SCESHORTCUT: EFFICIENT SCENE-AWARE MOTION SYNTHESIS VIA SHORTCUT MODELING

Anonymous authors
Paper under double-blind review



Figure 1: **Inference efficiency comparison.** With one-step sampling, *SceShortcut* achieves approximately $50 \sim 100\times$ speedups over recent advanced baselines, while maintaining motion plausibility and comparable motion-scene alignment. Inference times are measured on an NVIDIA A100 GPU.

ABSTRACT

Previous works on scene-aware human motion synthesis have mainly focused on motion-scene compatibility, with less attention to runtime efficiency. In this work, we present *SceShortcut*, a shortcut-based framework that accelerates scene-aware motion synthesis through one-step or few-step sampling while producing plausible and scene-compatible motion. To preserve fine-grained scene adaptation under a limited sampling budget, we introduce a global-to-local architecture with two stages trained jointly end-to-end. Specifically, at each sampling step, conditioned on the text instruction and global geometric-semantic scene features, the global stage produces intermediate motion features and predicts a root transport velocity. This velocity yields a coarse clean root motion estimate for querying fine-grained local scene geometry from a dense height field. The local stage incorporates text conditioning and the queried geometry into these intermediate features to predict transport velocities for the remaining motion components and a residual correction to the global-stage root transport velocity. To improve coarse root motion estimates and thereby enhance generation quality, we augment shortcut training with additional root-transport-velocity supervision and root-residual regularization. On TRUMANS and HUMANISE, one-step sampling with *SceShortcut* achieves approximately $50 \sim 100\times$ speedups over recent advanced open-source baselines while producing plausible motion with comparable motion-scene alignment.

1 INTRODUCTION

Scene-aware human motion synthesis, an important research direction for AR/VR, robotics, and embodied intelligence, requires generated motion to both align with semantic instructions and re-

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.

main compatible with the surrounding 3D environment (Tevet et al., 2023; Guo et al., 2024; Zhao et al., 2023; Gong et al., 2026), including obstacle avoidance during navigation (Jiang et al., 2024b) and appropriate contact during object interaction (Deng et al., 2026; Ghosh et al., 2026). Early methods (Hassan et al., 2021; Zhang & Tang, 2022; Wang et al., 2022) use conditional variational autoencoders (CVAEs) (Kingma & Welling, 2013) for motion generation under different conditioning settings, including motion history, spatial goals, action labels, and language descriptions.

With advances in motion generation architectures (Tevet et al., 2023; Jiang et al., 2023), scene-aware motion synthesis approaches have mainly adopted diffusion-based generation (Huang et al., 2023), autoregressive motion-token prediction (Wang et al., 2025), or hybrid architectures (Wang et al., 2026). To achieve fine-grained scene adaptation, some methods adopt a two-stage architecture, first generating global representations such as root trajectories (Cen et al., 2024; Yi et al., 2024) or scene affordance maps (Wang et al., 2024), and then synthesizing full-body motion based on these intermediate representations. Other approaches synthesize motion segments incrementally, generating short motion windows (Jiang et al., 2024b;a) or sequentially decoding motion tokens (Ghosh et al., 2026), using local scene geometry queried around character or target locations to improve geometric alignment between the generated motion and the scene.

Despite their progress in motion-scene compatibility, generating complete motion sequences at low inference cost remains challenging. First, current diffusion-based methods (Cen et al., 2024; Yi et al., 2024; Wang et al., 2024) often involve hundreds or even thousands of denoising steps. Second, segment-wise generation introduces sequential dependencies (Jiang et al., 2024b;a), while repeatedly extracting local geometric features from 3D scene points can further increase inference cost (Ghosh et al., 2026). As a result, these methods often take at least a few seconds to generate a complete motion sequence, as shown in Fig. 1. This raises a central question: how can we improve inference efficiency while generating plausible and scene-compatible motion?

To reduce inference cost, few-step generative approaches, such as latent consistency models (Luo et al., 2023) and shortcut models (Frans et al., 2025), have been applied to motion generation without 3D scene conditioning (Dai et al., 2024; Liu et al., 2025). However, reducing sampling steps alone does not eliminate the sequential dependencies of segment-wise synthesis. For full-sequence generation without these dependencies, the challenge is to align motion with fine-grained local geometry at the character's future locations under a limited sampling budget.

To address this challenge, we introduce *SceShortcut*, a shortcut-based framework for one-step or few-step full-sequence motion synthesis. For fine-grained scene adaptation under a limited sampling budget, it uses a global-to-local architecture with two jointly trained stages. Specifically, conditioned on text and global semantic-geometric scene features, the global stage produces intermediate motion features and predicts a root transport velocity. The predicted velocity yields a coarse clean root motion estimate that guides local geometry extraction from a dense height field. Using text and the extracted geometry, the local stage further processes these motion features to predict transport velocities for the remaining motion components and a residual correction to the initially predicted root transport velocity. To improve coarse root estimates and thus generation quality, we augment shortcut training with additional supervision on the combined root transport velocity and a penalty on the local root residual magnitude. With these designs, *SceShortcut* achieves approximately $50 \sim 100\times$ speedups over recent advanced open-source baselines while producing plausible motion with comparable motion-scene alignment, as shown in Fig. 1. Our main contributions are:

- We propose a shortcut-based framework, *SceShortcut*, for efficient one-step and few-step generation of plausible and scene-compatible motion sequences.
- To enable fine-grained scene adaptation with limited sampling steps, we introduce a within-step global-to-local architecture that uses coarse root motion estimates to query local scene geometry for body transport velocity prediction and coarse transport velocity refinement.
- To improve coarse root estimates and thus generation quality, we augment shortcut training with additional root-transport-velocity supervision and root-residual regularization.

2 Related Work

Scene-Aware Human Motion Synthesis. Scene-conditioned motion generation (Starke et al., 2019; Zhang & Tang, 2022; Zhao et al., 2022; 2023) aims to synthesize motion compatible with 3D en-

vironments under conditions such as semantic instructions or target locations. From a generative-model perspective, conditional variational autoencoders (CVAEs) (Kingma & Welling, 2013) have first been used under different conditioning settings, including motion history and spatial goals (Hassan et al., 2021; Zhang & Tang, 2022) or language descriptions together with scene representations (Wang et al., 2022). With advances in generative modeling (Ho et al., 2020; Song et al., 2020; Achiam et al., 2023; Rempe et al., 2026), recent scene-aware motion generation methods increasingly adopt diffusion-based generation (Huang et al., 2023), autoregressive motion-token prediction (Wang et al., 2025), or hybrid token-diffusion architectures (Wang et al., 2026). To achieve fine-grained scene adaptation, some methods adopt two-stage architectures that first generate global trajectories (Wang et al., 2021; Cen et al., 2024; Yi et al., 2024), affordance cues (Wang et al., 2024), coarse motion (Song et al., 2026), or key-joint paths (Hwang et al., 2025a) and then complete or refine detailed full-body motion. Other approaches synthesize motion segments incrementally, generating short motion windows (Jiang et al., 2024b;a; Gong et al., 2026) or sequentially decoding motion tokens into four-frame segments (Ghosh et al., 2026). Despite progress in scene adaptation, iterative diffusion sampling requires multiple model evaluations, while segment-wise generation retains sequential dependencies and may incur substantial costs from repeatedly processing local scene conditions, limiting the efficiency of full-sequence motion synthesis.

