current bin, tracked as an exponential moving average with decay 0.98, reaches 0.6, or after 4000 resets without progress. A fraction of 0.2 of the episodes starts anywhere in the clip.

Domain randomization. Robot link masses are scaled by factors in $[0.9, 1.1]$. Foot and wrist contacts use static friction in $[0.3, 1.6]$, dynamic friction in $[0.3, 1.2]$, and restitution in $[0, 0.2]$. The pelvis and torso centers of mass are shifted by up to 2 cm, joint offsets are perturbed by up to 0.01 rad, and object friction is drawn from $[0.3, 1.0]$. Actuation delay, action filtering, and observation noise are randomized as described in Appendix C.1.

Optimization. Table 11 lists the training hyperparameters of PPO (Schulman et al., 2017). Observations are normalized with running statistics.

Table 11: Training hyperparameters of CoLM-Tracker, shared by all reference clips.

Hyperparameter	Value
Parallel environments	4096
Environment steps	3×10^8
Rollout length	32
Epochs per iteration	5
Minibatches per epoch	8
Clip ratio	0.2
Discount γ	0.99
GAE λ	0.95
Learning rate	10^{-4}, adapted to a KL target of 0.01
Entropy coefficient	0.001
Gradient-norm clipping	1.0
Initial action standard deviation	1.5
Value-target normalization	Running statistics
Hardware	1 NVIDIA RTX A6000 GPU per reference clip

D Real-World Deployment

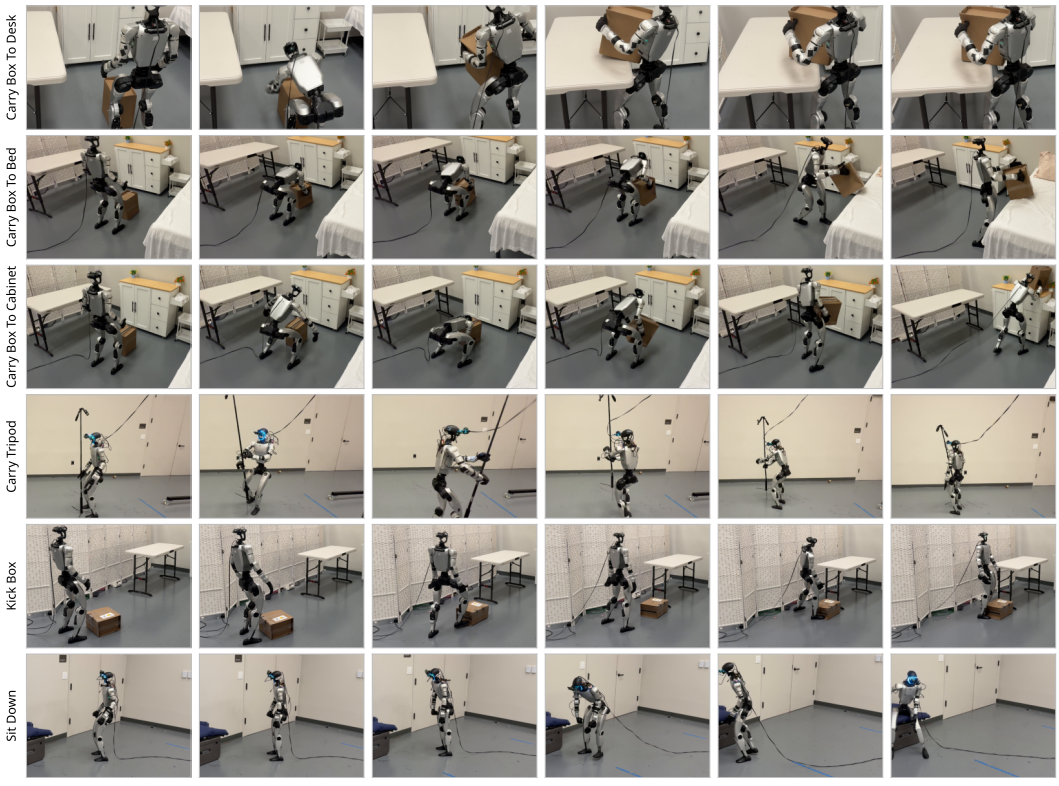


Figure 9: **Real-world deployment.** Key frames of each deployed task, in time order from left to right. Carrying the box is shown once per commanded target pose. Each panel follows the robot, so the viewpoint is fixed but the crop tracks it across the room.

