SCALABLE RF SIMULATION IN GENERATIVE 4D WORLDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Radio Frequency (RF) sensing has emerged as a powerful, privacy-preserving alternative to vision-based methods for various perception tasks. However, building high-quality RF datasets in dynamic and diverse environments remains a major challenge. To address this, we introduce WAVEVERSE, a prompt-based, scalable framework that simulates realistic RF signals from generated indoor scenes with human motions. WAVEVERSE introduces a language-guided 4D world generator and a physics-based signal simulator that enables realistic simulation of RF signals in diverse environments. Experiments validate the effectiveness of our method, and we present two case studies showing WAVEVERSE not only enables data generation for highly flexible RF imaging configurations for the first time, but also consistently achieves performance gains in both data-limited and data-rich scenarios.

1 INTRODUCTION

Radio frequency (RF) sensing has emerged as a compelling modality for tasks such as 3D imaging, human activity recognition, and health monitoring (Singh et al., 2019; Zhao et al., 2021; Lai et al., 2024). In safety-critical or low-visibility scenarios, RF-based methods Sun et al. (2021a;b); Lai et al. (2024) offer high-resolution imaging despite fog, smoke, or occlusion. At the same time, RF sensors do not capture images or videos, making them inherently privacy-preserving and well-suited for contactless and continuous health monitoring, including vital sign monitoring (Zhao et al., 2016; Ha et al., 2020), sleep analysis (Zhao et al., 2017; He et al., 2025), and mental health assessment (Ha et al., 2021; Liang et al., 2023). Despite these advantages, acquiring large-scale and high-quality RF sensing datasets remains challenging. Building such datasets requires capturing a wide range of room layouts, human activities, and individual differences, all of which demand significant cost and effort. Worse still, RF sensing systems differ widely in hardware configurations (i.e., bandwidth, antenna layout, and signal modulation), making it difficult to share or reuse data across systems. As a result, unlike vision or audio, RF sensing lacks standardized and unified benchmark datasets, limiting generalization across systems and slowing research progress.

Recent efforts have explored both physics-based simulation (Cai et al., 2020; Zhang et al., 2022) and learning-based synthesis (Chen & Zhang, 2023; Chi et al., 2024) to address the challenges. However, existing approaches focus on signal interactions with human bodies while neglecting the surrounding environment. This is particularly problematic for RF sensing, where multipath propagation (i.e., multi-bounce reflections with surrounding structures like walls, floors, and objects) significantly affects the received signal and is a key factor limiting generalization (Wang et al., 2020; Zhang et al., 2023a). Moreover, learning-based synthesis (Chen & Zhang, 2023; Chi et al., 2024) still requires a large training dataset to begin with and do not generalize beyond a specific sensor configuration.

In this paper, we introduce WAVEVERSE, a hybrid generation–simulation framework for synthesizing realistic and diverse RF signals. As illustrated in Fig. 1, WAVEVERSE combines 4D world generation with physics-based RF simulation. Specifically, it leverages the emergent capabilities of Large Language Models (LLMs) (Achiam et al., 2023; Hurst et al., 2024) to generate diverse 3D indoor environments and dynamic human motions within them. Given the 4D world (i.e., a 3D environment with dynamic human motions), WAVEVERSE employs a ray tracing engine that accurately models multipath propagation and provides phase-accurate signals across antennas and over time. This hybrid design combines the best of both worlds: generative diversity from 4D synthesis and physical realism from RF simulation. The use of explicit mesh representations for 3D layouts provides additional benefits. It enables aligned supervision for RF learning tasks (e.g., depth estimation, semantic

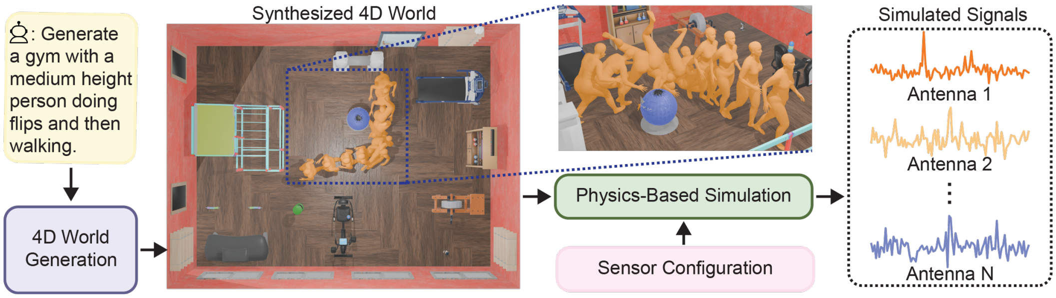


Figure 1: Given input text and sensor configuration, WaveVerse generates 4D worlds with moving humans in 3D environments and simulates the received RF signals using physics-based modeling.

segmentation, human poses) and supports RF simulation with flexible sensor configurations for a wide range of downstream applications, which are difficult to realize with existing methods.

WaveVerse introduces two key innovations to achieve these new capabilities. The first is about spatial conditioning for human motion generation. Prior approaches Xie et al. (2024); Dai et al. (2024) condition motion generation on *trajectories*, which are *time-indexed* sequences of joint positions. These trajectories prescribe not only *where* a person moves but also *when and how fast*, encoding velocities, durations, and frame-level details. While effective for strict and fine-grained control, this formulation is over-constrained, requires substantial manual effort to design, and ultimately restricts generative models from producing diverse motions conditioned on the trajectories. In contrast, we introduce a *path*-based conditioning strategy that provides spatial guidance *without temporal assignment*. A path is defined as a set of waypoints specifying *where* the motion should occur, while leaving velocity, style, and duration flexible. This simpler representation enables automatic path generation and eliminates the need for manual trajectory design. It also allows the path generator to focus on high-level semantics (i.e., aligning motion intent with the generated environments) while leaving motion details to a separate human motion generator. As a result, path-based conditioning achieves spatially realistic yet diverse and natural motion generation.

Our second innovation is a physics-based simulation framework with phase-coherent ray tracing, which enables accurate and consistent modeling of signal phase. Prior methods (Ren et al., 2024; Chen & Zhang, 2023) neglect spatial and temporal phase coherence. Yet such coherence is essential for many RF sensing tasks including imaging and vital sign monitoring. In contrast, our simulator explicitly preserves phase information across space and time, ensuring stable beamforming, Doppler estimation, and other phase-sensitive applications. Grounded in physical modeling, our approach generates high-fidelity signals directly, without requiring post-hoc learning-based signal refinement.

We evaluate WaveVerse through extensive experiments. In microbenchmarks, we evaluate human motion generation under text and path conditioning, showing our state-aware causal transformer outperforms baselines, including diffusion-based models. We further show, with this method, WaveVerse generates human motions that are diverse and aligned with environments. We then compare our phase-coherent ray tracing with conventional RF ray tracing and observe significant improvements in phase-sensitive tasks, including circular beamforming imaging, respiration monitoring, and Doppler estimation, yielding high-fidelity signals. Finally, we conduct two case studies on high-resolution RF imaging and human activity recognition, showing that WaveVerse, using only ray tracing without neural network refinement, enables data simulation for highly flexible RF imaging configurations for the first time and achieves consistent performance gains in both data-limited and data-rich settings.

2 RELATED WORK

RF Simulation. Ray tracing has been widely used for radio propagation modeling, with early efforts addressing communication-centric applications such as signal coverage in static scenes (Yun & Iskander, 2015; Hoydis et al., 2023; Yun & Iskander, 2024). For applications in RF sensing, prior work (Erol et al., 2020; Ahuja et al., 2021; Zhang et al., 2022; Xue et al., 2023) focused on the signal interaction with human bodies neglecting environments and requires learning-based signal refinement. Some methods (Ren et al., 2024; Chen et al., 2025) use ray tracing for signal simulation but fall short of modeling spatial and temporal phase coherence. Inspired by the progress in image generation (Kingma et al., 2013; Goodfellow et al., 2020; Ho et al., 2020), data-driven methods (Chen & Zhang, 2023; Chi et al., 2024) combine ray tracing with neural networks for signal synthesis. However, they rely on large annotated datasets, offer limited controllability, lack physical interpretability, and cannot ensure multipath effects or phase coherence. Full-wave solvers like HFSS (Cendes, 2016) provide accurate simulation but are computationally prohibitive for large-scale, dynamic indoor scenes. In contrast, our work develops ray tracing with explicit spatial and temporal phase coherence, enabling high-fidelity RF simulation without additional learning-based refinement.

Human Motion Generation. The generation of human motion has long been studied, with recent efforts focusing on enhancing controllability. Text-based conditioning Guo et al. (2022a;b); Zhang et al. (2023b); Tevet et al. (2023); Shafir et al. (2023); Jiang et al. (2023); Guo et al. (2024; 2025) offers an intuitive interface but ignores environment context, yielding unrealistic movements under spatial constraints. To address this issue, several methods Tevet et al. (2023); Shafir et al. (2023); Wan et al. (2024); Xie et al. (2024); Dai et al. (2024) additionally introduce trajectories of explicit joint positions at designated frames. While effective, such time-indexed trajectories are over-constrained, as they require predefined durations, velocities, and careful alignment with text conditions, making the process labor-intensive, difficult to scale, and limiting generalization. Alternative approaches Yi et al. (2024); Liu et al. (2024); Hwang et al. (2025) have explored motion generation directly within 3D scenes, but they either lack text conditioning capabilities or still require time-indexed inputs such as joint poses at specific frames and motion durations. Similar to trajectory-based methods, they impose substantial preparation overhead and ultimately limit the scalability of practical generation pipelines. Conversely, our path conditioning inherently addresses these issues and allows for practical and scalable generation while providing high diversity and generalization.

3 METHOD

WaveVerse is an automated LLM-powered framework for simulating realistic RF signals in 3D indoor environments with human motions. As a prompt-driven framework, WaveVerse can be used either interactively by a human user or fully automated by an LLM agent. Given a text description of an indoor environment, WaveVerse generates a text-aligned indoor environment with dynamic human activities, and finally simulates RF signals of the scene. This section describes the two core components of WaveVerse: (1) a 4D scene generator that synthesizes diverse indoor 4D scenes (Sec. 3.1), and (2) a phase-coherent ray tracing engine for signal simulation (Sec. 3.2).

3.1 4D WORLD GENERATION

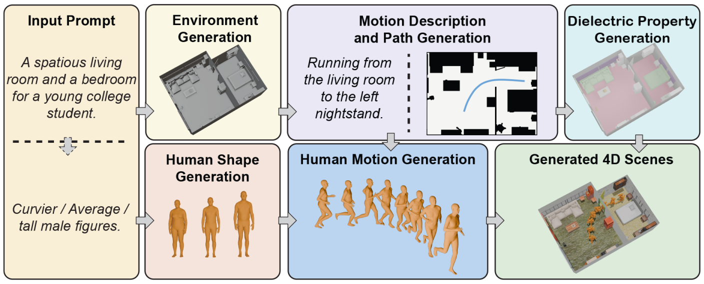


Figure 2: Overview of 4D World Generation.

WaveVerse utilizes a prompt-based pipeline to enable fully automated 4D world generation. Given a text description of the desired environment, whether provided by a user or an LLM, WaveVerse first constructs a semantically aligned 3D environment along with corresponding human body shapes. To generate realistic motion within a scene, WaveVerse first generates text descriptions and paths automatically, which are then used as conditions for our state-aware causal transformer for motion generation. In addition, to support realistic and physics-based RF simulation (Sec. 3.2), WaveVerse also assigns dielectric properties to scene objects with an LLM.

3D Environment and Human Shape Generation. WaveVerse begins with a text description of the environment (Fig. 2). We build on an existing generation pipeline Yang et al. (2024) to produce a structured layout, including floor plans, object categories, and placements, ultimately yielding a mesh representation of the indoor environment. This explicit 3D representation serves as a foundation for simulating RF signals as well as other modalities such as RGB images and depth maps. For human modeling, WaveVerse uses the SMPL model (Loper et al., 2023), a parametric human mesh that can be animated by adjusting pose and shape parameters. The shape parameters can be manually specified or automatically generated using a finetuned LLM (Árbol & Casas, 2024), which is conditioned on plausible body descriptions inferred from the input environment text by a general-purpose LLM.