Efficient Generative Motion Modeling. Efficient generative modeling seeks to reduce the number or cost of model evaluations during sampling. Flow matching formulates generation as continuous transport (Lipman et al., 2023). Consistency models enable one-step and few-step sampling by learning consistent mappings from states along the same generative trajectory to a common clean endpoint (Song et al., 2023; Geng et al., 2025b; Luo et al., 2023).Shortcut models learn step-size-conditioned transport updates by enforcing consistency between a larger update and the composition of smaller updates (Frans et al., 2025; Nguyen et al., 2025). Other advances include average-velocity formulations (Geng et al., 2025a; Zhang et al., 2026) and phased distribution-matching distillation (Fan et al., 2026). These techniques have also advanced efficient motion modeling. Flow matching has been applied to motion synthesis and editing (Hu et al., 2023), as well as interactive and reactive motion generation (Gupta et al., 2026), while flow distillation enables one-step trajectory forecasting (Fu et al., 2025). MotionLCM uses latent consistency distillation for controllable one-step and few-step motion synthesis (Dai et al., 2024), and MotionPCM adopts phased consistency learning for efficient motion generation (Jiang et al., 2025). GestureLSM combines latent shortcut learning with spatiotemporal modeling for efficient co-speech gesture generation (Liu et al., 2025). For autoregressive motion synthesis, AAMDM combines diffusion-GAN generation with diffusion refinement (Li et al., 2024), while ARDY employs a hybrid motion representation with few-step denoising (Zhao et al., 2026). However, efficient motion generators largely operate without fine-grained 3D scene constraints, and extending them to full-sequence synthesis with scene adaptation remains underexplored. In this work, we combine shortcut modeling with a within-step global-to-local architecture for efficient scene adaptation and introduce additional root-transport-velocity supervision and root-residual regularization for more reliable coarse root motion estimates.

3 METHOD

As shown in Fig. 2 (a), given a text instruction $\boldsymbol{P}$, a static 3D scene $\boldsymbol{S}$, and an optional initial human state $\boldsymbol{X}_{\text{init}}$, our *SceShortcut* efficiently synthesizes a complete N-frame motion sequence $\boldsymbol{X}_1 = \{\boldsymbol{x}_1^i\}_{i=1}^N$ using one-step or few-step sampling while maintaining motion-scene alignment. For clarity, we first review shortcut modeling and introduce our motion representation in Sec. 3.1. Then, we describe text and global scene condition encoding in Sec. 3.2 (Fig. 2 (b)), followed by the within-step global-to-local architecture in Sec. 3.3 (Fig. 2 (c)-(e)). Finally, the training and inference procedures of our method are presented in Sec. 3.4.

3.1 PRELIMINARIES: SHORTCUT MODELING AND MOTION REPRESENTATION

Shortcut Modeling. Shortcut models (Frans et al., 2025) learn transport updates for variable step sizes, enabling the same model to support both multi-step and one-step generation. Let $\boldsymbol{X}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ denote Gaussian noise and $\boldsymbol{X}_1$ a clean motion sequence. Following flow matching (Lipman et al., 2023), we construct an intermediate state by linear interpolation:

$$\boldsymbol{X}_t = (1-t)\boldsymbol{X}_0 + t\boldsymbol{X}_1, \qquad t \in [0,1]. \tag{1}$$

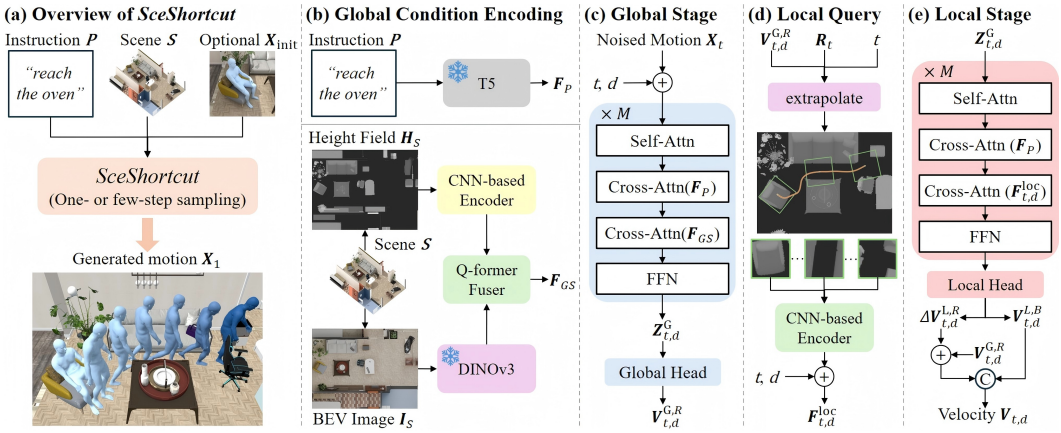

Figure 2: **Overview of our method.** (a) Overview of *SceShortcut*. (b) Text and global scene condition encoding. (c)-(e) Our within-step global-to-local architecture. Within each sampling step, the global stage predicts a coarse root transport velocity, which guides local geometry queries for body transport velocity prediction and coarse root transport velocity refinement.

Here, t denotes generative transport time, while each $\boldsymbol{X}_t$ represents a noisy motion sequence. Given the current state $\boldsymbol{X}_t$, transport time t, step size d, and external condition $\boldsymbol{C}$, a shortcut model predicts a step-size-normalized transport velocity:

$$\boldsymbol{V}_{t,d} = S_\theta\left(\boldsymbol{X}_t, t, d \mid \boldsymbol{C}\right), \qquad \boldsymbol{X}_{t+d} = \boldsymbol{X}_t + d\boldsymbol{V}_{t,d}. \tag{2}$$

A step size of $d = 1$ at $t = 0$ transports noise to a complete motion sequence in one model evaluation, while smaller step sizes enable few-step sampling.

Motion Representation. Following the motion representation of SceMoS (Ghosh et al., 2026), we represent each motion frame $\boldsymbol{x}_1^i$ using a 265-dimensional vector defined as

$$\boldsymbol{x}_1^i = \left[\boldsymbol{p}_{\text{root}}^i, \boldsymbol{q}_{\text{root}}^i, \boldsymbol{q}_{\text{body}}^i, \boldsymbol{p}_{\text{local}}^i, \boldsymbol{v}_{\text{joint}}^i, \boldsymbol{c}^i\right]. \tag{3}$$

Here, $\boldsymbol{p}_{\text{root}}^i$ denotes the global root translation. $\boldsymbol{q}_{\text{root}}^i$ and $\boldsymbol{q}_{\text{body}}^i$ are the global root and local body-joint orientations in the 6D rotation representation (Zhou et al., 2019), respectively. $\boldsymbol{p}_{\text{local}}^i$ and $\boldsymbol{v}_{\text{joint}}^i$ denote the root-relative joint offsets and velocities, respectively. $\boldsymbol{c}^i$ denotes foot contact states.

3.2 Text and Global Scene Condition Encoding

Text Encoding. Given a text instruction $\boldsymbol{P}$, we use a frozen pretrained T5 encoder (Raffel et al., 2020) to obtain token features $\boldsymbol{F}_P$, which guide the semantic content of the generated motion.

Global Scene Condition Encoding. For the static 3D scene $\boldsymbol{S}$, we extract global geometric and semantic features. Specifically, we construct a dense height field $\boldsymbol{H}_S$ aligned with the horizontal scene plane and encode it using a CNN to obtain geometric features $\boldsymbol{F}_{\text{geo}}$. Following SceMoS (Ghosh et al., 2026), we encode a bird's-eye-view (BEV) rendering $\boldsymbol{I}_S$ using a frozen pretrained DINOv3 encoder (Siméoni et al., 2025) to obtain semantic features $\boldsymbol{F}_{\text{sem}}$. A Q-Former-based network fuses $\boldsymbol{F}_{\text{geo}}$ and $\boldsymbol{F}_{\text{sem}}$ into global scene features $\boldsymbol{F}_{GS}$ to guide coarse root motion estimation in the global stage, while $\boldsymbol{H}_S$ is retained for fine-grained local queries, as described in Sec. 3.3.

3.3 Within-Step Global-to-Local Architecture

To enable fine-grained scene adaptation under a limited sampling budget, we design a jointly trained global-to-local architecture. Within each sampling step, the global stage estimates coarse clean root motion to guide local geometry queries, while the local stage uses the queried features to predict body transport velocities and a residual correction to the global-stage root transport velocity.