System. We deploy on a Unitree G1 with 29 body degrees of freedom, using Dex3-1 hands for *Carry Tripod* and rubber hands for the other tasks. The onboard stack is the deployment code released with HDMI (Weng et al., 2025): a bridge that exchanges state and commands with the robot at 200 Hz, and a policy process that runs one ONNX forward pass at 50 Hz, with joint-level PD executed by the motor controllers.

What the policy is given is deliberately thin. The object's initial pose is estimated once, before the episode, with FoundationPose (Wen et al., 2024), and the policy holds its object observation at that single reading for the whole episode, exactly as in training (Appendix C.1). The robot's own root pose is never measured during execution: there is no state estimator and no odometry, and the only quantities entering the policy are the IMU orientation and angular velocity, the joint encoders, the last actions, and the reference clip, which advances on its own clock. One world-frame pelvis pose is read at the instant the operator hands control over, and is used only to express that single object reading in the robot's frame. The policy therefore replays the reference open loop and corrects only through proprioception.

Objects. Every physical object is measured in the room and rebuilt as a mesh, and none of them appears in the motion dataset. The carried carton is $20 \times 29 \times 35$ cm; it is carried standing upright and, for *Kick Box*, laid on its 29×35 face so that it stands only 20 cm tall. The tripod is 1.75 m tall. For *Sit Down* the robot sits on a plastic storage tote, $0.51 \times 0.645 \times 0.38$ m, with a folded blanket on top that brings the seat to 0.465 m; the clip was generated against a chair whose seat is at 0.46 m, and the tote replaced it at deployment because the chair slid on the floor as soon as the robot leaned on it.

Tasks. Every task starts from the robot's standing pose, which the standing-pose prefix of Appendix B.4 trains the generator to begin from, and is generated by the same checkpoint used everywhere else. The three carrying settings share one start scene, with the carton upright on the floor 0.5 m ahead, and one prompt, "Lift the largebox, move the largebox, and put down the largebox."; the only input that differs among them is the commanded endpoint (Table 12). The commanded height is the surface height plus the box's 17.5 cm half-height, so the box lands on the surface rather than in it, and the commanded yaw equals the approach azimuth, so the box ends square to the surface instead of across the robot's chest. The clip length grows with the carry distance d as $3.3 + 1.2d$ seconds, a rule shared by all three settings rather than a separate input. The other three tasks command no object pose and are generated from the text and the first frame alone. *Carry Tripod* is prompted with "Lift the tripod, move the tripod, and put down the tripod." over 8.0 s, and the model itself chooses where to set the tripod down, 1.21 m from where it stood, lifting it 18 cm with both hands. *Kick Box* is prompted with "Kick the smallbox, and set it back down." on the flattened carton 0.45 m ahead, over 5.6 s. *Sit Down* walks to a seat 1.2 m ahead and turns $135°$ before sitting, over 5.5 s.

Table 12: Commanded object endpoints of the three carrying settings, in the robot's start frame. The start pose is identical in all three.

Destination	Commanded position (m)	Yaw	Carry distance	Clip length
Desk	$(1.50,\ 0.00,\ 0.955)$	$0°$	1.00 m	4.5 s
Bed	$(0.00,\ 1.60,\ 0.815)$	$+90°$	1.68 m	5.6 s
Cabinet	$(0.00,\ -1.72,\ 1.075)$	$-90°$	1.79 m	5.8 s

Motion selection and tracking. A deployed clip is not the first sample drawn. For each task we draw multiple samples under a fixed seed with the DDIM schedule of Table 9, adding the endpoint guidance of Appendix B.5 over a 30-frame window for the three carrying settings, and keep one of them. The kept sample is not chosen by eye: it is the top-ranked one under an automatic score computed on the MuJoCo geometry, which rewards reaching the commanded endpoint, holding the object with both hands, and setting it down flat on the intended surface, and penalizes penetration of the floor and the furniture and any motion of the object while it is not held. The kept motion becomes the reference of one tracking policy, trained with exactly the settings of Appendix C that are used for the simulation experiments of Sec. 4.3.

