1. Additional related work

3D scene understanding. There has been extensive research in the field of visual 3D scene understanding. Previous studies have primarily focused on training models using accurate 3D labels [6, 21], addressing tasks such as 3D object classification [25], 3D object detection [4, 5], and 3D semantic and instance segmentation [3, 10]. OpenScene [19] introduces a zero-shot method for understanding 3D scenes with an open vocabulary. This approach leverages CLIP embeddings to calculate dense features for 3D points, co-embedded with text strings and image pixels, to facilitate 3D semantic segmentation.

In comparison to existing ML-based approaches that rely solely on RF measurements, **Diffusion**^{2} requires significantly fewer RF measurements for training due to the use of the 3D environment model. 2) By using the 3D environment model as input, it supports various environments without significant measurement overhead or retraining, whereas approaches like NeRF^2 and RF-Diffusion necessitate extensive new measurements and retraining whenever the environment changes. 3) It supports multiple frequencies. 4) It can generate RF heatmaps for both static and dynamic 3D scenes. In short, **Diffusion**^{2} combines the strengths of both ray tracing and ML-based approaches to achieve high accuracy, fast performance, flexibility (supporting multiple frequencies and both RF heatmap images and videos), ease of use, and requires minimal training data.

2. Modeling RF propagation with diffusion models

Predicting radio frequency (RF) propagation is challenging. While the underlying physics is deterministic, real-world environments introduce significant uncertainty from factors like noisy 3D scans, unknown material properties, and complex multipath interference. Consequently, exact, path-based simulations are often computationally intractable, and simple regression models struggle to capture the full range of possible outcomes [28].

We propose a diffusion-based framework that learns a *distribution over plausible RF fields* rather than predicting a single, deterministic outcome. This approach embraces uncertainty and decomposes the complex problem of field prediction into a sequence of manageable denoising steps. This paradigm has proven effective in other physics-grounded domains, such as world modeling in Cosmos [2], by progressively refining an output to ensure it remains physically plausible. Our work extends this concept to RF propagation, demonstrating that diffusion models can accurately fit simulated data while respecting the physical principles of wave propagation.

2.1. Overcoming the limitations of prior models

Previous attempts to model RF propagation with generative models often fell short. As noted in the NeRF^2 [28], models like DCGANs and VAEs failed to generalize because they treated RF heatmaps as static spatial *signatures* tied to a transmitter's location. Instead of learning the physics of propagation, they simply memorized geometric patterns. NeRF^2 made progress by incorporating a more physically grounded radiance field representation.

Our model, which we call **Diffusion**^{2}, builds on this insight. We structure the diffusion process to explicitly mimic the temporal dynamics of wave propagation. Our model initiates the process with high signal intensity concentrated near the transmitter, which then gradually diffuses outward. This behavior is not just a generative artifact; it is an emergent property that aligns with physical reality.

This physically grounded approach is crucial for learning true propagation semantics. We observed that when key components of our architecture were removed (*e.g.*, in a single-frequency baseline), the model reverted to overfitting, reproducing spatial artifacts of the environment (*e.g.*, apartment layouts) without modeling genuine signal dynamics. In contrast, our full model generates coherent and physically plausible propagation trajectories.

2.2. The advantage of a probabilistic framework

The core advantage of diffusion over deterministic methods is its ability to *represent a distribution over possible RF fields*. This probabilistic approach provides inherent robustness to the uncertainties and incomplete observations common in real-world scenarios, such as material variations or missing geometry in a 3D scan. By learning a range of plausible outcomes, the model generalizes more effectively.

In summary, diffusion offers a physics-aligned and uncertainty-aware framework that bridges the gap between computationally expensive deterministic simulations and brittle pattern-matching approaches.

3. Diffusion process with condition

The overall diffusion consists of two processes: **forward** noising $q(.)$ and **reverse** denoising $p(.)$.

