PIPE: Teaching WiFi Sensing to Ignore Position

Kenzo Heijman[1]

K.E.Heijman@student.tudelft.nl

Supervisor(s): Fabian Portner[1]**, Arash Asadi**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 24, 2026

Name of the student: Kenzo Heijman
Final project course: CSE3000 Research Project
Thesis committee: Fabian Portner, Arash Asadi, Hayley Hung

Abstract

While WiFi signals are primarily intended for data transmission, their interaction with the environment causes subtle changes in the received signal. Receivers can measure these changes and use them to infer events occurring within the environment. For example, we can detect when a subject is jumping or standing still. However, the observed signal depends strongly on geometry and the subject's position. As a result, a model trained on one set of positions can fail entirely when the subject moves elsewhere in the same room. In recent research, contrastive learning has been used to combat this problem, with some success. Within this domain, current solutions use a distant proxy of the subject's pose, like activity labels, which might yield subpar results. Using a more direct proxy, like using a subject's pose as seen by a Motion Capture system, should result in a system that generalizes better. In this paper, we introduce **PIPE**, a contrastive learning approach to WiFi sensing which defines positive pairs using motion-capture derived pose similarity. PIPE achieves similar performance to SHARP [16] on Human Activity Recognition (HAR), slightly beating it on unseen positions with an unseen subject. However, pose- and label-derived supervision for PIPE do not substantially differ in performance on HAR. We attribute this result to the activity label aligning directly with the objective of this task.

1 Introduction

WiFi devices are commonplace nowadays, connecting many wireless devices by sending signals over the air. These WiFi devices communicate by transmitting electromagnetic waves. As these waves travel, the environment alters them through scattering, reflection, and other effects. To communicate reliably, a WiFi router estimates this wireless channel and corrects for it, capturing and correcting for the environment's impact on the signal.

A new promising technology called WiFi sensing reverses this idea: instead of removing these disturbances, we analyze them to learn things about the environment, such as the current body pose or movement of a subject.

Working solutions could be deployed in situations where camera feeds are unwanted for privacy reasons, and would work no matter the lighting conditions. For example, to detect when an elderly person falls. However, there are still several problems that need to be solved before this technology becomes widely usable.

A major problem in WiFi sensing is that a signal seen at a router when a subject performs an activity is heavily dependent on the position of this subject in the room. This makes it hard to create a model that generalizes, as such a model has to account for how the signals interact with the room for every activity at all positions they could be performed. This makes naive approaches unusable, unless both the WiFi device and the subject never move. We therefore need a way to teach the model what stays the same across positions.

In general machine learning settings, this is often done with a technique called contrastive learning. The basic idea of contrastive learning is to train a model to learn a transformation that makes related examples look similar. More specifically, the model is trained with examples that should be similar and examples that should remain different. It then learns to map the similar examples close together, while keeping the different examples apart.

Some research has tried to apply this to the WiFi sensing domain, and these methods mainly differ in how positive samples are defined. For example, DT-Pose [4] defines positive samples as time-adjacent, and WiGr [27] uses activity labels.

The main problem with these approaches is that to determine if a pose is similar, a proxy of the actual pose is used, which may lead to subpar representations. The open question is whether a more direct proxy, like motion capture (MoCap), can better help the model generalize to unseen positions. Furthermore, this approach would be task-independent, meaning that the samples map to a general representation of a pose which could be used for any number of tasks.

To test this idea, we introduce PIPE, a **P**osition-**I**nvariant **P**ose **E**mbedder. PIPE uses motion capture to define which samples should be treated as similar during contrastive training. We show that this signal leads to downstream position generalization, but does not perform substantially better than Supervised Contrastive learning [11] on Human Activity Recognition (HAR). PIPE achieves similar performance to the state-of-the-art SHARP [16], and retains slightly more accuracy in the most difficult setting when transferring to unseen positions with an unseen subject. We further analyze which design choices impact this approach, including pose-based and velocity-based supervision, normalization, and encoder window size.

2 Technical Background

Channel State Information In a room, a signal can arrive through several paths: for example, directly from transmitter to receiver, or indirectly after reflecting off walls, furniture, or a person. Each path has a different strength and delay, and the receiver sees a combination of all paths that reach it. A person standing or moving in the room changes these paths. For example, when a person moves to another position, they may block one path, create a new reflection, or make some paths longer. The combination of all these paths as seen by the receiver is called *Channel State Information* (CSI).

Each position in the room has its own view of how the transmitted signal reaches it, which determines the CSI at that position. CSI can be wildly different based on

the position of the router in the room. Likewise, CSI also changes based on a subject's activity or pose in the room. This is the change in signal WiFi Sensing techniques rely on. For an example, see figure 2.

An important effect of this is that when a subject performs a movement in one part of the room, the channel will change differently from when this same movement is done in another part of the room. This is called *position dependence*, and the reason why it is difficult to create a model that generalizes across positions.

Contrastive Learning The general idea behind contrastive learning [3] is that we have data samples that are in some way related, and other samples that are not (or less) related. The goal of contrastive learning is to teach a Machine Learning model to compress all related samples into a similar representation vector while giving unrelated samples a different representation vector. Hence the full name 'Contrastive Representation Learning'. See figure 1 for an overview of this.

Contrastive learning has shown strong results in domains where the input contains a lot of nuisance variation alongside a comparatively weak signal, including vision [3] and supervised classification [11]. Recently, in the WiFi sensing domain, contrastive learning has been shown to create pose regression models that generalize over multiple rooms [4].