Motion Description and Path Generation. To enable scalable motion generation for RF-based applications, WaveVerse animates SMPL with sequences of generated joint positions, referred to as motion throughout this paper. Our focus aligns with RF tasks (Singh et al., 2019; Pan et al., 2024), which require diverse whole-body dynamics rather than object-centric interactions or simple locomotion. The key challenge is to generate human motion that matches the semantic context of the environment while ensuring diversity and spatial realism, such as avoiding wall penetrations. One approach to achieve such control is to pair text descriptions with time-indexed *trajectory* inputs, that is, precise 3D joint positions specified at key frames, consistent with the text and environment. However, it requires careful alignment with text and extensive manual specification, making it labor-intensive and difficult to scale. Additionally, by fixing joint positions, durations, and velocities in advance, it effectively predetermines the motion and reduces flexibility, limiting generalization.

To address this challenge, we decompose motion generation into two stages. Given a text prompt of the environment, an LLM first produces a motion description, like “wave the arm”, and specifies the start and end 2D positions on the floor, which can also be provided by users. We replace trajectory constraints with *paths*, a set of L spatial waypoints that guide where the person should move without prescribing velocity or duration. Such paths can be readily generated with path-finding algorithms given start and end points. For model training, we derive paths by downsampling and projecting the pelvis trajectory from dataset motions into $L = 64$ evenly spaced 2D waypoints. We delegate the motion generation task to later models, while LLMs focus on high-level reasoning.

Conditional Human Motion Generation. The second step is to generate motion sequences conditioned on the input texts and paths. Since the path does not specify motion duration, we adopt an autoregressive model that dynamically determines when to terminate the sequence, unlike existing methods that generate human motion with a pre-defined duration (Tevet et al., 2023; Xie et al., 2024; Dai et al., 2024). Specifically, motion sequences are first tokenized using VQ-VAE (Van Den Oord et al., 2017), achieving motion tokens $X = [m_1, m_2, \ldots, m_n, m_{\text{end}}]$, where $m_i \in \{1, \ldots, M\}$ indexes a learned codebook, and m_{end} denotes the end of the sequence. The motion description is encoded using CLIP (Radford et al., 2021), while the 2D waypoint sequence is processed through an MLP-based position encoder, producing condition embeddings $c = (c_{\text{text}}, c_{\text{path}_0}, \ldots, c_{\text{path}_L})$.

While existing autoregressive models (Zhang et al., 2023b) generate motion via next-token prediction, learning the distribution $P(m_n \mid c, m_0, \ldots, m_{n-1})$, we find this formulation struggles to align motion with the input path. Inspired by reinforcement learning (Kaelbling et al., 1996), we view next-token prediction as a sequential decision-making process, where each token is an action. We argue that the absence of explicit spatial context at each decision step limits path adherence. To address this, we introduce a *state-aware causal transformer* shown in Fig. 3, conditioning each prediction on the current spatial state. Formally, the next-token distribution is modeled as $P(m_n \mid c, m_0, s_0, \ldots, m_{n-1}, s_{n-1})$, where s_i encodes the 2D position at the final frame up to token m_i, with the same position encoder.

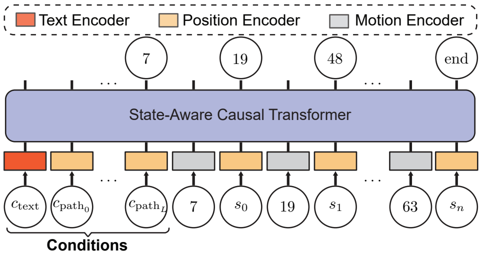


Figure 3: State-Aware Causal Transformers.

Despite the benefits of spatial state conditioning, we observe the model overfits by relying heavily on path information, resulting in poor text alignment. To mitigate this and promote balanced conditioning, we introduce a path-masking strategy during training. We first sample a masking ratio $[r_{\min}, r_{\max}]$ to determine the number of waypoints to mask. Then, we iteratively select and mask random contiguous segments of length ℓ. If further masking is needed and no full segments remain, we continue by randomly masking individual waypoints until the target ratio is reached. We find that this sequential masking strategy improves generalization and enhances text-motion alignment (Sec. 4.1).

Dielectric Property Generation. To further enhance physical realism, WaveVerse models dielectric properties following the ITU-R P.2040-2 recommendation (Series, 2015), which provides frequency-dependent parametric models for permittivity and conductivity along with validated parameter sets for 14 common materials. These parameters define physically meaningful dielectric constants that we use directly in the simulator. To extend beyond these 14 materials, we sample objects from our asset library and prompt the LLM to propose additional material categories and follow the same ITU parametric model. We retain only categories whose dielectric values fall within documented physical ranges, resulting in a library of 24 materials. During scene generation, instead of generating dielectric constants from scratch, we prompt LLM to simply assign each object to the most appropriate category from this curated library. This two-stage approach, physics-based

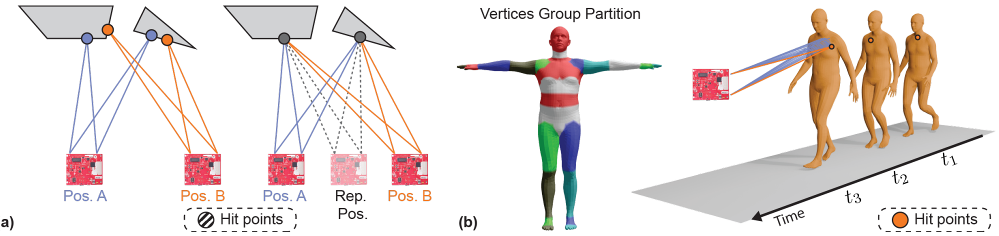


Figure 4: Illustration of phase-coherent ray tracing. (a) shows spatial coherence by tracing consistent paths across two radar locations. (b) depicts temporally coherent ray tracing for a moving person across timestamps t_1, t_2, and t_3. For clarity, the rays at t_2 and t_3 are omitted.

parametric modeling followed by LLM-based categorization, ensures that all dielectric values remain physically validated while still allowing semantic description to guide material selection.

3.2 RF SIGNAL SIMULATION

Given the generated 4D scenes, WaveVerse employs ray tracing to simulate RF signals. Existing RF ray tracing engines, however, inherit practices from computer graphics, where the focus is on modeling signal amplitude and rays are cast stochastically (Cook, 1986; Nimier-David et al., 2019). As a result, RF simulation similarly casts rays randomly over a spherical or conical distribution, resulting in inconsistent ray-surface interactions across frames and radar positions (Chen & Zhang, 2023; Ren et al., 2024). This inconsistency poses significant challenges for RF applications, where signal phase plays a critical role. For example, high-resolution RF imaging distinguishes objects at the same range but different angles by leveraging phase differences through beamforming. Similarly, Doppler-based velocity estimation relies on phase shifts across frames caused by object motion. To address this, we introduce *phase-coherent ray tracing* that operates on the scene mesh and ensures consistent ray-surface interactions across different radar positions and over time as objects move. This preserves signal phase coherence, enabling accurate simulation of phase-dependent RF phenomena.

RF Simulation with Ray Models. Ray tracing models wave propagation as a collection of discrete paths connecting the transmitter (Tx) and receiver (Rx) through the environment. Let $\{\mathcal{P}_k\}_{k=1}^{K}$ denote the set of valid propagation paths identified by ray tracing. Each path $\mathcal{P}_k$ is characterized by four parameters: the propagation delay τ_k; a complex coefficient a_k, whose magnitude encodes attenuation due to path loss and interactions with scene surfaces, and whose angle represents accumulated phase shifts; the angle of departure (AoD) θ_k at Tx; and the angle of arrival (AoA) φ_k at Rx. The channel impulse response (CIR) $h(t)$, which describes how an impulse propagates from Tx to Rx, is modeled as the superposition of all paths: $h(t) = \sum_k a_k \cdot G_{\text{Tx}}(\theta_k) \cdot G_{\text{Rx}}(\varphi_k) \cdot \delta(t - \tau_k)$, where G_{Tx} and G_{Rx} denote the antenna gain patterns of the transmitter and receiver, capturing their directionality, and $\delta(t)$ is the Dirac delta function. Any signal received by Rx can then be computed as the convolution between the transmitted signal and the CIR.

Phase-Coherent Ray Tracing. As mentioned above, conventional ray tracing methods fall short in preserving phase coherence, as they cast rays stochastically, resulting in different ray-surface interactions even for nearby radar positions (Fig. 4(a), left). This issue becomes more severe in dynamic scenes with moving humans, where changes in geometry cause rays to strike entirely different surface points across frames, breaking temporal phase coherence shown in Fig. 4(b).

To overcome these challenges, we propose phase-coherent ray tracing that ensures consistent ray-surface interactions across space and time. To achieve *spatially-coherent ray tracing*, i.e., ensuring coherent phase variation across different radar locations, we generate paths for each radar from a fixed set traced from a representative reference radar. Specifically, assume we are synthesizing signals for N radars with poses $\{(\mathbf{t}_n, \mathbf{r}_n)\}$ for $n = 1, \ldots, N$, where $\mathbf{t}_n$ and $\mathbf{r}_n$ denote the position and rotation of the transmitter and receiver. We define a reference $(\mathbf{t}_0, \mathbf{r}_0)$ as the geometric center of all radar positions, and trace rays uniformly over a sphere, to obtain paths $\{\mathcal{P}_k\}$ between $\mathbf{t}_0$ and $\mathbf{r}_0$. For each path $\mathcal{P}_k$, we represent it with a sequence of 3D points $\mathcal{P}_k = [\mathbf{t}_0, \mathbf{p}_0, \cdots, \mathbf{p}_{D_k}, \mathbf{r_0}]$ where $\mathbf{p}_d$ denotes the d-th surface interaction point along the path and D_k denotes the number of encountered surfaces.

To generate paths for a radar with poses $(\mathbf{t}_n, \mathbf{r}_n)$, we modify each reference path $\mathcal{P}_k$ by replacing the original transmitter and receiver positions with the current ones, as shown in the right side of Fig. 4(a). We then compute the CIR for each modified path using updated propagation delay, attenuation, phase, AoD, and AoA. Occlusion checks are further performed on the achieved paths, and any blocked paths are discarded. By preserving consistent surface interaction points, our approach ensures spatial phase coherence across radars with various poses while avoiding redundant ray tracing.

To obtain *temporally-coherent ray tracing*, i.e., coherent phase changes as humans move within the scene, we remap ray-surface interactions from individual vertices to semantically or spatially coherent groups. While ray tracing is performed independently over time, we enable temporal coherence by expanding ray hits over stable vertex groups that persist across frames.

Specifically, we partition all vertices $\mathcal{V} = \{\mathbf{v}_m\}_{m=1}^{M}$ of a human mesh into G disjoint, semantically coherent groups $\{\mathcal{V}_g\}_{g=1}^{G}$, where $\cup_g \mathcal{V}_g = \mathcal{V}$, with a grouping function $\mathcal{G} : \mathbf{v}_m \mapsto \{1, \ldots, G\}$. At each timestamp t, ray tracing yields a set of paths $\{\mathcal{P}_k^{(t)}\}$. When a path $\mathcal{P}_k^{(t)}$ intersects the human mesh at point $\mathbf{p}_d^{(t)}$, we associate it with a representative vertex $\hat{\mathbf{p}}_d^{(t)}$, a fixed vertex of the intersected face, to enable consistent grouping. We then expand the path by replacing the hit point with all vertices within the same group, i.e., those satisfying $\mathcal{G}(\mathbf{v}_m) = \mathcal{G}(\hat{\mathbf{p}}_d^{(t)})$. More specifically, the following set of paths $\left\{ \left[\mathbf{t}, \ldots, \mathbf{p}_{d-1}^{(t)}, \mathbf{v}_m, \mathbf{p}_{d+1}^{(t)}, \ldots, \mathbf{r}\right] \,\middle|\, \mathcal{G}(\mathbf{v}_m) = \mathcal{G}\left(\hat{\mathbf{p}}_d^{(t)}\right) \text{ for } m \in \{1, \ldots, M\} \right\}$ are created to replace $\mathcal{P}_k^{(t)}$. We perform occlusion checks on the expanded paths and denote the number of valid paths as N_{valid}. For each valid path, we compute propagation delay, attenuation, phase, AoD and AoA. To preserve overall signal energy, attenuation is further divided by N_{valid}. In practice, we expand only the first hit point from the transmitter, both to avoid exponential growth from higher-order reflections and because single-bounce paths typically dominate received energy due to lower propagation loss.