Spatial control. The three carrying settings receive the same prompt and start scene, and the task input changes only the commanded object pose; the clip length then follows from the carry distance by the shared rule above. Fig. 8 shows the result: the same prompt and start scene produce a carry of 1.00, 1.68, or 1.79 m, ending on a surface ahead of, to the left of, or to the right of the robot, each at its own height and heading.

Results. Each task is run for 20 trials. A trial succeeds if the robot executes the motion to the end and the object's final position lies within 20 cm of the reference's final object position. This uses the 20 cm threshold of the simulation criterion (Appendix E.1), but applies it to the final position only, whereas simulation averages the error over the whole rollout. Fig. 8 reports the outcome: across the six settings, 111 of 120 trials succeed. The two carrying tasks that place the object straight ahead,

on the desk and with the tripod, succeed in every trial, while the two settings that require turning before placing, the bed and the cabinet, lose two trials each.

E Evaluation Protocols

E.1 Common Setup

Contact F1. For generation, the generated clip and its paired ground-truth clip are decoded to robot configurations and placed in the same MuJoCo scene. At every frame, each of the 15 keypoint links of Appendix B.1 is in contact when its convex collision geometry lies within 2 cm of the object's convex geometry, measured with `mj_geomDistance`. For each clip, F1 compares the generated and ground-truth contact tables over their common length, and a clip in which neither side touches the object scores 1; we report the mean over clips. For retargeting, the reference label comes from the source human mesh against the object mesh (Appendix A.2) and the robot label applies the same 2 cm criterion to the rubber-hand and ankle links of the two hands and feet; F1 is again computed per clip and averaged over clips.

Gap, Jitter, and Foot Skate. Gap averages the distance between the robot link and the object surface over ground-truth contact frames at which the robot label is false. It is computed per clip and averaged over the clips that lose at least one contact, and distances are capped at 15 cm, beyond which the convex-mesh distance query is unreliable, so larger gaps are counted as 15 cm. Jitter is the mean magnitude of the third time derivative of the global link positions at 30 fps, reported in 10^2 m/s^3; because it depends on frame rate and skeleton, it is only comparable with the ground-truth row of the same table. Foot Skate is the mean speed of the ankle and toe links over frames labelled as in contact by the foot-contact channel of the representation (Table 7), reported in cm/s.

Joint Limit. Joint Limit is the fraction of joint-angle samples outside the URDF limits of the G1. The generator output is de-normalized without smoothing, joint-limit clipping, or foot re-grounding, so this metric measures the raw generated motion.

Object pose errors. Translation error is the Euclidean distance between the predicted and reference object positions in centimetres, and orientation error is the geodesic angle between the predicted and reference rotations in degrees. The superscript "end" denotes the final frame; errors without it are averaged over all frames. For methods that generate a human and retarget it, all errors are measured after retargeting, in the robot feature space. For tracking, $T_{\text{obj}}^{\text{end}}$ and $O_{\text{obj}}^{\text{end}}$ compare the object pose when a rollout ends, at the end of the clip or at termination, with the reference pose at the final frame of the clip, and are averaged over environments.

Semantic metrics. FID (Heusel et al., 2017) and the R@3, MM-Dist, and Diversity metrics of Guo et al. (2022) are computed with a TMR-style evaluator (Petrovich et al., 2023) trained on the training split only, following the HumanML3D and TMR protocols, so the clips it scores were never seen during its own training. It uses ACTOR-style Transformer encoders with 4 layers, 4 heads, a latent width of 256, and a feed-forward width of 1024, on top of a frozen DistilBERT (Sanh et al., 2019) text backbone. It is trained for 800 epochs with batch size 32 and learning rate 10^{-4} on clips of 40 to 300 frames, using TMR's contrastive, reconstruction, KL, and latent losses with weights 0.1, 1, 10^{-5}, and 10^{-5} and a contrastive temperature of 0.1; negatives whose captions exceed a sentence-embedding similarity of 0.85 are filtered. Its motion input concatenates the 369 body channels, the 18 object channels, and a per-frame BPS of the object mesh with 128 basis points. The interaction field is withheld so that all ablations are scored on the same input. FID compares evaluator embeddings of generated clips with those of the ground-truth clips truncated to the same lengths. R@3 retrieves the matching description within pools of 32 clips, MM-Dist is the distance between a motion embedding and its own description, and Diversity follows the InterAct implementation on evaluator embeddings.