Forward process. Following a Markov chain, a forward process generates z_t starting from the original signal latent z_0 by sequentially adding a Gaussian noise distribution t times. The forward process finally generates the random noisy latent z_T, which becomes a normal distribution $\mathcal{N}(0, I)$. However, since the diffusion step T is usually set over 1,000, forwarding all steps sequentially is inefficient from a computing resource perspective. So, DDPM applies the reparameterization trick that samples with some steps skipped to process directly from z_0 to z_t as follows:

$$q(z_t|z_0) := \mathcal{N}(z_t; \sqrt{\bar{\alpha}_t}z_0, (1-\bar{\alpha}_t)I) \quad (1)$$
$$:= \sqrt{\bar{\alpha}_t}z_0 + \sqrt{1-\bar{\alpha}_t}\epsilon \quad (2)$$

where $\alpha_t = \prod_{i=0}^{t} \alpha_i$, $\alpha_t = 1 - \beta_t$, and $\epsilon \sim \mathcal{N}(0, I)$. β represents noise variance schedule and ϵ denotes sampled noise from a normal distribution.

Reverse process. In the denoising process, we use the same normal distribution as the forward process and assign the task of predicting a mean μ and a diagonal covariance matrix Σ of the distribution to neural networks as follows:

$$p_\theta(z_{t-1}|z_t) := \mathcal{N}(z_{t-1}; \mu_\theta(z_t, t), \Sigma_\theta(z_t, t)) \quad (3)$$

where μ denotes the predicted mean of the distribution and Σ represents the predicted variance. We append the symbol θ indicating it is trained through neural networks. With this process, we can finally infer the original signal latent z_0 from random noisy latent z_T.

Visual-condition guided denoising process. To consider RF signal information during the denoising process, we reformulate Eq. 3 by adding a visual condition c [8]:

$$p_\theta(z_{t-1}|z_t, \mathbf{c}) := \mathcal{N}(z_{t-1}; \mu_\theta(z_t, t, \mathbf{c}), \Sigma_\theta(z_t, t)) \quad (4)$$

The visual condition c, which represents our *RF-3D Features*, turns the probability formula in Eq. 3 into a conditional probability Eq. 4. It requires that every step of the diffusion process adheres to the given conditioning c, which reflects the real physical environment. The design of conditioning c is critical as it determines whether we can provide rich input signals to feedback environmental information, thereby making the generated RF signal map as consistent with the real scenario as possible.

Inference acceleration with DDIM. DDPM follows a Markov chain, so the inference is slow because generating a single image requires passing T, typically over 1,000 diffusion steps. DDIM notices that the objective function of DDPM depends directly on the marginal distribution $q(z_t|z_0)$ not the joint distribution $q(z_{1:T}|z_0)$ and introduces the non-Markov chain to speed up the reverse process with little performance degradation. DDIM reformulates the forward process as follows:

$$q(z_{1:T}|z_0) := q(z_T|z_0) \prod_{t=2}^{T} q(z_{t-1}|z_t, z_0). \quad (5)$$

According to Bayes' theorem, $q(z_{t-1}|z_t, z_0)$ is also a Gaussian distribution, and the mean and variance are determined to ensure $q(z_t|z_0) := \mathcal{N}(z_t; \sqrt{\bar{\alpha}_t}z_0, (1-\bar{\alpha}_t)I)$ according to Eq. 1 for all $t > 1$ as follows:

$$q(z_{t-1}|z_t, z_0) := \mathcal{N}(z_{t-1}; \tilde{\mu}(z_t, z_0), \tilde{\beta}_t I) \quad (6)$$

where

$$\tilde{\mu}(z_t, z_0) = \frac{\sqrt{\bar{\alpha}_t - 1}\beta_t}{1-\bar{\alpha}_t}z_0 + \frac{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}z_t, \quad (7)$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \quad (8)$$

Note that the forward process of DDIM is a non-Markovian process because z_t is dependent on not only z_{t-1} but also z_0. We adopt the improved inference process [22] by fixing the variance schedulers α and β during the forward process and setting Σ to 0 during the reverse process. In other words, our neural networks focus on predicting μ_θ in the denoising process to generate deterministic outputs.

