

GenHPE: Generative Counterfactuals for 3D Human Pose Estimation with Radio Frequency Signals

Shuokang Huang Julie A. McCann

Imperial College London

s.huang21@imperial.ac.uk j.mccann@imperial.ac.uk

Abstract

Human pose estimation (HPE) detects the positions of human body joints for various applications. Compared to using cameras, HPE using radio frequency (RF) signals is non-intrusive and more robust to adverse conditions, exploiting the signal variations caused by human interference. However, existing studies focus on single-domain HPE confined by domain-specific confounders, which cannot generalize to new domains and result in diminished HPE performance. Specifically, the signal variations caused by different human body parts are entangled, containing subject-specific confounders. RF signals are also intertwined with environmental noise, involving environment-specific confounders. In this paper, we propose GenHPE, a 3D HPE approach that generates counterfactual RF signals to eliminate domain-specific confounders. GenHPE trains generative models conditioned on human skeleton labels, learning how human body parts and confounders interfere with RF signals. We manipulate skeleton labels (i.e., removing body parts) as counterfactual conditions for generative models to synthesize counterfactual RF signals. The differences between counterfactual signals approximately eliminate domain-specific confounders and regularize an encoder-decoder model to learn domain-independent representations. Such representations help GenHPE generalize to new subjects/environments for cross-domain 3D HPE. We evaluate GenHPE on three public datasets from WiFi, ultra-wideband, and millimeter wave. Experimental results show that GenHPE outperforms state-of-the-art methods and reduces estimation errors by up to 52.2mm for cross-subject HPE and 10.6mm for cross-environment HPE.

1. Introduction

Human pose estimation (HPE) is a fundamental technique to predict the coordinates of human body joints for diverse applications, such as healthcare [3], autonomous driving [53], and augmented/virtual reality [52]. Recent works have

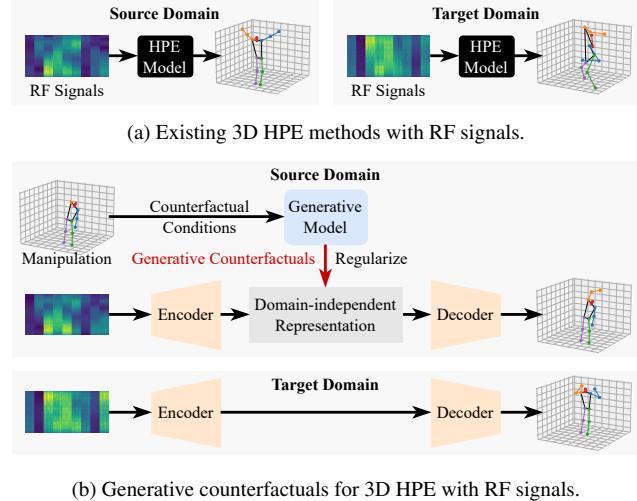


Figure 1. Comparison between (a) existing 3D HPE methods and (b) GenHPE in cross-domain scenarios. Existing methods train HPE models confined by domain-specific confounders and suffer from severe performance degradation in new domains. GenHPE introduces a generative model to help eliminate domain-specific confounders and regularizes an encoder-decoder model to learn domain-independent representations for cross-domain 3D HPE.

explored various signal sources for HPE, including cameras [6, 31, 32], wearable sensors [15, 33, 42], and radio frequency (RF) [5, 39, 41]. In contrast to cameras and wearable sensors, RF-based HPE does not visually record human subjects or attach sensors to them, which is less intrusive and offers a degree of privacy protection [17], making it widely applicable. Moreover, most cameras are highly sensitive to occlusion or darkness, whereas RF signals can continue detection even in these adverse conditions [11, 48, 50].

The principle of using RF signals for HPE is that human bodies interfere with RF signals and lead to signal variations [36, 38]. Such variations implicitly contain human features and therefore can be exploited for HPE [49]. For example, WiFi devices record signal variations in Channel State Information (CSI), which can be utilized for WiFi-based

HPE [41] with Recurrent Neural Networks (RNNs) [22], Convolutional Neural Networks (CNNs) [40], Transformers [39, 55], etc. Similarly, ultra-wideband (UWB) radars have been used for HPE with CNNs [1, 5, 45]. Existing methods have also explored millimeter wave (mmWave) signals for HPE [41] based on Transformers [47, 51], Graph Convolution Networks (GCNs)[10], and diffusion models [10, 12].

Despite remarkable advancements, current RF-based HPE methods are limited to single domains and struggle to generalize in cross-domain scenarios [7]. Specifically, single-domain HPE methods are inevitably biased towards domain-specific confounders, which do not exist in new domains and hamper the performance of cross-domain HPE. For example, compared to single-domain scenarios, the HPE errors of PiW-3D [39] increase by 39.9mm in cross-subject scenarios and 534.7mm in cross-environment scenarios. Since all body parts of human subjects interfere with RF signals simultaneously, their corresponding signal variations entangle and include subject-specific confounders. Environmental noise also induces environment-specific confounders in RF signals. These confounders hinder models from generalizing to new subjects/environments and motivate us to investigate:

How to eliminate domain-specific confounders in RF signals and represent the signal variations caused solely by each human body part for cross-domain HPE?

The core challenge is that, in reality, it is impossible to separate body parts from a living person and study each body part individually. Meanwhile, excessive environmental noise makes it difficult to eliminate environment-specific confounders. To tackle these challenges, we propose GenHPE, a 3D HPE method using generative models to learn how body parts and confounders interfere with RF signals. GenHPE trains generative models to synthesize RF signals conditioned on ground-truth human skeleton labels. After the training, we manipulate skeleton labels (*i.e.*, removing human body parts iteratively) as counterfactual¹ conditions, under which generative models synthesize counterfactual RF signals. A series of counterfactual signals simulate the variations if each body part is “removed”. We calculate the differences between these signals to eliminate domain-specific confounders and aggregate these differences to approximate signals related to all body parts yet independent of domains. The aggregated signals regularize an encoder to learn domain-independent representations, which are fed to a decoder for HPE. Since the encoder extracts representations independent of domains, the encoder-decoder model can generalize to new subjects/environments (target domains) for cross-domain 3D HPE with RF signals (Figure 1). We summarize our contributions as follows:

- We propose GenHPE, to the best of our knowledge, the

¹We use the term “counterfactual” to describe what cannot happen in reality (*e.g.*, a living person without head).

first work of domain generalization for RF-based 3D HPE and the first attempt using generative models to help RF-based models generalize across domains.

- We devise a difference-based scheme to remove domain-specific confounders with counterfactual RF signals and regularize an encoder-decoder model to learn domain-independent representations for cross-domain HPE.
- Experimentally, we apply three generative models (*i.e.*, two diffusion models and a generative adversarial network) in GenHPE for three RF sources (*i.e.*, WiFi, UWB, and mmWave) on public datasets. Compared with state-of-the-art methods, GenHPE remarkably reduces 3D HPE errors in cross-subject/environment scenarios.

2. Related Work

2.1. 3D Human Pose Estimation with RF Signals

Recent years have witnessed the rapid growth of HPE using RF signals, thanks to its lower intrusiveness and its robustness to challenging situations. Existing approaches have utilized RF signals from WiFi, UWB, and mmWave.

WiFi-based 3D HPE. WiFi has been studied for 3D HPE because WiFi signals are ubiquitous for device-free sensing without dedicated deployments. WiPose [22] and GoPose [27] combine CNNs and RNNs to estimate human poses from WiFi CSI, showing the practicability of WiFi-based 3D HPE. Winect [26] and WPNet [40] further establish ResNet models [16] to improve 3D HPE performance using WiFi. Given the success of Transformers [34], WPFormer [55] and PiW-3D [39] leverage the attention mechanism to achieve state-of-the-art WiFi-based 3D HPE. Recent works have further explored the use of WiFi for hand skeleton estimation [21], human mesh construction [37], etc.

UWB-based 3D HPE. UWB-based human sensing, which is less popular due to the need for additional devices, has used higher-frequency signals for hand gesture recognition [1] and respiration monitoring [45]. OPERANet [5] has demonstrated its feasibility for 3D HPE with UWB signals.

mmWave-based 3D HPE. mmWave radars gain increasing popularity for higher-resolution 3D HPE using higher signal frequencies (over 30 GHz) than WiFi (2.4/5 GHz) and UWB (3.1~10.6 GHz) [2, 41]. Given the 3D point clouds from mmWave radars, Point Transformer [47] and PoseFormer [51] develop Transformer-based methods for 3D HPE. DiffPose [12] and mmDiff[10] adopt diffusion models for more accurate estimation with mmWave signals.

Despite their promising performance, the above 3D HPE methods are limited to single domains and fall short in domain generalization. Recent works have examined cross-domain human sensing with RF signals, such as RF-Net [8] which studies environment adaptation for human activity recognition, as well as Widar3.0 [54] and AirFi [35] for environment-independent gesture recognition. However,

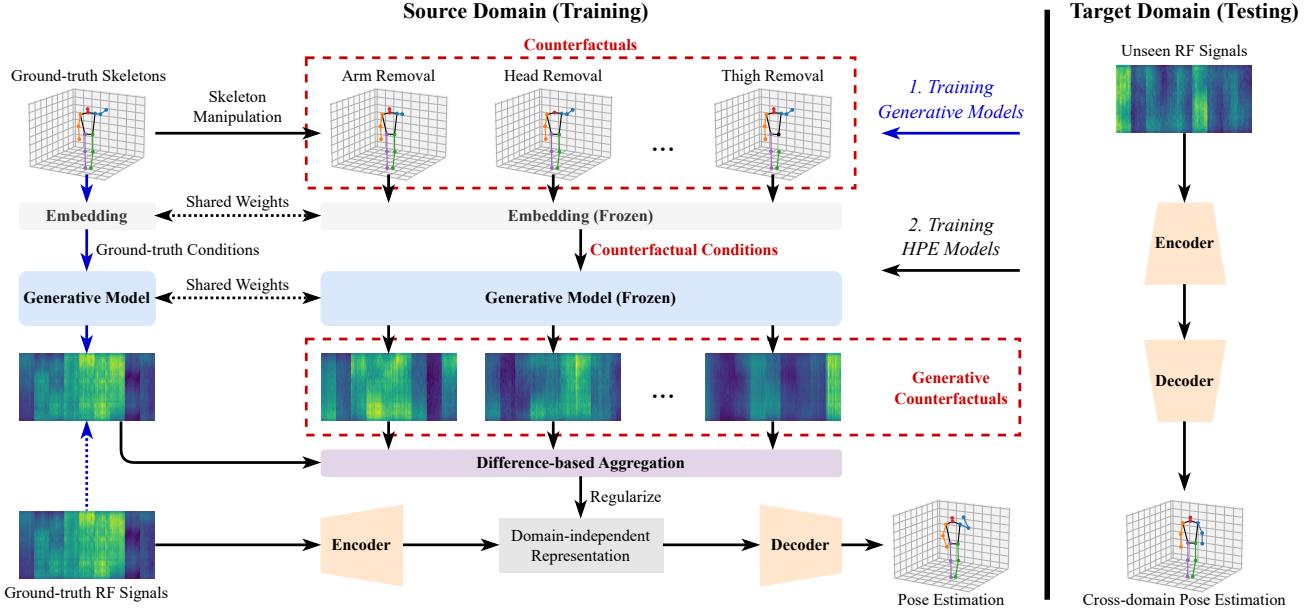


Figure 2. Overview of the proposed GenHPE. **Step 1:** In the source domain, we train a generative model using ground-truth RF signals and skeleton labels. **Step 2:** We manipulate skeleton labels by removing each body part (*e.g.*, arms, head, and thighs) and use them as counterfactual conditions for the generative model to synthesize counterfactual RF signals. The differences between counterfactual signals eliminate domain-specific confounders and are aggregated to approximate signals only related to body parts, regularizing an encoder-decoder model to learn domain-independent representations. Such representations generalize to both source and target domains for HPE.

they only focus on solving coarse-grained classification problems, with finite and discrete outputs. Cross-domain 3D HPE with RF signals is finer-grained, requiring infinite and continuous outputs, and remains unexplored. Furthermore, domain adaptation methods [8] need data from new (target) domains for adaptations, requiring more effort for data collection. Our method aims to directly learn domain-independent representations which can generalize to new subjects/environments without domain adaptation.