Tracking success. A tracking rollout succeeds if it reaches the end of the clip without termination and its mean object position error is at most 20 cm. Each policy is evaluated with 1024 parallel environments, each running one episode from the first frame after 64 warm-up steps, and SR is the fraction of successful environments. The Tracking SR column of the generation table is measured the same way, on the 100 clips of the tracking benchmark (Appendix E.5): a method's generated motion for a clip becomes the reference of one policy, trained with the settings of Appendix C. The policies for CoLM-Generator are the ones reported in Sec. 4.3, so the two averages agree.

E.2 RETARGETING

We compare with GMR, which is also the first stage of our pipeline and keeps the recorded object trajectory, and with OmniRetarget as implemented in holosoma, using its interaction-mesh optimization with default constraints. GMR and our refinement return a solution for every clip, so the comparison is limited by OmniRetarget: all methods are evaluated on the $9{,}130$ of the $10{,}412$ clips for which it returns a feasible solution. A clip in which neither the reference nor the robot ever touches the object has no defined F1 and is left out of that average.

E.3 TEXT-CONDITIONED GENERATION

Test clips. We evaluate on the 597 test clips of at least 40 frames (Appendix A.1). Each method generates one motion per clip from the clip's description, object, and first frame, at the clip's own length capped at 300 frames; 79 clips reach this cap. Every method is scored in the 504-dimensional G1 feature space against the same ground-truth clips. Our samples use the sampler, step count, and guidance scale of Table 9.

Baselines. DreamControl generates the G1 body but no object motion, so its body is paired with the ground-truth object trajectory. We use the first of its samples for each clip, and resample its 20 fps output to 30 fps; the model emits 196 frames, so longer clips are truncated and scored against equally truncated ground truth. Kimodo is conditioned with its own end-effector constraints extracted from the ground truth at every frame: the positions and orientations of the effectors in contact, chosen per clip from the ground-truth geometry, together with the root path, root height, and facing direction. Its body is likewise paired with the ground-truth object trajectory. CHOIS + Retarget generates a human and the object, and its motion is retargeted to G1 with the pipeline used to build our dataset. CHOIS (G1) denotes direct generation on the G1 skeleton without subsequent human-to-robot retargeting. The only change to CHOIS is the skeleton: its human representation is replaced by the body and object features of CoLM-Generator (Table 7), and everything else follows the original method and its released settings. It is trained on the same clips and split as our model, so it differs from it in architecture and objective rather than in what it represents. Ours (Human) + Retarget uses the same architecture, training clips, split, text encoder, and sampling schedule as CoLM-Generator on 22 SMPL-H joints; its samples share the clip order and noise with our model, and they are retargeted with the same pipeline.

E.4 SPATIAL CONTROL

Commands are the ground-truth object poses of each test clip: the final pose in target mode, additionally frames 29, 59, and 89 in waypoint mode, and every frame in full-trajectory mode. At evaluation, waypoint conditions use zero-based frame indices 29, 59, and 89 when within the valid sequence length, together with the final frame. The same waypoint indices are used for both CHOIS variants and our method. CHOIS conditions on these waypoints through its input; both CHOIS variants are included only in waypoint mode. Ours (Human) + Retarget imposes each command with the same clean overwrite in human space, and its errors are measured after retargeting.

For the letter patterns in Fig. 6, each box is generated as a separate sequence conditioned on a first frame from the test set and a commanded final pose at a grid cell of the letter. A planner places the start of each box on a ring of 0.6 to 1.0 m around its goal and orients the robot to carry the box forward. The generated segments are then transformed from their canonical frames into a shared world frame.

E.5 TRACKING

The tracking benchmark contains 100 test clips, 20 for each of carry, push, pull, kick, and sit. A test clip of 60 to 300 frames is a candidate for a category when two conditions hold. First, the verbs of its description name that category and no other: lift, pick, carry, or raise for carry, push for push, pull or drag for pull, kick for kick, and sit for sit. Second, its reference motion performs the interaction. For carry, push, and pull, the hands touch the object and move it at least 0.3 m horizontally from its start, lifting it for carry and keeping it on the ground for push and pull. For kick, a foot touches the object and the hands do not, and for sit, the pelvis touches the object. A body part touches the object when the mesh contact of Appendix A.2 holds on at least 20% of the frames, and the object is lifted

when its lowest point is more than $5\,\mathrm{cm}$ above its lowest position in the clip on at least 20% of the frames. Within each category, clips are drawn from one object at a time, with the objects in a random order under a fixed seed, so that the category covers as many objects as it offers. The benchmark contains 48 objects from all six sources, and its contacting body parts are both hands in 33 clips, one hand in 27, one foot in 20, and the pelvis in 20. The reference motions are generated by CoLM-Generator from these clips, and every tracker trains one policy per reference motion. Table 13 lists the benchmark.