3.1. Network training

Transform to signal latent space. Training and inference of diffusion models directly based on pixel space require a lot of computing resources and time for optimization [20]. Following the latent designs [8, 20], we encode the pixel space into latent signal space before the diffusion process and decode it backward to generate the RF signal map on a pixel-by-pixel basis. The latent encoder consists of two sequentially connected 2D convolution layers and Tanh as an activation function, while the latent decoder has one sequentially connected 2D transposed convolution layer, one 2D convolution layer, and a sigmoid function as an activation. This transformation into latent space allows for in-depth analysis of the relationships between pixels, which are receivers (RXs) in our problem. The neural networks of the decoder and encoder are indirectly trained by minimizing the signal loss calculated pixel by pixel, not latent space, as shown in Eq. 9.

Loss function. We have neural networks to train, denoted as θ, in the reverse process. Since we set $\Sigma_\theta(z_t, t)$ to 0 for deterministic predictions, we only consider the L2 loss for the denoising prediction and diffusion output, as follows:

$$L_D = ||z_{t-1} - \mu_\theta(z_t, t, \mathbf{c})||^2 \quad (9)$$

where z_{t-1} is calculated based on Eq. 2. We also include two pixel-wise signal losses between the ground truth and the prediction result using L1 and L2 as follows:

$$L_T = \sum_{i,j} |z_0(i,j) - \hat{z}_0(i,j)| + \sum_{i,j} (z_0(i,j) - \hat{z}_0(i,j))^2 \quad (10)$$

where z_0 is the ground truth and $\hat{z_0}$ is the predicted signal map. i and j represent the pixel indices. Lastly, we have pre-measured map input that works as the baseline for prediction. So, we apply the mean squared error to calculate

point-wise loss between the pre-measured map and our prediction as:

$$L_{Pre} = \frac{1}{N}\sum_{i,j}(p(i,j) - \hat{z}_0(i,j))^2 \tag{11}$$

where p represents the pre-measured signal map and N is the number of actually measured points in p. Finally, we get our loss with the scaling factor λ as follows:

$$Loss = \lambda_1 L_D + \lambda_2 L_T + \lambda_3 L_{Pre}. \tag{12}$$

3.2. Generating RF heatmap video

Following the approach in [20], we incorporate one Conv3D temporal layer and one temporal attention layer for both the noisy images z_t and the *RF-3D Features*, labeled T1 and T2 in Fig. 1a. While the spatial layer processes information within each individual frame, these two temporal layers manipulate the feature dimensions across frames. Specifically, the initial input shape is (b, f, c, h, w), where b is the batch size, f is the frame index, c is the image channel, and h and w are the height and width of the images. The spatial layer processes this input for each frame individually.

The Conv3D temporal layer reshapes the input in the following steps:

$$(b, f, c, h, w) \rightarrow (b, \mathbf{c}, \mathbf{f}, h, w) \rightarrow (b, f, c, h, w)$$

The temporal attention layer reshapes as follows:

$$(b, f, c, h, w) \rightarrow (b, \mathbf{h}, \mathbf{w}, \mathbf{f}, \mathbf{c}) \rightarrow (b, f, c, h, w)$$

These temporal layers mix the features across frames by reordering the dimensions, rather than stacking layers across frames or batch units. Importantly, although these layers perform reshaping internally, the final output shape matches the original input shape, allowing these temporal layers to be seamlessly integrated into the architecture without altering the existing design.

4. Implementation details

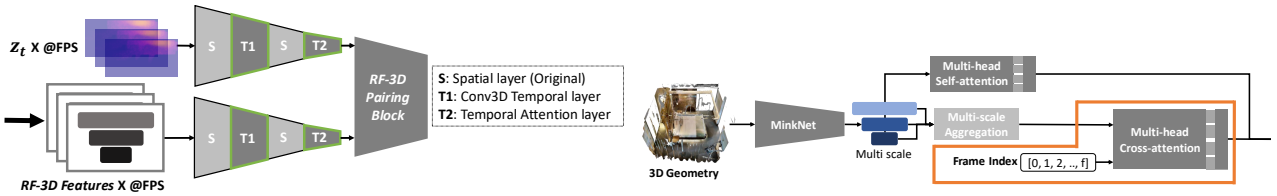
3D dataset. Our problem requires a 3D environment dataset that can be used to place the transmitter (TX) in the appropriate position. Therefore, each object should be stored separately to facilitate manipulation. However, popular datasets like Matterport [24] and ScanNet [7] only offer a unified mesh file for the entire environment, lacking the desired granularity. Consequently, we adopt 3D-FRONT [9], a dataset that aligns with our requirements and features synthetic indoor scenes with professionally crafted layouts, encompassing 18,797 rooms with diverse objects.