Global Root Transport Prediction. The global stage uses a M-layer transformer-based network G with separate cross-attention layers for text and global scene features, followed by a global head H_G.

We decompose the current motion state as $\boldsymbol{X}_t = [\boldsymbol{R}_t, \boldsymbol{B}_t]$, where $\boldsymbol{R}_t$ contains the root translation and orientation, and $\boldsymbol{B}_t$ contains the remaining local body motion components. Given $\boldsymbol{X}_t$, time t, and step size d, the network produces intermediate motion features $\boldsymbol{Z}^G_{t,d}$ conditioned on $\boldsymbol{F}_P$ and $\boldsymbol{F}_{GS}$ (ignoring the optional $\boldsymbol{X}_{\text{init}}$). Then, H_G maps $\boldsymbol{Z}^G_{t,d}$ to a root transport velocity $\boldsymbol{V}^{G,R}_{t,d}$:

$$\boldsymbol{Z}^G_{t,d} = \mathrm{G}\left(\boldsymbol{X}_t \mid t, d, \boldsymbol{F}_P, \boldsymbol{F}_{GS}\right), \qquad \boldsymbol{V}^{G,R}_{t,d} = H_G\left(\boldsymbol{Z}^G_{t,d}\right). \tag{4}$$

Root-Motion-Guided Local Geometry Queries. To capture fine-grained geometry around the character, we first obtain a coarse clean root motion estimate $\widehat{\boldsymbol{R}}^G_1$ by extrapolating the predicted velocity $\boldsymbol{V}^{G,R}_{t,d}$ from the current root motion state $\boldsymbol{R}_t$ over the remaining transport interval:

$$\widehat{\boldsymbol{R}}^G_1 = \boldsymbol{R}_t + (1-t)\boldsymbol{V}^{G,R}_{t,d}. \tag{5}$$

Then, we convert the coarse root motion estimate $\widehat{\boldsymbol{R}}^G_1$ into 2D root positions and yaw angles to guide the extraction of local height patches from the dense scene height field $\boldsymbol{H}_S$. For each query frame, we construct a regular $g \times g$ grid over a square of side length ℓ, centered at the estimated 2D root position and aligned with the yaw. We then bilinearly sample the height field at all query grids and encode the resulting patches into local geometry tokens by a CNN-based encoder. These tokens represent local geometric information around a coarse approximation of the final root trajectory and guide the local stage in aligning the generated motion with the surrounding environment. To account for the varying uncertainty of coarse global estimates across transport times and step sizes, we augment the geometry tokens with the embeddings of t and d, yielding $\boldsymbol{F}^{\text{loc}}_{t,d}$.

Local Body Transport and Root Refinement. The local stage also uses a M-layer transformer-based network L to continue processing the intermediate motion features $\boldsymbol{Z}^G_{t,d}$, with separate cross-attention layers for text features $\boldsymbol{F}_P$ and local geometry tokens $\boldsymbol{F}^{\text{loc}}_{t,d}$. A local head H_L maps the resulting motion features $\boldsymbol{Z}^L_{t,d}$ to a root transport residual $\Delta\boldsymbol{V}^{L,R}_{t,d}$ and transport velocities $\boldsymbol{V}^{L,B}_{t,d}$ for the remaining local body motion components:

$$\boldsymbol{Z}^L_{t,d} = \mathrm{L}\left(\boldsymbol{Z}^G_{t,d} \mid \boldsymbol{F}_P, \boldsymbol{F}^{\text{loc}}_{t,d}\right) \qquad (\Delta\boldsymbol{V}^{L,R}_{t,d}, \boldsymbol{V}^{L,B}_{t,d}) = H_L(\boldsymbol{Z}^L_{t,d}). \tag{6}$$

We add this residual to the global root transport velocity $\boldsymbol{V}^{G,R}_{t,d}$, then concatenate the resulting root velocity with the body transport velocity to update the motion state:

$$\boldsymbol{V}^R_{t,d} = \boldsymbol{V}^{G,R}_{t,d} + \Delta\boldsymbol{V}^{L,R}_{t,d}, \qquad \boldsymbol{V}_{t,d} = [\boldsymbol{V}^R_{t,d}, \boldsymbol{V}^{L,B}_{t,d}], \qquad \boldsymbol{X}_{t+d} = \boldsymbol{X}_t + d\boldsymbol{V}_{t,d}. \tag{7}$$

Together, the two stages jointly produce one full-sequence transport update using text, global scene features, and local scene geometry queried along the estimated coarse root trajectory, without requiring separate trajectory sampling, external spatial guidance, or short-segment-wise autoregression.

3.4 Training and Inference

Since local geometry queries rely on coarse root-motion estimates, improving these estimates can enhance the quality of generated motion. We therefore augment shortcut training with additional supervision on the combined root transport velocity and regularization of the local root residual.

Base Shortcut Supervision. To support generation with different sampling budgets, we combine flow-matching supervision at the smallest supported step size with self-consistency supervision at larger step sizes. For the smallest step size $d_{\min}$, the target follows the linear interpolation in Eq. 1:

$$\boldsymbol{V}^\star_{t,d_{\min}} = \boldsymbol{X}_1 - \boldsymbol{X}_0. \tag{8}$$

For a larger step size d, a copy of the model being trained, $S_{\bar{\theta}}$, performs two consecutive updates of size $d/2$ starting from $\boldsymbol{X}_t$. The resulting transport velocities $\overline{\boldsymbol{V}}_1$ and $\overline{\boldsymbol{V}}_2$ define the target $\boldsymbol{V}^\star_{t,d}$:

$$\overline{\boldsymbol{V}}_1 = S_{\bar{\theta}}\left(\boldsymbol{X}_t, t, \tfrac{d}{2}\right), \quad \overline{\boldsymbol{V}}_2 = S_{\bar{\theta}}\left(\boldsymbol{X}_t + \tfrac{d}{2}\overline{\boldsymbol{V}}_1, t + \tfrac{d}{2}, \tfrac{d}{2}\right), \quad \boldsymbol{V}^\star_{t,d} = \mathrm{sg}\left[\frac{\overline{\boldsymbol{V}}_1 + \overline{\boldsymbol{V}}_2}{2}\right], \tag{9}$$

where sg denotes stop-gradient. Each teacher evaluation recomputes the coarse clean root motion estimate and the corresponding local geometry conditions, allowing the student to approximate two smaller updates with a single larger update. For either supervision branch, the shortcut loss is

$$\mathcal{L}_{\text{shortcut}} = \left\|\boldsymbol{V}_{t,d} - \boldsymbol{V}^\star_{t,d}\right\|_2^2. \tag{10}$$

Additional Root Supervision. To improve the accuracy of the combined root update, we introduce an additional loss on $\boldsymbol{V}^R_{t,d}$, complementing the root channel supervision in $\mathcal{L}_{\text{shortcut}}$. Let $\boldsymbol{V}^{\star,R}_{t,d}$ denote the root channels of the shortcut target $\boldsymbol{V}^{\star}_{t,d}$. The root loss combines weighted supervision of the root transport velocity, its second-order temporal differences, and the updated root states:

$$\mathcal{L}_{\text{root}} = \lambda_{\text{vel}}\mathcal{L}_{\text{vel}} + \lambda_{\text{temp}}\mathcal{L}_{\text{temp}} + \lambda_{\text{pos}}\mathcal{L}_{\text{pos}} + \lambda_{\text{yaw}}\mathcal{L}_{\text{yaw}}. \tag{11}$$

Here, $\mathcal{L}_{\text{vel}}$ measures the mean squared error between $\boldsymbol{V}^R_{t,d}$ and $\boldsymbol{V}^{\star,R}_{t,d}$, while $\mathcal{L}_{\text{temp}}$ measures the mean L1 error between their second-order differences along the motion-frame dimension. To supervise root positions and headings after the transport update, we convert the updated root states $\boldsymbol{R}_t + d\boldsymbol{V}^R_{t,d}$ and $\boldsymbol{R}_t + d\boldsymbol{V}^{\star,R}_{t,d}$ into global 3D root positions and yaw angles. $\mathcal{L}_{\text{pos}}$ and $\mathcal{L}_{\text{yaw}}$ measure the mean L1 position error and the mean cosine yaw error, respectively.