Table 13: **Tracking benchmark.** The 100 reference clips of Appendix E.5, 20 per interaction category. Contact names the body parts whose mesh contact selects the clip.

Category	Source	Sequence	Object	Frames	Contact
Carry	BEHAVE	Date03_Sub05_backpack_3500	backpack	187	Both hands
	BEHAVE	Date07_Sub04_boxtiny_1131	boxtiny	144	Left hand
	BEHAVE	Date03_Sub04_chairblack_liftreal_1230	chairblack	187	Both hands
	BEHAVE	Date07_Sub04_plasticcontainer_229	plasticcontainer	190	Right hand
	BEHAVE	Date05_Sub06_suitcase_lift_901	suitcase	249	Both hands
	BEHAVE	Date03_Sub04_tablesquare_lift_1001	tablesquare	199	Both hands
	BEHAVE	Date06_Sub07_toolbox_0	toolbox	300	Right hand
	IMHD	20230827_zhangjy_bat_bat_holdhandle_hit_2_510_0	baseball	173	Both hands
	IMHD	20231014_dujsh_dumbbell_dumbbell_right_lunges2_0_0_2	dumbbell	192	Right hand
	IMHD	20231014_dujsh_kettlebell_kettlebell_swing1_0_0	kettlebell	68	Both hands
	InterCap	Sub01_Object04_Seg_0_umbrella_0	umbrella	180	Right hand
	NeuralDome	subject10_monitor_188	monitor	289	Both hands
	NeuralDome	subject02_table_275	table	135	Both hands
	NeuralDome	subject04_trashcan_370_1	trashcan	213	Left hand
	NeuralDome	subject04_trolleycase_1440	trolleycase	85	Both hands
	OMOMO	sub11_floorlamp_001	floorlamp	216	Left hand
	OMOMO	sub7_largebox_025	largebox	215	Both hands
	OMOMO	sub2_largetable_004	largetable	206	Left hand
	OMOMO	sub1_suitcase_013	suitcase	235	Both hands
	OMOMO	sub1_tripod_000	tripod	123	Both hands
Push	IMHD	20230901_songzn_suitcase_suitcase_twohands_push_0_769_0	suitcase	262	Both hands
	OMOMO	sub10_largebox_004	largebox	131	Both hands
	OMOMO	sub16_largebox_002	largebox	150	Right hand
	OMOMO	sub1_largetable_051	largetable	150	Both hands
	OMOMO	sub1_largetable_053	largetable	132	Both hands
	OMOMO	sub15_plasticbox_000	plasticbox	135	Both hands
	OMOMO	sub15_plasticbox_004	plasticbox	95	Both hands
	OMOMO	sub16_plasticbox_000	plasticbox	133	Right hand
	OMOMO	sub16_plasticbox_005	plasticbox	136	Both hands
	OMOMO	sub1_plasticbox_009	plasticbox	123	Both hands
	OMOMO	sub8_plasticbox_000	plasticbox	144	Right hand
	OMOMO	sub14_smallbox_000	smallbox	130	Left hand