3D dataset augmentation. 3D-FRONT provides about 18K rooms, but the structure of each room is quite similar, and the number of datasets is not enough, limiting its ability to train our large-scale diffusion model. Therefore, we apply two data augmentation methods. First, the structure of the 3D-FRONT dataset contains one apartment, which is divided into several rooms such as a living room and bathroom. We extract different rooms from the apartments and generate more apartments by randomly combining rooms. Second, our environment requires one TX to be located. Therefore, we enhance the diversity of the dataset by randomly placing TXs inside the room. In particular, this augmentation is suitable for our problem because signal propagation plays a crucial role in predictions both indoors and outdoors. As a result, we secure over 55k rooms with a variety of layouts and an appropriately located TX.

RF signal dataset. We utilize the wireless channel simulator of AUTOMS [16] to generate the amplitude and phase of the 3D environments. We generate the RF signal map for both Wi-Fi and mmWave considering the multiple channels. For Wi-Fi, we consider 2.4 GHz and 10 different channels for 5 GHz as follows: 5.16, 5.18, 5.20, ..., and 5.34 GHz. For mmWave, we divide into 10 different channels based on the frequency equation within each sweep [17]: $f = f_{min} + \frac{B \times t}{T_c}$ where B is the signal bandwidth, t is a sweep index, and T_c is the chirp length. t is determined by sampling rate R_s as $[0 : 1/R_s : T]$. All variables except T_c are fixed according to the board specifications, *i.e.*, $B = 4e9$, $S_r = 25e6$, and $f_{min} = 77e9$. We set T_c as $20e$-6 to chirp into 501 frequencies from mmWave and select the first 10 frequencies for our dataset, *i.e.*, 77, 77.008, 77.016, ..., and 77.072 GHz. We extensively evaluate **Diffusion**^{2} by varying the combinations of frequencies in the input dataset, such as using 1 to 10 frequencies. Note that we fix the total number of training datasets across all evaluations to ensure a fair comparison.

2D feature. The Swin Transformer [15] is employed to generate visual conditions as multi-scale layers for the overview image and pre-measured map, and we incorporate these features using a hierarchical aggregation and heterogeneous interaction [13]. This multi-scale feature embedding is particularly effective for RF signal estimation because it spans small to large scales, similar to signal propagation properties. Additionally, a feature pyramid neck (FPN) [14] is utilized to consolidate features into diffusion conditions. For the Swin Transformer, we specify channel dimensions as [192, 384, 768, 1536]. Also, we randomly choose 15 points in the RF signal map for the pre-measured map.

3D feature. We use the pre-trained MinkNet model [6] with 21 classes for 3D geometry embedding. We use four levels of multi-scale features before the final layer and apply the FPN for these features to align with the 2D multi-scale embedding. We then use interpolation to unify the feature size at each level. The interpolation size is $(fea, coords) =$



(a) Two temporal layers denoted as T1 and T2 in processing both *RF-3D Features* and noisy latent z_t

(b) A multi-head cross-attention layer for understanding the relationship between frames

Figure 1. Diffusion model for video generation

$(64, 30000)$, where fea is the feature size and $coords$ represents the 3D coordinates. In addition, we apply multi-head self-attention for the last layer from the MinkNet model using Perceiver IO [11]. We use 512 latent dimensions and 12 heads for cross-attention and latent self-attention.

Hyperparameters. We employ the PyTorch framework [18] and conduct training with a batch size of 16 over 20 epochs with four NVIDIA A100 GPUs. We use the Adam optimizer [12] and a linear learning rate warm-up strategy for the first 15% of iterations. The initial learning rate is 10^{-3} and decreases sequentially over 10, 15, and 20 epochs, applying a multiplicative gamma factor of 0.8, 0.2, and 0.04, respectively. We set an equal ratio of L_D, L_T, and L_{Pre} in the loss function.