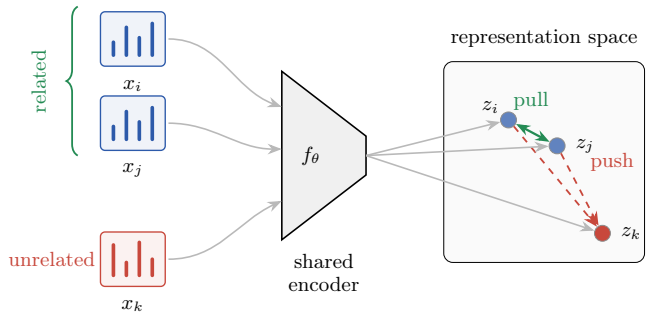

Figure 1: Illustration of contrastive learning. Shared encoder f_θ maps samples to a feature space. Related samples pulled together, unrelated samples pushed apart.

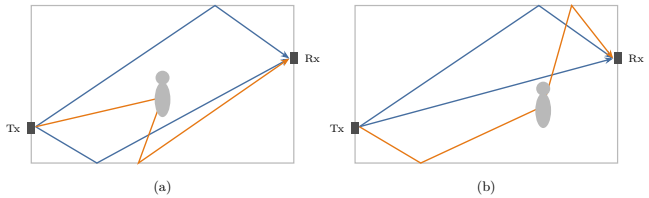

Figure 2: Position dependence of CSI. (a) The subject intercepts the direct path, which scatters off the body and reaches the receiver via a longer detour. (b) The subject intercepts another path instead when in a different position.

3 Related Work

There are two main generalization problems in the WiFi Sensing domain: (1) *Domain-independence* asks whether a model trained in one room can work in another. (2) *Position-independence* asks whether a model trained with the subject at one set of positions in a room can work when the subject moves to a different position in the same room. In both cases, the same pose produces a different CSI signature, so a model trained on raw CSI cannot reliably tell whether it has learned about the pose or about the geometry of the room.

Approaches for domain independence include hand-engineered domain-invariant signal features [28, 16], feature distribution alignment [29, 22], adversarial removal of environment and subject information [9], cross-view consistency across co-located antennas [24], and self-supervised representation learning on CSI [25, 1, 4].

For position-independence, existing approaches rely on hand-engineered position-invariant features [28, 15], meta-learning [5], conditioning on calibrated transceiver geometry [8], or implicit invariance through multi-device cross-view consistency [13]. UniFi [13] introduces a mutual information-maximization regularizer that encourages representations of the same gesture observed by different receivers to become more similar in the embedding space.

Contrastive learning has been particularly effective at producing embeddings that transfer across rooms. DT-Pose [4] treats temporally adjacent CSI frames as positive pairs and uses the resulting embeddings to train a final supervised model for pose regression. The temporal proximity signal defines positive pairs by when samples were captured, so the same pose performed at a different time or in a different position is never treated as a positive pair. WiGr [27] learns a similar embedding space but defines positives by activity label rather than by pose. These methods differ mainly in what defines a positive pair: temporal proximity [4], the same activity seen by multiple receivers [2], or the class label [27, 11]. Absent from this is using MoCap derived positive pairs.

Some work has used WiFi for pose estimation with motion-capture or video [10, 26, 7, 23, 18], but in that setting pose simply serves as a regression target or label.

No approach generates task-independent embeddings, while removing the subject position. Using proxies to do this, like class labels or time-closeness might result in subpar representations. For temporally adjacent windows, position is held constant within positive pairs and the encoder is never forced to discard it. Class labels are a coarse proxy that collapses within-class pose variation.

The hypothesis is that adding a more direct proxy like pose-similarity as the supervision metric, gives a finer and position-decoupled supervision signal, allowing a contrastive model to generate better task-independent, position-invariant embeddings.

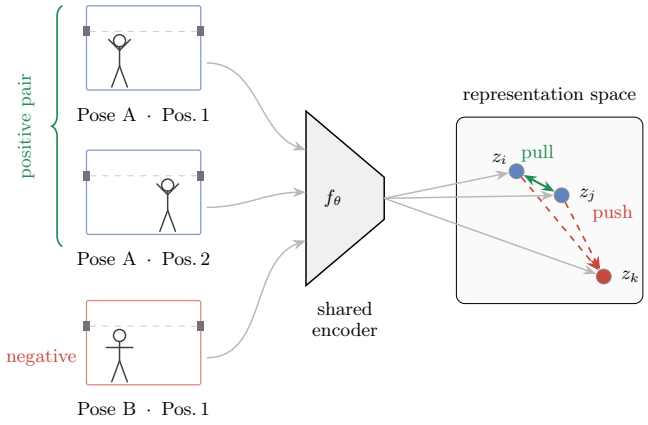

Figure 3: Contrastive learning approach based on CSI and subject pose.

4 Method

The goal is to train a model that compresses a subject's pose into a vector, where similar poses are similar vectors and different poses are different vectors, regardless of the subject position.

To do this, we train a Transformer model to take a sample of CSI data over a short window (for example 100 ms), alongside the subject MoCap data over that period.

Then, we train a model to make two representations similar when a similar pose or movement is performed in both samples. We do this by defining a special loss function that operates over multiple similar and dissimilar samples. Ideally, this would eventually allow our model to filter out all irrelevant information from a sample, and only give us the subject pose & movement information. An example can be seen in figure 3.