Local Root Residual Regularization. To discourage excessive local corrections that may misalign generated motion with the queried geometry, we penalize the magnitude of the local root residual:

$$\mathcal{L}_{\text{res}} = \left\| \Delta \boldsymbol{V}^{L,R}_{t,d} \right\|_2^2. \tag{12}$$

Given $\mathcal{L}_{\text{shortcut}}$, $\mathcal{L}_{\text{root}}$, and $\mathcal{L}_{\text{res}}$, the complete training objective is

$$\mathcal{L} = \mathcal{L}_{\text{shortcut}} + \mathcal{L}_{\text{root}} + \lambda_{\text{res}}\mathcal{L}_{\text{res}}, \tag{13}$$

where λ_{res} controls the strength of residual regularization.

Inference. Our model supports both one-step and multi-step generation. For K-step sampling, we perform K uniform updates starting from Gaussian noise $\boldsymbol{X}_0$. At step $k = 0, \ldots, K-1$, we set $t = k/K$ and $d = 1/K$ and apply Eq. 7. Each step produces a new coarse clean root motion estimate, queries the corresponding local geometry, and updates the complete motion sequence.

4 Experiments

4.1 Experimental Setup

Implementation Details. We set the number of transformer layers in each stage to $M = 6$. Together with the other trainable feature encoders, the complete model contains 63.78 M trainable parameters. To reduce computational cost, we query local geometry at the first motion frame and at subsequent frames sampled along the estimated coarse root trajectory with a temporal stride of $\tau = 4$. We evaluate *SceShortcut* on the TRUMANS (Jiang et al., 2024b) and HUMANISE (Wang et al., 2022) datasets. Details of global scene preprocessing, including bird's-eye-view (BEV) image preparation and height-field construction, are provided in Appendix A. On both datasets, local geometry queries use a regular $g \times g$ grid ($g = 64$) covering a $2\,\text{m} \times 2\,\text{m}$ square, centered and oriented according to the estimated root position and yaw, respectively. We train separate models on each dataset for 300 epochs using the Adam optimizer with a batch size of 16 and an initial learning rate of 10^{-4} under cosine annealing. For shortcut learning, we sample the flow-matching and self-consistency branches with equal probability. The loss weights in Eqs. 11 and 13 are set to $\lambda_{\text{vel}} = 1$, $\lambda_{\text{temp}} = 0.25$, $\lambda_{\text{pos}} = 1$, $\lambda_{\text{yaw}} = 0.5$, and $\lambda_{\text{res}} = 1$. We use $d_{\min} = 1/32$ and support sampling with step sizes $d \in \{1/32, 1/16, 1/8, 1/4, 1/2, 1\}$. Setting $(t, d) = (0, 1)$ enables one-step generation. Training and testing are conducted on a single NVIDIA A100 80GB GPU.

Baselines. On the TRUMANS dataset, we compare *SceShortcut* with TRUMANS (Jiang et al., 2024b), TeSMo (Yi et al., 2024), and SceMoS (Ghosh et al., 2026). TRUMANS is a powerful autoregressive diffusion model for motion generation, but it can generate only 16 frames at a time and requires a final pelvis position for each window and frame-wise action labels in addition to the initial position. TeSMo uses a diffusion model conditioned on scene points, text, and an additional final pelvis position. SceMoS generates discrete motion tokens with an autoregressive planner and decodes them into motion segments conditioned on local heightmaps queried around the root position in the preceding segment's final frame. However, we encounter some issues when running inference with its open-source code, as detailed in Appendix B. For quantitative comparisons with SceMoS, we report the conditioning requirements and evaluation results from both the SceMoS paper and our evaluation using its released inference code, while all qualitative results of SceMoS in this work are

Table 1: **Quantitative comparison on the TRUMANS dataset.** We compare *SceShortcut* with existing advanced methods in terms of additional spatial conditioning requirements, motion quality, and average inference time. The best and second-best results are **bold** and underlined, respectively.

Method	Add. SCond.	FID↓	Div.→	PFC↓	Cont.↑	P_{mean} ↓	P_{max} ↓	Time(s)↓
TRUMANS	GT Sparse pelvis pos.	0.34	2.71	**0.58**	0.98	1.83	11.75	6.048
TeSMo	GT Final pelvis pos.	0.48	2.02	0.65	0.92	2.11	11.98	31.06
SceMoS (officially reported)	None	0.31	2.67	0.61	0.98	1.81	**11.12**	8.000
SceMoS (Our evaluation)	GT root trajectories	0.32	2.59	0.60	0.98	1.79	11.74	2.969
***SceShortcut*-1step (Ours)**	None	0.32	2.69	0.61	0.98	1.77	11.48	**0.058**
***SceShortcut*-2step (Ours)**	None	0.29	2.72	0.60	0.98	1.69	11.27	0.082
***SceShortcut*-4step (Ours)**	None	**0.27**	2.70	0.59	**0.99**	**1.63**	**11.12**	0.131

Table 2: **Quantitative comparison on the HUMANISE dataset.** We compare *SceShortcut* with existing advanced methods in terms of additional spatial conditioning requirements, motion quality, and average inference time. The best and second-best results are **bold** and underlined, respectively.

Method	Add. SCond.	FID↓	APD↑	Goal dist.↓	Cont.↑	Non-coll.↑	Time(s)↓
HUMANISE	None	1.46	**6.271**	0.750	63.52	**99.84**	0.158
Afford@2	None	1.20	2.625	0.151	**95.17**	99.71	3.347
MoST	Pred. target mask	1.80	2.108	0.148	78.78	99.75	3.202
***SceShortcut*-1step (Ours)**	None	1.20	2.356	0.144	93.74	99.80	**0.059**
***SceShortcut*-2step (Ours)**	None	1.02	2.275	0.147	94.54	99.81	0.084
***SceShortcut*-4step (Ours)**	None	**0.91**	2.301	**0.142**	94.72	99.82	0.131

based on motions generated with this code. On the HUMANISE dataset, we compare our method with HUMANISE (Wang et al., 2022), Afford-Motion (Wang et al., 2024) (Afford@2), and MoST. HUMANISE uses a conditional variational autoencoder to generate motion conditioned on the scene point clouds and language. Afford-Motion adopts a two-stage diffusion framework that first predicts an affordance map and then generates motion conditioned on the generated map. MoST generates trajectories and body motion through separate diffusion models conditioned on an externally supplied target-object location. More details on baseline evaluation are provided in Appendix C.

Evaluation Metrics. On the TRUMANS dataset, we follow the evaluation protocol of SceMoS (Ghosh et al., 2026) and report Fréchet Inception Distance (FID), diversity (Div.), physical foot contact score (PFC), scene contact score (Cont.), and mean and maximum penetration depths (P_{mean} and P_{max}). FID measures the distributional difference between generated and ground-truth motion features, while Div. measures motion diversity. PFC evaluates the physical plausibility of foot motion relative to the ground, Cont. evaluates human-scene contact, and the penetration metrics quantify the depth of human-scene intersections. On the HUMANISE dataset, we use the evaluation script provided by Afford-Motion (Wang et al., 2024) to compute average pairwise distance (APD), goal distance (Goal dist.), contact score (Cont.), and non-collision score (Non-coll.). APD measures diversity across generated motions, and Goal dist. evaluates how closely the motion reaches the target object. Cont. and Non-coll. evaluate human-scene contact and collision avoidance, respectively. FID is additionally reported on HUMANISE to assess motion distribution quality. We also report the average inference time in seconds to evaluate generation efficiency.