Flexible Configuration. WaveVerse generalizes to a wide range of radar configurations, including arbitrary antenna positions and orientations, gain patterns, frequency bands, and sampling rates, making it adaptable to diverse hardware setups. This flexibility arises because our CIR modeling inherently accounts for these factors, allowing the same formulation to be applied consistently across different configurations. By simulating received signals through convolution with the transmitted waveform, WaveVerse supports diverse RF protocols. Additionally, relying on explicit physical modeling, WaveVerse scales to unseen conditions while achieving accurate and reliable signal behavior, offering robustness and scalability that are difficult to achieve with data-driven methods.

4 EXPERIMENTS

In this section, we evaluate the 4D world generation and the signal simulation in WaveVerse. We begin with benchmarks and ablation studies of the proposed state-aware causal transformer for text and path conditioned motion generation, and analyze the generated 4D world. We then assess the benefits of phase-coherent ray tracing over existing baselines on three phase-sensitive RF tasks. Finally, we demonstrate the utility of WaveVerse in two real-world case studies.

4.1 PERFORMANCE OF HUMAN MOTION GENERATION

Dataset and Evaluation Metrics. We evaluate models over the HumanML3D (Guo et al., 2022a) dataset for benchmarks. It contains 14,146 captioned human motion sequences. In all experiments, we fix $L = 64$. More details can be found in Appendix A.1. Following the evaluation protocol from OmniControl (Xie et al., 2024), we report *R-Precision* to quantify the alignment between the text and the motion, the *Frechet Inception Distance (FID)* to assess motion quality, and the *Diversity* score to measure motion variability. To assess spatial alignment with the path condition, we define the *Path Error* as the average per-point L_2-distance between the generated path and the ground-truth path, and the *Ending Error* as the deviation at the last timestamp. For both errors, we use the mean and the percentage of samples exceeding thresholds of 20 and 60 cm to characterize the distribution.

Comparison to Baselines. We adopt four open-source, state-of-the-art motion generation methods as baselines, selected as the closest in design to ours: the diffusion-based MDM (Tevet et al., 2023), OmniControl (Xie et al., 2024) and MotionLCM (Dai et al., 2024), and the autoregressive T2M-GPT (Zhang et al., 2023b). Details of the model adaptations are provided in Appendix A.1.

As shown in Tab. 1, our method consistently outperforms all baselines across R-Precision, FID, and path-following metrics, demonstrating better motion quality and alignment with input conditions. We defer detailed analysis to Appendix A.1, but emphasize that the gains come from our proposed designs rather than the autoregressive structure. Crucially, T2M-GPT, on which our method is built, underperforms diffusion-based baselines, whereas our approach achieves better performance.

Method	Architecture	R-Prec. ↑	FID ↓	Div. →	Path Error ↓		Ending Error ↓	
					> 20 cm	> 60 cm	> 20 cm	> 60 cm
Ground Truth		0.797	0.002	9.503	0.	0.	0.	0.
MDM	Diffusion	0.719	0.295	**9.462**	0.547	0.207	0.666	0.367
OmniControl	Diffusion	0.751	0.319	9.279	0.239	0.083	0.330	0.152
MotionLCM	Diffusion	0.739	0.754	9.588	0.315	0.055	0.468	0.177
T2M-GPT	Autoregressive	0.691	0.377	9.736	0.406	0.127	0.545	0.255
Ours	Autoregressive	**0.755**	**0.238**	9.445	**0.208**	**0.045**	**0.325**	**0.111**

Table 1: Text and path conditioned motion generation performance. **Bold** for the best and underline for the second best. R-Prec.: R-Precision; Div.: Diversity.

Setting	Model / Variant	R-Precision ↑	FID ↓	Mean Path Error ↓	Mean Ending Error ↓
Components	Ours	0.755	0.238	0.151	0.287
	w/o Mask	0.643	0.747	0.192	0.325
	w/o State	0.757	0.422	0.250	0.460
	w/o Mask & State	0.691	0.377	0.274	0.528
Masking Rate	[0.5, 0.9]	0.755	0.238	0.151	0.287
	[0.1, 0.5]	0.691	0.396	0.171	0.312
	[0.1, 0.9]	0.713	0.298	0.160	0.303
Segment Length	5 Points	0.755	0.238	0.151	0.287
	10 Points	0.763	0.342	0.207	0.393
	15 Points	0.776	0.339	0.228	0.403

Table 2: Ablation study for key components and hyperparameters in our model.

Physical Plausibility. In addition to text and path alignment, the generated motions should exhibit realistic physical behavior. Beyond achieving a low FID score, which indicates that the generated motions are natural and plausible in terms of velocity, we further quantify physical plausibility using two additional metrics: *Skating Ratio*, which measures foot sliding, and *Bone-Length Variance*, which measures the temporal stability of skeletal geometry. Our model attains a skating ratio of 0.067, closely matching 0.057 for real data, and a bone-length variance of 1.78 cm^2, indicating stable limb lengths over time. These results confirm that our motions are not only well aligned with the input texts and paths, but also physically plausible and closely consistent with real human dynamics.

Ablation Studies. We validate the key components of the state-aware causal transformer introduced in Sec. 3.1 through three ablation studies, summarized in Tab. 2. First, without path masking, the model overfits to path conditions, resulting in degraded motion quality and worse text alignment, evidenced by higher FID and lower R-Precision. Path masking alleviates this issue and also improves generalization in path following. We explore alternative approaches in Appendix A.1, but they yield suboptimal performance. We also show removing state information markedly reduces path-following capability, underscoring its importance. Second, for the masking rate range $[r_{\min}, r_{\max}]$, our choice $[0.5, 0.9]$ outperforms both $[0.1, 0.5]$ and $[0.1, 0.9]$, showing that higher rates better balance reliance on path and text. Finally, varying contiguous masking length ℓ reveals a trade-off. Shorter segments (5 points) enhance path alignment and lower FID, whereas longer ones (10 or 15 points) improve text alignment but substantially degrade path following. We therefore adopt 5 points in our model.

4.2 PERFORMANCE OF GENERATED 4D SCENES

Scene Evaluation. We evaluate both the quality and diversity of the generated 4D scenes, which couple indoor environments with human motion. Across 120 trials, our pipeline achieves a 95.83% success rate, producing 115 unique environments spanning a broad range of room types and objects, with two motion sequences synthesized per scene. Failures are mainly due to floor plan errors in environment generation or overly constrained layouts that hinder path generation. In total, the dataset contains 920 unique objects (averaging 25 per scene), 47 room categories, and 24 dielectric materials, with an average motion duration of 4.5 seconds. To assess spatial compatibility, we report a collision ratio of 2.35%, the fraction of motion frames with collisions with the environment, and a cumulative collision depth of 12.23 cm on average, indicating total interpenetration per motion. These results show that the generated motions conform well to the environments. We further provide qualitative results in Fig. 5, showcasing diverse room layouts, furniture configurations, and semantically consistent human motion, highlighting the scalability of our pipeline. For computation cost, we report a detailed breakdown of the execution time of each component in Appendix A.2.

Figure 5: Examples of generated 4D scenes. Texts above each scene are prompts for the environment (top) and motion (bottom) generation. See Appendix A.2 for more details.

4.3 PERFORMANCE OF PHASE-COHERENT RAY TRACING

To evaluate the effectiveness of our phase-coherent ray tracing, we conduct three benchmarks that require phase coherence. The baseline ray tracing follows existing methods (Ren et al., 2024; Chen et al., 2025), reflecting standard practice in prior work. Our method differs only by incorporating spatial and temporal phase coherence, ensuring fair comparison. In addition, we compare the simulated signals with real-world measurements and Ansys HFSS simulations, as detailed in Appendix A.3.

Spatial Phase Coherence. To evaluate spatial phase coherence, we adopt the panoramic imaging setup of Lai et al. (2024), combining signals from 1,200 radar positions and orientations arranged along a circular path. Fig. 6 shows the imaging results from two random environments generated in Sec. A.3 using the beamforming algorithm. The improved image clarity highlights the role of spatial phase coherence, ensuring that wavefronts remain aligned across all radar poses. Notably, the ghost reflections in the imaging result show our simulation captures multipath effects that are difficult to guarantee in learning-based methods. These results demonstrate the importance of spatial phase coherence for downstream RF applications. More examples are provided in Appendix A.3.

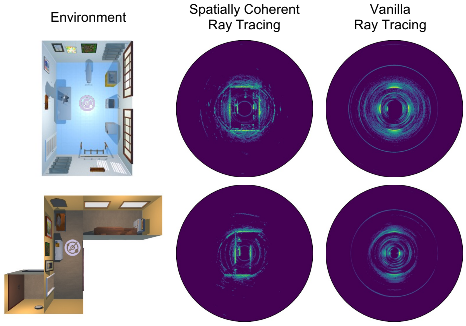


Figure 6: Panoramic imaging results with and without spatial phase coherence.

Temporal Phase Coherence. As discussed in Sec. 3.2, stochastic ray casting fails to preserve phase in dynamic settings, making it fundamentally unsuitable. For the baseline, we instead fix rays cast across time, though it deviates from standard practice. We validate temporal coherence on a respiration tracking task by animating the SMPL with real breathing signals from Li et al. (2024), generating 500 seconds of data across 40 sequences. As the chest moves, minute changes to the radar are captured by phase (Zhao et al., 2016). We extract this phase from simulated signals and convert it into distance change. With temporal coherence, the reconstructed curves achieve 0.08 RMSE and 8.89 DTW against ground truth, significantly outperforming the baseline, 0.14 and 12.68. We further simulate a sinusoidally moving sphere and generate range–Doppler heatmaps via range and Doppler FFT, where our approach again outperforms the baseline. We provide qualitative results in Appendix A.3. These experiments highlight the importance of coherent ray-surface mapping for stable phase tracking and reliable Doppler estimation in dynamic RF scenes.

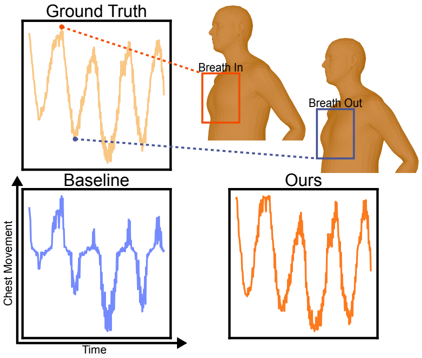


Figure 7: Recovered chest motion with and without temporal phase coherence.

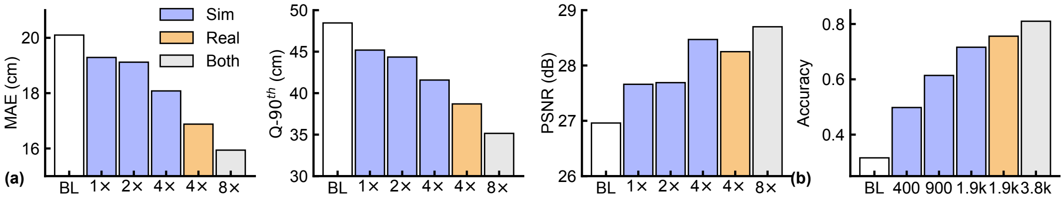


Figure 8: Performance comparison over the baseline with varying amounts of additional real and simulated data on: (a) high-resolution RF imaging and (b) human activity recognition. BL: Baseline.