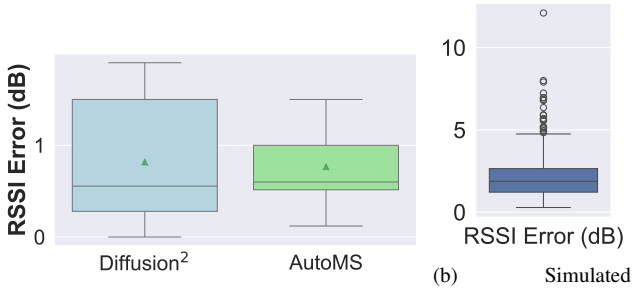
Diffusion setup. We use the improved sampling process [22] with 1,000 diffusion steps for training and 20 inference steps for inference. The learning rate is 10^{-4} for image diffusion and 10^{-3} for video diffusion. The maximum signal strength of the amplitude is 70 for all experiments. The resolution of the results is 352×705 and 52×72 for image and video, respectively. Our video generation model outputs 8 frames. Our model requires approximately 40 GB of GPU memory during training and completes training in about one day.

Human locomotion dataset. To collect a dataset for video diffusion, we use DIMOS [26], which generates human locomotion in a 3D environment. DIMOS uses a Markov decision process to create reasonable human movements while avoiding collisions between surrounding objects. We extract 8 snapshots for each room through DIMOS and generate an amplitude map according to each snapshot environment through the wireless channel simulator [16].

5. Evaluation details

5.1. Real-World Measurement Setup

We conduct experiments in three indoor scenarios. We use Polycam [1] to obtain the 3D models of the experiment environment. We use two Acer Travelmate P658 laptops with Qualcomm QCA6320 chipset-based 60 GHz commercial Wi-Fi cards to measure the mmWave received signal strength indicator (RSSI). The access point (AP) and station use a 6×6 uniform planar array (UPA) with a 120° field-of-view [23] and 4 corner antennas deactivated. The antenna element spacing is 0.58λ [27]. Each antenna has a 1-bit switch (on or off) and a 2-bit phase shifter. All antennas share a single RF chain. The central carrier frequency is 60.48 GHz. We also conduct real measurements for the RF signal video where an object 1.5 meters in height moves in Scenario III. We place the wireless receiver at designated locations for measurement.



(a) Real-world (Scenario III)

(b) Simulated (mmWave)

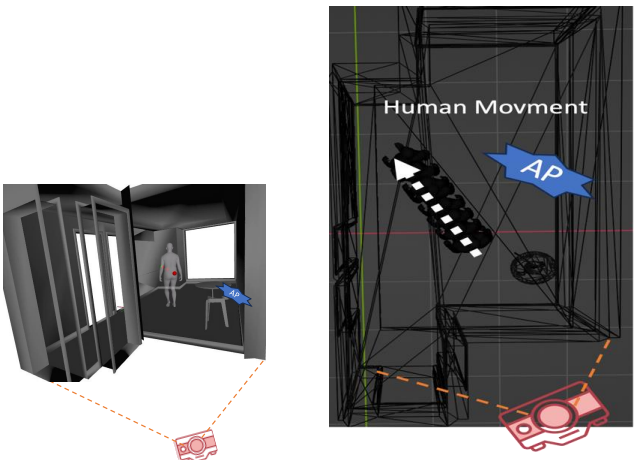
Figure 2. Evaluation of amplitude video generation for simulated and real-world measurements.

5.2. Amplitude video

We compare amplitude video results with the ground truth using the mmWave signal frequency, as shown in Fig. 2. The real-world measurement in Scenario III is shown in Fig. 2a, while the simulated mmWave result is presented in Fig. 2b. NeRF^2, MRI, and Wireless InSite are excluded, as they do not support video output. In the real-world evaluation, **Diffusion**^{2} achieves comparable accuracy and a slightly improved median error, outperforming AUTOMS by 0.05 dB. On the simulated dataset, **Diffusion**^{2} attains a median RSSI error of 2.07 dB, effectively captures dynamic human locomotion, and adapts flexibly to changes in the 3D environment through our video diffusion.