2.2. Generative Counterfactuals

Generative models have exhibited remarkable ability for data synthesis. Based on adversarial learning, Generative Adversarial Networks (GANs) [13] can perform high-quality data synthesis, deriving Conditional GANs (CGANs) [24], Wasserstein GAN (WGANs) [4], *etc.* Compared to GANs using one-step synthesis from noise to data, diffusion models have been introduced to synthesize data by multiple-step denoising to outperform GANs, including Denoising Diffusion Probabilistic Models (DDPMs) [18] and Denoising Diffusion Implicit Models (DDIMs) [30]. Specifically, generative models contribute to counterfactual synthesis for various tasks, such as model explanation [14, 44] and data augmentation [19, 25, 43].

Recently, generative models have served to synthesize counterfactual features as interventions for representation learning. For example, GANs have been employed to in-

tervene an HPE model for advanced generalization ability [46]. GaitGCI [9] enhances camera-based gait recognition by maximizing the likelihood difference between ground-truth/synthetic features. In this paper, we pioneer the use of generative counterfactuals to enhance domain generalization for RF-based 3D HPE.

3. GenHPE

We outline GenHPE in Figure 2, which learns domain-independent representations with generative counterfactuals and difference-based aggregation for cross-domain HPE.

3.1. Preliminary

Given a raw RF signal sample \mathbf{x} , 3D HPE aims to predict the ground-truth coordinates of N human body joints $\mathbf{y} \in \mathbb{R}^{N \times 3}$, where $\mathbf{y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$ and $\mathbf{y}_n \in \mathbb{R}^3$. A 3D HPE model $f(\cdot)$ takes \mathbf{x} as input to output $\hat{\mathbf{y}} = f(\mathbf{x})$, and we aim to minimize the error between $\hat{\mathbf{y}}$ and \mathbf{y} . From the aspect of probability distributions, $f(\cdot)$ models a distribution $\hat{\mathbf{y}} \sim p_f(\hat{\mathbf{y}}|\mathbf{x})$ to estimate the ground-truth $\mathbf{y} \sim p_{gt}(\mathbf{y}|\mathbf{x})$.

Domain-specific Confounders. We can use a joint distribution $p_{gt}(\mathbf{x}^h, \mathbf{x}^d)$ to represent RF signal samples, where \mathbf{x}^h represents human-related signal variations and \mathbf{x}^d represents domain-specific confounders. Therefore, the 3D HPE model actually learns a distribution $\hat{\mathbf{y}} \sim p_f(\hat{\mathbf{y}}|\mathbf{x}^h, \mathbf{x}^d)$ including the confounders \mathbf{x}^d . In the source domain d_0 , exist-

ing works can model $p_f(\hat{\mathbf{y}}|\mathbf{x}^h, \mathbf{x}^{d_0})$ to estimate the ground-truth $p_{gt}(\mathbf{y}|\mathbf{x}^h, \mathbf{x}^{d_0})$ with minimized errors. However, if we apply $f(\cdot)$ in a new (target) domain d_1 , there exists a gap between $p_f(\hat{\mathbf{y}}|\mathbf{x}^h, \mathbf{x}^{d_0})$ and $p_{gt}(\mathbf{y}|\mathbf{x}^h, \mathbf{x}^{d_1})$ due to the difference between confounders \mathbf{x}^{d_0} and \mathbf{x}^{d_1} . To solve these issues in cross-domain 3D HPE, we study how to *approximately eliminate domain-specific confounders \mathbf{x}^d using the differences between generative counterfactual RF signals*.

3.2. Generative Counterfactual Synthesis

We transform human body joints \mathbf{y} into skeleton vectors \mathbf{h} , whose embeddings \mathbf{c} are used as conditions to train a generative model $g(\cdot)$ for the synthesis of RF signals $\hat{\mathbf{x}}$. With the trained generative model $g(\cdot)$, we manipulate skeleton vectors as counterfactual conditions for $g(\cdot)$ to synthesize RF signals $\bar{\mathbf{x}}^{\mathbf{h}_k}$ which removes the impact of the k -th skeleton vector \mathbf{h}_k . Using the difference between $\hat{\mathbf{x}}$ and $\bar{\mathbf{x}}^{\mathbf{h}_k}$, we draw out \mathbf{r}_k to eliminate domain-specific confounders and represent the signals *only* related to \mathbf{h}_k . Given K skeleton vectors, we aggregate all \mathbf{r}_k for $k \in [1, \dots, K]$ as \mathbf{r} to approximate skeleton-related yet domain-independent signals.

Skeleton Embedding. Typically, HPE labels are in the form of discrete human body joint descriptions, but what really interfere with RF signals are different human body parts. Therefore, we embody body parts with skeleton embeddings and use them for generative models to synthesize RF signals. We adopt a skeleton map $\mathbf{s} \in \mathbb{R}^{K \times 2}$ to indicate K bones in each skeleton, where the k -th bone is connected between the joints indexed by $s_{k,1}$ and $s_{k,2}$. Given N human body joints $\mathbf{y} \in \mathbb{R}^{N \times 3}$, the corresponding k -th bone can be represented by $\mathbf{h}_k = (\mathbf{y}_{s_{k,1}}, \mathbf{y}_{s_{k,2}}) \in \mathbb{R}^{2 \times 3}$. Accordingly, we can transform \mathbf{y} into K skeleton vectors as:

$$\mathbf{h} = \{\mathbf{h}_k \mid k \in [1, 2, \dots, K]\} \in \mathbb{R}^{K \times 2 \times 3}. \quad (1)$$

We project skeleton vectors \mathbf{h} into an embedding space using a multilayer perceptron $\varphi(\cdot)$ to derive a skeleton embedding vector $\mathbf{c} = \varphi(\mathbf{h})$. We equip $\varphi(\cdot)$ with two linear layers, and each linear layer is followed by a Sigmoid Linear Unit (SiLU) activation function.

Conditional Generative Models. Diverse conditional generative models have been proposed to synthesize specific samples, including conditional diffusion models [28] and CGANs [24]. We employ generative models conditioned on skeleton embeddings \mathbf{c} to synthesize RF signals $\hat{\mathbf{x}}$. Specifically, most generative models perform synthesis from noise \mathbf{z} to data. Therefore, we formulate a conditional generative model $g(\cdot)$ for RF signal synthesis as:

$$\hat{\mathbf{x}} = g(\mathbf{z}, \mathbf{c}) = g(\mathbf{z}, \varphi(\mathbf{h})). \quad (2)$$

From the aspect of probability distributions, $g(\cdot)$ essentially learns a generative distribution p_g as:

$$\begin{aligned} \hat{\mathbf{x}} \sim p_g(\hat{\mathbf{x}} \mid \mathbf{h}) &= p_g(\hat{\mathbf{x}}^{\mathbf{h}_1}, \dots, \hat{\mathbf{x}}^{\mathbf{h}_K}, \hat{\mathbf{x}}^d \mid \mathbf{h}_1, \dots, \mathbf{h}_K) \\ &= p_g(\hat{\mathbf{x}}^{\mathbf{h}_{1:K}}, \hat{\mathbf{x}}^d \mid \mathbf{h}_{1:K}), \end{aligned} \quad (3)$$

where $\hat{\mathbf{x}}^{\mathbf{h}_k}$ represents the signal variations caused by \mathbf{h}_k , and $\hat{\mathbf{x}}^d$ denotes the learned domain-specific confounders. Such a generative distribution can simulate how each body part \mathbf{h}_k interferes with RF signals, and we need to manipulate skeletons \mathbf{h} to further eliminate the confounders $\hat{\mathbf{x}}^d$.

Counterfactual Skeleton Manipulation. In reality, it is impossible to separate different body parts from a living person for study. However, we can manipulate skeleton vectors \mathbf{h} by removing each vector \mathbf{h}_k . A manipulation function $m_k(\cdot)$ removes the k -th vector from \mathbf{h} to yield counterfactual skeleton vectors $\bar{\mathbf{h}}_k$ as:

$$\bar{\mathbf{h}}_k = m_k(\mathbf{h}) = \{\mathbf{h}_i \mid i \in [1, 2, \dots, K] \text{ and } i \neq k\}. \quad (4)$$

We use such counterfactuals as conditions for generative models to synthesize RF signals. Specifically, we project $\bar{\mathbf{h}}_k$ into the embedding space as $\bar{\mathbf{c}}_k = \varphi(\bar{\mathbf{h}}_k)$ and feed $\bar{\mathbf{c}}_k$ into $g(\cdot)$ to sample counterfactual RF signals $\bar{\mathbf{x}}^{\mathbf{h}_k}$ as:

$$\bar{\mathbf{x}}^{\mathbf{h}_k} = g(\mathbf{z}, \bar{\mathbf{c}}_k) = g(\mathbf{z}, \varphi(\bar{\mathbf{h}}_k)), \quad (5)$$

where the parameters of $\varphi(\cdot)$ and $g(\cdot)$ are frozen in this phase. Such a counterfactual synthesis approximately samples RF signals from the following distribution:

$$\bar{\mathbf{x}}^{\mathbf{h}_k} \sim p_g(\hat{\mathbf{x}}^{\mathbf{h}_{1:k-1}}, \hat{\mathbf{x}}^{\mathbf{h}_{k+1:K}}, \hat{\mathbf{x}}^d \mid \mathbf{h}_{1:k-1}, \mathbf{h}_{k+1:K}), \quad (6)$$

which represents the signal variations if the k -th human body part is “removed”. In practice, $\varphi(\cdot)$ and $g(\cdot)$ require inputs with unchanged dimensions during training and sampling, and thus we can set the values of \mathbf{h}_k as 0 as removal. Note that the domain-specific confounders $\hat{\mathbf{x}}^d$ still exist in $\bar{\mathbf{x}}^{\mathbf{h}_k}$, and we devise a difference-based aggregation scheme to approximately eliminate these confounders.

Difference-based Aggregation. Comparing Equation (3) and (6), $\hat{\mathbf{x}}$ represents the signal variations caused by the confounders $\hat{\mathbf{x}}^d$ and the complete skeleton $\mathbf{h}_{1:K}$, while $\bar{\mathbf{x}}^{\mathbf{h}_k}$ represents the signal variations without the impact of \mathbf{h}_k . Intuitively, we can *represent the signal variations only caused by \mathbf{h}_k* based on the difference between $\hat{\mathbf{x}}$ and $\bar{\mathbf{x}}^{\mathbf{h}_k}$. We apply a distance function $\phi(\cdot)$ to calculate such a difference \mathbf{r}_k , which approximates $\hat{\mathbf{x}}^{\mathbf{h}_k}$ only related to \mathbf{h}_k as:

$$\mathbf{r}_k = \phi(\hat{\mathbf{x}}, \bar{\mathbf{x}}^{\mathbf{h}_k}) \approx \hat{\mathbf{x}}^{\mathbf{h}_k} \sim p_g(\hat{\mathbf{x}}^{\mathbf{h}_k} \mid \mathbf{h}_k). \quad (7)$$

In practice, we can simply implement the distance function with element-wise subtraction $\phi(a, b) = a - b$. Such a difference removes environmental noise and disentangles concurrent signal variations caused by multiple body parts.

Therefore, \mathbf{r}_k is independent of domains (*i.e.*, environments or subjects) and only related to a single body part \mathbf{h}_k . We can calculate \mathbf{r}_k for $k \in [1, 2, \dots, K]$ iteratively and aggregate them as \mathbf{r} . Accordingly, \mathbf{r} represents the impacts of all body parts and is also independent of domains. Note that RF signals from various sources have different

meanings. For example, WiFi and UWB collect CSI data which are multichannel sequences, while mmWave collects radar point clouds in 3D space. We concatenate \mathbf{r}_k for $k \in [1, 2, \dots, K]$ in a new dimension and apply a linear layer to aggregate them as $\mathbf{r} = \text{linear}(\text{concat}(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_K))$. Appendix B.1 provides more details about data dimensions.