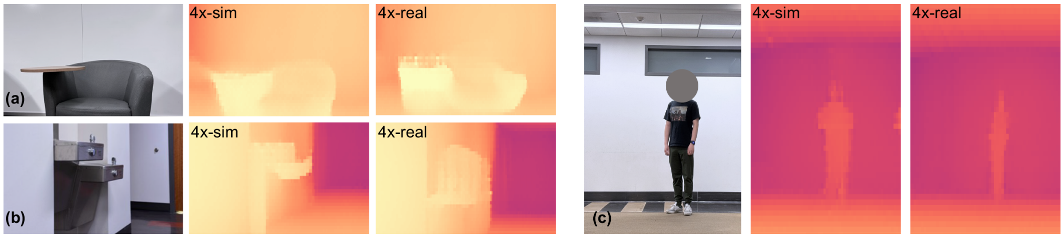


Figure 9: Improved imaging quality over objects. (a) Armchair. (b) Water fountain. (c) Human.

4.4 CASE STUDIES

Having established the effectiveness of individual components, we now evaluate the full pipeline of WAVEVERSE in real-world scenarios. To this end, we conduct two case studies on RF-based applications with publicly available data: high-resolution imaging (Lai et al., 2024) and human activity recognition (Singh et al., 2019). We evaluate each task under two conditions: a limited-data setting that reflects practical constraints, and a data-rich setting where more real-world samples are available, and compare with prior work to highlight the advantages of our approach.

High-Resolution Imaging. We first evaluate dense depth prediction from RF signals with the ML model of Lai et al. (2024), which employs a rotating radar setup adopted earlier in Sec. 4.3. We apply the same cross-building protocol, where the model is trained on RF data from 11 buildings and evaluated on 1,000 frames from a held-out building. To improve depth prediction under limited data, we augment training with simulated RF signals and depth supervision generated by WAVEVERSE. We sample 1,000 real frames as a baseline dataset and progressively introduce 1×, 2×, and 4× simulated data from 115 diverse scenes synthesized in Sec. A.3.

Fig. 8(a) shows consistent improvements in MAE, 90th percentile error, and PSNR as simulated training data increases, outperforming the baseline trained on limited real data. With 4× simulated data, MAE and 90th percentile error drop by 2.02 cm and 6.88 cm, while PSNR improves by 1.51 dB. The gains show that simulated data alone can enhance performance in data-limited settings. For comparison, we include a 4×-real setting trained with 4,000 additional real samples. Notably, the simulated data captures 73.33% of the improvement in 90th percentile error, and surpasses it in PSNR. Our analysis shows the model excels in high-quality ranges. 12.1% of predictions exceed 35 dB, nearly double the 6.6% baseline. For the broader 30 dB threshold, the proportion rises from 41.9% to 45.4%. We visualize improvements in imaging quality over objects in Fig. 9, and we attribute these gains to the rich object diversity in our scenes. Finally, combining simulated and real data yields the best performance, with an additional gain of 3.55 cm in 90th percentile error and 0.45 dB in PSNR, highlighting the value of WAVEVERSE-generated signals in both limited and rich data scenarios.

To further demonstrate the benefits of WAVEVERSE-generated signals, we compare it against a Standard Ray Tracing (RT) baseline with the same augmentation design. Existing learning-based methods such as RF Genesis Chen & Zhang (2023) are not applicable here due to their fixed radar assumptions and lack of support for continuous rotational trajectories required in our setup. Standard RT adopts the simulation paradigm as WAVEVERSE but omits phase coherence modeling, resembling traditional MATLAB-style ray tracers. As shown in tab. 3, WAVEVERSE consistently outperforms Standard RT across all metrics and augmenta-

Metric	Method	Real only	+1× sim	+2× sim	+4× sim
MAE (↓)	WAVEVERSE	20.10	19.29	19.12	18.08
	Standard RT	20.10	21.45	21.89	22.28
Q-90th (↓)	WAVEVERSE	48.46	45.19	44.35	41.58
	Standard RT	48.46	49.98	50.24	53.29
PSNR (↑)	WAVEVERSE	26.96	27.66	27.69	28.47
	Standard RT	26.96	27.01	26.85	26.89

Table 3: Comparison with Standard RT.

tion levels. The performance improves steadily as more WAVEVERSE-generated data is added, while adding more Standard RT data yields no improvement and even degrades at higher ratios, suggesting that phase-incoherent simulation produces unreliable signals that do not benefit learning.

While the above evaluation demonstrates the value of WAVEVERSE for providing additional simulated training data, we also explicitly evaluate how well the simulated RF signals align with the real measurements. Following standard practice in generative tasks, we compute both FID and the Jensen-Shannon divergence of the TR margin (JS Div.) (Gong et al., 2025) between simulated and collected RF signals to evaluate the fidelity. Concretely, we train a U-Net on the RF imaging task, compute FID on features extracted at the bottleneck, and compute JS Div. following Gong et al. (2025). WAVEVERSE achieves an FID of 2.879 and a JS Div. of 0.365, values that are on par with those reported for strong generative models (Tian et al., 2024; You et al., 2025; Gong et al., 2025). In contrast, removing phase coherence degrades these metrics to an FID of 5.495 and a JS Div. of 0.430, indicating a substantially larger gap to real signals. Taken together with the task-level improvements above, these results provide strong evidence that WAVEVERSE produces RF signals that are both effective for downstream models and closely aligned with the real-world measurements.

Human Activity Recognition. We further evaluate WAVEVERSE on an open-source human activity classification task (Singh et al., 2019), which maps RF signal sequences to activities. To synthesize motions, we use an LLM to generate diverse descriptions for the five activities in the dataset: walking, standing, squatting, jumping, and jumping jacks. A classifier trained on 100 real samples and tested on 500 held-out samples achieves a baseline accuracy of 31.6% (Fig. 8(b)). Augmenting with 400, 900, and 1900 simulated samples progressively improves accuracy to 49.8%, 61.4%, and 71.6%, approaching the 75.6% from training on all 2,000 real samples. Finally, combining all simulated and real data yields the best performance of 81.0%.

Additionally, we provide an explicit comparison between WAVEVERSE and RF Genesis (Chen & Zhang, 2023) under the same augmentation strategy. Tab. 4 reports accuracies when augmenting 100 real samples with 4×, 9×, and 19× simulated samples. Across all these settings, WAVEVERSE consistently provides larger accuracy gains at every augmentation level. While RF Genesis yields some improvement at low augmentation ratios, its performance plateaus when more simulated data is added. In contrast, WAVEVERSE continues to scale effectively, demonstrating that its physically grounded and environment-aware simulation produces higher-fidelity signals.

Method	Real only	+ 4× sim	+ 9× sim	+ 19× sim
WAVEVERSE	31.6%	49.8%	61.4%	71.6%
RF Genesis	31.6%	46.6%	55.8%	54.6%

Table 4: Accuracy Comparison with RF Genesis.

5 LIMITATIONS AND FUTURE WORK

While WAVEVERSE demonstrates strong performance, some limitations remain. First, the current 4D generative pipeline focuses on whole-body dynamics, which is sufficient for most RF sensing tasks but does not yet capture fine-grained interactions such as typing or manipulating small objects. As a result, the applicability of WAVEVERSE in interaction-centric scenarios is still limited. However, as world-generation and motion-generation models improve, WAVEVERSE, as a unified generation-and-simulation framework, can be naturally extended to handle fine-grained human-object interactions. Second, our simulation is built on ray tracing with reflection modeling, which dominates indoor RF propagation and supports most RF sensing tasks. However, more complex phenomena like diffraction around sharp edges and refraction through objects are currently simplified, as in prior work (Cai et al., 2020; Ren et al., 2024). Extending the simulator with UTD-based diffraction and Fresnel-based refraction is a promising direction to reduce this gap, and we leave this for future work. Lastly, while our signal generation pipeline is fully simulation-based, we agree that lightweight, real-data-driven refinement could further enhance fidelity. We consider integrating such refinement into WAVEVERSE as an interesting direction for future work.

6 CONCLUSION

We present WAVEVERSE, a prompt-based, scalable framework that generates dynamic 4D environments with human motion and simulates realistic RF signals via phase-coherent ray tracing. Comprehensive evaluations and case studies demonstrate the practical utility of WAVEVERSE in enabling high-fidelity RF data generation and enhancing performance in both data-limited and data-rich scenarios. We will release our code and simulator to support future research.

7 ETHICS STATEMENT

We strictly adhere to the ICLR Code of Ethics. In this paper, we introduce WAVEVERSE, a scalable and physically grounded framework for simulating RF signals in dynamic environments. The ability to generate realistic RF data with diverse human motion and scene layouts has several potential positive societal impacts. It can facilitate progress in privacy-preserving sensing, indoor navigation, and health monitoring by reducing reliance on vision-based sensors. By supporting high-fidelity simulation under varied conditions, WAVEVERSE may also help broaden access to RF research, lowering the barrier to entry for institutions without expensive hardware or large-scale data collection pipelines. However, WAVEVERSE may also entail potential negative societal impacts. Since WAVEVERSE relies on LLMs to generate human motions and semantic scene layouts, it inherits the risks associated with LLMs, such as biases in generated content and unintended reinforcement of stereotypes, which users should pay attention to.

8 REPRODUCIBILITY STATEMENT

To ensure reproducibility, the Method section of the main paper presents a detailed description of our approach, while the Experiment section provides the implementation details. Comprehensive information on the dataset and the adopted splits is reported in both the main paper and the Appendix. For the baselines, we carefully document their implementation to enable fair and transparent comparison. Additionally, we will release our code and simulator to facilitate future research.

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.

Karan Ahuja, Yue Jiang, Mayank Goel, and Chris Harrison. Vid2doppler: Synthesizing doppler radar data from videos for training privacy-preserving activity recognition. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–10, 2021.

Baldomero R Árbol and Dan Casas. Bodyshapegpt: Smpl body shape manipulation with llms. *arXiv preprint arXiv:2410.03556*, 2024.

Hong Cai, Belal Korany, Chitra R Karanam, and Yasamin Mostofi. Teaching rf to sense without rf training measurements. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(4):1–22, 2020.

Zoltan Cendes. The development of hfss. In *2016 USNC-URSI Radio Science Meeting*, pp. 39–40. IEEE, 2016.

Xingyu Chen and Xinyu Zhang. Rf genesis: Zero-shot generalization of mmwave sensing through simulation-based data synthesis and generative diffusion models. In *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*, pp. 28–42, 2023.

Xingyu Chen, Zihao Feng, Kun Qian, and Xinyu Zhang. Radio frequency ray tracing with neural object representation for enhanced rf modeling. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 21339–21348, 2025.

Guoxuan Chi, Zheng Yang, Chenshu Wu, Jingao Xu, Yuchong Gao, Yunhao Liu, and Tony Xiao Han. Rf-diffusion: Radio signal generation via time-frequency diffusion. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, pp. 77–92, 2024.

Robert L Cook. Stochastic sampling in computer graphics. *ACM Transactions on Graphics (TOG)*, 5 (1):51–72, 1986.

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. Springer, 2024.

Baris Erol, Sevgi Z Gurbuz, and Moeness G Amin. Exploitation of motion capture data for improved synthetic micro-doppler signature generation with adversarial learning. In *Big Data II: Learning, Analytics, and Applications*, volume 11395, pp. 30–37. SPIE, 2020.

Chen Gong, Bo Liang, Wei Gao, and Chenren Xu. Data can speak for itself: Quality-guided utilization of wireless synthetic data. In *Proceedings of the 23rd Annual International Conference on Mobile Systems, Applications and Services*, pp. 209–222, 2025.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5152–5161, 2022a.

Chuan Guo, Xinxin Zuo, Sen Wang, and Li Cheng. Tm2t: Stochastic and tokenized modeling for the reciprocal generation of 3d human motions and texts. In *European Conference on Computer Vision*, pp. 580–597. Springer, 2022b.

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

Chuan Guo, Inwoo Hwang, Jian Wang, and Bing Zhou. Snapmogen: Human motion generation from expressive texts. *arXiv preprint arXiv:2507.09122*, 2025.

Unsoo Ha, Salah Assana, and Fadel Adib. Contactless seismocardiography via deep learning radars. In *Proceedings of the 26th annual international conference on mobile computing and networking*, pp. 1–14, 2020.

Unsoo Ha, Sohrab Madani, and Fadel Adib. Wistress: Contactless stress monitoring using wireless signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5 (3):1–37, 2021.