References

[1] Polycam - LiDAR & 3D Scanner, 2026. https://poly.cam/. 4



(a) A camera view (b) The human trajectory

(c) AUTOMS (d) **Diffusion**^{2}

Figure 3. Video diffusion examples from the synthetic dataset. (c) and (d) are snapshots from the video.

[2] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. 1

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. *IEEE/CVF ICCV*, pages 9297–9307, 2019. 1

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. Nuscenes: A multimodal dataset for autonomous driving. *IEEE/CVF CVPR*, pages 11621–11631, 2020. 1

[5] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. *ECCV*, pages 202–221, 2020. 1

[6] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. *IEEE/CVF CVPR*, pages 3075–3084, 2019. 1, 3

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *IEEE/CVF CVPR*, pages 5828–5839, 2017. 3

[8] Yiquan Duan, Xianda Guo, and Zheng Zhu. Diffusiondepth: Diffusion denoising approach for monocular depth estimation. *ECCV*, pages 432–449, 2024. 2

[9] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. *IEEE/CVF ICCV*, pages 10933–10942, 2021. 3

[10] Jingwei Huang, Haotian Zhang, Li Yi, Thomas Funkhouser, Matthias Nießner, and Leonidas J Guibas. Texturenet: Consistent local parametrizations for learning from high-resolution signals on meshes. *IEEE/CVF CVPR*, pages 4440–4449, 2019. 1

[11] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv:2107.14795*, 2021. 4

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 4

[13] Zhenyu Li, Zehui Chen, Xianming Liu, and Junjun Jiang. Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation. *Machine Intelligence Research*, pages 1–18, 2023. 3

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. *IEEE/CVF CVPR*, pages 2117–2125, 2017. 3

[15] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *IEEE/CVF ICCV*, pages 10012–10022, 2021. 3

[16] Ruichun Ma, Shicheng Zheng, Hao Pan, Lili Qiu, Xingyu Chen, Liangyu Liu, Yihong Liu, Wenjun Hu, and Ju Ren. Automs: Automated service for mmwave coverage optimization using low-cost metasurfaces. *ACM MobiCom*, 2024. 3, 4

[17] Wenguang Mao, Jian He, and Lili Qiu. Cat: high-precision acoustic motion tracking. *ACM MobiCom*, pages 69–81, 2016. 3

[18] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019. 4

[19] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. *IEEE/CVF CVPR*, pages 815–824, 2023. 1

[20] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image syn-

thesis with latent diffusion models. *IEEE/CVF CVPR*, pages 10684–10695, 2022. 2, 3

[21] Jonas Schult, Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. Dualconvmesh-net: Joint geodesic and euclidean convolutions on 3d meshes. *IEEE/CVF CVPR*, pages 8612–8622, 2020. 1

[22] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *NeurIPS*, 33:12438–12448, 2020. 2, 4

[23] Yiwen Song, Changhan Ge, Lili Qiu, and Yin Zhang. 2ace: Spectral profile-driven multi-resolutional compressive sensing for mmwave channel estimation. *ACM MobiHoc*, pages 41–50, 2023. 4

[24] Mohamad Zaidi Sulaiman, Mohd Nasiruddin Abdul Aziz, Mohd Haidar Abu Bakar, Nur Akma Halili, and Muhammad Asri Azuddin. Matterport: virtual tour as a new marketing approach in real estate business during pandemic covid-19. *International Conference of Innovation in Media and Visual Design*, pages 221–226, 2020. 3

[25] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. *IEEE/CVF CVPR*, pages 1912–1920, 2015. 1

[26] Kaifeng Zhao, Yan Zhang, Shaofei Wang, Thabo Beeler, and Siyu Tang. Synthesizing diverse human motions in 3d indoor scenes. *IEEE/CVF ICCV*, pages 14738–14749, 2023. 4

[27] Renjie Zhao, Timothy Woodford, Teng Wei, Kun Qian, and Xinyu Zhang. M-cube: A millimeter-wave massive mimo software radio. *ACM MobiCom*, pages 1–14, 2020. 4

[28] Xiaopeng Zhao, Zhenlin An, Qingrui Pan, and Lei Yang. Nerf2: Neural radio-frequency radiance fields. *ACM MobiCom*, pages 1–15, 2023. 1