4.1 Model Architecture

The architecture we propose (figure 4) consists of three components: a per-receiver CSI encoder, a self-attention stage, and a class-attention stage. Its output is a final PIPE embedding that can be used for downstream tasks.

The CSI encoder uses a separate sub-encoder for each receiver, because every receiver sits at a different position and therefore observes a different static channel, and because receivers have different hardware-dependent biases such as antenna type and silicon imperfections. For each timestep, the encoder stacks all subcarriers across antennas into a single 456 dimensional vector and compresses it with an MLP into a 256 dimensional representation. This yields a per-receiver, per-timestep CSI embedding.

These embeddings are stacked into an array. A sinusoidal (sin-cos) embedding is added along the time axis so that tokens from different timesteps can be distinguished, and a learnable per-receiver embedding is added to all tokens from a given receiver.

The resulting tokens are passed through a self-attention block [21], allowing every token to exchange information with every other. A learnable pose token is then introduced into a final cross-attention (class-attention) block [6], where it attends over all tokens to extract the pose-relevant information into a single vector: the final PIPE embedding.

One issue remains. Each window spans multiple timesteps and therefore corresponds to many slightly different MoCap poses. For some activities this is harmless, as for example the pose barely changes over a 0.1 s window; for others, such as boxing, the pose can move by 60–120 cm [14]. We therefore compute the contrastive loss only on the pose of the final timestep. This means the last frame of the window is the pose we aim to embed, while the preceding frames serve as temporal context. This also motivates a short stride, since consecutive samples correspond to different poses. The code and training hyperparameters can be found in the project repository.

4.2 Contrastive Objectives

In order to compare two poses, we normalize the MoCap data: we estimate the subject's yaw angle from the left-to-right hip axis, all joints are rotated to a canonical facing direction, and MoCap marker positions are expressed relative to the hip midpoint. Since both a subject's movement and pose are important for an activity, we test two similarity metrics focusing on each of these.

Soft-InfoNCE loss. Standard InfoNCE [19] requires discrete positive/negative pairs. Since MoCap provides continuous pose similarity, we use Soft-InfoNCE [12], which replaces the binary positive indicator with a soft target matrix $S \in [0,1]^{B\times B}$ encoding graded similarity between all pairs in a batch. The encoder produces a raw embedding $\mathbf{z}_i$, which we use for downstream HAR. Before the loss, we pass $\mathbf{z}_i$ through a projection head [3] to give $\mathbf{p}_i$. The loss is:

$$\mathcal{L} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{j\neq i} S_{ij}\log\frac{\exp\left(\mathbf{p}_i^\top \mathbf{p}_j / \tau\right)}{\sum_{k\neq i}\exp\left(\mathbf{p}_i^\top \mathbf{p}_k / \tau\right)} \qquad (1)$$

Pose similarity. The soft target S_{ij} is computed from the instantaneous (last-frame) joint position vectors $\mathbf{q}_i$ using bandwidth σ_p, then row-normalized:

$$S_{ij} = \frac{\exp(-\|\mathbf{q}_i - \mathbf{q}_j\|_2 / \sigma_p)}{\sum_{k\neq i}\exp(-\|\mathbf{q}_i - \mathbf{q}_k\|_2 / \sigma_p)}, \quad S_{ii} = 0 \qquad (2)$$

Windows with similar body configurations should thus attract in embedding space, regardless of where in the room they occurred.

Velocity similarity. To also capture motion dynamics, we can use a velocity similarity metric in place of or combined with the pose similarity. Here, each window is summarized by a per-joint mean absolute speed $\mathbf{v}_i$ (mm/frame), using bandwidth σ_v:

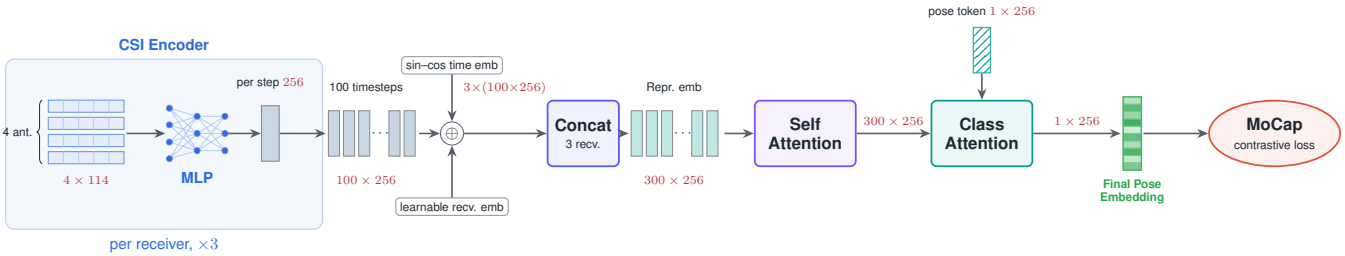

Figure 4: Overview of the proposed architecture. A per-receiver MLP encoder maps each timestep's stacked subcarriers to an embedding; sin–cos time and learnable receiver embeddings are added, the three receivers are concatenated into a token sequence, refined by self-attention, and pooled by a class-attention block with a learnable pose token into the final PIPE embedding, which is trained with the MoCap-supervised Soft-InfoNCE contrastive loss.

$$S_{ij}^{\mathrm{vel}} = \frac{\exp(-\|\mathbf{v}_i - \mathbf{v}_j\|_2 / \sigma_v)}{\sum_{k \neq i} \exp(-\|\mathbf{v}_i - \mathbf{v}_k\|_2 / \sigma_v)}, \quad S_{ii}^{\mathrm{vel}} = 0 \qquad (3)$$

Body-state similarity The *Body-state* similarity combines both signals via the element-wise product $S_{ij} = S_{ij}^{\mathrm{pose}} \cdot S_{ij}^{\mathrm{vel}}$, pulling together windows that share both a similar body configuration and movement speed.