3.3. Domain-independent Representation Learning

After eliminating domain-specific confounders, \mathbf{r} is independent of domains and represents the signal variations caused solely by human body parts, which can approximate skeleton-related yet domain-independent signals as:

$$\mathbf{r} \approx \hat{\mathbf{x}}^{\mathbf{h}} \sim p_g(\hat{\mathbf{x}}^{\mathbf{h}_{1:K}} | \mathbf{h}_{1:K}). \quad (8)$$

Compared to Equation (3), \mathbf{r} eliminates \mathbf{x}^d and thus can regularize models to generalize across domains. To this end, we build up an encoder-decoder 3D HPE model. The encoder $f_{\text{en}}(\cdot)$ takes ground-truth RF signal samples \mathbf{x} as input and learns domain-independent representations $\mathbf{v} = f_{\text{en}}(\mathbf{x})$ with the same dimensions as \mathbf{r} . The decoder $f_{\text{de}}(\cdot)$ further projects \mathbf{v} into the 3D space for HPE as:

$$\hat{\mathbf{y}} = f_{\text{de}}(\mathbf{v}) = f_{\text{de}}(f_{\text{en}}(\mathbf{x})). \quad (9)$$

We optimize the encoder-decoder model using a pose estimation loss \mathcal{L}_{pe} between $\hat{\mathbf{y}}$ and \mathbf{y} . Meanwhile, we apply counterfactual regularization \mathcal{L}_{cr} based on the likelihood between \mathbf{v} and \mathbf{r} , enforcing the encoder to learn domain-independent representations. The overall loss \mathcal{L} of GenHPE can be formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{pe}}(\hat{\mathbf{y}}, \mathbf{y}) + \lambda \cdot \mathcal{L}_{\text{cr}}(\mathbf{v}, \mathbf{r}), \quad (10)$$

where λ is a ratio to balance pose estimation loss and counterfactual regularization. Since the encoder has learned representations that are independent of domains, the trained encoder-decoder model can generalize to target domains (new subjects/environments) without domain adaptation.

3.4. Overall Training in GenHPE

Training Generative Models. GenHPE explores three generative models that have been widely discussed and time-tested for high-quality data synthesis, including DDPMs [18, 28], DDIMs [30], and CGANs [4, 24]. More details about generative models are presented in Appendix A.1.

Conditional DDPM. DDPM [18] introduces a forward process where a fixed T -step Markov chain converts raw data \mathbf{x}_0 to Gaussian noise $\hat{\mathbf{x}}_T$. Each step in the forward process is a fixed Gaussian transition $q(\mathbf{x}_t | \mathbf{x}_{t-1})$. Conversely, DDPM devises a reverse process using T steps to synthesize $\hat{\mathbf{x}}_0$ from noise $\hat{\mathbf{x}}_T$. Each step in the reverse process is a learnable Gaussian transition $p_\theta(\hat{\mathbf{x}}_{t-1} | \hat{\mathbf{x}}_t)$. Given the data distribution $q(\mathbf{x}_0 | \mathbf{c})$ under conditions \mathbf{c} , conditional DDPM [28] aims to maximize the likelihood $\mathbb{E}[p_\theta(\mathbf{x}_0 | \mathbf{c})]$.

In practice, conditional DDPM minimizes the negative log likelihood $\mathbb{E}[-\log p_\theta(\mathbf{x}_0 | \mathbf{c})]$ by minimizing the divergence between $p_\theta(\hat{\mathbf{x}}_{t-1} | \hat{\mathbf{x}}_t, \mathbf{c})$ and $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0, \mathbf{c})$. Since $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0, \mathbf{c})$ is a Gaussian distribution with a fixed mean $\tilde{\mu}_t$, $p_\theta(\hat{\mathbf{x}}_{t-1} | \hat{\mathbf{x}}_t, \mathbf{c})$ can learn $\tilde{\mu}_t$ with a parameterized mean μ_θ . Such a process can be simplified as training a model $\epsilon_\theta(\cdot)$ to estimate $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ as:

$$\mathcal{L}(\theta) := \mathbb{E}_{\mathbf{x}_0} \left[\left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t, \mathbf{c}) \right\|^2 \right], \quad (11)$$

After training $\epsilon_\theta(\cdot)$, conditional DDPM synthesizes $\hat{\mathbf{x}}_0$ from $\hat{\mathbf{x}}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ by iterating $t \in [T, \dots, 1]$ as:

$$\hat{\mathbf{x}}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\hat{\mathbf{x}}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\hat{\mathbf{x}}_t, t, \mathbf{c}) \right) + \sigma_t \mathbf{z}. \quad (12)$$

DDPM needs many steps (e.g., $T = 1000$) for high-quality synthesis, which is time-consuming. Hence, DDIM [30] focuses on faster synthesis with fewer steps.

Conditional DDIM. The training of DDIM [30] is the same as that of DDPM, and thus we can reuse the trained $\epsilon_\theta(\cdot)$ in DDPM. DDIM uses T_S synthesis steps which are flexible and can be fewer than T in training $\epsilon_\theta(\cdot)$. Given a trained $\epsilon_\theta(\cdot)$, conditional DDIM synthesizes $\hat{\mathbf{x}}_0$ from $\hat{\mathbf{x}}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with $t \in [T_S, \dots, 1]$ as²:

$$\begin{aligned} \hat{\mathbf{x}}_{t-1} = & \sqrt{\bar{\alpha}_{t-1}} \left(\frac{\hat{\mathbf{x}}_t - \sqrt{1 - \bar{\alpha}_t} \cdot \epsilon_\theta(\hat{\mathbf{x}}_t, t, \mathbf{c})}{\sqrt{\bar{\alpha}_t}} \right) \\ & + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \epsilon_\theta(\hat{\mathbf{x}}_t, t, \mathbf{c}) + \sigma_t \mathbf{z}, \end{aligned} \quad (13)$$

where η is a factor to control variances. Typically, we can use $T_S = 100$ or $T_S = 50$ for efficient synthesis.

Conditional GAN. GAN [13] learns data synthesis by the adversarial learning between a generator $g_\theta(\cdot)$ and a discriminator $f_\omega(\cdot)$. In CGAN [24], a generator takes noise \mathbf{z} and conditions \mathbf{c} as inputs to synthesize samples $g_\theta(\mathbf{z}, \mathbf{c})$, while a discriminator distinguishes real samples \mathbf{x} from synthetic samples $g_\theta(\mathbf{z}, \mathbf{c})$. Based on the Wasserstein loss [4], the objective of CGAN can be formulated as:

$$\mathcal{L}(\theta, \omega) := \mathbb{E}_{\mathbf{x}} [f_\omega(\mathbf{x}, \mathbf{c})] - \mathbb{E}_{\mathbf{z}} [f_\omega(g_\theta(\mathbf{z}, \mathbf{c}), \mathbf{c})], \quad (14)$$

where $f_\omega(\mathbf{x}, \mathbf{c})$ denotes the outputs of $f_\omega(\cdot)$ when its inputs are real samples, and $f_\omega(g_\theta(\mathbf{z}, \mathbf{c}), \mathbf{c})$ denotes the outputs of $f_\omega(\cdot)$ when its inputs are synthetic samples. CGAN trains $f_\omega(\cdot)$ and $g_\theta(\cdot)$ alternately by $\min_\theta \max_\omega \mathcal{L}(\theta, \omega)$, where $f_\omega(\cdot)$ aims to maximize the distances between \mathbf{x} and $g_\theta(\mathbf{z}, \mathbf{c})$, while $g_\theta(\cdot)$ aims to minimize such distances. After training $f_\omega(\cdot)$ and $g_\theta(\cdot)$, we can use the generator to synthesize $\hat{\mathbf{x}} = g_\theta(\mathbf{z}, \mathbf{c})$.

Training Encoder-decoder Models. After training a generative model, GenHPE exploits it to synthesize RF signals conditioned on counterfactual skeletons as Equations

²The DDIM paper [30] uses the notation α_t to represent $\bar{\alpha}_t$, while we keep using $\bar{\alpha}_t$ to be consistent with the notations in the DDPM paper [18].

(4)~(6). The synthetic RF signals are aggregated into r , which regularizes an encoder-decoder model for 3D HPE as Equations (9)~(10). We design an encoder with a simplified U-Net [29] and construct a two-stream decoder with self-attention layers augmented by convolutional layers. We only use data in source domains for training, while encoder-decoder models can generalize to target domains thanks to domain-independent representation learning with generative counterfactuals and difference-based aggregation. More details about models are provided in [Appendix A.2](#).

4. Experiments

4.1. Datasets

We evaluate GenHPE on three public datasets of different RF signal sources, including WiFi, UWB, and mmWave. (1) The **WiFi** dataset [39] includes 7 subjects in 3 environments, where we extract 30707 CSI samples for evaluation. (2) The **UWB** dataset [5] collects about 8 hours of data from 6 subjects in 2 environments. As suggested by the authors, we segment the UWB data into 293930 samples. (3) The **mmWave** dataset [41] consists of over 320000 point cloud frames from 40 subjects in 4 environments. In line with the authors, we combine frames to obtain 95666 samples.

We apply three data split strategies to evaluate GenHPE in different scenarios. (1) **Random**: We shuffle each dataset and randomly split it into a training set (80%), a validation set (10%), and a test set (10%). (2) **Cross-Subject**: For each dataset, we randomly select the data of certain subjects as a training set and the data of other subjects as a test set. (3) **Cross-Environment**: For each dataset, we randomly select the data from certain environments for training and the data from other environments for testing. Table 1 shows the details of Cross-Subject/Environment splits, where we further use 10% of each training set as a validation set. We train models on training sets and apply validation sets to select the best models for evaluation on test sets. We further introduce these datasets and data splits in [Appendix B.1](#).

4.2. Baselines

For 3D HPE with WiFi and UWB, whose signals are multichannel sequences, we compare GenHPE with four state-of-the-art methods. (1) **WiPose** [22] combines CNNs and RNNs to learn spatial-temporal features. (2) **WPNet** [40] implements a ResNet [16] to extract deep implicit features. (3) **WPFormer** [55] establishes Transformer layers for advanced RF-based 3D HPE. (4) **PiW-3D** [39] further exploits a CSI encoder, a pose decoder, and a refine decoder to outperform other 3D HPE methods.

For 3D HPE with mmWave, whose signals are point cloud data, we use four latest methods for comparison. (1) **Point Transformer** [47] supports 3D HPE from mmWave point clouds using self-attention layers. (2) **Pose**

Data Split	WiFi [39]		UWB [5]		mmWave [41]	
	Train	Test	Train	Test	Train	Test
Cross-Subject	6	1	5	1	32	8
Cross-Environment	2	1	1	1	3	1

Table 1. Numbers of subject(s)/environment(s) in training/test sets following previous works [39, 41, 54] for comparison.

Former [51] designs a spatial-temporal transformer to further strengthen 3D HPE. (3) **DiffPose** [12] formulates 3D HPE as a reverse diffusion process with GCNs as backbones, significantly reducing estimation errors. (4) **mmDiff** [10] integrates diffusion models with global-local contexts to achieve better HPE accuracy than counterparts.

4.3. Evaluation Metrics

Following previous works [10, 39], we apply three metrics for evaluation, including Mean Per Joint Position Error (**MPJPE**), Procrustes-aligned MPJPE (**PA-MPJPE**), and Mean Per Joint Dimension Location Error (**MPJDLE**). We further describe these metrics in [Appendix B.2](#).

4.4. Implementation Details

We explore DDPM [18], DDIM [30], and CGAN [24] for counterfactual synthesis. To optimize DDPM, we adopt a linear spaced noise schedule where $\beta_t \in [10^{-5}, 10^{-1}]$ with diffusion steps $T = 1000$. The learning rate of DDPM is 10^{-4} . DDIM reuses the trained neural networks in DDPM with fewer synthesis steps. We employ DDIM with synthesis steps $T_S = 100$ or $T_S = 50$ and set the variance factor $\eta = 0$. To optimize CGAN, the generator is trained with the learning rate of 2×10^{-4} , and the discriminator is trained with the learning rate of 10^{-4} . All generative models are trained for 1000 epochs with the batch size of 128.

Both generative models and encoder-decoder models are optimized by Adam [23] on a single Nvidia RTX 4090 GPU. We employ Mean Squared Error (MSE) as \mathcal{L}_{pe} and \mathcal{L}_{cr} with $\lambda = 1.0$. We conduct a hyper-parameter sensitivity study to discuss different values of λ in [Appendix B.3](#). We use the learning rate of 10^{-4} to optimize encoder-decoder models for 200 epochs with the batch size of 32. All experiments are based on PyTorch 2.0.1 with Python 3.9.12. We repeat each experiment 10 times with random seeds to report the means and standard deviations of results. Note that each repetition randomly chooses subjects/environments for model evaluation with *diverse test domains*.

4.5. Comparison with State-of-the-art Methods

GenHPE outperforms state-of-the-art methods with significantly decreased estimation errors in cross-subject and cross-environment scenarios, as shown in Table 2.