Hao He, Chao Li, Wolfgang Ganglberger, Kaileigh Gallagher, Rumen Hristov, Michail Ouroutzoglou, Haoqi Sun, Jimeng Sun, M Brandon Westover, and Dina Katabi. What radio waves tell us about sleep! *Sleep*, 48(1):zsae187, 2025.

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

Jakob Hoydis, Fayçal Aït Aoudia, Sebastian Cammerer, Merlin Nimier-David, Nikolaus Binder, Guillermo Marcus, and Alexander Keller. Sionna rt: Differentiable ray tracing for radio propagation modeling. In *2023 IEEE Globecom Workshops (GC Wkshps)*, pp. 317–321. IEEE, 2023.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Inwoo Hwang, Bing Zhou, Young Min Kim, Jian Wang, and Chuan Guo. Scenemi: Motion inbetweening for modeling human-scene interactions. *arXiv preprint arXiv:2503.16289*, 2025.

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.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

Haowen Lai, Gaoxiang Luo, Yifei Liu, and Mingmin Zhao. Enabling visual recognition at radio frequency. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 388–403, 2024.

Yong-Xian Li, Jiong-Ling Huang, Xin-Yu Yao, Si-Qi Mu, Shou-Xin Zong, and Yan-Fei Shen. A ballistocardiogram dataset with reference sensor signals in long-term natural sleep environments. *Scientific Data*, 11(1):1091, 2024.

Kun Liang, Anfu Zhou, Zhan Zhang, Hao Zhou, Huadong Ma, and Chenshu Wu. mmstress: Distilling human stress from daily activities via contact-less millimeter-wave sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 7(3):1–36, 2023.

Xinpeng Liu, Haowen Hou, Yanchao Yang, Yong-Lu Li, and Cewu Lu. Revisit human-scene interaction via space occupancy. In *European Conference on Computer Vision*, pp. 1–19. Springer, 2024.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pp. 851–866. 2023.

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

Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (ToG)*, 38(6):1–17, 2019.

Liang Pan, Jingbo Wang, Buzhen Huang, Junyu Zhang, Haofan Wang, Xu Tang, and Yangang Wang. Synthesizing physically plausible human motions in 3d scenes. In *2024 International Conference on 3D Vision (3DV)*, pp. 1498–1507. IEEE, 2024.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

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.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Zhenyu Ren, Guoliang Li, Chenqing Ji, Chao Yu, Shuai Wang, and Rui Wang. Caster: A computer-vision-assisted wireless channel simulator for gesture recognition. *IEEE Open Journal of the Communications Society*, 5:3185–3195, 2024.

P Series. Effects of building materials and structures on radiowave propagation above about 100 mhz. *recommendation itu-r*, pp. 2040–1, 2015.

Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H Bermano. Human motion diffusion as a generative prior. *arXiv preprint arXiv:2303.01418*, 2023.

Soyong Shin, Juyong Kim, Eni Halilaj, and Michael J Black. Wham: Reconstructing world-grounded humans with accurate 3d motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2070–2080, 2024.

Akash Deep Singh, Sandeep Singh Sandha, Luis Garcia, and Mani Srivastava. Radhar: Human activity recognition from point clouds generated through a millimeter-wave radar. In *Proceedings of the 3rd ACM Workshop on Millimeter-wave Networks and Sensing Systems*, pp. 51–56, 2019.

Tadeusz Stolarski, Yuji Nakasone, and Shigeka Yoshimoto. *Engineering analysis with ANSYS software*. Butterworth-Heinemann, 2018.

Yue Sun, Zhuoming Huang, Honggang Zhang, Zhi Cao, and Deqiang Xu. 3drimr: 3d reconstruction and imaging via mmwave radar based on deep learning. In *2021 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pp. 1–8. IEEE, 2021a.

Yue Sun, Honggang Zhang, Zhuoming Huang, and Benyuan Liu. Deeppoint: A deep learning model for 3d reconstruction in point clouds via mmwave radar. *arXiv preprint arXiv:2109.09188*, 2021b.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=SJ1kSyO2jwu`.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

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.

Weilin Wan, Zhiyang Dou, Taku Komura, Wenping Wang, Dinesh Jayaraman, and Lingjie Liu. Tlcontrol: Trajectory and language control for human motion synthesis. In *European Conference on Computer Vision*, pp. 37–54. Springer, 2024.

Ge Wang, Chen Qian, Kaiyan Cui, Xiaofeng Shi, Han Ding, Wei Xi, Jizhong Zhao, and Jinsong Han. A universal method to combat multipaths for rfid sensing. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 277–286. IEEE, 2020.

Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. Omnicontrol: Control any joint at any time for human motion generation. In *The Twelfth International Conference on Learning Representations*, 2024.

Hongfei Xue, Qiming Cao, Chenglin Miao, Yan Ju, Haochen Hu, Aidong Zhang, and Lu Su. Towards generalized mmwave-based human pose estimation through signal augmentation. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pp. 1–15, 2023.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16227–16237, 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. Springer, 2024.

Zebin You, Jingyang Ou, Xiaolu Zhang, Jun Hu, Jun Zhou, and Chongxuan Li. Effective and efficient masked image generation models. *arXiv preprint arXiv:2503.07197*, 2025.

Zhengqing Yun and Magdy F Iskander. Ray tracing for radio propagation modeling: Principles and applications. *IEEE access*, 3:1089–1100, 2015.

Zhengqing Yun and Magdy F Iskander. Radio propagation modeling and simulation using ray tracing. In *The advancing world of applied electromagnetics: In honor and appreciation of Magdy Fahmy Iskander*, pp. 251–279. Springer, 2024.

Bin-Bin Zhang, Dongheng Zhang, Yadong Li, Yang Hu, and Yan Chen. Unsupervised domain adaptation for rf-based gesture recognition. *IEEE Internet of Things Journal*, 10(23):21026–21038, 2023a. doi: 10.1109/JIOT.2023.3284496.

Jianrong Zhang, Yangsong Zhang, Xiaodong Cun, Yong Zhang, Hongwei Zhao, Hongtao Lu, Xi Shen, and Ying Shan. Generating human motion from textual descriptions with discrete representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14730–14740, 2023b.

Xiaotong Zhang, Zhenjiang Li, and Jin Zhang. Synthesized millimeter-waves for human motion sensing. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pp. 377–390, 2022.

Mingmin Zhao, Fadel Adib, and Dina Katabi. Emotion recognition using wireless signals. In *Proceedings of the 22nd annual international conference on mobile computing and networking*, pp. 95–108, 2016.

Mingmin Zhao, Shichao Yue, Dina Katabi, Tommi S Jaakkola, and Matt T Bianchi. Learning sleep stages from radio signals: A conditional adversarial architecture. In *International conference on machine learning*, pp. 4100–4109. PMLR, 2017.

Mingmin Zhao, Kreshnik Hoti, Hao Wang, Aniruddh Raghu, and Dina Katabi. Assessment of medication self-administration using artificial intelligence. *Nature medicine*, 27(4):727–735, 2021.

A APPENDIX

In this appendix, we present additional details and results of WAVEVERSE from three perspectives, conditional human motion generation, 4D world generation, and RF simulation, organized consistently with the structure of the main paper, along with the LLM usage in our paper writing. We also include videos of qualitative results in the Supplementary Material, accessible via **main.html** in the accompanying ZIP file.

A.1 CONDITIONAL HUMAN MOTION GENERATION

Model Details. Our model comprises two main components: a VQ-VAE (Van Den Oord et al., 2017) tokenizer and the proposed state-aware causal transformer. The VQ-VAE is built with 1D convolution layers, residual blocks, and ReLU activations in both the encoder and decoder along the temporal dimension. It applies a temporal downsampling rate of 4 and uses a codebook of size 512×512. The tokenizer is trained for 300K iterations with a batch size of 256 using the AdamW optimizer (Loshchilov & Hutter, 2017) ($\beta_1 = 0.9$, $\beta_2 = 0.99$). Following Van Den Oord et al. (2017); Zhang et al. (2023b), the training objective combines reconstruction, embedding, and commitment losses. To enhance motion quality and training stability, we adopt velocity regularization, exponential moving average updates, and codebook resetting as in Zhang et al. (2023b). The learning rate is initialized at 2e-4 and decayed by a factor of 0.05 after 200K iterations with a MultiStepLR scheduler.

The state-aware causal transformer consists of 8 transformer layers (Vaswani et al., 2017), each with 8 attention heads and a hidden dimension of 512. Temporal causality is enforced by applying causal self-attention (Radford et al., 2018) across the network. Text conditions are encoded with CLIP (Radford et al., 2021), while path conditions and spatial states are encoded by a 3-layer MLP with a hidden dimension of 256. The model is trained to maximize the likelihood of token sequences using cross-entropy loss with a batch size of 128. Optimization is performed using the Adam optimizer ($\beta_1 = 0.5$, $\beta_2 = 0.9$) for 300K iterations. The learning rate is initialized at 1e-4 and decayed by a factor of 0.05 after 150K iterations using a MultiStepLR scheduler.

Dataset and Baseline Details. We adopt HumanML3D (Guo et al., 2022a) as our dataset, which contains 14,616 motion sequences annotated with 44,970 text descriptions. To extract path information, we downsample the pelvis trajectory into 64 evenly spaced 2D waypoints on the floor, which serve as the path condition. Notably, the path encodes only directional guidance and excludes duration or velocity information. The dataset is split following the standard protocol as that in Tevet et al. (2023); Xie et al. (2024); Zhang et al. (2023b); Dai et al. (2024).

We adopt four open-source, state-of-the-art motion generation methods as baselines, selected as the closest in design to ours: the diffusion-based MDM (Tevet et al., 2023), OmniControl (Xie et al., 2024) and MotionLCM (Dai et al., 2024), and the autoregressive T2M-GPT (Zhang et al., 2023b). MDM, OmniControl, and MotionLCM support trajectory-conditioned motion generation, which is close to our path-conditioned framework, whereas T2M-GPT serves as the base model for our approach. For MDM, OmniControl, and MotionLCM, we follow their original setup, providing target motion length during both training and inference. T2M-GPT dynamically determines when to terminate the sequence by outputting an `[end]` token. To incorporate path conditioning, we apply only the necessary modifications while keeping all other components unchanged, as described below.

For MDM, we incorporate path conditions by adding the encoded path features to its original conditioning inputs. For OmniControl, we make a minimal change by replacing the per-frame joint encodings with a shared global path feature that is applied uniformly to all joints. We adopt the MLP design as before for a fair comparison. We experimented with both max pooling and mean pooling for aggregating path features, and found that max pooling consistently yields better performance. Thus, we apply max pooling when encoding paths for MDM and OmniControl. In addition, we retain the spatial guidance of OmniControl by similarly applying an analytic function that evaluates how closely the generated motion path aligns with the desired path. The gradient of this function is then used to explicitly perturb the predicted mean at each denoising step, guiding the generated motions to follow the specified path. For MotionLCM, we preserve its stacked transformer layers to encode path signals, as originally designed for trajectory encoding, and leverage the extracted features in the same way as in the original implementation. We also retain its original trajectory-alignment loss but apply it with paths, explicitly penalizing deviations between the generated and desired paths

Method	Architecture	R-Prec. ↑	FID ↓	Div. →	Path Error ↓		Ending Error ↓	
					> 20 cm	> 60 cm	> 20 cm	> 60 cm
Ground Truth		0.797	0.002	9.503	0.	0.	0.	0.
MDM	Diffusion	0.719	0.295	**9.462**	0.547	0.207	0.666	0.367
OmniControl	Diffusion	0.751	0.319	9.279	0.239	0.083	0.330	0.152
MotionLCM	Diffusion	0.739	0.754	9.588	0.315	0.055	0.468	0.177
T2M-GPT	Autoregressive	0.691	0.377	9.736	0.406	0.127	0.545	0.255
Ours	Autoregressive	**0.755**	**0.238**	9.445	**0.208**	**0.045**	**0.325**	**0.111**

Table 5: Text and path conditioned motion generation performance. **Bold** for the best and underline for the second best. R-Prec.: R-Precision; Div.: Diversity.

during training. For T2M-GPT, we extend the input sequence by appending path tokens after the text tokens, mirroring our own path condition encoding to ensure fairness in comparison. All other settings, including hyperparameters, follow the original configurations reported in the respective papers. All models are implemented in PyTorch and trained on an NVIDIA L40 GPU.