5 Experimental setup

5.1 Data and preprocessing

Dataset. We use a dataset of 156 minutes of recordings collected in a single room with three ASUS WiFi receivers sampling at 1 kHz and a motion capture system. One subject performed ten activities in a continuous manner: squat, jumping jack, jump, boxing, walk, stand, run, stir a pot, raise left foot, and raise left arm. These activities are presegmented and we take multiple 2-second parts to produce multiple training samples per activity sample.

CSI preprocessing Each receiver captures CSI amplitude per antenna per subcarrier per timestep. We segment raw CSI into 100-sample windows (100 ms) with a stride of 10 samples. We then normalize each window by subtracting its temporal mean and dividing by its temporal standard deviation per subcarrier, both computed along the time axis. See table 1 for a comparison of multiple considered preprocessing methods.

5.2 Activity Classifier

The embeddings are meant to be task-agnostic and could be used to extract a wide range of information like pose and activity. As a simple test, we evaluate whether they can be used to classify presegmented activity data. This task is called Human Activity Recognition (HAR). For this, we use a classifier model to predict a subject's activity based on the embedded CSI over multiple windows.

We use a single-layer Transformer encoder [21] with an MLP head. Activity segments are typically long and contain multiple repetitions. The classifier is trained from scratch on the frozen embeddings produced by the pretrained Position-Invariant Pose Embedder (see figure 5). This classifier architecture and training procedure are held fixed across all conditions, so that the embeddings are evaluated equally. The training settings and this encoder can be found in the project repository.

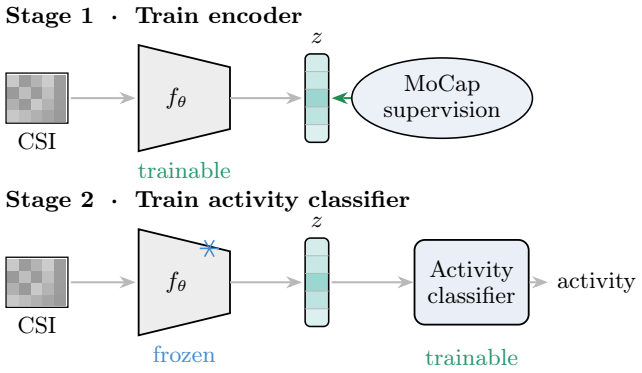

Figure 5: Two-stage setup. The encoder f_θ is trained contrastively under MoCap supervision, then frozen while a HAR classifier is trained on its embeddings.

5.3 Evaluation Protocols

It is important that the model is evaluated in a manner that can show the effect of the subject position on the downstream accuracy. Two evaluation protocols are used: namely, a *Position-based holdout split* within the training dataset, and a *Zero-shot transfer* to a completely unseen subject and certainly unseen position.

Position-based holdout split via K-means. To evaluate whether our learned embedding generalizes to unseen positions, we want to have a holdout set of positions that are different from what the model has seen in training.

To do so, we first need to determine the subject's position during the performed activities. We take the median of all MoCap hip-XZ positions per activity sample in the dataset. We then put these on a 2D graph as seen in figure 6. Finally, we use K-means to get three distinct

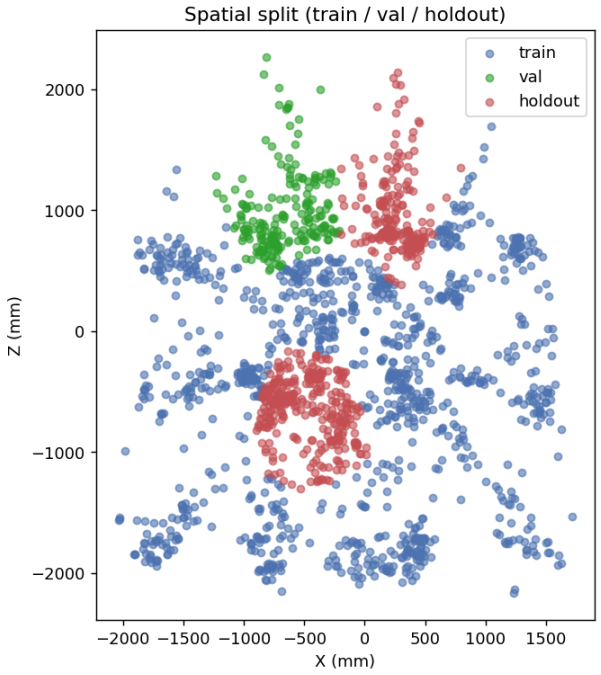

Figure 6: Position-based holdout split via K-means (K=12) on median XZ-MoCap positions grouped into three splits.

clusters of positions, which are then assigned to train, validation, and holdout splits.

The contrastive encoder is only trained on the train & validation samples, so that the Holdout set measures performance in positions that both the encoder and classifier have not yet seen. For activities that have a lot of XZ movement (running), there could be some contamination in the position, which is a known confound.

Zero-shot evaluation. To assess generalization of the model, a new subject performs all activities at three positions: the room middle, room edge (in front of the transmitter), and outside the nominal capture area. Zero-shot means the encoder and classifier are used without any fine-tuning or adaptation on these new domains.