In random split scenarios, we evaluate 3D HPE performance in known domains, which have been widely dis-

Methods	Random			Cross-Subject			Cross-Environment		
	MPJPE	PA-MPJPE	MPJDLE	MPJPE	PA-MPJPE	MPJDLE	MPJPE	PA-MPJPE	MPJDLE
WiFi [39]									
WiPose [22]	105.66 \pm 1.03	49.52 \pm 0.53	49.42 \pm 0.46	274.03 \pm 32.30	125.75 \pm 24.21	129.63 \pm 15.47	444.72 \pm 169.37	203.36 \pm 59.98	214.59 \pm 82.33
WPNet [40]	98.24 \pm 1.97	46.02 \pm 0.83	45.60 \pm 0.88	259.74 \pm 34.87	121.83 \pm 33.81	122.12 \pm 16.02	239.48 \pm 5.10	112.91 \pm 6.68	112.23 \pm 2.48
WPFormer [55]	93.93 \pm 3.32	41.99 \pm 0.64	43.68 \pm 1.71	272.74 \pm 33.79	125.75 \pm 25.65	127.72 \pm 15.98	259.38 \pm 24.10	122.80 \pm 11.68	121.15 \pm 12.77
PiW-3D [39]	91.78 \pm 2.03	43.00 \pm 1.01	42.44 \pm 0.94	268.50 \pm 32.90	125.78 \pm 24.85	126.17 \pm 15.79	252.49 \pm 7.89	123.47 \pm 7.77	119.62 \pm 4.66
GenHPE ₁ (Ours)	89.16\pm0.98	41.40\pm0.36	41.12\pm0.42	211.81 \pm 2.84	95.21\pm1.74	101.94 \pm 1.67	228.88\pm0.56	105.81\pm0.90	108.17\pm0.24
GenHPE ₂ (Ours)	89.42 \pm 1.11	42.09 \pm 0.59	41.39 \pm 0.53	211.68\pm2.69	95.21\pm1.74	101.88\pm1.59	228.88 \pm 0.57	105.81\pm0.90	108.17 \pm 0.25
UWB [5]									
WiPose [22]	108.42 \pm 1.21	70.62 \pm 0.61	51.40 \pm 0.56	332.96 \pm 37.50	147.10 \pm 8.19	151.95 \pm 14.33	316.42 \pm 25.58	145.00 \pm 6.19	145.41 \pm 10.49
WPNet [40]	94.44 \pm 1.30	60.92 \pm 0.75	44.73 \pm 0.59	331.34 \pm 38.04	144.02 \pm 6.31	150.81 \pm 14.70	303.34 \pm 28.19	137.79 \pm 7.62	139.46 \pm 11.67
WPFormer [55]	91.52 \pm 0.96	59.09 \pm 0.52	43.51 \pm 0.45	335.79 \pm 38.01	146.58 \pm 7.73	153.12 \pm 14.44	308.93 \pm 26.30	141.53 \pm 6.96	142.43 \pm 10.69
PiW-3D [39]	76.57 \pm 1.22	52.98 \pm 0.65	36.76 \pm 0.57	336.68 \pm 38.61	147.85 \pm 8.24	153.50 \pm 14.89	310.67 \pm 27.24	144.24 \pm 6.81	143.43 \pm 11.20
GenHPE ₁ (Ours)	75.59 \pm 1.02	51.43\pm0.47	36.14 \pm 0.47	279.17\pm3.99	124.42\pm1.10	128.79\pm1.46	292.80\pm12.66	133.97 \pm 6.52	135.36\pm5.84
GenHPE ₂ (Ours)	75.24\pm0.64	51.48 \pm 0.49	36.03\pm0.31	279.25 \pm 3.12	125.28 \pm 1.03	129.01 \pm 1.06	302.44 \pm 20.76	133.44\pm7.72	139.23 \pm 8.34
mmWave [41]									
Point Transformer [47]	106.47 \pm 2.50	65.46 \pm 1.40	52.33 \pm 1.29	123.09 \pm 3.53	71.71 \pm 2.14	60.36 \pm 1.66	130.39 \pm 10.89	77.72 \pm 4.98	63.95 \pm 5.40
PoseFormer [51]	99.96 \pm 1.42	62.12 \pm 0.99	49.03 \pm 0.70	115.77 \pm 3.62	67.56 \pm 1.49	56.70 \pm 1.66	121.43 \pm 14.36	69.87 \pm 4.57	59.57 \pm 7.19
DiffPose [12]	89.36 \pm 2.48	53.37 \pm 1.69	43.91 \pm 1.21	106.72\pm3.48	58.80 \pm 0.83	52.25\pm1.54	113.04\pm11.98	62.28 \pm 4.57	55.42\pm5.78
mmDiff [10]	79.38\pm2.03	48.22\pm1.28	39.03\pm0.99	107.50 \pm 3.47	57.01\pm1.99	52.65 \pm 1.55	113.20 \pm 14.19	60.06\pm5.72	55.47 \pm 6.97
GenHPE ₁ (Ours)	78.36 \pm 1.95	47.51 \pm 1.04	38.44 \pm 0.96	107.54 \pm 2.88	57.93 \pm 1.47	52.67 \pm 1.40	112.12 \pm 2.21	58.83 \pm 1.12	54.56 \pm 1.07
GenHPE ₂ (Ours)	77.30\pm2.19	45.84\pm0.82	37.96\pm1.07	105.83\pm2.09	56.68\pm1.30	51.89\pm1.05	111.19\pm2.77	57.61\pm1.10	54.13\pm1.35

Table 2. Performance of 3D HPE methods in three scenarios with WiFi, UWB, and mmWave. GenHPE₁ adopts DDPM [18] with 1000 synthesis steps. GenHPE₂ adopts DDIM [30] with 100 synthesis steps. **Bold**: the best results. **Blue**: the best baseline results. (Unit: mm)

cussed [10, 39]. GenHPE outperforms the best baselines with slightly lower estimation errors. This is reasonable because in random split scenarios, existing methods can directly learn the confounder distributions of all domains. Thus, the effectiveness of confounder elimination in GenHPE seems less notable. However, GenHPE still makes contributions to the generalization ability of 3D HPE.

In cross-subject scenarios, we can evaluate how GenHPE eliminates subject-specific confounders and generalizes to new subjects. In contrast to the best baselines, GenHPE improves WiFi-based 3D HPE with 48.06mm lower MPJPE, 26.62mm lower PA-MPJPE, and 20.24mm lower MPJDLE. With UWB, GenHPE reduces three metrics by 52.17mm, 19.60mm, and 22.02mm, respectively, compared to the best baselines. We can observe the significantly lower standard deviations of GenHPE (\leq 2.69mm with WiFi and \leq 3.99mm with UWB) against those of baselines (\leq 34.87mm with WiFi and \leq 38.04mm with UWB). The improvement of GenHPE with mmWave is less obvious, since point cloud signals from high-frequency mmWave contain less confounders, but GenHPE still outperforms all baselines. These results illustrate that GenHPE indeed eliminates subject-specific confounders for more robust cross-subject 3D HPE.

In cross-environment scenarios, we can validate how GenHPE eliminates environment-specific confounders and generalizes to new environments. WiFi-based 3D HPE benefits from GenHPE with 10.60mm lower MPJPE, 7.10mm lower PA-MPJPE, and 4.06mm lower MPJDLE, against the

best baselines. With UWB, GenHPE outperforms the best baselines by 10.54mm on MPJPE, 4.35mm on PA-MPJPE, and 4.1mm on MPJDLE. Similar to cross-subject scenarios, GenHPE also results in obviously lower standard deviations than baselines. GenHPE also achieves state-of-the-art performance for mmWave-based 3D HPE. These results prove that GenHPE can effectively tackle environment-specific confounders for more robust cross-environment 3D HPE.

4.6. Ablation Study

Table 3 presents the ablation study of GenHPE to discuss the contributions of different components in our method.

Generative Models. Generally, GenHPE achieves similar performance with different generative models. In most cases, using diffusion models leads to slightly better performance than using CGANs. Compared with DDPMs, using DDIMs not only enables effective counterfactual synthesis for GenHPE but also requires less synthesis steps as few as $T_S = 50$. Such results demonstrate that most generative models are applicable in our method, and GenHPE does not rely on specific models to achieve state-of-the-art 3D HPE.

Skeleton Embedding. We remove Skeleton Embedding from GenHPE and directly use ground-truth/counterfactual human body joints as conditions for DDPMs with 1000 synthesis steps. After such ablation, GenHPE results in higher estimation errors in most cases, which supports our hypothesis that what really interfere with RF signals are different human body parts but not discrete human body joints.

Methods	Random			Cross-Subject			Cross-Environment		
	MPJPE	PA-MPJPE	MPJDLE	MPJPE	PA-MPJPE	MPJDLE	MPJPE	PA-MPJPE	MPJDLE
WiFi [39]									
(1) w/ DDPM ($T = 1000$)	89.16±0.98	41.40±0.36	41.12±0.42	211.81±2.84	95.21±1.74	101.94±1.67	228.88±0.56	105.81±0.90	108.17±0.24
(2) w/ DDIM ($T_S = 100$)	89.42±1.11	42.09±0.59	41.39±0.53	211.68±2.69	95.21±1.74	101.88±1.59	228.88±0.57	105.81±0.90	108.17±0.25
(3) w/ DDIM ($T_S = 50$)	89.51±1.30	42.10±0.63	41.43±0.58	211.69±2.79	95.21±1.74	101.89±1.65	228.88±0.57	105.81±0.90	108.17±0.25
(4) w/ CGAN	91.28±1.48	43.07±0.61	42.30±0.69	211.46±2.71	95.21±1.74	101.78±1.60	228.87±0.56	105.80±0.90	108.17±0.25
(5) w/o Skeleton Embedding	90.70±1.50	42.49±0.54	41.93±0.72	211.83±2.81	95.21±1.75	101.96±1.67	258.56±10.07	117.98±7.15	123.68±6.01
(6) w/o Gen. Counterfactuals	93.26±0.73	44.48±0.32	43.37±0.34	260.59±4.43	112.64±1.13	122.62±2.15	261.86±12.97	117.22±7.80	126.25±8.01
(7) w/ Decoder Only	122.31±2.02	56.46±0.86	56.73±0.93	255.63±5.45	113.55±1.36	120.07±2.35	254.77±6.77	126.18±7.08	120.39±4.30
UWB [5]									
(1) w/ DDPM ($T = 1000$)	75.59±1.02	51.43±0.47	36.14±0.47	279.17±3.99	124.42±1.10	128.79±1.46	292.80±12.66	133.97±6.52	135.36±5.84
(2) w/ DDIM ($T_S = 100$)	75.24±0.64	51.48±0.49	36.03±0.31	279.25±3.12	125.28±1.03	129.01±1.06	302.44±20.76	133.44±7.72	139.23±8.34
(3) w/ DDIM ($T_S = 50$)	74.97±0.49	51.18±0.42	35.91±0.26	277.45±1.86	125.02±1.01	128.34±0.79	294.95±10.53	134.60±4.24	136.04±4.47
(4) w/ CGAN	77.60±0.81	52.69±0.54	37.12±0.37	277.83±4.19	125.11±1.18	128.51±1.56	294.41±7.52	134.35±3.69	136.05±3.65
(5) w/o Skeleton Embedding	77.37±0.79	52.57±0.41	36.97±0.37	279.48±3.98	125.58±0.92	129.08±1.41	291.14±10.54	134.25±5.99	134.65±5.02
(6) w/o Gen. Counterfactuals	83.34±0.31	57.09±0.19	39.93±0.16	298.99±2.61	135.73±0.45	138.81±1.00	337.45±3.45	146.74±0.49	153.21±1.14
(7) w/ Decoder Only	97.31±0.94	65.79±0.54	46.56±0.44	303.16±2.55	136.35±0.60	139.86±0.94	332.82±2.70	146.31±0.36	151.46±1.02
mmWave [41]									
(1) w/ DDPM ($T = 1000$)	78.36±1.95	47.51±1.04	38.44±0.96	107.54±2.88	57.93±1.47	52.67±1.40	112.12±2.21	58.83±1.12	54.56±1.07
(2) w/ DDIM ($T_S = 100$)	77.30±2.19	45.84±0.82	37.96±1.07	105.83±2.09	56.68±1.30	51.89±1.05	111.19±2.77	57.61±1.10	54.13±1.35
(3) w/ DDIM ($T_S = 50$)	77.70±1.45	46.14±0.73	38.15±0.71	105.82±2.00	55.96±1.01	51.90±0.99	112.26±2.11	58.50±1.78	54.68±0.98
(4) w/ CGAN	79.10±2.24	47.70±1.27	38.82±1.07	107.49±1.54	57.44±0.76	52.68±0.73	114.56±2.25	60.51±1.30	55.73±1.09
(5) w/o Skeleton Embedding	78.17±2.28	46.22±0.94	38.37±1.12	109.83±2.21	60.46±1.33	53.82±1.12	112.61±1.63	59.44±1.21	54.81±0.80
(6) w/o Gen. Counterfactuals	84.17±1.37	50.11±0.82	41.34±0.66	109.44±1.49	59.76±1.17	53.69±0.67	113.43±3.49	60.38±1.53	55.30±1.69
(7) w/ Decoder Only	86.05±2.57	51.32±1.11	42.28±1.27	109.11±1.97	60.49±1.03	53.55±1.01	113.88±2.15	61.00±0.74	55.54±1.10

Table 3. Ablation study of GenHPE with (w/) or without (w/o) different components. **Bold** highlights the best results. (Unit: mm)

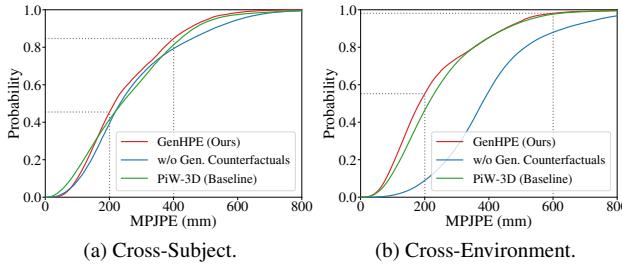


Figure 3. Cumulative distributions of MPJPE (mm) for cross-subject/environment 3D HPE with WiFi [39].