4.2 Comparisons with State-of-the-Art Baselines

Results on TRUMANS. As shown in Tab. 1, our method achieves substantially faster generation than existing advanced diffusion-based or autoregressive methods using only one or a few sampling steps, while attaining comparable PFC and P_{max}, with even better FID, Cont., and P_{mean}. The qualitative results in Fig. 3 further show that our method can, in some cases, produce more plausible scene-aware motions, such as walking to the desk, sitting on the chair, or lying on the bed.

Results on HUMANISE. As shown in Tab. 2, our method achieves substantially faster generation than existing diffusion-based methods using only one or a few sampling steps, while attaining comparable Cont. and Non-coll., with even better FID and Goal dist. The qualitative results in Fig. 4 further show that our method can, in some cases, produce more plausible scene-aware motion, such as walking to the desk, sitting on the chair, or lying on the bed, than existing advanced methods.

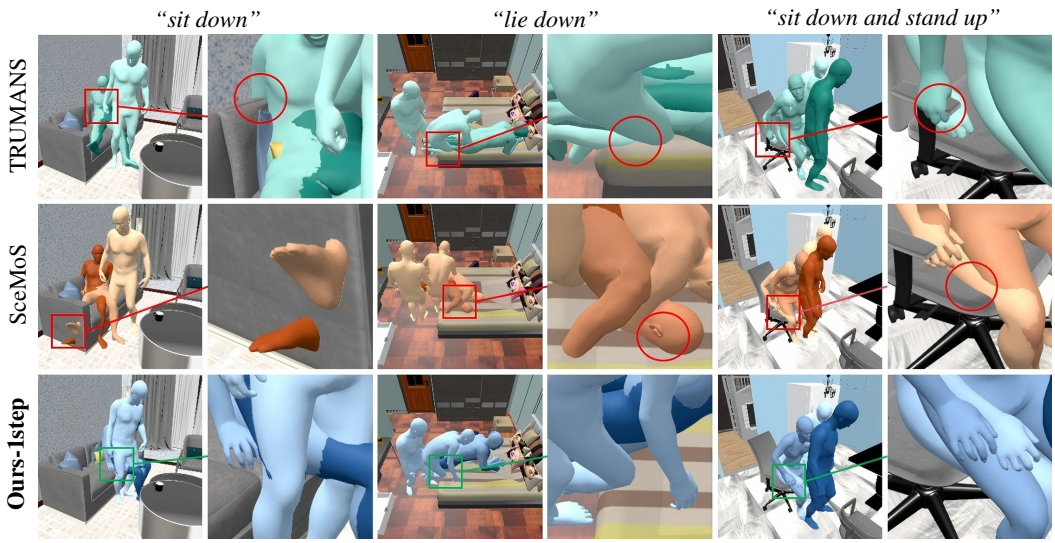

Figure 3: **Qualitative comparison on the TRUMANS dataset.** Despite requiring only one sampling step, the adjacent enlarged views show that our method can, in some cases, produce more plausible human-scene interactions than advanced baselines.

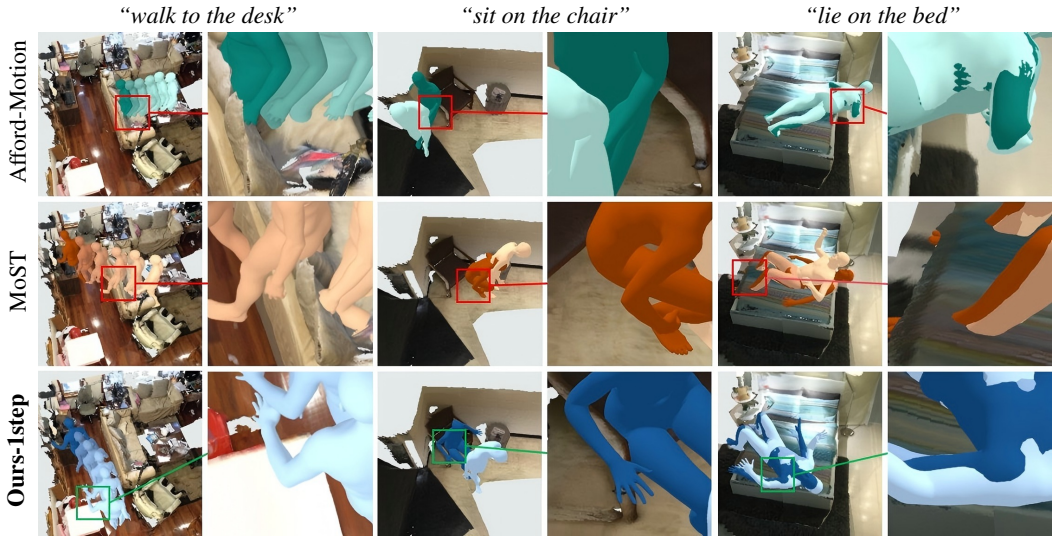

Figure 4: **Qualitative comparison on the HUMANISE dataset.** Despite requiring only one sampling step, the adjacent enlarged views show that our method can, in some cases, produce more plausible human-scene interactions than advanced baselines.

4.3 Ablation Studies

We ablate our method using one-step sampling on the TRUMANS dataset with respect to the architecture design and training loss. To directly evaluate the reliability of the coarse root estimate, we additionally report translation and rotation errors (TE and RE), measured in meters and degrees, respectively, for both the final generated root trajectory (FTE and FRE) and the coarse root-trajectory estimate (CTE and CRE) relative to the ground-truth root trajectory. More evaluations of scene conditioning and feature injection are provided in Appendix D.

Architecture Design. We evaluate three variants under the same shortcut training protocol, with total trainable parameter counts comparable to ours: (1) a single-stage baseline that directly predicts the full transport velocity conditioned on text and global scene features, without local geometry queries or root residual correction; (2) a two-model architecture that generates root and body motion separately, without intermediate feature passing or root residual correction; and (3) our architecture

Table 3: **Ablation studies of (a) network architecture and (b) training loss for our *SceShortcut*.** The best and second-best results are **bold** and underlined, respectively.

Configuration	FID↓	PFC↓	Cont.↑	P_{mean} ↓	P_{max} ↓	FTE↓	FRE↓	CTE↓	CRE↓	Time (s)↓
(a) Architecture Design										
Single-stage shortcut model	0.46	0.67	0.95	2.01	12.72	0.32	24.6	/	/	**0.055**
Separate root and body models	0.38	**0.61**	0.97	1.85	11.79	0.30	**20.8**	/	/	0.061
w/o local root residual correction	0.35	0.62	**0.99**	1.83	11.51	0.31	23.2	/	/	0.058
(b) Training loss										
w/o additional root supervision	0.56	0.71	0.94	2.17	13.35	0.37	28.3	0.40	30.7	0.059
w/o root residual regularization	0.36	0.62	0.96	1.87	11.74	0.31	22.7	0.35	26.3	0.058
Full model (Ours)	**0.32**	**0.61**	0.98	**1.77**	**11.48**	**0.29**	21.7	**0.31**	**23.4**	0.058

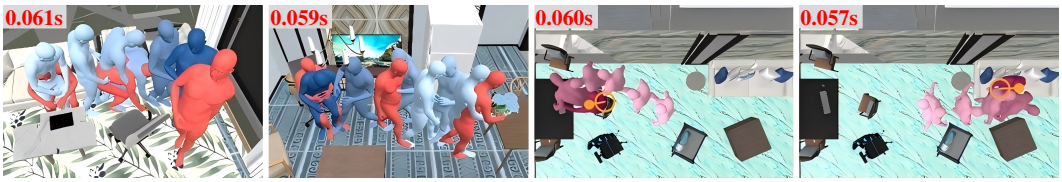

Figure 5: **Extensions to scene-aware motion control.** The first two examples show scene-aware motion inbetweening, with keyframes highlighted in red, whereas the last two show endpoint pelvis-position control from the same initial pose, with target pelvis positions marked in orange. All results are generated by our method using one-step sampling.

without local root residual correction. Compared with them, our full model achieves the best performance in terms of FID, PFC, and scene penetration, supporting the effectiveness of our architecture.