This evaluates whether the learned representations and classifier transfer across subjects. Additionally, with this dataset we can measure whether there is a difference in accuracy between (seen) middle positions and (certainly unseen) outside positions. See figure 7.

6 Results

6.1 Normalization methods

In general, normalization is extremely important in the Machine Learning domain. It lets models focus more on the signal contained in the data, instead of overfitting on the absolute magnitude of feature values.

We evaluated six strategies: no normalization, taking the difference over consecutive frames, removing the mean per subcarrier per frame, whitening (mean removal

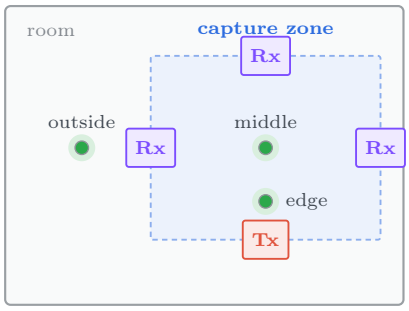

Figure 7: Zero-shot evaluation positions. The subject stands at the *middle* of the room, at the *edge* (in front of the transmitter), or *outside* the capture zone entirely.

Table 1: Effect of CSI preprocessing on HAR accuracy. All runs: pose-only loss, $L = 2$, 2 epochs.

Preprocessing	Val acc.	Holdout acc.
None	0.729	0.278
Temporal diff	0.658	0.389
Mean removal	0.718	0.483
ℓ_1 frame norm	0.748	0.270
ℓ_1 + whitening	0.832	0.678
Whitening	**0.858**	**0.711**

followed by division by standard deviation per subcarrier per frame), per-frame ℓ_1 normalization [17], and ℓ_1 normalization combined with whitening.

Normalization results. Table 1 shows when full whitening normalization is applied, the validation-holdout generalization gap reduces by 67%.

Interestingly, ℓ_1 normalization scores the lowest, even below no preprocessing. This is surprising, as it was the best performing method in Portner et al. [17]. An explanation for this could be that their paper targets cross-device deployment, where AGC varies across hardware families.

6.2 Supervision Objectives

To see if pose and velocity contain different types of information that could be combined for better embeddings, we tested three variations of the similarity metric: the *Pose* similarity, the *Velocity* similarity and the *Body-state* similarity.

Table 2 shows all three losses perform better when the right hyperparameter value is chosen. However, it is not clear one of the supervision methods is fundamentally better than the others, as the relative differences are not substantial. What can be seen is that the *Body-state* similarity performs slightly worse, but is generally more stable over the hyperparameter values, and could therefore be a better loss choice even though the performance difference is unclear.

SupCon supervision objective. Since we want to know the impact of adding detailed pose data into the supervision method, we test our existing *Body-state*

Table 2: HAR accuracy per bandwidth hyperparameter. $L = 2$ encoder layers, epoch 2. Body-state σ_p=150. 100 ms frames.

Similarity	Bandwidth	Val	Holdout
Pose	$\sigma_p = 75\,\text{mm}$	0.848	**0.716**
	$\sigma_p = 150\,\text{mm}$	0.848	0.658
	$\sigma_p = 300\,\text{mm}$	0.804	0.614
Velocity	$\sigma_v = 0.10\,\text{mm}$	0.764	0.682
	$\sigma_v = 0.25\,\text{mm}$	0.813	0.752
	$\sigma_v = 0.5\,\text{mm}$	**0.832**	**0.790**
	$\sigma_v = 2\,\text{mm}$	0.652	0.608
Body-state	$\sigma_v = 0.25\,\text{mm}$	0.807	0.740
	$\sigma_v = 0.5\,\text{mm}$	0.828	**0.748**
	$\sigma_v = 1\,\text{mm}$	**0.851**	0.705
	$\sigma_v = 2\,\text{mm}$	0.850	0.725

Table 3: In-distribution holdout and zero-shot accuracy for direct-label SupCon vs. the best Body-state (MoCap) encoder. σ_p=150, σ_v=0.5. Both: $L = 2$, whitening, 100 ms windows.

Supervision	Holdout	Zero-shot
Body-state	**0.795**	0.407
Direct labels (SupCon)	0.760	**0.490**

loss against supervised contrastive learning, using class labels, as in SupCon [11].

Table 3 shows SupCon works nearly as well as the *Body-state* similarity for the Holdout set and transfers better to the zero-shot dataset. This is an important finding, because it means that the MoCap supervision objective is not adding clear value to the embeddings in this evaluation scenario.
However, since we are only testing the embeddings usefulness using HAR, which has as goal to predict the class label, it might be the case that SupCon performs exceptionally well for only this task.

6.3 Window Size

A smaller window size means less temporal information and less tolerance for fluctuations in CSI amplitude values, but more granular embeddings.

To measure the impact of the window size, we test window sizes 10, 50, 100 and 200 ms given the previously highest scoring hyperparameters. This window size also influences 2 major parts of the model: the velocity loss, and the normalization preprocessing.

Firstly, the velocity loss is calculated as a per-window mean absolute speed. A larger window means this speed gets generally more smoothened, as a short burst of movement within this window will result in the same value as a slow movement during the entire window.

Secondly, the normalization preprocessing is done per window. This means the window size directly influences the window size that is used for normalization, meaning more or less values are used.

Table 4: Window-size sweep: in-distribution HAR accuracy. Body-state loss (σ_p=150, σ_v=1), whitening, stride fixed at 10 ms.