Comparison to Baselines. As shown in Tab. 5, our method consistently outperforms all baselines across R-Precision, FID, and path-following metrics, demonstrating superior motion quality and alignment with input conditions. MDM supports trajectory-conditioned motion generation by formulating it as an inpainting task. However, it explicitly leverages known keyframes during denoising, which are unavailable in our path-conditioned framework, and it lacks explicit mechanisms to guide or evaluate motions against the desired path, ultimately limiting its ability to satisfy path conditions. For OmniControl, its original per-frame joint control signals are replaced with global path conditions, which removes localized frame-wise guidance. Furthermore, its analytic path function computes a weighted sum over joint positions, where the weights can vary across denoising steps, leading to instability and higher rates of large path-following errors, despite competitive accuracy at lower thresholds. MotionLCM leverages a path-supervision loss between the predicted and ground-truth paths, which reduces high-level path errors but still results in only moderate performance overall.

In contrast, our method generates motions in a stable, end-to-end autoregressive manner with spatial state feedback, enabling precise and controllable motion generation. Importantly, it does not require predefined duration, making it more scalable in practice. We also emphasize that the gains arise from our proposed modules rather than from the autoregressive structure itself. Crucially, T2M-GPT, on which our method is built, underperforms diffusion-based baselines, whereas our approach achieves better performance. This highlights the effectiveness of our proposed designs, validated by the ablation study in Sec. 4.1.

Training and Inference Time. We report and compare our method with baselines for both the training and inference time. We summarize the training time required in the Table 6 below:

Ours	T2M-GPT	MDM	OmniControl	MotionLCM
27.1	19.7	20.9	47.1	23.2

Table 6: Comparison of training time (hours).

In practice, we adopt the VQ-VAE checkpoint provided by Zhang et al. (2023b) for a fair comparison, and thus exclude its training time (7.1 hours) from the table. For OmniControl and MotionLCM, we also do not include the training time required by pretrained models, as we directly use the released checkpoints. While our method requires longer training time than T2M-GPT due to the longer tokens, it is significantly more efficient than OmniControl, which achieves the closest performance.

Ours	T2M-GPT	MDM	OmniControl	MotionLCM
(0.16, 0.47)	(0.09, 0.27)	7.43	117.26	0.05

Table 7: Comparison of inference time (s). For autoregressive models, we report the time required to generate motion sequences of lengths 64 (first) and 196 (second).

We report the inference time required to generate a sequence of human motion in Table 7. Since the inference time for our method and T2M-GPT depends on the motion sequence length, we provide results for sequences of length 64 and 196. Experiments show that both autoregressive methods are significantly faster than diffusion-based MDM and OmniControl. MotionLCM achieves faster inference through its one-step latent consistency model, but this comes at the cost of neutral performance compared to our method. While our method is slightly slower than T2M-GPT due to the need to compute spatial states on the fly, yet it delivers substantially better performance.

Ablation Study on Addressing Path Overfitting. Apart from the path masking strategy discussed in Sec. A.3, we also investigated alternative approaches and variants. We hypothesize that the primary source of overfitting is the imbalance between inputs: the path is represented by 64 tokens, whereas the text condition is compressed into a single token by CLIP. To address this, we evaluated mean and max pooling to compress all path features into a single token, following the approach of Reimers & Gurevych (2019), but observed a decline in performance. To preserve both model effectiveness and simplicity, we therefore retain all path tokens and let the transformer learn attention over them. Since we ultimately retain all path tokens, we further explored independent masking(IM), which masks each token independently without segment-level masking, and input perturbation as regularization during development. We report additional ablation results on pooling, masking, and perturbation, with varied masking rates and noise levels, based on our current model.

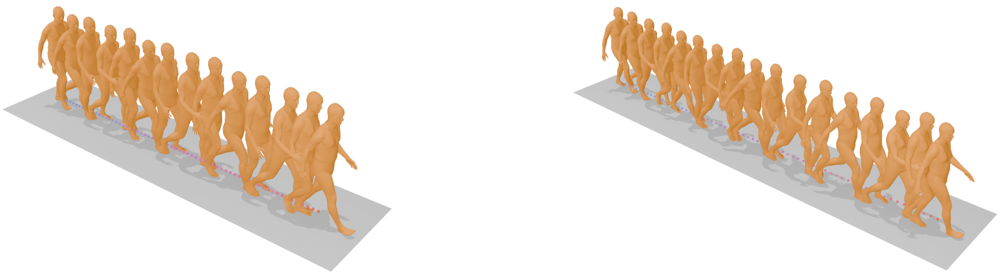
Model	R-Prec.↑	FID↓	Mean Path Err↓	Mean Ending Err↓	Path Err > 60cm↓	Ending Err > 60cm↓
Mean Pooling	0.749	0.477	0.201	0.391	0.081	0.176
Max Pooling	0.707	0.395	0.214	0.362	0.088	0.157
IM 10%	0.670	0.658	0.177	0.301	0.056	0.107
IM 50%	0.728	0.298	0.156	0.284	0.037	0.099
IM 90%	0.744	0.283	0.203	0.389	0.082	0.173
Perturbation 10%	0.662	0.651	0.188	0.311	0.061	0.105
Perturbation 50%	0.671	0.501	0.173	0.277	0.051	0.092
Perturbation 90%	0.695	0.448	0.165	0.273	0.042	0.094
Ours	0.755	0.238	0.151	0.287	0.045	0.111

Table 8: Ablation study on addressing path overfitting.

As shown above, our full method consistently outperforms variants with mean or max pooling in both text alignment and path-following accuracy. Compared to independent masking and input perturbation to mitigate overfitting, our method achieves a significant improvement in text alignment while maintaining strong path-following performance, striking a better balance between the two objectives. It also achieves a lower FID, indicating higher motion quality.

(a) Path length: 1m

(b) Path length: 3m



(c) Path length: 5m

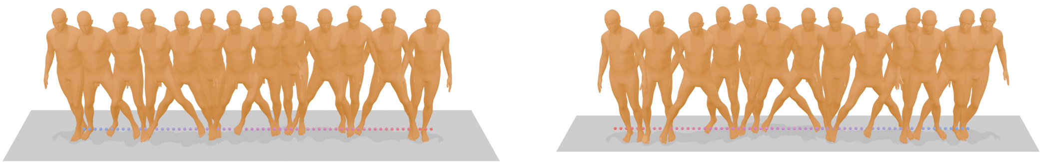
(d) Path length: 7m

Figure 10: Visualization of generated human motions given the same text description and path direction but different path lengths.

Qualitative Results. Lastly, we present qualitative results of our method for text and path conditioned human motion generation. We begin with customized conditions to highlight the capabilities of our model, followed by qualitative results from the test set of the HumanML3D dataset.

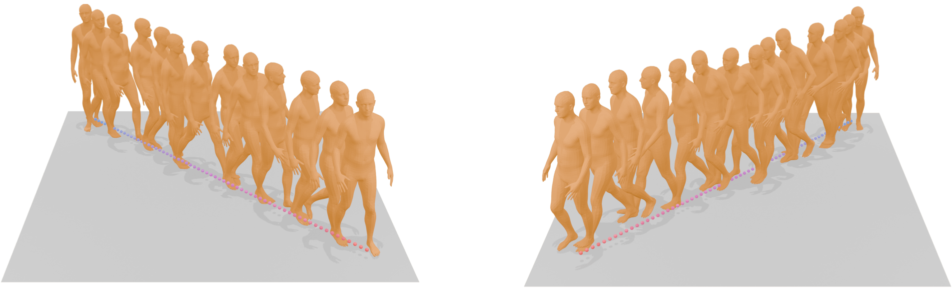
We begin by presenting qualitative results that demonstrate the model's ability to follow diverse path lengths. To this end, we use the same text condition, *walk*, while varying paths of lengths [1, 3, 5, 7] meters, all oriented in the same direction. As shown in Fig. 10, the input paths are visualized with colored points transitioning from **blue** to **red** to indicate temporal order. The generated motions closely follow the given paths while remaining consistent with the texts.

We then change the text prompt to *slowly walk* and fix both the text and path length while varying the path direction by angles of $\pm 90°$, $\pm 45°$, and $\pm 30°$. The corresponding visualizations are shown in Fig. 11. The generated motions exhibit slower velocities compared to those in previous examples, reflecting the semantics of the updated text conditions. We refer the reviewer to the accompanying video for a clearer comparison. Additionally, the visualizations show that the generated motions accurately follow paths with varying directions, demonstrating strong path adherence.



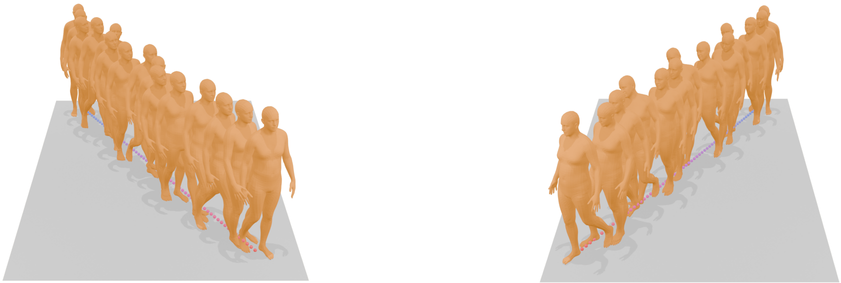
(a) Path direction: $-90°$

(b) Path direction: $90°$



(c) Path direction: $-45°$

(d) Path direction: $45°$

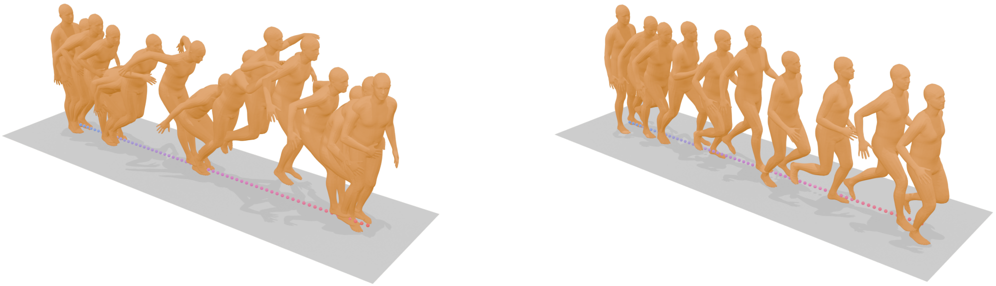


(e) Path direction: $-30°$

(f) Path direction: $30°$

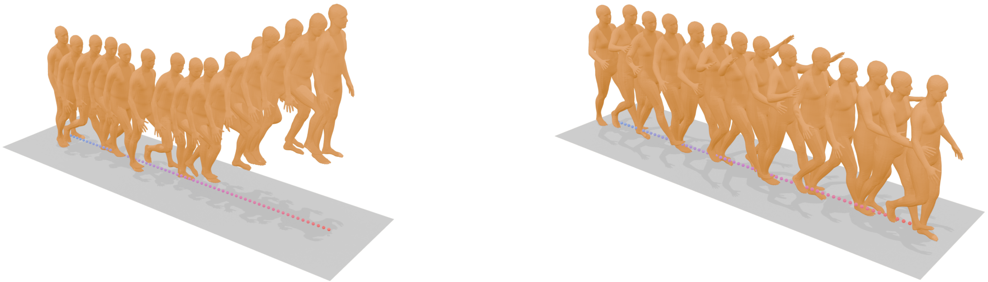
Figure 11: Visualization of generated human motions given the same text and path length but different path directions.

Next, we demonstrate the model's ability to align motion with varying textual descriptions. Using a fixed path, we condition the model on different texts such as *jump*, *run*, *walk as if there are stairs in the front*, and *wave their arms*. As shown in Fig. 12, the generated motions faithfully follow the same path while accurately reflecting the semantics of each instruction.



(a) Text: Jump.

(b) Text: Run.