Training Loss. Removing the additional root supervision substantially increases both final and coarse root-trajectory errors, resulting in degradation across all motion-quality and scene-compatibility metrics. By incorporating root residual regularization, we further obtain consistent improvements across all metrics. These results show that additional supervision on the combined root transport velocity and regularization of the local root residual enhance the reliability of coarse root motion estimates and thereby improve overall generation quality.

4.4 Extensions to Scene-Aware Motion Control

SceShortcut can be extended to common applications such as scene-aware motion inbetweening (Hwang et al., 2025b) and endpoint pelvis-position control (Yi et al., 2024). As shown in Fig. 5, it can efficiently generate plausible motion sequences that align with the control signals in a single sampling step, demonstrating the extensibility of our method.

5 Limitations and Conclusion

Limitation. Our method employs shortcut modeling to reduce sampling steps and substantially improves generation speed. However, very few sampling steps, particularly one-step generation, may occasionally introduce motion artifacts, such as mild scene penetration or jitter, as shown in Appendix E. Increasing inference steps can alleviate these issues but incurs additional computational cost. Future work could explore training on larger, higher-quality datasets to mitigate the limitations.

Conclusion. In this work, we presentd a shortcut-based framework, *SceShortcut*, for the efficient one-step and few-step generation of plausible and scene-compatible motion sequences. To enable fine-grained scene adaptation with limited sampling steps, we introduced a within-step global-to-local architecture that uses coarse root motion estimates to query local scene geometry for body transport velocity prediction and coarse transport velocity refinement. To improve coarse root motion estimates and thus generation quality, we augmented shortcut training with additional root-transport-velocity supervision and root-residual regularization. Extensive experiments demonstrate that *SceShortcut* substantially improved generation efficiency over recent advanced open-source baselines while producing plausible motion with comparable motion-scene alignment.

AI USE STATEMENT

In this work, we used generative AI tools for: "design or provide feedback on research methodology or experiments, implement methods, assist with translation". We have not used generative AI tools for: "generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, clean and reformat dataset, support qualitative and thematic data analysis, interpret results". Additionally, we used generative AI tools for: "draft parts of a research paper, summarize or analyse existing literature, discover research topics or identify gaps, sourcing/searching for information, identify relevant literature." We have reviewed all AI-assisted work. For example, we manually verified the existence and accuracy of references suggested by LLMs, and all LLM-generated code was reviewed and tested for correctness by the authors. We take responsibility for the final content of this work, including any text, claims, code, and video demonstrations produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we will release the source code for our method upon acceptance of the paper. Section 3 describes the model architecture, motion representation, conditioning mechanisms, training objectives, and inference procedure, while the experimental section (Sec. 4) reports the datasets, baseline configurations, evaluation metrics, and implementation settings used in our experiments. Appendix A documents the scene-preprocessing procedures, and Appendix C provides further details on the dataset splits, motion-clip construction, and baseline evaluation protocols. We additionally document our reproduction of SceMoS in Appendix B. Together, these materials provide the implementation and experimental details needed to reproduce our main results.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Zhi Cen, Huaijin Pi, Sida Peng, Zehong Shen, Minghui Yang, Shuai Zhu, Hujun Bao, and Xiaowei Zhou. Generating human motion in 3d scenes from text descriptions. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1855–1866. IEEE, 2024.

Wenxun Dai, Ling-Hao Chen, Jingbo Wang, Jinpeng Liu, Bo Dai, and Yansong Tang. MotionLCM: Real-time controllable motion generation via latent consistency model. In *European Conference on Computer Vision*, pp. 390–408, 2024.

Zekai Deng, Ye Shi, Kaiyang Ji, Lan Xu, Shaoli Huang, and Jingya Wang. Human-object interaction via automatically designed vlm-guided motion policy. In *International Conference on Learning Representations*, volume 2026, pp. 100691–100718, 2026.

Xiangyu Fan, Zesong Qiu, Zhuguanyu Wu, Fanzhou Wang, Zhiqian Lin, Tianxiang Ren, Dahua Lin, Ruihao Gong, and Lei Yang. Phased dmd: Few-step distribution matching distillation via score matching within subintervals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41667–41676, 2026.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, 2025.

Yuxiang Fu, Qi Yan, Lele Wang, Ke Li, and Renjie Liao. Moflow: One-step flow matching for human trajectory forecasting via implicit maximum likelihood estimation based distillation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17282–17293. IEEE, 2025.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *Advances in Neural Information Processing Systems*, 38:75460–75482, 2025a.

Zhengyang Geng, Ashwini Pokle, Weijian Luo, Justin Lin, and Zico Kolter. Consistency models made easy. In *International Conference on Learning Representations*, volume 2025, pp. 75158–75185, 2025b.

Anindita Ghosh, Vladislav Golyanik, Taku Komura, Philipp Slusallek, Christian Theobalt, and Rishabh Dabral. SceMoS: Scene-aware 3d human motion synthesis by planning with geometry-grounded tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16443–16453, 2026.

Jingyu Gong, Kunkun Tong, Zhuoran Chen, Chuanhan Yuan, Mingang Chen, Zhizhong Zhang, Xin Tan, and Yuan Xie. Human motion synthesis in 3d scenes via unified scene semantic occupancy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4248–4256, 2026.

Chuan Guo, Yuxuan Mu, Muhammad Gohar Javed, Sen Wang, and Li Cheng. Momask: Generative masked modeling of 3d human motions. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1900–1910. IEEE, 2024.

Prerit Gupta, Shourya Verma, Ananth Grama, and Aniket Bera. Unified multi-modal interactive and reactive 3d motion generation via rectified flow. In *International Conference on Learning Representations*, 2026.

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

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

Vincent Tao Hu, Wenzhe Yin, Pingchuan Ma, Yunlu Chen, Basura Fernando, Yuki M. Asano, Efstratios Gavves, Pascal Mettes, Björn Ommer, and Cees G. M. Snoek. Motion flow matching for human motion synthesis and editing. *arXiv preprint arXiv:2312.08895*, 2023.

Siyuan Huang, Zan Wang, Puhao Li, Baoxiong Jia, Tengyu Liu, Yixin Zhu, Wei Liang, and Song-Chun Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16750–16761, 2023.

Inwoo Hwang, Jinseok Bae, Donggeun Lim, and Young Min Kim. Goal-driven human motion synthesis in diverse tasks. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2911–2921. IEEE, 2025a.

Inwoo Hwang, Bing Zhou, Young Min Kim, Jian Wang, and Chuan Guo. Scenemi: Motion inbetweening for modeling human-scene interactions. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6034–6045. IEEE, 2025b.

Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36:20067–20079, 2023.

Lei Jiang, Ye Wei, and Hao Ni. Motionpcm: Real-time motion synthesis with phased consistency model. *arXiv preprint arXiv:2501.19083*, 2025.

Nan Jiang, Zimo He, Zi Wang, Hongjie Li, Yixin Chen, Siyuan Huang, and Yixin Zhu. Autonomous character-scene interaction synthesis from text instruction. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024a.

Nan Jiang, Zhiyuan Zhang, Hongjie Li, Xiaoxuan Ma, Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, and Siyuan Huang. Scaling up dynamic human-scene interaction modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1737–1747, 2024b.

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

Tianyu Li, Calvin Qiao, Guanqiao Ren, KangKang Yin, and Sehoon Ha. Aamdm: accelerated auto-regressive motion diffusion model. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1813–1823. IEEE, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Pinxin Liu, Luchuan Song, Junhua Huang, Haiyang Liu, and Chenliang Xu. Gesturelsm: Latent shortcut based co-speech gesture generation with spatial-temporal modeling. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10929–10939. IEEE, 2025.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Anh Nguyen, Viet Nguyen, Duc Vu, Trung Dao, Chi Tran, Toan Tran, and Anh Tran. Improved training technique for shortcut models. *Advances in Neural Information Processing Systems*, 38: 92816–92846, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

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.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.