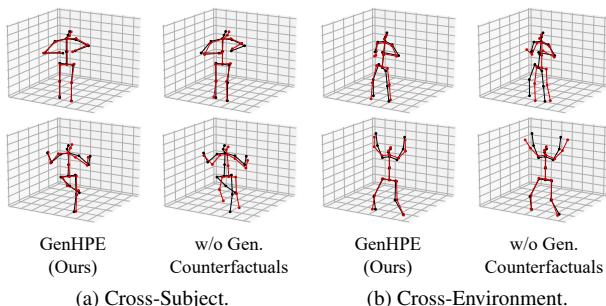


Figure 4. Performance of GenHPE with/without generative counterfactuals for cross-subject/environment 3D HPE with mmWave [41]. Black: ground-truth human poses. Red: model estimations.

Generative (Gen.) Counterfactuals. We directly examine encoder-decoder models for 3D HPE without generative counterfactuals. Such ablation leads to severe performance declines, especially in cross-subject and cross-environment scenarios. Figure 3 compares the cumulative distributions of MPJPE with WiFi to highlight the effectiveness of generative counterfactuals. Figure 4 further visualizes the performance of GenHPE with/without generative counterfactuals on the mmWave dataset. We further investigate the performance of using decoders only and observe even worse 3D HPE performance. These results demonstrate that the regularization of generative counterfactuals indeed helps GenHPE generalize to new subjects and environments.

5. Conclusion

GenHPE pioneers RF-based cross-domain 3D HPE by learning how human body parts and confounders interfere with RF signals. GenHPE synthesizes counterfactual RF signals conditioned on manipulated skeleton labels which “remove” human body parts. A difference-based scheme eliminates domain-specific confounders using counterfactual signals, by which GenHPE regularizes an encoder-decoder model to learn domain-independent representations. Such representations help GenHPE generalize to new subjects/environments, thereby outperforming state-of-the-art methods in cross-subject/environment scenarios.

GenHPE: Generative Counterfactuals for 3D Human Pose Estimation with Radio Frequency Signals

Supplementary Material

A. Models in GenHPE

GenHPE explores three generative models to synthesize counterfactual RF signals and implements an encoder-decoder model for 3D HPE. Herein, we provide more details about these models.

A.1. Generative Models

GenHPE leverages three generative models, including conditional DDPM [18, 28], DDIM [30], and CGAN [4, 24].

A.1.1. Conditional DDPM

DDPM [18] aims to approximate a data distribution $q(\mathbf{x}_0)$ by learning a model distribution $p_\theta(\mathbf{x}_0)$. Specifically, DDPM introduces a fixed forward process and a learnable reverse process. The forward process converts $q(\mathbf{x}_0)$ to a Gaussian distribution $q(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$ in T steps. The reverse process learns to convert a Gaussian distribution $p(\hat{\mathbf{x}}_T) = \mathcal{N}(\hat{\mathbf{x}}_T; \mathbf{0}, \mathbf{I})$ to the model distribution $p_\theta(\hat{\mathbf{x}}_0)$ in T steps. The objective of DDPMs is to maximize the likelihood function $\mathbb{E}[p_\theta(\mathbf{x}_0)]$.

The forward process is a fixed Markov chain that gradually adds noise to $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ with an increasing variance schedule $[\beta_1, \dots, \beta_T]$. Each step in the forward process is a fixed Gaussian transition $q(\mathbf{x}_t | \mathbf{x}_{t-1})$ as:

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_{t-1}; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}\right), \quad (15)$$

where $\beta_t \in (0, 1)$ is the variance. Accordingly, the forward process can be formulated as:

$$q(\mathbf{x}_{1:T} | \mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1}). \quad (16)$$

The reverse process aims to synthesize samples $\hat{\mathbf{x}}_0$ from Gaussian noise $\hat{\mathbf{x}}_T \sim \mathcal{N}(\hat{\mathbf{x}}_T; \mathbf{0}, \mathbf{I})$ with a learnable T -step Markov chain as:

$$p_\theta(\hat{\mathbf{x}}_{0:T}) := p(\hat{\mathbf{x}}_T) \prod_{t=1}^T p_\theta(\hat{\mathbf{x}}_{t-1} | \hat{\mathbf{x}}_t), \quad (17)$$

where $p_\theta(\hat{\mathbf{x}}_{t-1} | \hat{\mathbf{x}}_t)$ is a learnable Gaussian transition as:

$$p_\theta(\hat{\mathbf{x}}_{t-1} | \hat{\mathbf{x}}_t) := \mathcal{N}(\hat{\mathbf{x}}_{t-1}; \boldsymbol{\mu}_\theta(\hat{\mathbf{x}}_t, t), \boldsymbol{\Sigma}_\theta(\hat{\mathbf{x}}_t, t)). \quad (18)$$

θ denotes the learnable parameters of Gaussian transition with the mean $\boldsymbol{\mu}_\theta(\hat{\mathbf{x}}_t, t)$ and the variance $\boldsymbol{\Sigma}_\theta(\hat{\mathbf{x}}_t, t)$.

To maximize the likelihood $\mathbb{E}[p_\theta(\mathbf{x}_0)]$, it is equivalent to minimize the negative log likelihood $\mathbb{E}[-\log p_\theta(\mathbf{x}_0)]$.

Given a variational upper bound on $\mathbb{E}[-\log p_\theta(\mathbf{x}_0)]$, to minimize the negative log likelihood is equivalent to minimize the Kullback-Leibler (KL) divergence D_{KL} as:

$$\min_{\theta} D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\hat{\mathbf{x}}_{t-1} | \hat{\mathbf{x}}_t)), \quad (19)$$

between $p_\theta(\hat{\mathbf{x}}_{t-1} | \hat{\mathbf{x}}_t)$ and the forward process posterior $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$. Using the notations $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$, $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$ can be presented as a fixed Gaussian distribution as:

$$\begin{aligned} q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) &= \mathcal{N}\left(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t, \tilde{\beta}_t \mathbf{I}\right) \\ \text{where } \tilde{\boldsymbol{\mu}}_t &:= \frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1 - \bar{\alpha}_t} \mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{x}_t \\ \text{and } \tilde{\beta}_t &:= \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t. \end{aligned} \quad (20)$$

Since the forward process is a fixed Markov chain, we have:

$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}\right), \quad (21)$$

which can be parameterized as $\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ denotes the added noise in each step. Accordingly, we can replace \mathbf{x}_0 with \mathbf{x}_t in Equation (20) to derive $\tilde{\boldsymbol{\mu}}_t$ as:

$$\tilde{\boldsymbol{\mu}}_t = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon} \right). \quad (22)$$

Given $\mathbf{x}_T \sim \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$, $\tilde{\boldsymbol{\mu}}_t$ and $\tilde{\beta}_t$ indicate how to reverse the forward process by Gaussian transitions, but $\boldsymbol{\epsilon}$ added in each step is unknown. To minimize the KL divergence between $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$ and $p_\theta(\hat{\mathbf{x}}_{t-1} | \hat{\mathbf{x}}_t)$, the reverse process learns Gaussian transitions $p_\theta(\hat{\mathbf{x}}_{t-1} | \hat{\mathbf{x}}_t)$ with means $\boldsymbol{\mu}_\theta(\hat{\mathbf{x}}_t, t)$ and variances $\boldsymbol{\Sigma}_\theta(\hat{\mathbf{x}}_t, t)$ to fit $\tilde{\boldsymbol{\mu}}_t$ and $\tilde{\beta}_t$. Specifically, $\boldsymbol{\mu}_\theta(\hat{\mathbf{x}}_t, t)$ and $\boldsymbol{\Sigma}_\theta(\hat{\mathbf{x}}_t, t)$ are parameterized as:

$$\begin{aligned} \boldsymbol{\mu}_\theta(\hat{\mathbf{x}}_t, t) &= \frac{1}{\sqrt{\alpha_t}} \left(\hat{\mathbf{x}}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\hat{\mathbf{x}}_t, t) \right), \\ \boldsymbol{\Sigma}_\theta(\hat{\mathbf{x}}_t, t) &= \sigma_t^2 \mathbf{I} \text{ where } \sigma_t^2 = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t. \end{aligned} \quad (23)$$

To fit $\tilde{\boldsymbol{\mu}}_t$ with $\boldsymbol{\mu}_\theta(\hat{\mathbf{x}}_t, t)$, it is equivalent to learn a noise model $\boldsymbol{\epsilon}_\theta(\hat{\mathbf{x}}_t, t)$ to predict $\boldsymbol{\epsilon}$ from $\hat{\mathbf{x}}_t$. For the training of $\boldsymbol{\epsilon}_\theta(\hat{\mathbf{x}}_t, t)$, DDPM uses $\hat{\mathbf{x}}_t = \mathbf{x}_t$ and formulates the loss function as:

$$\begin{aligned} \mathcal{L}(\theta) &:= \mathbb{E}_{\mathbf{x}_0} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2 \right] \\ &= \mathbb{E}_{\mathbf{x}_0} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}, t)\|^2 \right], \end{aligned} \quad (24)$$

Algorithm 1 Training Conditional DDPM [18]

repeat

- 1: sample ground-truth 3D HPE labels \mathbf{y}
- 2: obtain ground-truth RF signal samples $\mathbf{x} \sim q(\mathbf{x}|\mathbf{y})$
- 3: convert \mathbf{y} to skeleton vectors \mathbf{h}
- 4: calculate skeleton embeddings $\mathbf{c} = \varphi(\mathbf{h})$
- 5: sample $t \sim \text{Uniform}(\{1, \dots, T\})$
- 6: sample $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 7: take gradient descent steps on

$$\nabla_{\theta} \|\epsilon - \epsilon_{\theta}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t, \mathbf{c})\|^2$$

until converged

which can be optimized by gradient descent. After training $\mu_{\theta}(\hat{\mathbf{x}}_t, t)$, it can be used to synthesize $\hat{\mathbf{x}}_0$ from $\hat{\mathbf{x}}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ by iterating $t \in [T, \dots, 1]$ as:

$$\begin{aligned} \hat{\mathbf{x}}_{t-1} &= \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\hat{\mathbf{x}}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_{\theta}(\hat{\mathbf{x}}_t, t) \right) + \sigma_t \mathbf{z} \\ \text{and } \sigma_t &= \sqrt{\frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}} \sqrt{\beta_t}, \end{aligned} \quad (25)$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ when $t > 1$ and $\mathbf{z} = 0$ when $t = 1$.