continued on next page

Table 13 continued from previous page

Category	Source	Sequence	Object	Frames	Contact
	OMOMO	sub8_smallbox_000	smallbox	132	Right hand
	OMOMO	sub8_smallbox_001	smallbox	145	Both hands
	OMOMO	sub12_smalltable_056	smalltable	200	Both hands
	OMOMO	sub9_smalltable_050	smalltable	122	Both hands
	OMOMO	sub1_suitcase_005	suitcase	150	Left hand
	OMOMO	sub8_suitcase_041	suitcase	180	Left hand
	OMOMO	sub6_trashcan_000	trashcan	229	Both hands
	OMOMO	sub8_trashcan_001	trashcan	141	Both hands
Pull	BEHAVE	Date07_Sub04_chairblack_hand_490	chairblack	270	Both hands
	BEHAVE	Date03_Sub04_tablesquare_hand_215	tablesquare	159	Both hands
	NeuralDome	subject01_trolleycase_1115_1	trolleycase	177	Both hands
	NeuralDome	subject01_trolleycase_1115_3	trolleycase	127	Right hand
	OMOMO	sub11_clothesstand_022	clothesstand	195	Both hands
	OMOMO	sub16_clothesstand_010	clothesstand	181	Right hand
	OMOMO	sub9_floorlamp_024	floorlamp	170	Both hands
	OMOMO	sub10_largebox_074	largebox	149	Right hand
	OMOMO	sub12_largebox_062	largebox	289	Both hands
	OMOMO	sub6_largetable_000_1	largetable	199	Both hands
	OMOMO	sub4_monitor_018	monitor	199	Left hand
	OMOMO	sub16_plasticbox_041	plasticbox	141	Right hand
	OMOMO	sub3_smalltable_010	smalltable	140	Left hand
	OMOMO	sub1_suitcase_068	suitcase	179	Both hands
	OMOMO	sub8_suitcase_036	suitcase	161	Right hand
	OMOMO	sub11_trashcan_049	trashcan	134	Left hand
	OMOMO	sub9_tripod_011	tripod	132	Right hand
	OMOMO	sub10_whitechair_036	whitechair	145	Right hand
	OMOMO	sub5_whitechair_023	whitechair	261	Right hand
	OMOMO	sub3_woodchair_013	woodchair	166	Left hand
Kick	InterCap	Sub04_Object10_Seg_1_stool_0	stool	220	Right foot
	InterCap	Sub04_Object10_Seg_1_stool_220	stool	170	Right foot
	OMOMO	sub1_clothesstand_023_0	clothesstand	202	Right foot
	OMOMO	sub11_floorlamp_042	floorlamp	128	Right foot
	OMOMO	sub11_floorlamp_043	floorlamp	134	Left foot
	OMOMO	sub1_largetable_063	largetable	166	Left foot
	OMOMO	sub1_largetable_070	largetable	275	Right foot
	OMOMO	sub6_largetable_043	largetable	202	Left foot
	OMOMO	sub12_plasticbox_035_0	plasticbox	182	Right foot
	OMOMO	sub8_plasticbox_031	plasticbox	137	Left foot
	OMOMO	sub7_smallbox_039	smallbox	247	Right foot
	OMOMO	sub12_smalltable_065	smalltable	249	Right foot
	OMOMO	sub9_smalltable_060	smalltable	158	Left foot
	OMOMO	sub15_suitcase_032	suitcase	127	Right foot
	OMOMO	sub15_suitcase_033	suitcase	138	Right foot
	OMOMO	sub11_trashcan_063	trashcan	127	Left foot
	OMOMO	sub10_whitechair_099	whitechair	140	Right foot
	OMOMO	sub16_whitechair_035	whitechair	229	Right foot
	OMOMO	sub12_woodchair_016	woodchair	166	Right foot
	OMOMO	sub14_woodchair_056	woodchair	246	Right foot
Sit	BEHAVE	Date04_Sub05_chairwood_380	chairwood	180	Pelvis
	BEHAVE	Date03_Sub04_tablesquare_sit_151	tablesquare	299	Pelvis
	CHAIRS	Sub1338_Obj104_Seg0_333	104	177	Pelvis

continued on next page

Table 13 continued from previous page

Category	Source	Sequence	Object	Frames	Contact
	CHAIRS	Sub0471_Obj109_Seg0_0_2	109	235	Pelvis
	CHAIRS	Sub1894_Obj110_Seg0_315_1	110	226	Pelvis
	CHAIRS	Sub1294_Obj111_Seg0_0	111	255	Pelvis
	CHAIRS	Sub0524_Obj117_Seg0_0_0	117	179	Pelvis
	CHAIRS	Sub1749_Obj121_Seg0_486	121	207	Pelvis
	CHAIRS	Sub1407_Obj158_Seg0_186	158	153	Pelvis
	CHAIRS	Sub1437_Obj162_Seg0_480	162	153	Pelvis
	CHAIRS	Sub1660_Obj17_Seg0_465	17	225	Pelvis
	CHAIRS	Sub0159_Obj36_Seg0_240	36	270	Pelvis
	CHAIRS	Sub0862_Obj68_Seg0_0_3	68	214	Pelvis
	CHAIRS	Sub0825_Obj81_Seg0_0_1	81	173	Pelvis
	CHAIRS	Sub1027_Obj96_Seg0_159	96	213	Pelvis
	CHAIRS	Sub1876_Obj99_Seg0_264	99	264	Pelvis
	IMHD	20231015_zhaochf_chair_freestyle1_0_0_0	chair	215	Pelvis
	InterCap	Sub09_Object10_Seg_0_stool_0	stool	177	Pelvis
	NeuralDome	subject04_bigsofa_779_1	bigsofa	255	Pelvis
	NeuralDome	subject07_chair_540	chair	240	Pelvis