Wenfeng Song, Shi Zheng, Xinyu Zhang, Xingliang Jin, Aimin Hao, Fei Hou, Xia Hou, and Shuai Li. Intentmotion: Learning intent-aware human motion from language in 3d scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2065–2073, 2026.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Sebastian Starke, He Zhang, Taku Komura, and Jun Saito. Neural state machine for character-scene interactions. *ACM Transactions on Graphics*, 38(6):178:1–178:14, 2019.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-Or, and Amit Haim Bermano. Human motion diffusion model. In *International Conference on Learning Representations*, 2023.

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

Yuan Wang, Yali Li, Xiang Li, and Shengjin Wang. Hsi-gpt: A general-purpose large scene-motion-language model for human scene interaction. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7147–7157. IEEE, 2025.

Yuan Wang, Xiang Li, Yali Li, Xuege Hou, and Shengjin Wang. Hsi-gpt2: A dual-granularity large motion reasoning model with diffusion refinement for human-scene interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16432–16442, 2026.

Zan Wang, Yixin Chen, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. HUMANISE: Language-conditioned human motion generation in 3d scenes. In *Advances in Neural Information Processing Systems*, volume 35, pp. 14959–14971, 2022.

Zan Wang, Yixin Chen, Baoxiong Jia, Puhao Li, Jinlu Zhang, Jingze Zhang, Tengyu Liu, Yixin Zhu, Wei Liang, and Siyuan Huang. Move as you say, interact as you can: Language-guided human motion generation with scene affordance. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 433–444. IEEE, 2024.

Hongwei Yi, Justus Thies, Michael J. Black, Xue Bin Peng, and Davis Rempe. Generating human interaction motions in scenes with text control. In *European Conference on Computer Vision*, pp. 246–263, 2024.

Huijie Zhang, Aliaksandr Siarohin, Willi Menapace, Michael Vasilkovsky, Sergey Tulyakov, Qing Qu, and Ivan Skorokhodov. Alphaflow: Understanding and improving meanflow models. In *International Conference on Learning Representations*, volume 2026, pp. 144056–144094, 2026.

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

Kaifeng Zhao, Shaofei Wang, Yan Zhang, Thabo Beeler, and Siyu Tang. COINS: Compositional human-scene interaction synthesis with semantic control. In *European Conference on Computer Vision*, 2022.

Kaifeng Zhao, Yan Zhang, Shaofei Wang, Thabo Beeler, and Siyu Tang. Synthesizing diverse human motions in 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14738–14749, 2023.

Kaifeng Zhao, Mathis Petrovich, Haotian Zhang, Tingwu Wang, Siyu Tang, and Davis Rempe. Autoregressive diffusion with hybrid representation for interactive human motion generation. *ACM Transactions on Graphics (TOG)*, 45(4):1–14, 2026.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5745–5753, 2019.

A GLOBAL SCENE PREPROCESSING

For TRUMANS, we directly use the BEV images provided by SceMoS (Ghosh et al., 2026) without re-rendering them. To construct the height field, we project each scene's original 3D voxel grid, with cubic voxels of side length $2\,\mathrm{cm}$, onto the horizontal plane, yielding a 400×300 heightmap, in which each cell stores the maximum height of the occupied voxels in the corresponding vertical column. For HUMANISE, we follow the data processing protocol of Wang et al. (2024). Specifically, we crop the 3D scenes to a horizontal extent of $4\,\mathrm{m} \times 4\,\mathrm{m}$ and obtain the corresponding spatially aligned human motion. We render BEV images using a top-down camera positioned directly above the horizontal center of each cropped scene and oriented vertically downward. The heightmap is constructed by rasterizing each cropped scene onto a 400×400 horizontal grid with a cell size of $1\,\mathrm{cm} \times 1\,\mathrm{cm}$ and assigning each cell the 99th percentile of the heights of the scene points within it.

B REPRODUCTION OF SCEMOS

Discrepancy Between Paper Description and Released Code. We identify a discrepancy between the inference dependency described in SceMoS (Ghosh et al., 2026) and the default geometry-conditioning path in the released code. The paper describes a planner that generates motion tokens, a geometry-conditioned decoder that produces motion segments token by token conditioned on local heightmaps queried based on previously generated poses, and a trajectory regressor that refines root motion. However, we find that the released code differs from the paper's description in both the geometric conditions supplied to the VQ-VAE decoder and the way these conditions are obtained.

Additional Contact Conditioning. In addition to the local heightmap described in Equation (5) of the paper, the code supplies a contact field consisting of the distances from the 2,000 scene points closest to the human surface at a given time to the pelvis. This contact field is concatenated with the heightmap to condition VQ-VAE decoding, which is inconsistent with the paper's description.

Offline and Online geometry. The official implementation supports two geometry-conditioning settings: *offline* and *online*. When generating a complete motion sequence under the offline setting, which is the default inference setting in the released code, the aforementioned geometric conditions are first precomputed from the ground-truth reference motion and then temporally subsampled and shifted. The model then generates all motion tokens for the sequence and jointly decodes them using the precomputed geometric conditions. *Consequently, this setting leaks ground-truth trajectory and motion information into the model during inference, contrary to the paper's description of querying local geometry from previously generated poses.* The online setting instead computes local heightmaps and contact maps from the initial pose and subsequently generated motions in the static scene, updating these conditions as generation proceeds. It therefore follows the dependency structure described in the paper more closely.

Results and Interpretation. Tab. 4 compares the results reported in SceMoS (Ghosh et al., 2026) with our evaluation results obtained using its released inference code. We find that the offline setting in the official code yields metrics close to those reported in the paper. For example, its FID of 0.32 is close to the reported value of 0.31. In contrast, the online results are substantially worse than the reported results. For example, its FID of 6.67 is approximately 21.5 times the reported value. Fig. 6 provides qualitative examples, showing that offline inference produces substantially smoother motion and much more plausible human-scene interactions than online inference. The gap motivates the hypothesis that the published experiments may have used the official code's offline setting, which exposes ground-truth trajectory and motion information from the test data to the model. This hypothesis is based solely on evaluation results using its released inference code. We remain unable to verify the experimental setup used to obtain the published results or the actual performance of SceMoS described in the original paper.

Results Reported in the Main Paper. Since offline inference yields metrics close to those reported in the SceMoS paper, we use this setting for all SceMoS evaluations in the main paper, including the quantitative results in Tab. 1 and the qualitative results in the corresponding figures.

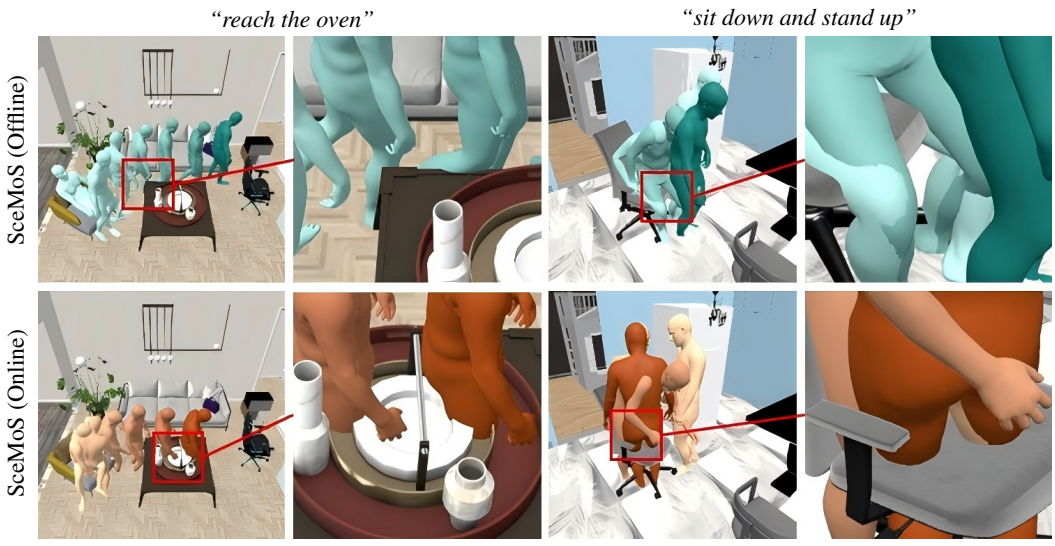

Figure 6: **Qualitative comparison of the offline and online inference settings of SceMoS.** The offline setting yields substantially smoother motion and much more plausible human-scene interactions than online inference.