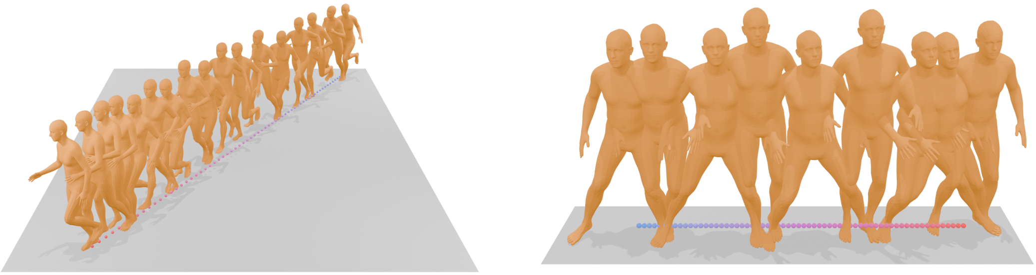


(c) Text: Walk as if there are stairs in the front.

(d) Text: Wave their arms.

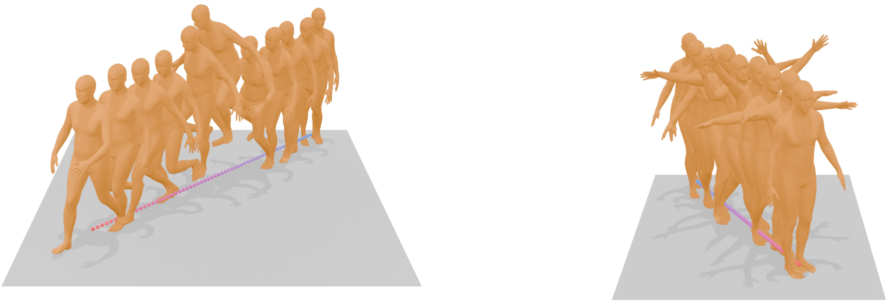
Figure 12: Visualization of generated human motions given the same path direction and length but different text descriptions.

Moreover, we showcase generalization to random combinations of different texts, path lengths and directions in Fig. 13.



(a) Conditions: Run; 8m; 30°

(b) Conditions: Jump; 2.5m; $-90°$

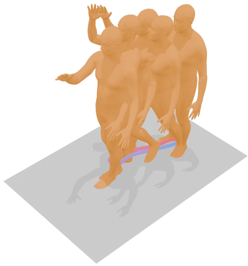


(c) Conditions: Walk as if there are stairs in the front; 3.5m; 45°

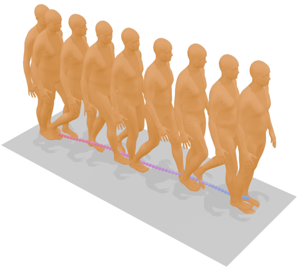
(d) Conditions: Wave their arms; 2.0m; $-30°$

Figure 13: Visualization of generated human motions under varying texts, path lengths, and path directions.

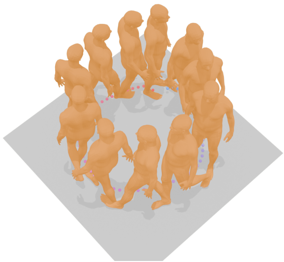
Finally, we present qualitative results of our model on the test set of the HumanML3D dataset shown in Fig. 14. These results highlight the alignment between the generated motions and more complex text and path conditions, demonstrating the model's ability to produce coherent and contextually accurate human motion.



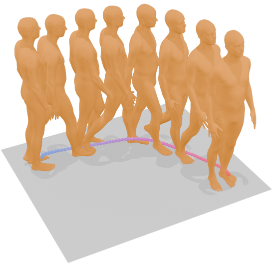
(a) Text: The person takes a step and waves his right hand back and forth.



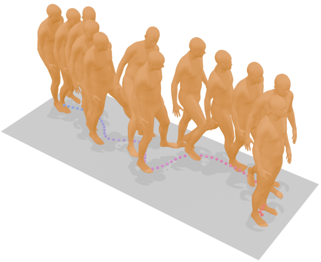
(b) Text: A man walks backwards and then stops.



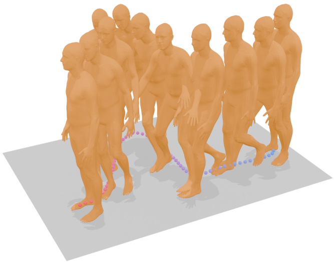
(c) Text: A person walks in a circular motion.



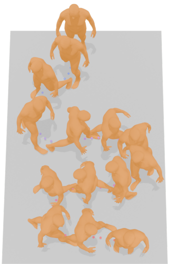
(d) Text: A person bends to the right.



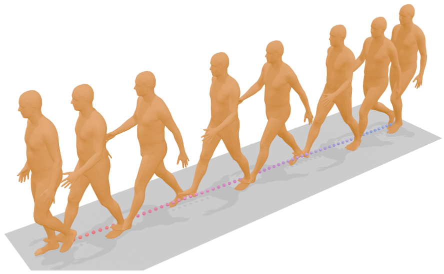
(e) Text: A person begins walking forward first with their left foot, taking wide awkward steps as if they are stepping around or over something; begins walking towards the right and then slowly continues to walk to the left, then continues to walk towards the right coming to a stop off to the right side.



(f) Text: The person was pushed but did not fall.



(g) Text: A figure tip toes around while walking in a slolam like motion.



(h) Text: A person who is walking moves forward taking six confident strides.

Figure 14: Qualitative results from the HumanML3D test set. Text conditions are shown in the subcaptions, and path conditions are visualized as points within the scene.

A.2 4D World Generation

Execution Time. WaveVerse is fully automatic and does not require human interaction, and its components can be parallelized across scenes and radar positions to improve throughput. We report the execution time of each WaveVerse module (Fig. 2), measured on a desktop equipped with an RTX 3090 GPU and an i9-11900 CPU, averaged over 10 runs.

- *Input Prompt*. Generating scene and human-shape descriptions takes 1.37 s and 0.56 s, respectively, with most of the latency coming from OpenAI API communication rather than local computation.
- *Environment and Human Shape Generation*. Environment generation and human-shape generation take 105.47 s and 5.17 s, respectively. This includes API calls, mesh creation, object selection and placement in Unity, and loading the fine-tuned LLM checkpoint for human shapes. Importantly, this cost is incurred once per environment and can be amortized over many motion sequences and simulated signals.
- *Motion Description and Path Generation*. Generating motion descriptions and planning paths within the environment (again via API + path search) takes on average 7.03 s.
- *Human Motion Generation*. The Human Motion Generation module takes 20.79 s in total, though only 0.48 s comes from motion generation with our state-aware transformer. The dominant cost is SMPL fitting for the human mesh, which can be further optimized with faster implementations in computer vision.
- *Dielectric Property Generation*. Dielectric properties are precomputed, and the time is already included in environment generation.

We report the time for *RF signal simulation*. Generating a raw measurement for a single radar with 3 transmitters and 4 receivers takes 0.86 seconds for 100k cast rays. In our high-resolution imaging case study with 1,200 radar poses, we reduce the runtime to 8.97 seconds with a custom CUDA kernel.

Prompts. We provide the adopted prompt in Fig. 15 for the generation of motion descriptions and begin/end points. We also provide the prompts for the human shape generation in Fig. 16 and the dielectric property generation in Fig. 17, respectively.

Motion Description and Path Generation Prompt: You are an experienced human motion designer, expert in drafting realistic daily human motions within a given environment while considering the context of the environment. Please assist me in drafting descriptions of daily human motions. You need to give a text description of motion, including the description of the motion itself, the start and the end positions. The environment is generated from an environment prompt which will be provided. Please ensure that the motion description is feasible within the given environment, like the action can be done by a person within the environment and the start and end points are in the environment. Below is an example of an environment prompt, the details of the generated environment, and examples of human motion descriptions. Note: Units for the coordinates are in meters.

```
For example:
Environment prompt:
A living room.
Environment details:
Floor plan:
living room | [(0, 0), (0, 6), (7, 6), (7, 0)]
Wall height: 2.7
Doors:
door|0|exterior|living room | exterior | living room | [(2.08, 6), (4.08, 6)]
Windows:
window|wall|living room|south|3|0|0 | living room | [(5.27, 0), (6.75, 0)]
window|wall|living room|south|3|1|1 | living room | [(2.70, 0), (4.18, 0)]
window|wall|living room|south|3|2|2 | living room | [(0.27, 0), (1.75, 0)]
Floor objects:
sectional_sofa-0 (living room) | living room | [(5.89, 0, 2.84), (7.05, 0.72,
5.55)]
tv_stand-0 (living room) | living room | [(0, 0, 3.34), (0.54, 0.74, 5.06)]
bookshelf-0 (living room) | living room | [(6.51, 0, 0.13), (7.05, 1.92, 1.27)]
armchair-0 (living room) | living room  | [(3.77, 0, 3.44), (4.62, 1.00, 4.96)]
Wall objects:
```

painting-0 (living room) | living room | [(6.97, 3.87), (7.00, 4.63)]
wall-mounted_shelf-0 (living room) | living room | [(4.15, 5.56), (4.95, 6.00)]
Small objects:
55 inch tv-0|tv_stand-0 (living room) | living room | [(0.24, 0.93, 3.80)]
coaster-0|side_table-0 (living room) | living room | [(4.49, 0.73, 5.80)]

Here are some guidelines for you to understand the above environment details:
1. The space is represented in a ‘X, Y, Z‘ coordinate system, where Y represents the height.
2. Whenever there are only two numbers for a coordinate, it represents ‘(X,Z)‘, omitting height Y.
3. The detailed environment consists of six parts: Floor plan, Doors, Windows, Floor objects, Wall objects, and Small objects.
4. The floor plan is represented as: room name | four coordinates of four corners.
5. Doors are represented as: door name | room 1 | room 2 | two coordinates of the projected doors on X-Z plane (line). The room1 and room2 indicate which rooms the door connects.
6. Windows are represented as: window name | room | two coordinates of the projected doors on X-Z plane (line). The room indicates which room the window is located in.
7. Floor objects are represented as: floor object name | room | two 3D coordinates which compose the 3D bounding box for the object. The room indicates which room the floor object is located in.
8. Wall objects are represented as: wall object name | room | two 2D coordinates which compose the 2D bounding box for the projected object on X-Z plane. The room indicates which room the wall object is located in.
9. The object category is included in its name; you can infer size or height from the name if needed.
10. Do not take the Small objects into consideration when designing the motion.

Motion description examples:
A person walks and gets things from the ‘sectional_sofa-0 (living room)‘ to the ‘tv_stand-0 (living room)‘, from position ‘(5.30,4.20)‘ to position ‘(0.27,2.80)‘.
A person waves hands from the middle of the ‘living room‘ to the ‘window|wall|living room|south|3|2|2‘, from position ‘(2.03,3.02)‘ to position ‘(1.00,0.20)‘.

Motion Design Guidelines:
1. The generated motion description is expected to provide the begin point and the end point; they can be around objects in the scene or spare spaces in the environment.
2. You need to provide the 2D coordinates of these points on the X-Z plane.
3. You should derive the spatial relations among all objects in the environment.
4. You need to consider the space between objects to ensure that the motion (path) is feasible without moving objects. In general, more open-space motions are preferred.
5. Objects in the scene do not interact with humans.
6. There might be multiple rooms; you can design a motion from one room to another.
7. Infer from the context to generate diverse actions (run, slip, wave, etc.).
8. Follow the example format exactly: include a complete motion description, optionally provide the start and end position names, and always include the coordinates in the form ‘from position (x1,z1) to position (x2,z2)‘.

Now, you need to design actions for the below prompts:
Environment prompt:
{environment_prompt}
Environment details:
{environment_details}

Generate {motion_number} possible motions for the motion description generation, which should be as diverse as possible. Strictly follow the format provided in the example. Your response should be direct and without additional text at the beginning or end.

Figure 15: Prompt for Motion Description and Path Generation.

Human Shape Generation Prompt: Infer one plausible human body shape for the scene {environment_prompt} and return exactly one description listing key physical attributes, with no extra text. Example: “Average height, tall neck, long arms, and broad shoulders.”

Figure 16: Prompt for Human Shape Generation.