Original DDPM is unconditional, while recent works [28] have further introduced conditions to control the target outputs of DDPM. Given input conditions \mathbf{c} , conditional DDPM formulates the loss function as:

$$\begin{aligned} \mathcal{L}(\theta) &:= \mathbb{E}_{\mathbf{x}_0} [\|\epsilon - \epsilon_{\theta}(\mathbf{x}_t, t, \mathbf{c})\|^2] \\ &= \mathbb{E}_{\mathbf{x}_0} [\|\epsilon - \epsilon_{\theta}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t, \mathbf{c})\|^2], \end{aligned} \quad (26)$$

After training $\epsilon_{\theta}(\cdot)$, a sample $\hat{\mathbf{x}}_0$ conditioned on \mathbf{c} can be synthesized from $\hat{\mathbf{x}}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with $t \in [T, \dots, 1]$ as:

$$\hat{\mathbf{x}}_{t-1} = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\hat{\mathbf{x}}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_{\theta}(\hat{\mathbf{x}}_t, t, \mathbf{c}) \right) + \sigma_t \mathbf{z}. \quad (27)$$

In GenHPE, we apply skeleton embeddings $\mathbf{c} = \varphi(\mathbf{h})$ as conditions, while \mathbf{h} can be ground-truth or manipulated skeleton vectors. Algorithm 1 illustrates the training of conditional DDPM in GenHPE, and Algorithm 2 illustrates the synthesis with conditional DDPM in GenHPE.

A.1.2. Conditional DDIM

Since DDPM relies on a Markov chain for synthesis, it typically requires many steps ($T = 1000$) to generate high-quality samples. Meanwhile, the number of synthesis steps must be the same as that in the training of DDPM. These drawbacks reduce the efficiency and flexibility of DDPM.

To address the drawbacks of DDPM, DDIM [30] focuses on faster synthesis with fewer steps. DDIM generalizes DDPM with non-Markov diffusion processes that lead to the same training objective. Therefore, DDIM uses the

Algorithm 2 Synthesis with Conditional DDPM [18]

Input: ground-truth or manipulated skeleton vectors \mathbf{h}

- 1: sample $\hat{\mathbf{x}}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 2: calculate skeleton embeddings $\mathbf{c} = \varphi(\mathbf{h})$
- 3: **for** $t = T, \dots, 1$ **do**
- 4: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$ else $\mathbf{z} = 0$
- 5: $\sigma_t = \sqrt{\frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}} \sqrt{\beta_t}$
- 6: $\hat{\mathbf{x}}_{t-1} = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\hat{\mathbf{x}}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_{\theta}(\hat{\mathbf{x}}_t, t, \mathbf{c}) \right) + \sigma_t \mathbf{z}$
- 7: **end for**

return $\hat{\mathbf{x}}_0$

Algorithm 3 Synthesis with Conditional DDIM [30]

Input: ground-truth or manipulated skeleton vectors \mathbf{h}

- 1: sample $\hat{\mathbf{x}}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 2: calculate skeleton embeddings $\mathbf{c} = \varphi(\mathbf{h})$
- 3: **for** $t = T_S, \dots, 1$ **do**
- 4: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 5: $\sigma_t = \eta \sqrt{(1 - \bar{\alpha}_{t-1}) / (1 - \bar{\alpha}_t)} \sqrt{\beta_t}$
- 6: $\hat{\mathbf{x}}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} (\hat{\mathbf{x}}_t - \sqrt{1 - \bar{\alpha}_t} \cdot \epsilon_{\theta}(\hat{\mathbf{x}}_t, t, \mathbf{c})) / \sqrt{\bar{\alpha}_t}$
+ $\sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \epsilon_{\theta}(\hat{\mathbf{x}}_t, t, \mathbf{c}) + \sigma_t \mathbf{z}$
- 7: **end for**

return $\hat{\mathbf{x}}_0$

same training algorithm as DDPM, and the trained $\epsilon_{\theta}(\cdot)$ in DDPM can be reused by DDIM. Specifically, DDIM can use T_S synthesis steps which can be fewer than T in training $\epsilon_{\theta}(\cdot)$. Given a trained $\epsilon_{\theta}(\cdot)$, conditional DDIM synthesizes $\hat{\mathbf{x}}_0$ from $\hat{\mathbf{x}}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with $t \in [T_S, \dots, 1]$ as:

$$\begin{aligned} \hat{\mathbf{x}}_{t-1} &= \sqrt{\bar{\alpha}_{t-1}} \left(\frac{\hat{\mathbf{x}}_t - \sqrt{1 - \bar{\alpha}_t} \cdot \epsilon_{\theta}(\hat{\mathbf{x}}_t, t, \mathbf{c})}{\sqrt{\bar{\alpha}_t}} \right) \\ &\quad + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \epsilon_{\theta}(\hat{\mathbf{x}}_t, t, \mathbf{c}) + \sigma_t \mathbf{z}, \end{aligned} \quad (28)$$

where $\sigma_t = \eta \sqrt{(1 - \bar{\alpha}_{t-1}) / (1 - \bar{\alpha}_t)} \sqrt{\beta_t}$ is the variance controlled by the factor η .

In GenHPE, we use conditional DDIM with $T_S = 50$ or $T_S = 100$ under the conditions of $\mathbf{c} = \varphi(\mathbf{h})$. Meanwhile, we use $\eta = 0$ for deterministic synthesis. Algorithm 3 illustrates the synthesis with conditional DDIM in GenHPE.

A.1.3. CGAN

GAN [13] designs a minimax two-player game to optimize a generator $g_{\theta}(\cdot)$ and a discriminator $f_{\omega}(\cdot)$ by adversarial learning. $g_{\theta}(\cdot)$ takes noise $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ as input to synthesize samples $g_{\theta}(\mathbf{z})$, which implicitly defines a generative distribution $p_g(\mathbf{z})$. $f_{\omega}(\cdot)$ takes samples \mathbf{x} as input and outputs the probability $f_{\omega}(\mathbf{x})$ that \mathbf{x} comes from ground-truth distribution $p_{\text{gt}}(\mathbf{x})$ rather than $p_g(\mathbf{x})$. $f_{\omega}(\cdot)$ aims to distinguish $\mathbf{x} \sim p_{\text{gt}}(\mathbf{x})$ from $\mathbf{z} \sim p_g(\mathbf{z})$, while $g_{\theta}(\cdot)$ aims

Algorithm 4 Training Conditional GAN [24]

repeat

- 1: sample ground-truth 3D HPE labels \mathbf{y}
- 2: obtain ground-truth RF signal samples $\mathbf{x} \sim q(\mathbf{x}|\mathbf{y})$
- 3: convert \mathbf{y} to skeleton vectors \mathbf{h}
- 4: calculate skeleton embeddings $\mathbf{c} = \varphi(\mathbf{h})$
- 5: **for** discriminator steps **do**
- 6: sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 7: take gradient descent steps on

$$\nabla_{\omega} \mathbb{E}_{\mathbf{x}, \mathbf{z}} [-f_{\omega}(\mathbf{x}, \mathbf{c}) + f_{\omega}(g_{\theta}(\mathbf{z}, \mathbf{c}), \mathbf{c})]$$
- 8: **end for**
- 9: take gradient descent steps on

$$\nabla_{\theta} \mathbb{E}_{\mathbf{z}} [-f_{\omega}(g_{\theta}(\mathbf{z}, \mathbf{c}), \mathbf{c})]$$

until converged

to maximize the similarity between $p_g(\mathbf{z})$ and $p_{\text{gt}}(\mathbf{x})$. The distance between $p_g(\mathbf{z})$ and $p_{\text{gt}}(\mathbf{x})$ can be represented by the Jensen–Shannon (JS) divergence D_{JS} as:

$$D_{\text{JS}}(p_{\text{gt}} \| p_g) = \frac{1}{2} D_{\text{KL}}(p_{\text{gt}} \| p_m) + \frac{1}{2} D_{\text{KL}}(p_g \| p_m), \quad (29)$$

where $p_m = \frac{p_{\text{gt}} + p_g}{2}$. D_{JS} can be further formulated as an objective function:

$$\begin{aligned} \mathcal{L}(\theta, \omega) := & \mathbb{E}_{\mathbf{x}} [\log f_{\omega}(\mathbf{x})] \\ & + \mathbb{E}_{\mathbf{z}} [\log (1 - f_{\omega}(g_{\theta}(\mathbf{z})))]. \end{aligned} \quad (30)$$

The discriminator and the generator are optimized alternately with $\min_{\theta} \max_{\omega} \mathcal{L}(\theta, \omega)$, where $f_{\omega}(\cdot)$ aims to maximize the difference between its outputs $f_{\omega}(\mathbf{x})$ from ground-truth inputs and its outputs $f_{\omega}(g_{\theta}(\mathbf{z}))$ from synthetic inputs. On the contrary, $g_{\theta}(\cdot)$ aims to minimize their difference. However, this objective function lacks of continuity and usually leads to unstable training and mode collapse. Therefore, Wasserstein loss [4] has been proposed to solve these issues, which can be formulated as:

$$\mathcal{L}(\theta, \omega) := \mathbb{E}_{\mathbf{x}} [f_{\omega}(\mathbf{x})] - \mathbb{E}_{\mathbf{z}} [f_{\omega}(g_{\theta}(\mathbf{z}))]. \quad (31)$$

Original GAN and WGAN are unconditional, while a common practice of conditional GAN (CGAN) is to feed conditions into $g_{\theta}(\cdot)$ and $f_{\omega}(\cdot)$. Accordingly, the objective function of CGAN can be formulated as:

$$\mathcal{L}(\theta, \omega) := \mathbb{E}_{\mathbf{x}} [f_{\omega}(\mathbf{x}, \mathbf{c})] - \mathbb{E}_{\mathbf{z}} [f_{\omega}(g_{\theta}(\mathbf{z}, \mathbf{c}), \mathbf{c})], \quad (32)$$

In practice, CGAN is optimized by gradient descent, and thus $\max_{\omega} \mathcal{L}(\theta, \omega)$ is equivalent to perform gradient descent on $\nabla_{\omega} \mathbb{E}_{\mathbf{x}, \mathbf{z}} [-f_{\omega}(\mathbf{x}, \mathbf{c}) + f_{\omega}(g_{\theta}(\mathbf{z}, \mathbf{c}), \mathbf{c})]$. $\min_{\theta} \mathcal{L}(\theta, \omega)$ is equivalent to perform gradient descent on $\nabla_{\theta} \mathbb{E}_{\mathbf{z}} [-f_{\omega}(g_{\theta}(\mathbf{z}, \mathbf{c}), \mathbf{c})]$.

In GenHPE, we use skeleton embeddings $\mathbf{c} = \varphi(\mathbf{h})$ as conditions. After the training of CGAN, we can directly employ $\hat{\mathbf{x}} = g_{\theta}(\mathbf{z}, \mathbf{c})$ to synthesize RF signals from noise \mathbf{z} . Algorithm 4 presents the training of CGAN in GenHPE.

A.1.4. Backbone of Generative Models

We use a simplified ResNet [?] as the backbone of all generative models (*i.e.*, $\epsilon_{\theta}(\cdot)$ in diffusion models and $g_{\theta}(\cdot)$ in CGANs). This backbone has three blocks, and each block includes two convolutional layers. The numbers of filters in these layers over three blocks are {256, 512, 256}. The kernel sizes of convolutions over three blocks are {11, 7, 5}. The backbone of discriminator $f_{\omega}(\cdot)$ is implemented by an one-layer CNN whose convolutional layer has 64 filters with the kernel size of 3. All convolutional layers are followed by LeakyReLU activation functions.

A.2. Encoder-decoder Models

GenHPE implements an encoder-decoder model to estimate the coordinates of human body joints in 3D space from RF signals. Herein, we describe the architecture of encoder-decoder models, as shown in Supplement Figure 1 and 2.

A.2.1. Encoder

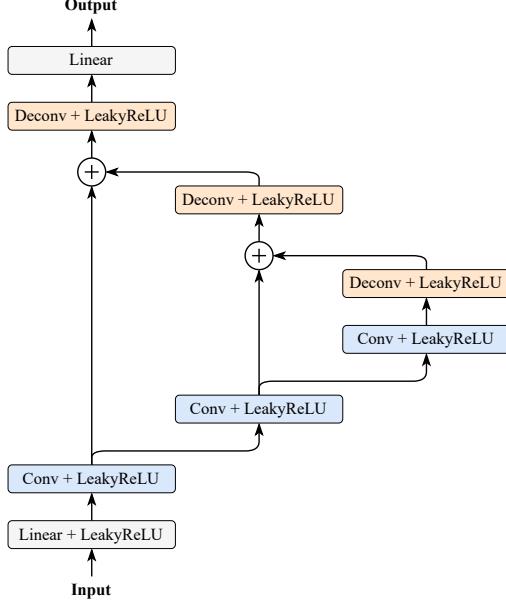
We design a simplified U-Net model [29] as the backbone of encoder. Supplement Figure 1 presents the network architecture of encoder, including three convolutional (Conv) layers and three deconvolutional (Deconv) layers. Specifically, each convolutional layer has 256 filters, while the kernel sizes are {7, 5, 3} for three layers, respectively. On the contrary, the kernel sizes of three deconvolutional layers are {3, 5, 7}, while each layer also has 256 filters.