Table 4: **Published and reproduced SceMoS results on TRUMANS.** The published quality metrics are transcribed from the Table 1 of SceMoS (Ghosh et al., 2026).The paper reports approximately 8 seconds for 80 frames in its Section 4.1, which is not our timing measurement.

Method	Add. SCond.	FID↓	Div.→	PFC↓	Cont.↑	P_{mean} ↓	P_{max} ↓	Time(s)↓
SceMoS (Officially reported)	None	0.31	2.67	0.61	0.98	1.81	11.12	8.000
SceMoS-Offline (Our evaluation)	GT root trajectories	0.32	2.59	0.60	0.98	1.79	11.74	2.969
SceMoS-Online (Our evaluation)	None	6.67	1.88	0.87	0.89	2.94	17.74	3.307

C Additional Details on Baseline Evaluation

C.1 Details of Evaluation on the TRUMANS Dataset

For evaluation on the TRUMANS dataset, we use the data-processing code released by SceMoS (Ghosh et al., 2026) to construct the training and test sets. Specifically, the resulting splits contain 1,269 long motion sequences for training and 408 for testing. Following SceMoS, we further divide these sequences into 80-frame motion clips using a sliding window for both training and evaluation. Given the text and scene conditions associated with each clip, SceMoS and our method natively generate an entire 80-frame motion clip.

In contrast, TRUMANS generates only 16 frames per window, of which the first two are history frames; thus, each autoregressive window produces only 14 new frames. In addition to these history frames, TRUMANS requires a target position for every window and frame-wise action labels. For evaluation, we provide all of these inputs: the ground-truth first two frames as the initial history, the ground-truth action label for each frame, and the ground-truth final pelvis position of every window. Consequently, TRUMANS receives substantially stronger ground-truth constraints than our *SceShortcut*. Because repeated TRUMANS inference cannot produce exactly 80 frames, five autoregressive windows yield 72 frames, whereas six yield 86 frames. We therefore run six windows and retain the first 80 frames for evaluation. The only exception is inference time, for which we report the total runtime of the first five autoregressive windows, averaged over all evaluated clips. Nevertheless, when computing the evaluation metrics using the code released by SceMoS, we find that TRUMANS obtains an FID of 0.94, approximately 2.8 times the reported value of 0.34, while its other metrics are comparable to or better than those reported in the SceMoS paper, as shown in Tab. 5. Because SceMoS does not specify the exact protocol used to evaluate TRUMANS, we use the results reported in its paper for our main comparison and discuss our reproduction here.

Table 5: **Reported and reproduced TRUMANS results on the TRUMANS dataset.** We compare the TRUMANS results reported by SceMoS (Ghosh et al., 2026) with those from our evaluation, for which the metrics are computed using the code released by SceMoS. For inference time, we report the total runtime of the first five autoregressive windows, averaged over all evaluated clips.

Method	Add. SCond.	FID↓	Div.→	PFC↓	Cont.↑	P_{mean} ↓	P_{max} ↓	Time (s)↓
TRUMANS (reported by SceMoS)	GT Sparse pelvis pos.	0.34	2.71	0.58	0.98	1.83	11.75	–
TRUMANS (our evaluation)	GT Sparse pelvis pos.	0.94	2.61	0.57	0.98	1.71	11.49	6.048

Table 6: **Additional ablation studies of feature injection on the TRUMANS dataset.** We evaluate alternative mechanisms for injecting conditioning features into *SceShortcut*. The best and second-best results are **bold** and underlined, respectively.

Configuration	FID↓	PFC↓	Cont.↑	P_{mean} ↓	P_{max} ↓	FTE↓	FRE↓	CTE↓	CRE↓	Time (s)↓
(c) Scene conditioning										
w/o global semantic features	0.39	0.62	0.97	1.86	11.93	0.31	21.9	0.32	23.5	0.053
w/o global height-field features	0.38	**0.60**	0.97	1.88	11.84	0.32	23.1	0.34	24.3	**0.052**
3D occupancy instead of heightmaps	**0.31**	0.61	0.98	**1.74**	11.53	**0.29**	22.7	**0.30**	24.1	0.064
(d) Feature injection										
Concatenation + cross-attention	0.33	0.61	0.97	1.81	11.75	0.30	22.3	0.33	23.8	0.056
Separate cross-attention + MoE fusion	0.37	0.61	**0.99**	1.85	12.01	0.30	22.5	0.32	23.9	0.061
Full model (Ours)	0.32	0.61	0.98	1.77	**11.48**	**0.29**	**21.7**	0.31	**23.4**	0.058

For TeSMo (Yi et al., 2024), SceMoS states that its interaction branch is fine-tuned before quantitative evaluation but does not provide the specific fine-tuning or evaluation settings. We therefore directly report the TeSMo results presented in SceMoS (Ghosh et al., 2026).

C.2 Details of Evaluation on the HUMANISE Dataset

For HUMANISE, we follow the train-test split used by Afford-Motion (Afford@2) (Wang et al., 2024), comprising 16,523 training and 3,125 test motion-scene pairs. Because the longest motion sequence in HUMANISE contains 132 frames, we train our model to support sequences of up to 132 frames. Shorter sequences are zero-padded, and the padded frames are excluded from attention using an attention mask. We compute all quantitative metrics for our method and the baselines using the evaluation code provided by Afford-Motion.

D Additional Ablations

Here we provide additional ablations on scene conditioning and feature injection, as shown in Tab. 6.

Scene Conditioning. Removing either global semantic features or global height-field features degrades performance in terms of FID, Cont., and scene penetration, demonstrating the benefits of both types of global scene information. We also replace the 2D height field with a 3D voxel occupancy representation, following TRUMANS. This alternative achieves comparable overall quality at a higher inference cost, supporting our choice of the height-field representation.

Feature Injection. We evaluate two alternative mechanisms for injecting text and global or local scene features into the motion model: (1) concatenating the text and scene tokens and jointly fusing them with the noisy motion tokens through a single cross-attention operation; and (2) using parallel cross-attention branches to fuse the text and scene modalities separately with the noisy motion tokens, followed by a weighted mixture-of-experts (MoE) fusion module. Compared with these variants, our full model achieves better overall motion quality and scene compatibility, with lower FID and scene penetration while maintaining competitive PFC and Cont., supporting the effectiveness of our feature-injection design.

E LIMITATIONS

Although shortcut modeling substantially reduces the number of sampling steps and accelerates motion generation, using an extremely small sampling budget may occasionally introduce local motion artifacts. As illustrated in Fig. 7, one-step generation can exhibit mild human-scene penetration or jitter in fine-grained interaction regions. Increasing the number of inference steps allows the model to progressively refine the motion and can alleviate these artifacts, although it incurs additional computational cost. Training on larger and higher-quality datasets may mitigate this limitation.



Figure 7: **Qualitative illustration of the limitations under different sampling budgets.** From left to right, we show motions generated using one, two, and four sampling steps, with enlarged views highlighting local human-scene interaction details. One-step generation may occasionally exhibit local artifacts, while increasing the number of sampling steps can alleviate them.