Window	Val	Holdout
10 ms	0.606	0.664
50 ms	**0.829**	**0.825**
100 ms	0.817	0.749
200 ms	0.714	0.620

Table 5: Window-size sweep: zero-shot accuracy by position (%). Body-state loss (σ_p=150, σ_v=1).

Window	Middle	Edge	Outside	Overall
10 ms	52.9	20.7	55.7	43.1
50 ms	**59.3**	**37.1**	**61.4**	**52.6**
100 ms	50.0	28.6	44.3	40.9
200 ms	49.3	21.4	52.9	41.2

As can be seen in tables 4 & 5, window size has a major effect on the Holdout score and Zero-shot dataset result.

The 50 ms window scores higher on every position. Edge sees the biggest improvement, which could be due to the fact that drastic CSI changes are common when the subject is close to the transmitter, which could be potentially helped by more granular normalization.

6.4 Position invariance

In order to check if the features are truly position-invariant, we train an 8-layer MLP to predict the mean Holdout hip-XZ position from both the pure CSI per frame and the PIPE embedding in that frame. In order to compare samples of the same dimension, PCA-256 was taken of each 50 ms CSI sample.

Table 6 shows position is slightly more recoverable from the PIPE embeddings than from raw CSI under explicit MLP decoding (a 15% reduction in median error), though at 74.4 cm the error remains large. We note that recoverability under a powerful decoder is a strict test: position information can persist in a low-variance subspace of the embedding without affecting the contrastive objective or the downstream classifier. The embeddings are therefore not invariant in the strict decoding sense, even if position is not the dimension the HAR classifier relies on.

Holdout vs Zero-shot per activity We compute the per class difference for a full PIPE + classifier model pipeline, comparing the Holdout sets and the Zero-shot dataset in table 7. We see some classes transfer significantly worse than others. This could be because the embeddings only embed activity data for some activities, or the new subject performed the activities differently.

However, comparing the (seen) middle positions with the (completely unseen) outside positions, we see that

Table 6: Cross-position hip-position regression.

Representation	Median error (mm) ↓
Naive (mean train position)	905
Whitened CSI (PCA-256)	876
PIPE embedding (256)	**744**

Table 7: Per-activity difference between in-distribution holdout accuracy and zero-shot accuracy per position zone. 2s window. Frame size 50.

Activity	ΔMiddle	ΔEdge	ΔOutside
stand	+0.126	+0.054	+0.126
stir a pot	+0.099	−0.543	−0.329
run	+0.080	+0.080	+0.080
walk	+0.031	−0.041	−0.041
raising left arm	+0.016	−0.413	+0.016
jumping jack	−0.062	−0.848	−0.633
raising left foot	−0.250	−0.750	−0.107
boxing	−0.592	−0.592	+0.337
jump	−0.812	−0.669	−0.812
squat	−0.877	−0.877	−0.877

Table 8: SHARP fused amplitude-Doppler baseline accuracy per 2 s window, on train/val/holdout.

Split	Accuracy	N
Train	0.82	7763
Val	0.71	1505
Holdout	0.74	3285

Table 9: SHARP zero-shot transfer to unseen positions, accuracy per 2 s window.

Position	Accuracy	N
Middle	0.65	560
Edge	0.35	560
Outside	0.56	560
Overall	**0.52**	1680

some classes transfer very well to the unseen positions. We can therefore conclude that, for some activities on some unseen positions, the embeddings retain as much discriminative information as they do for the same activities on seen positions. This is important, because a general model that achieves this across all classes would, by definition, be position-invariant.

6.5 Comparison with SHARP

To see how this model performs against a state of the art position-invariant approach, we test PIPE against SHARP [16], which uses Doppler frequencies and Convolutional Neural Networks, but no contrastive learning. Tables 8 & 9 show SHARP performs very similar to PIPE. It can be seen that SHARP loses more accuracy than PIPE when transferring to unseen positions, but has a slightly higher accuracy to start with. This could suggest the contrastive learning method is slightly more position-invariant than SHARP, but trades in-distribution performance to do so.

7 Responsible Research

We release all code written for the evaluations together with instructions on its usage:
`https://github.com/Blagues/pipe-research`

The dataset is part of ongoing research within the group and intended to be published with its publication. However, similar datasets including motion capture are also already publicly available [26].

It is important to acknowledge that there are real concerns about WiFi Sensing technology being used in nefarious ways. For this research, all subjects provided informed consent that the collected data may contain obfuscated information about them. Further work is advised to keep the privacy aspect in mind.

LLM usage Claude Opus 4.8 was used to write code for this project and help with researching and summarizing papers. In this paper, Claude was used to help format all tables, all graphs and images and partially help with the writing of sections: 2, 3, and 4. For the writing, Claude was used to format formulas and help with rewording of convoluted sentences only. See Appendix A for more details.

8 Discussion

An interesting finding is that the pose-derived similarity metrics resulted in only a minor improvement over a label-derived similarity metric. One possible explanation is that label-derived similarity aligns almost perfectly with the task of pre-segmented HAR. By using label-similarity, the contrastive model is trained directly for the downstream task, whereas pose-derived similarity acts as a more indirect proxy. As a result, improvements in pose awareness may not necessarily translate into better HAR performance. This suggests that the benefits of pose-based supervision may depend strongly on the downstream task.