A.2.2. Decoder

We devise a two-stream decoder with self-attention layers augmented by convolutional layers. Supplement Figure 2 presents the network architecture of decoder. Two streams have N_a and N_b blocks, respectively. Specifically, the first stream has 8 blocks ($N_a = 8$), where each self-attention layer has 8 heads, and the kernel sizes of convolutional layers over 8 blocks are {9, 7, 5, 3, 9, 7, 5, 3} with the dilation {2, 1, 2, 1, 2, 1, 2, 1}. The second stream takes transposed signals as input to model feature dependencies from another view, including 4 blocks ($N_b = 4$) with 4 heads in each self-attention layer. In the second stream, the kernel sizes of convolutional layers over 4 blocks are {3, 3, 3, 3} with the dilation {1, 1, 1, 1}. We concatenate the features from two streams and apply adaptive pooling to reduce feature dimensions. Finally, the features are fed into a multilayer perceptron (MLP) with two linear layers to output pose estimations, and the first linear layer is followed by a LeakyReLU activation function.

B. Experiments

Herein, we provide more details about our experiments, as well as more quantitative analysis and visualizations.



Supplement Figure 1. The network architecture of encoder.

B.1. Datasets

Our experiments include three public datasets, which collect different sources of RF signals, including WiFi, ultra-wideband (UWB), and millimeter wave (mmWave). Each dataset provides a set of RF signal samples and the corresponding annotations of human body joints in 3D space.

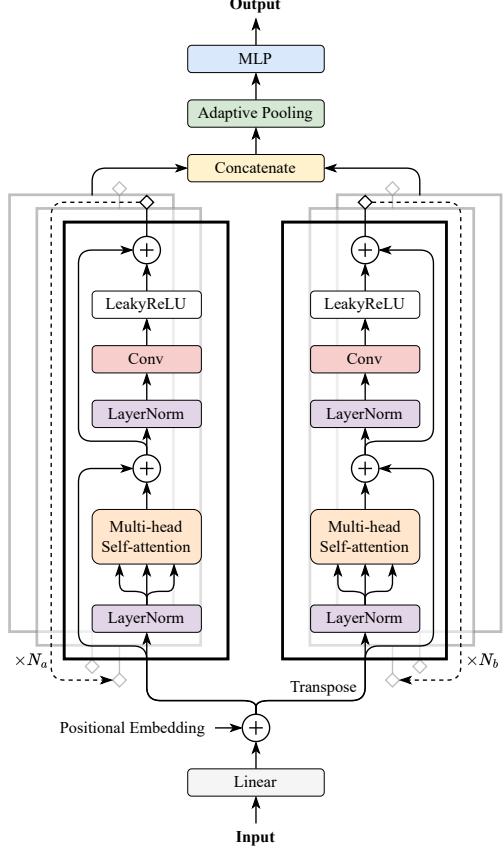
B.1.1. WiFi

The WiFi dataset [39] collects WiFi Channel State Information (CSI) using the carrier frequency of 5.64 GHz. The dimensions of each CSI sample (x in our paper) are 60×180 , annotated by a human pose label (y in our paper) whose dimensions are 14×3 (*i.e.*, 14 human body joints in 3D space). More descriptions about the dimensions of samples and labels are provided in the paper of this dataset [39].

There are 7 subjects and 3 environments in this dataset, from which we extract 30703 CSI samples for evaluation. (1) For Random splits, there are 24565 samples in the training set, 3071 samples in the validation set, and 3071 samples in the test set. (2) For Cross-Subject splits, there are 24746, 2750, and 3211 samples in the training, validation, and test sets, respectively. (3) For Cross-Environment splits, the training set includes 18914 samples, the validation set has 2102 samples, and the test set contains 9691 samples.

B.1.2. UWB

The UWB dataset [5] uses 3.99~4.49 GHz signals to collect RF signal samples, involving 6 subjects and 2 environments. Each UWB sample contains 35 channels and 40 time steps under the sample rate of 195~400 Hz, and each sample is annotated by 19 human body joints in 3D space. The di-



Supplement Figure 2. The network architecture of decoder.

mensions of each UWB sample (x in our paper) are 70×40 , including both the amplitudes and phases, while the dimensions of each human pose label (y in our paper) are 19×3 . More descriptions about the dimensions of samples and labels are provided in the paper of this dataset [5].

We extract 293930 samples from this dataset for evaluation. (1) For Random splits, we use 118150 samples as the training set, 14769 samples as the validation set, and 14769 samples as the test set. (2) For Cross-Subject splits, the numbers of samples are 101205, 11246, and 33791 for the training set, validation set, and test set, respectively. (3) For Cross-Environment splits, 53504, 5945, and 86793 samples are adopted for training, validation, and testing, respectively.

B.1.3. mmWave

The mmWave dataset [41] collects RF signals with 6 subjects and 4 environments using the signal frequency of 60~64 GHz. We extract 95666 samples by combining multiple frames in each 0.5-second window, and each sample includes a point cloud with the label of 17 human body joints in 3D space. In the point cloud sample, each point has 5 dimensions indicating its 3D coordinates, Doppler ve-

locity, and signal intensity. Since the maximum number of points is 493, the dimensions of each point cloud sample (\mathbf{x} in our paper) are 5×493 , and the dimensions of each human pose label (\mathbf{y} in our paper) are 17×3 . More descriptions about the dimensions of samples and labels are provided in the paper of this dataset [41].

(1) For Random splits, the training set includes 76532 samples, the validation set has 9567 samples, and the test set contains 9567 samples. (2) For Cross-Subject splits, there are 66830, 7426, and 21410 samples in the training, validation, and test sets, respectively. (3) For Cross-Environment splits, we employ 63373, 7042, and 25251 samples for training, validation, and testing, respectively.

B.2. Evaluation Metrics

Mean Per Joint Position Error (MPJPE). MPJPE computes the Euclidean distance between ground-truth and estimated human poses. Given a ground-truth label of N human body joints $\mathbf{y} \in \mathbb{R}^{N \times 3}$, the MPJPE of model estimation $\hat{\mathbf{y}}$ can be calculated as:

$$\begin{aligned} \text{MPJPE}(\hat{\mathbf{y}}, \mathbf{y}) &= \frac{1}{N} \sum_{i=1}^N \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|_2 \\ &= \frac{1}{N} \sum_{i=1}^N \sqrt{\sum_{j=1}^3 (\hat{\mathbf{y}}_{i,j} - \mathbf{y}_{i,j})^2}, \end{aligned} \quad (33)$$

where i is the joint index and j is the dimension index.

Procrustes-aligned MPJPE (PA-MPJPE). PA-MPJPE [20] adopts Procrustes alignment to transform the estimated pose $\hat{\mathbf{y}}$ into $\hat{\mathbf{y}}'$ according to the similarity between $\hat{\mathbf{y}}$ and \mathbf{y} . PA-MPJPE calculates the MPJPE of transformed pose estimation $\hat{\mathbf{y}}'$ as:

$$\text{PA-MPJPE}(\hat{\mathbf{y}}, \mathbf{y}) = \text{MPJPE}(\hat{\mathbf{y}}', \mathbf{y}). \quad (34)$$

Mean Per Joint Dimension Location Error (MPJDLE). MPJDLE [39] measures the mean absolute error of estimated pose $\hat{\mathbf{y}}$ in each dimension as:

$$\text{MPJDLE}_j(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^N |\hat{\mathbf{y}}_{i,j} - \mathbf{y}_{i,j}|, \quad (35)$$

and the overall MPJDLE of $\hat{\mathbf{y}}$ can be calculated as $\text{MPJDLE}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{1}{3} \sum_{j=1}^3 \text{MPJDLE}_j$.

B.3. Hyper-parameter Sensitivity Study

We further evaluate the impact of factor λ on the performance of GenHPE. Specifically, we vary $\lambda \in [0.0, 0.2, 0.4, 0.6, 0.8, 1.0]$ with WiFi and UWB in cross-subject and cross-environment scenarios.

As shown in Supplement Table 1, GenHPE consistently has lower estimation errors when $\lambda > 0.0$, which

	λ	MPJPE	PA-MPJPE	MPJDLE
Cross-Subject	0.0	260.59 \pm 4.43	112.64 \pm 1.13	122.62 \pm 2.15
	0.2	209.53 \pm 1.92	94.86 \pm 2.23	100.80 \pm 1.15
	0.4	211.64 \pm 2.84	95.22 \pm 1.77	101.87 \pm 1.68
	0.6	211.64 \pm 2.82	95.21 \pm 1.75	101.87 \pm 1.67
	0.8	211.64 \pm 2.82	95.21 \pm 1.74	101.86 \pm 1.66
	1.0	211.81 \pm 2.84	95.21 \pm 1.74	101.94 \pm 1.67
Cross-Environment	0.0	261.86 \pm 12.97	117.22 \pm 7.80	126.25 \pm 8.01
	0.2	230.30 \pm 2.46	106.71 \pm 0.81	109.26 \pm 0.92
	0.4	229.62 \pm 2.93	106.09 \pm 0.99	108.87 \pm 1.43
	0.6	228.27 \pm 1.33	106.04 \pm 0.82	108.08 \pm 0.80
	0.8	228.18 \pm 0.75	105.79 \pm 0.58	108.01 \pm 0.30
	1.0	228.88 \pm 0.56	105.81 \pm 0.90	108.17 \pm 0.24

Supplement Table 1. Performance of GenHPE regarding different regularization factors λ for 3D HPE with WiFi [39]. (Unit: mm)

	λ	MPJPE	PA-MPJPE	MPJDLE
Cross-Subject	0.0	298.99 \pm 2.61	135.73 \pm 0.45	138.81 \pm 1.00
	0.2	279.14 \pm 5.13	125.22 \pm 1.42	128.89 \pm 1.82
	0.4	277.37 \pm 3.28	125.23 \pm 1.32	128.27 \pm 1.18
	0.6	279.91 \pm 5.97	125.30 \pm 1.06	129.15 \pm 2.21
	0.8	278.29 \pm 2.61	125.12 \pm 0.62	128.44 \pm 1.02
	1.0	279.17 \pm 3.99	124.42 \pm 1.10	128.79 \pm 1.46
Cross-Environment	0.0	337.45 \pm 3.45	146.74 \pm 0.49	153.21 \pm 1.14
	0.2	295.37 \pm 10.39	135.48 \pm 6.27	136.49 \pm 4.76
	0.4	292.70 \pm 9.57	134.00 \pm 5.54	135.18 \pm 4.36
	0.6	290.32 \pm 5.91	132.38 \pm 2.98	133.93 \pm 2.45
	0.8	293.19 \pm 10.16	133.73 \pm 4.62	135.41 \pm 4.50
	1.0	292.80 \pm 12.66	133.97 \pm 6.52	135.36 \pm 5.84

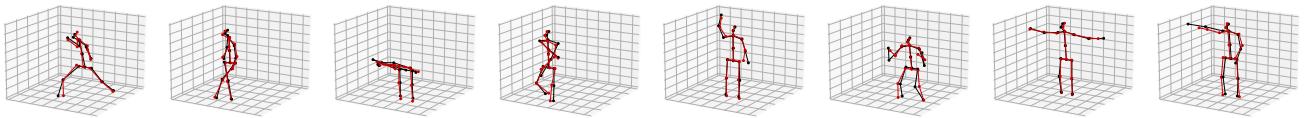
Supplement Table 2. Performance of GenHPE regarding different regularization factors λ for 3D HPE with UWB [5]. (Unit: mm)

demonstrates the efficacy of counterfactual regularization for domain-independent representation learning with WiFi. When we set $0.2 \leq \lambda \leq 1.0$, GenHPE shows stable performance for 3D HPE, proving its robustness for domain generalization with varying values of λ .

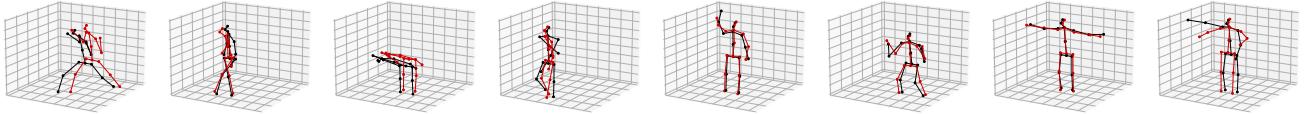
Supplement Table 2 discusses the performance of GenHPE with UWB using different values of λ . Results demonstrate that GenHPE achieves robust and state-of-the-art domain generalization performance for 3D HPE when $\lambda > 0.0$.