Dielectric Property Generation Prompt: I have a list of object materials from a 3D asset database: {list_of_object_materials} I need your help to group these materials into broader, high-level material categories. These categories will be used to define radio material properties in an RF simulation engine. Please identify and list appropriate high-level material categories (e.g., metal, plastic, wood, glass, etc.). The goal is to organize the materials in a way that helps assign RF properties during simulation. Use your best judgment based on common material characteristics.

Below is a list of radio materials with their corresponding RF response models and parameter values as defined by the ITU-R P.2040-2 recommendation. I’ve also included the table of parameters (a, b, c, d) used by the recommendation to model relative permittivity (ϵ_r) and conductivity (σ) as functions of frequency:

$\epsilon_r = a f_{\text{GHz}}^{b}$ and $\sigma = c f_{\text{GHz}}^{d}$.

All models assume non-ionized, non-magnetic materials (μ_r = 1).

{table_of_itu_material_models}.

For the following high-level material categories ({list_of_generated_rf_materials}), please:

1. Assign appropriate values for the parameters (a, b, c, d), following the same functional model as the ITU recommendation.

2. Use informed estimation or analogy with similar existing materials in the ITU-R P.2040-2 table.

The objective is to ensure all new materials have an associated RF response model that reflects the real physical responses to the best of your judgment.

You are tasked with selecting the most appropriate radio material from the following list based on an object description. You are only allowed to select one material name from the provided list of materials below.

Available materials: {list_of_all_rf_materials}.

Guidelines for selection: 1. First identify the most likely primary material of the object based on common manufacturing practices 2. Consider the bulk material that would dominate RF interactions, not surface coatings 3. For composite objects, select the material that makes up the largest volume 4. If multiple materials could apply, choose the one that would most affect RF propagation 5. Always select the closest matching material from the list only, even if it’s not an exact match

Output only the selected material name based on provided object description.

The object: {object_descriptions}.

Figure 17: Prompt for Dielectric Property Generation.

Path Planning. Given the input start and end points, we first generate a cost map from the scene layout, which is processed with morphological dilation. We then apply the A* algorithm to find a path between the start and end points. If no valid path is found, we regenerate the motion description along with new start and end points until a feasible path is obtained.

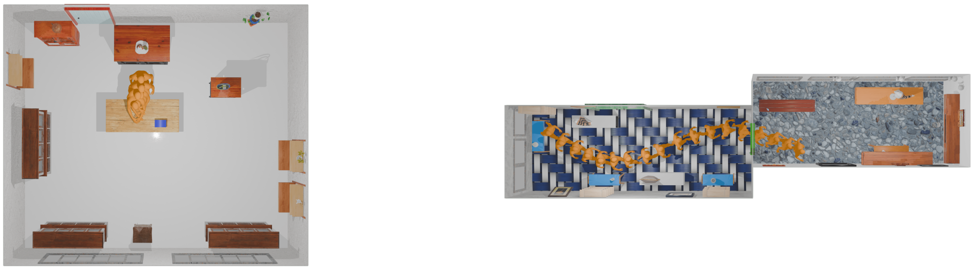
Qualitative Results. While WAVEVERSE can effortlessly generate shorter motions in open or less constrained spaces, we emphasize its ability to handle more challenging scenarios, producing long, semantically and spatially coherent motions within visually complex and spatially constrained environments. Figure 18 showcases qualitative results in such cases, including narrow hallways, intricate layouts, and long human motions. The generated motions align with the surrounding layout, navigating obstacles and fitting within the scene's geometry. Interestingly, the motions sometimes appear to interact with the scene, even though no explicit interaction is modeled. Text prompts are provided in the sub-captions, and we refer to the Supplementary Material for corresponding video visualizations.

(a) A broad gallery; Slowly tour around

(b) A hallway; Wave the arm

(c) A zigzag hallway; Navigate

(d) A keyhole-shaped hallway; Bend to pick something up



(e) A cozy cabin kitchen; Walk to retrieve items

(f) A winding corridor; Walk

(g) A L shape hallway; Quickly Move

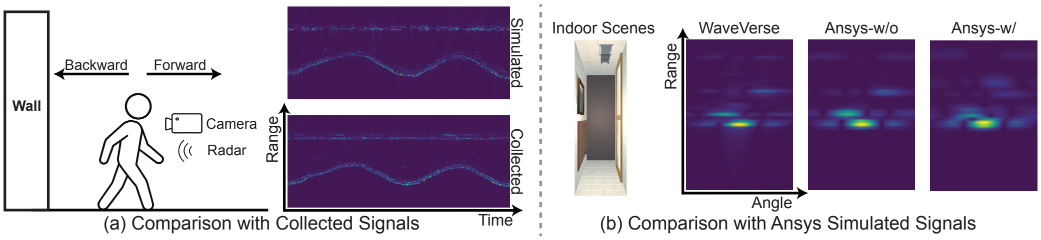
(h) A chic bathroom; Walk and almost slip

(i) A U-shaped hallway; Jump

(j) A classic music room; Dance

Figure 18: Visualization of the generated 4D world, with the environment and motion-generation prompts shown in the subcaptions.

A.3 RF Simulation



(a) Comparison with Collected Signals

(b) Comparison with Ansys Simulated Signals

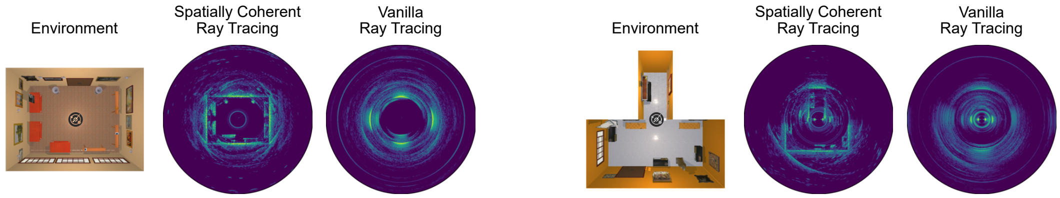
Figure 19: Comparison of WaveVerse generated RF signals with collected measurements and Ansys HFSS simulations. (a) We compare range–time heatmaps for a subject walking forward and backward in front of a wall, showing close agreement between WaveVerse-simulated and radar collected RF signals. (b) We evaluate the range–angle heatmaps simulated by WaveVerse against Ansys (Stolarski et al., 2018) HFSS with diffraction and refraction enabled (Ansys-w/) and with these effects disabled (Ansys-w/o).

Comparison with collected signals and Ansys HFSS simulations. We begin by validating WaveVerse against real RF measurements collected using a radar–camera setup, where a subject walks forward and backward in front of a wall. The synchronized camera video is processed with WHAM (Shin et al., 2024) to reconstruct a temporally consistent human mesh sequence. We then rebuild the surrounding environment, including walls, floors, and their spatial layout, and assign material properties based on the surfaces in the scene. Using this reconstructed 4D world, we simulate RF signals with WaveVerse and compare the resulting range–time spectrograms. The generated heatmaps, shown in Fig. 19(a), achieve a PSNR of 28.63 dB and a 93.65% similarity in energy distribution (computed as 1-Normalized RMSE), indicating strong alignment with the structural motion patterns and amplitude dynamics of the collected signals, and supporting the correctness and realism of the simulated signals from WaveVerse.

To further validate the correctness of the simulated signals, we compare WaveVerse with electromagnetic simulations with Ansys (Stolarski et al., 2018) HFSS (High Frequency Structure Simulator), a proprietary EM solver that models wave propagation solving Maxwell's equations. We sample four environments from previously generated indoor scenes and place four different poses within each, resulting in a total of 16 setups. For every setup, we run HFSS simulations both with and without diffraction and refraction effects enabled. We compare the range–angle spectrograms simulated by WaveVerse against both HFSS outputs. When diffraction and refraction are excluded, WaveVerse achieves a PSNR of 33.57 dB and 2.12% normalized RMSE. When these effects are included, the results are 31.25 dB and 2.76%, respectively. These findings confirm that WaveVerse closely approximates the HFSS-simulated signals with minimal degradation and show that the impact of diffraction and refraction is limited. Moreover, while HFSS requires over one hour per simulation, WaveVerse produces comparable results in under one second, offering a scalable alternative.

Gallery of Panoramic Imaging Results. We provide more comparisons of panoramic imaging results with and without spatial phase coherence as in Sec. 4.3, showing that our ray tracing generates high-fidelity signals that can be effectively used for downstream RF applications, whereas baseline simulations without phase coherence fail to produce data of sufficient quality. Notably, the ghost reflections in our results indicate that the simulation captures multipath effects, which learning-based methods struggle to reproduce or guarantee.

Qualitative Results of Velocity Estimation from Doppler Effects. We provide qualitative results of velocity estimation from Doppler effects in the video (Doppler_comparison.mp4) attached in the Supplementary Material. In this experiment, we simulate a rigid sphere moving back and forth along a straight line with sinusoidal velocity. A radar is positioned in front of the sphere, and velocity is estimated from Doppler shifts. This task requires precise tracking of phase changes induced by motion across different timestamps. The results are visualized as range–velocity maps at each timestamp, where we expect to observe a sinusoidal velocity pattern over time reflecting the sphere's periodic motion. In addition, a narrow velocity band should appear across several range bins, since the spatial extent of the sphere causes multiple ranges to share the same velocity. The video clearly demonstrates that our method, which preserves temporal phase coherence, produces substantially

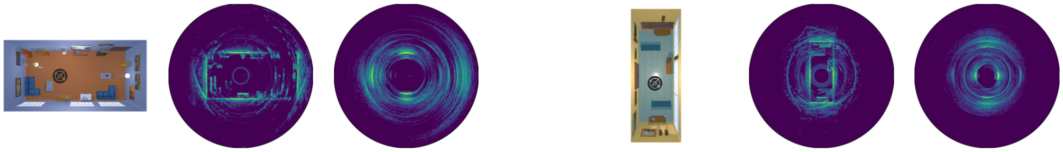


(a) An art gallery

(b) A T-shaped hallway

(c) A command center

(d) A mosaic hallway



(e) A polished wide hall

(f) A wide corridor

Figure 20: More examples of panoramic imaging results. Sensor locations are shown as black icons in environment images.

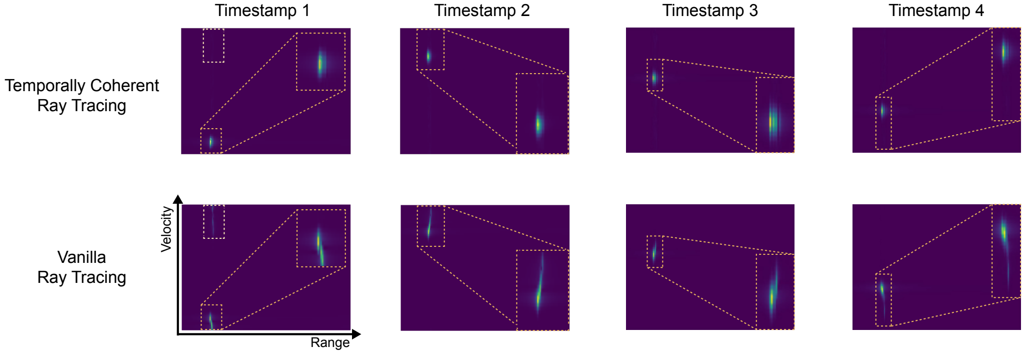


Figure 21: Comparison of velocity estimation from Doppler effects between our method and the baseline.

cleaner range-velocity maps compared to conventional ray tracing. For clarity, Fig. 21 also provides comparisons at four different timestamps, showing that our method outperforms conventional ray tracing.

License. Our proposed phase-coherent ray tracing can be integrated into conventional ray tracing-based simulators. In our implementation, we build on Hoydis et al. (2023), leveraging its underlying ray tracing engine. Hoydis et al. (2023) is released under the Apache 2.0 license. We adhere to the respective licensing terms in our use and will ensure proper attribution and compliance when open-sourcing our customized phase-coherent ray tracing simulator.

A.4 LLM IN PAPER WRITING

We only use LLMs to polish the writing. All retrieval, discovery, research ideation, and the content of this paper are entirely our own work.