Furthermore, the chosen approach has some limitations. Mainly, since we are focusing only on position-invariance within one room, one dataset and one training subject, this limits the strength of the claims that can be made about this approach. The use of Motion Capture in the dataset also makes collecting similar datasets expensive, which may hinder the creation of large-scale datasets for this method. This in itself might harm the practical applicability compared to other methods with easier-to-collect datasets.

Additionally, during experimentation it appeared that increasing the number of training epochs beyond a small number provided limited benefit. This may indicate that the available training data constrains further improvements, although a more systematic investigation would be required to confirm this.

Apart from this, the stated percentage is an average over all classes. Since some classes score near 100% and others score near 0%, this average does not fully convey the spread. In addition, there are no error bars, meaning we cannot fully tell if results are statistically significant or due to randomness.

Due to time constraints, PIPE was only compared to SHARP [16], a comparable state of the art approach. Thus, PIPE was not tested against other contrastive learning approaches like DT-Pose [4] or WiGr [27], which makes it harder to judge the approach in the relevant context of contrastive learning for WiFi Sensing.

Finally, ℓ_1-norm performing much worse than in Portner et al. [17] is unusual. Although no implementation errors were identified, this result should be further investigated.

9 Conclusions and Future Work

PIPE performs well on classifying presegmented HAR samples and produces embeddings that perform on par with the state of the art, SHARP [16], slightly exceeding it when tested on unseen positions with an unseen subject.

PIPE's best zero-shot cross-subject cross-position performance lies at around 61.4%, with some activities performing as well on cross-subject cross-position as in the original dataset. However, some activities break down on cross-subject seen-position data. The best seen-subject cross-position accuracy lies at 82.5%.

We find that normalization, specifically whitening, is extremely important for the proposed approach, as the contrastive learning method does practically not work without it.

The pose-derived similarity metrics performed similarly on HAR, with *Body-state* being the most stable.

While pose-derived supervision achieved competitive performance, it only provided a minor improvement over label-derived [11] supervision for the HAR task considered in this research. Therefore, the results do not provide sufficient evidence to conclude that pose-derived similarity is superior to label-derived similarity for presegmented HAR.

An interesting direction for future work is to evaluate PIPE on tasks where pose information is more directly relevant to the downstream objective. For example, in pose-regression tasks, pose-derived similarity may provide a stronger training signal than label-derived or temporally-derived similarity. Investigating whether the benefits of pose-based supervision become more apparent in such settings would help clarify when pose-similarity supervision offers an advantage over simpler contrastive objectives.

Overall, these results demonstrate that pose-derived supervision is a viable approach for learning position-invariant WiFi representations, while also highlighting the importance of evaluating such representations on tasks that are closely aligned with the supervisory signal used during training.

A Appendix

A.1 Extra results

See figure 8 & table 10.

A.2 LLMs for writing

Claude 4.8 Opus by Anthropic has been specifically helpful in writing this paper. LLMs in this paper were used to reword single sentences, and this was done for about 15% of the sentences in this paper.

For example, take the following format similar to actual usage: <context>. [sentence]. <context>. With prompt: "How can this be worded better, while keeping the content and style the same".

A.3 LLMs for Coding

Claude Opus 4.8 by Anthropic has been specifically helpful in coding for this paper. All code that went into the final repository was manually checked, understood and

Table 10: SHARP zero-shot per-class accuracy by position (per 2 s window, streams fused).

Activity	Middle	Edge	Outside
squat	0.70	0.00	0.04
jumping_jack	0.89	0.00	0.86
jump	0.93	0.93	0.64
boxing	0.02	0.00	0.23
walk	0.98	0.61	0.96
stand	1.00	1.00	1.00
run	0.84	0.98	0.98
stir_a_pot	0.05	0.00	0.20
raising_left_foot	0.52	0.00	0.32
raising_left_arm	0.59	0.00	0.34

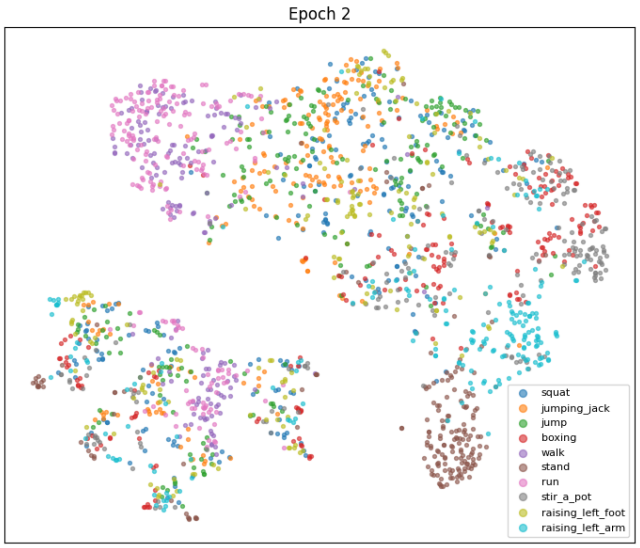

Figure 8: t-SNE [20] projection of the learned 256-d CSI embeddings, coloured by activity class, after 2 epochs of contrastive training.

made readable. Before any experiments were run, the validity of said code was checked.

All graphs and tables were also generated with code from LLMs. These are only visually checked.

A.4 Acknowledgements

Rune van Huffel for sharing their SHARP implementation code, which we have adapted to test our method against.

The WiFi Sensing team at TU Delft, for providing the dataset and lab used to train this model.