F Failure Mode Analysis

This appendix examines where each stage breaks down, using the same clips and metrics as the main experiments.

F.1 Retargeting

Contact preservation fails on a specific kind of clip rather than uniformly. Over the 9,130 shared clips of Appendix E.2, our per-clip Contact F1 has a median of 0.91 but a mean of 0.83, and 754 clips (8.3%) fall below 0.5. What separates those clips is how much the source human touches the object at all. Writing the reference contact density as the fraction of frame-region pairs the human mesh labels as contact, clips below 0.5 F1 have a median density of 0.078 against 0.310 for the rest, and F1 rises monotonically with density (Table 14; the correlation is 0.49).

Table 14: Retargeting Contact F1 against how much the source human touches the object. Density is the fraction of frame-region pairs labelled as contact in the reference.

Reference contact density	Clips	Mean Contact F1
< 0.05	388	0.27
0.05–0.15	979	0.66
0.15–0.30	3,245	0.85
≥ 0.30	4,465	0.89

Why sparse contact is hard. Two effects compound. The refinement recovers contact by pulling a robot link onto a recorded contact point, so an episode lasting a few frames gives the optimizer little to act on while the smoothness and regularization terms still pull toward the body-only solution. The metric is also fragile in this regime: with few true positives, a handful of frames in which a robot link passes within 2 cm of the object without the human touching it is enough to halve precision. Both effects push in the same direction, so a brief brush against an object is scored far more harshly than a sustained grasp.

Where it concentrates. CHAIRS is the weakest source, with a mean Contact F1 of 0.57 and 32.1% of its clips below 0.5, against 3.8% for OMOMO, 6.3% for IMHD, 8.3% for BEHAVE, 8.5% for InterCap, and 11.4% for NeuralDome. Its failing clips have a median density of 0.050. CHAIRS is dominated by sitting, where the load-bearing contact is the pelvis on the seat; the pelvis drives only the object alignment and is never reached for by the refinement (Appendix A.3), and Contact F1 scores only the hands and feet, which in these clips touch the chair briefly if at all.

CHAIRS also carries four times the share of foot contact of the other sources (10.2% and 10.9% of contact frames for the left and right foot, against 2–3% elsewhere), and a chair encloses the robot closely enough that a foot is often within the contact threshold without the human's foot being in contact.

F.2 GENERATION

The generator is trained on the object instances and the motion categories of Appendix A.1, and its failures outside that coverage are failures of interaction geometry rather than of appearance. For an object or an interaction it has not seen, it still produces a smooth, text-consistent motion, but one whose contact is wrong: the hands close where the object's surface is not, or the object follows the body without ever being held. The tracker cannot reliably correct these reference errors: following the prescribed body and object trajectories does not ensure that the physical interaction needed to execute them can be established.

F.3 TRACKING

The tracker reads the object's pose once, at the first frame, and holds that reading for the whole episode (Appendix C.1), while the reference clip advances on its own clock. The policy therefore receives no direct feedback about subsequent object displacement or rotation. Such deviations may affect proprioceptive observations, but the absence of live object-pose feedback limits the policy's ability to detect and correct interaction errors. Execution is open loop with respect to object-pose feedback, although the robot remains under closed-loop proprioceptive control. In the deployment results of Appendix D, the desk and tripod carrying settings achieve 100% success over 1.00 and 1.21 m, while the bed and cabinet settings both achieve 90% over 1.68 and 1.79 m. The latter two settings also require turning before placement. These results are consistent with increased difficulty for longer carries involving turns, but do not isolate the effects of distance, turning, or feedback availability.