B.4. Supplemental Visualizations

We discuss the effectiveness of generative counterfactuals in GenHPE with mmWave. Supplement Figure 3 and 4 visualize 3D HPE results with/without generative counterfactuals in cross-subject and cross-environment scenarios. Results intuitively illustrate that GenHPE achieves more precise 3D HPE by learning domain-independent representations for cross-domain 3D HPE.

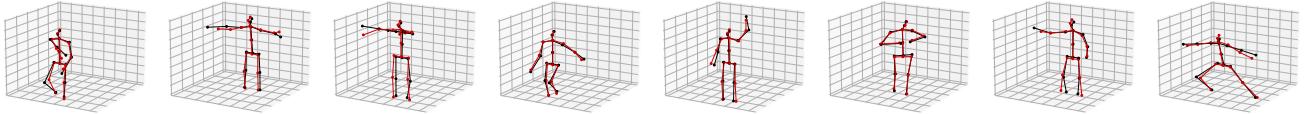


(a) Performance of GenHPE with generative counterfactuals using DDPMs ($T = 1000$).

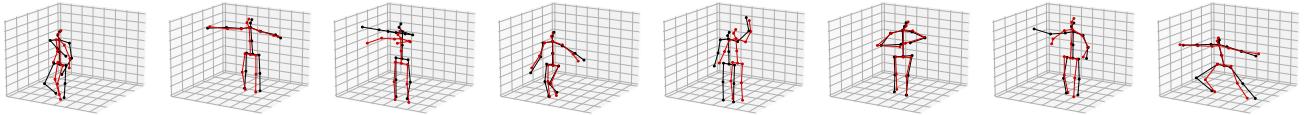


(b) Performance of GenHPE without generative counterfactuals.

Supplement Figure 3. Performance of GenHPE with/without generative counterfactuals in cross-subject scenarios. Black: ground-truth human poses. Red: model estimations.



(a) Performance of GenHPE with generative counterfactuals using DDPMs ($T = 1000$).



(b) Performance of GenHPE without generative counterfactuals.

Supplement Figure 4. Performance of GenHPE with/without generative counterfactuals in cross-environment scenarios. Black: ground-truth human poses. Red: model estimations.

References

- [1] Shahzad Ahmed, Dingyang Wang, Junyoung Park, and Sung Ho Cho. Uwb-gestures, a public dataset of dynamic hand gestures acquired using impulse radar sensors. *Scientific Data*, 8(1):102, 2021. [2](#)
- [2] Sizhe An and Umit Y Ogras. Fast and scalable human pose estimation using mmwave point cloud. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, pages 889–894, 2022. [2](#)
- [3] Sizhe An, Yin Li, and Umit Ogras. mri: Multi-modal 3d human pose estimation dataset using mmwave, rgb-d, and inertial sensors. *Advances in Neural Information Processing Systems*, 35:27414–27426, 2022. [1](#)
- [4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223. PMLR, 2017. [3, 5, 9, 11](#)
- [5] Mohammad J Bocus, Wenda Li, Shelly Vishwakarma, Roget Kou, Chong Tang, Karl Woodbridge, Ian Craddock, Ryan McConville, Raul Santos-Rodriguez, Kevin Chetty, et al. Operanet, a multimodal activity recognition dataset acquired from radio frequency and vision-based sensors. *Scientific Data*, 9(1):474, 2022. [1, 2, 6, 7, 8, 12, 13](#)
- [6] Yanlu Cai, Weizhong Zhang, Yuan Wu, and Cheng Jin. Posseirm: Enhance 3d human pose estimation on unseen camera settings via invariant risk minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2124–2133, 2024. [1](#)
- [7] Chen Chen, Gang Zhou, and Youfang Lin. Cross-domain wifi sensing with channel state information: A survey. *ACM Computing Surveys*, 55(11):1–37, 2023. [2](#)
- [8] Shuya Ding, Zhe Chen, Tianyue Zheng, and Jun Luo. Rfnet: A unified meta-learning framework for rf-enabled one-shot human activity recognition. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 517–530, 2020. [2, 3](#)
- [9] Huanzhang Dou, Pengyi Zhang, Wei Su, Yunlong Yu, Yining Lin, and Xi Li. Gaitgci: Generative counterfactual intervention for gait recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5578–5588, 2023. [3](#)
- [10] Junqiao Fan, Jianfei Yang, Yuecong Xu, and Lihua Xie. Diffusion model is a good pose estimator from 3d rf-vision. In

- European Conference on Computer Vision*. Springer, 2024. 2, 6, 7
- [11] Lijie Fan, Tianhong Li, Rongyao Fang, Rumen Hristov, Yuan Yuan, and Dina Katabi. Learning longterm representations for person re-identification using radio signals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10699–10709, 2020. 1
- [12] Jia Gong, Lin Geng Foo, Zhipeng Fan, QiuHong Ke, Hossein Rahmani, and Jun Liu. Diffpose: Toward more reliable 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13041–13051, 2023. 2, 6, 7
- [13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014. 3, 5, 10
- [14] Riccardo Guidotti. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, 38(5):2770–2824, 2024. 3
- [15] Vladimir Guzov, Aymen Mir, Torsten Sattler, and Gerard Pons-Moll. Human poseitioning system (hps): 3d human pose estimation and self-localization in large scenes from body-mounted sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4318–4329, 2021. 1
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2, 6
- [17] Steven M Hernandez and Eyuphan Bulut. Wifi sensing on the edge: Signal processing techniques and challenges for real-world systems. *IEEE Communications Surveys & Tutorials*, 25(1):46–76, 2022. 1
- [18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 3, 5, 6, 7, 9, 10
- [19] Shuoqiang Huang, Po-Yu Chen, and Julie Ann McCann. Diffar: Adaptive conditional diffusion model for temporal-augmented human activity recognition. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)*, pages 3812–3820, 2023. 3
- [20] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, 2013. 13
- [21] Sijie Ji, Xuanye Zhang, Yuanqing Zheng, and Mo Li. Construct 3d hand skeleton with commercial wifi. In *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems*, pages 322–334, 2023. 2
- [22] Wenjun Jiang, Hongfei Xue, Chenglin Miao, Shiyang Wang, Sen Lin, Chong Tian, Srinivasan Murali, Haochen Hu, Zhi Sun, and Lu Su. Towards 3d human pose construction using wifi. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020. 2, 6, 7
- [23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 6
- [24] Mehdi Mirza. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 3, 4, 5, 6, 9, 11
- [25] Silviu Pitis, Elliot Creager, Ajay Mandlekar, and Animesh Garg. Mocoda: Model-based counterfactual data augmentation. *Advances in Neural Information Processing Systems*, 35:18143–18156, 2022. 3
- [26] Yili Ren, Zi Wang, Sheng Tan, Yingying Chen, and Jie Yang. Winect: 3d human pose tracking for free-form activity using commodity wifi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(4):1–29, 2021. 2
- [27] Yili Ren, Zi Wang, Yichao Wang, Sheng Tan, Yingying Chen, and Jie Yang. Gopose: 3d human pose estimation using wifi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(2):1–25, 2022. 2
- [28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 4, 5, 9, 10
- [29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, proceedings, part III* 18, pages 234–241. Springer, 2015. 6, 11
- [30] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 3, 5, 6, 7, 9, 10
- [31] Vinkle Srivastav, Keqi Chen, and Nicolas Padoy. Selfpose3d: Self-supervised multi-person multi-view 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2502–2512, 2024. 1
- [32] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019. 1
- [33] Tom Van Wouwe, Seunghwan Lee, Antoine Falisse, Scott Delp, and C Karen Liu. Diffusionposer: Real-time human motion reconstruction from arbitrary sparse sensors using autoregressive diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2513–2523, 2024. 1
- [34] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 2
- [35] Dazhuo Wang, Jianfei Yang, Wei Cui, Lihua Xie, and Sumei Sun. Airfi: empowering wifi-based passive human gesture recognition to unseen environment via domain generalization. *IEEE Transactions on Mobile Computing*, 23(2):1156–1168, 2022. 2
- [36] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. Understanding and modeling of wifi signal based human activity recognition. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 65–76, 2015. 1

- [37] Yichao Wang, Yili Ren, Yingying Chen, and Jie Yang. Wi-mesh: A wifi vision-based approach for 3d human mesh construction. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, pages 362–376, 2022. 2
- [38] Bo Wei, Wen Hu, Mingrui Yang, and Chun Tung Chou. Radio-based device-free activity recognition with radio frequency interference. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, pages 154–165, 2015. 1
- [39] Kangwei Yan, Fei Wang, Bo Qian, Han Ding, Jinsong Han, and Xing Wei. Person-in-wifi 3d: End-to-end multi-person 3d pose estimation with wi-fi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 969–978, 2024. 1, 2, 6, 7, 8, 12, 13
- [40] Jianfei Yang, Yunjiao Zhou, He Huang, Han Zou, and Lihua Xie. Metafi: Device-free pose estimation via commodity wifi for metaverse avatar simulation. In *2022 IEEE 8th World Forum on Internet of Things*, pages 1–6. IEEE, 2022. 2, 6, 7
- [41] Jianfei Yang, He Huang, Yunjiao Zhou, Xinyan Chen, Yuecong Xu, Shenghai Yuan, Han Zou, Chris Xiaoxuan Lu, and Lihua Xie. Mm-fi: Multi-modal non-intrusive 4d human dataset for versatile wireless sensing. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2, 6, 7, 8, 12, 13
- [42] Xinyu Yi, Yuxiao Zhou, Marc Habermann, Soshi Shimada, Vladislav Golyanik, Christian Theobalt, and Feng Xu. Physical inertial poser (pip): Physics-aware real-time human motion tracking from sparse inertial sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13167–13178, 2022. 1
- [43] Yuxin Ying, Fuzhen Zhuang, Yongchun Zhu, Deqing Wang, and Hongwei Zheng. Camus: attribute-aware counterfactual augmentation for minority users in recommendation. In *Proceedings of the ACM Web Conference 2023*, pages 1396–1404, 2023. 3
- [44] Mehdi Zemni, Mickaël Chen, Éloi Zablocki, Hédi Ben-Younes, Patrick Pérez, and Matthieu Cord. Octet: Object-aware counterfactual explanations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15062–15071, 2023. 3
- [45] Fusang Zhang, Zhaoxin Chang, Jie Xiong, Junqi Ma, Jiazh Ni, Wenbo Zhang, Beihong Jin, and Daqing Zhang. Embracing consumer-level uwb-equipped devices for fine-grained wireless sensing. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(4):1–27, 2023. 2
- [46] Xiheng Zhang, Yongkang Wong, Xiaofei Wu, Juwei Lu, Mohan Kankanhalli, Xiangdong Li, and Weidong Geng. Learning causal representation for training cross-domain pose estimator via generative interventions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11270–11280, 2021. 3
- [47] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 2, 6, 7
- [48] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7356–7365, 2018. 1
- [49] Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh, Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, and Antonio Torralba. Rf-based 3d skeletons. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 267–281, 2018. 1
- [50] Mingmin Zhao, Yingcheng Liu, Aniruddh Raghu, Hang Zhao, Tianhong Li, Antonio Torralba, and Dina Katabi. Through-wall human mesh recovery using radio signals. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10112–10121, 2019. 1
- [51] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11656–11665, 2021. 2, 6, 7
- [52] Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. Deep learning-based human pose estimation: A survey. *ACM Computing Surveys*, 56(1):1–37, 2023. 1
- [53] Jingxiao Zheng, Xinwei Shi, Alexander Gorban, Junhua Mao, Yang Song, Charles R Qi, Ting Liu, Visesh Chari, Andre Cormann, Yin Zhou, et al. Multi-modal 3d human pose estimation with 2d weak supervision in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4478–4487, 2022. 1
- [54] Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. Zero-effort cross-domain gesture recognition with wi-fi. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 313–325, 2019. 2, 6
- [55] Yunjiao Zhou, He Huang, Shenghai Yuan, Han Zou, Lihua Xie, and Jianfei Yang. Metafi++: Wifi-enabled transformer-based human pose estimation for metaverse avatar simulation. *IEEE Internet of Things Journal*, 10(16):14128–14136, 2023. 2, 6, 7