References

[1] Borna Barahimi, Hina Tabassum, Mohammad Omer, and Omer Waqar. Context-Aware Predictive Coding: A Representation Learning Framework for WiFi Sensing. *IEEE Open Journal of the Communications Society*, 5, 2024.

[2] Mohammud J. Bocus, Hok-Shing Lau, Ryan McConville, Robert J. Piechocki, and Raul Santos-Rodriguez. Self-Supervised WiFi-Based Activity Recognition. In *2022 IEEE Globecom Workshops (GC Wkshps)*, pages 552–557. IEEE, 2022.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[4] Yang Chen and Jingcai Guo. Towards Robust and Realistic Human Pose Estimation via WiFi Signals. *arXiv preprint arXiv:2501.09411*, 2025.

[5] Xue Ding, Ting Jiang, Yi Zhong, Yan Huang, and Zhiwei Li. Wi-Fi-Based Location-Independent Human Activity Recognition via Meta Learning. *Sensors*, 21(8):2654, 2021.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[7] Yang He, Yingying Gao, et al. Can WiFi Estimate Person Pose? *arXiv preprint arXiv:1904.00277*, 2019.

[8] Songming Jia, Yan Lu, Bin Liu, Xiang Zhang, Peng Zhao, Xinmeng Tang, Yelin Wei, Jinyang Huang, Huan Yan, and Zhi Liu. Breaking Coordinate Overfitting: Geometry-Aware WiFi Sensing for Cross-Layout 3D Pose Estimation. *arXiv preprint arXiv:2601.12252*, 2026.

[9] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, Wenyao Xu, and Lu Su. Towards Environment Independent Device Free Human Activity Recognition. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18)*, pages 289–304. ACM, 2018.

[10] 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 (MobiCom '20)*, pages 1–14. ACM, 2020.

[11] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[12] Haochen Li, Xin Zhou, Luu Anh Tuan, and Chunyan Miao. Rethinking Negative Pairs in Code Search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[13] Yan Liu, Anlan Yu, Leye Wang, Bin Guo, Yang Li, Enze Yi, and Daqing Zhang. UniFi: A Unified Framework for Generalizable Gesture Recognition with Wi-Fi Signals Using Consistency-Guided Multi-View Networks. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 7(4), 2024.

[14] Irineu Loturco, Fábio Y. Nakamura, Guilherme G. Artioli, Ronaldo Kobal, Katia Kitamura, Cesar C. Cal Abad, Ivan F. Cruz, Felipe Romano, Lucas A. Pereira, and Emerson Franchini. Strength and Power Qualities Are Highly Associated with Punching Impact in Elite Amateur Boxers. *Journal of Strength and Conditioning Research*, 30(1):109–116, 2016.

[15] Yong Lu, Shaohe Lv, and Xiaodong Wang. Towards Location Independent Gesture Recognition with Commodity WiFi Devices. *Electronics*, 8(10):1069, 2019.

[16] Francesca Meneghello, Domenico Garlisi, Nicolò Dal Fabbro, Ilenia Tinnirello, and Michele Rossi. SHARP: Environment and Person Independent Activity Recognition with Commodity IEEE 802.11 Access Points. *IEEE Transactions on Mobile Computing*, 22(10):6160–6175, 2023.

[17] Fabian Portner, Francesco Gringoli, Matthias Hollick, and Arash Asadi. Same signal, different story: Demystifying receiver effects in Wi-Fi channel state information. *IEEE Internet of Things Journal*, 2026. Early Access.

[18] 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 (IMWUT)*, 6(2):1–25, 2022.

[19] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*, 2018.

[20] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[22] 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, 2024.

[23] Fei Wang, Sanping Zhou, Stanislav Panev, Jinsong Han, and Dong Huang. Person-in-WiFi: Fine-Grained Person Perception Using WiFi. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5451–5460. IEEE, 2019.

[24] Ke Xu, Jiangtao Wang, Hongyuan Zhu, and Dingchang Zheng. ARC-Fi: Exploiting Antenna Spatial Diversity for Label-Efficient Domain Generalization in Wi-Fi Sensing. *arXiv preprint arXiv:2310.06328*, 2023.

[25] Jianfei Yang, Xinyan Chen, Han Zou, Dazhuo Wang, and Lihua Xie. AutoFi: Towards Automatic WiFi Human Sensing via Geometric Self-Supervised Learning. *arXiv preprint arXiv:2205.01629*, 2022.

[26] Dongsheng Yuan, Xie Zhang, Weiying Hou, Sheng Lyu, Yuemin Yu, Luca Jiang-Tao Yu, Chengxiao Li, and Chenshu Wu. OctoNet: A Large-Scale Multi-Modal Dataset for Human Activity Understanding Grounded in Motion-Captured 3D Pose Labels. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*. The University of Hong Kong, 2025. OpenReview: z3TftXOizf; `https://github.com/aiot-lab/OctoNet`.

[27] Xie Zhang, C. Tang, Kai Yin, and Qiang Ni. WiFi-Based Cross-Domain Gesture Recognition via Modified Prototypical Networks. *IEEE Internet of Things Journal*, 9(11):8584–8596, 2022.

[28] Yi Zhang, Yue Zheng, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. Widar3.0: Zero-Effort Cross-Domain Gesture Recognition with Wi-Fi. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8671–8688, 2022.

[29] Yunjiao Zhou, Jianfei Yang, He Huang, and Lihua Xie. AdaPose: Towards Cross-Site Device-Free Human Pose Estimation with Commodity WiFi. *arXiv preprint arXiv:2309.16964*, 2023.