SIMPLICITY IS KEY: AN UNSUPERVISED PRETRAINING APPROACH FOR SPARSE RADIO CHANNELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised representation learning for wireless channel state information (CSI) reduces reliance on labeled data, thereby lowering annotation costs, and often improves performance on downstream tasks. However, state-of-the-art approaches take little or no account of domain-specific knowledge, forcing the model to learn well-known concepts solely from data. We introduce **Spa**rse pretrained **R**adio **Tran**sformer (SpaRTran), a hybrid method based on the concept of compressed sensing for wireless channels. In contrast to existing work, SpaRTran builds on a wireless channel model that constrains the optimization procedure to physically meaningful solutions and induces a strong inductive bias. Compared to the state of the art, SpaRTran cuts positioning error by up to 32% and increases top-1 codebook selection accuracy for beamforming by 26 percentage points. Our results indicate that modeling radio propagation sparsity as an unsupervised learning objective improves both network optimization and radio-localization performance.

1 INTRODUCTION

Deep learning for wireless communication and sensing networks is an active research area and achieves outstanding performance in tasks such as network optimization and localization (Zhang et al., 2019). However, due to limited data availability, these approaches often tend to overfit the training dataset, leading to poor generalization capabilities (Simeone, 2018). The reason is the unconstrained optimization in the large parameter space, which results in suboptimal solutions (Vapnik, 2013). Unsupervised representation learning has gained significant attention in domains such as natural language processing (Devlin et al., 2019; Radford et al., 2018) and computer vision (Grill et al., 2020; Caron et al., 2021; He et al., 2020; Chen et al., 2020), often requiring fewer labeled samples for finetuning by making use of large amounts of unlabeled data in advance. Intuitively, unsupervised training acts as a regularizer that biases the network parameters towards regions of the parameter space whose local minima exhibit better generalization performance (Erhan et al., 2010). Consequently, recent works successfully adapted the paradigm to greatly reduce labeling effort for wireless tasks without degrading performance (Ott et al., 2024; Salihu et al., 2024; Alikhani et al., 2024; Guler et al., 2025).

While existing unsupervised methods for wireless applications have proven effective, they largely disregard well-known physical models of the wireless channel (Ott et al., 2024; Alikhani et al., 2024) or incorporate it implicitly via data augmentations (Salihu et al., 2024; Guler et al., 2025). Wireless systems are well suited for hybrid model- and data-driven approaches as there exists extensive domain-specific knowledge in the form of analytical approximations, yet the wireless channel includes complex stochastic processes which are infeasible to capture with mathematical models (O'Shea et al., 2019). Incorporating physical models in deep learning procedures improves generalization capabilities and reduces the dependency on large datasets (Shlezinger & Eldar, 2023).

Our method, Sparse pretrained Radio Transformer (SpaRTran), is a physics-inspired representation learning approach for CSI measurements. We demonstrate that explicitly incorporating a sparse wireless channel model and using sparse recovery as a pretext task—describing signals with as few active components as possible while preserving relevant information—improves downstream performance. This introduces an inductive bias guiding pretraining toward simpler, more general solutions in parameter space, aligning with Occam's razor and the principle of parsimony (Chater & Vitányi, 2003), where "simple" characterizes the output structure. We draw inspiration from the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

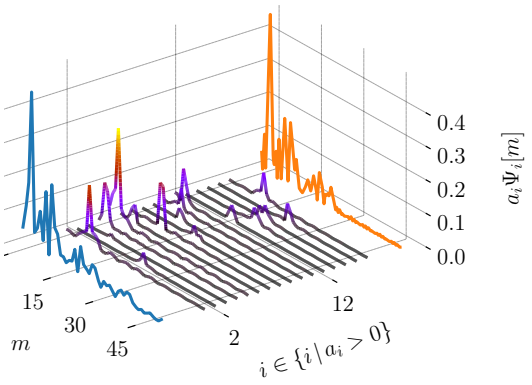


Figure 1: Example of learned sparse CSI decomposition, our unsupervised objective. The input signal (blue) is decomposed into a linear combination of basis functions (black) and subsequently recovered (orange).

compressed sensing (CS) framework for the architecture design and the pretraining objective. The central premise of CS is that sufficiently sparse representations reduce ambiguity, while in contrast, non-sparse representations typically contain numerous insignificant components, complicating both analysis and signal recovery (Donoho & Huo, 2001; Candes et al., 2006). To encode this inductive bias we employ a gating mechanism that selects active latent components, similar to sparse autoencoders (SAEs) (Rajamanoharan et al., 2024; Cunningham et al., 2023). We reinterpret the sparse autoencoder methodology for representation learning in wireless channels: rather than using sparsity to promote feature interpretability, we use it to bias the encoder toward simpler and more generalizable representations through sparse decomposition (see Fig. 1).

In contrast to the state-of-the-art, SpaRTran uses single-links between antenna pairs instead of accumulating all available links of a system into a single datapoint. While sacrificing the ability to learn spatial characteristics between antennas in an unsupervised manner, this design choice brings an important advantage: The learned representations are system agnostic, i.e., they do not depend on the number of antennas or system topologies. This is an important step towards larger, more general wireless basis models that act on many different systems.

2 Related Work

SpaRTran is an unsupervised representation learning method for pretraining on wireless signals that integrates techniques from compressed sensing and dictionary learning.

Unsupervised representation learning for CSI. Supervised deep learning has advanced the state-of-the-art in terms of accuracy in tasks such as wireless positioning (Salihu et al., 2022; Liu et al., 2022; Zhang et al., 2023) and beam-management (Ma et al., 2023). Recently, unsupervised learning and self-supervised learning (SSL) have attracted significant attention in this context aiming to reduce the dependence on costly labels. The idea is to leverage cost-effective unlabeled channel measurements to pretrain a task-agnostic basis model also known as a foundation model. Existing works transfer the pretraining objectives that have been established in domains such as computer vision or natural language processing to the wireless domain. Here, contrastive (Salihu et al., 2024) and predictive (Alikhani et al., 2024; Catak et al., 2025; Ott et al., 2024; Yang et al., 2025) methods have been studied as well as their combination (Pan et al., 2025; Guler et al., 2025). These approaches can be considered purely data-driven, learning the underlying structure of the wireless channel directly from measurements. To the best of our knowledge SpaRTran is the first unsupervised representation learning approach that incorporates a wireless channel model in both the architectural design and the pretraining objective.

Compressed sensing represents signals by a high-dimensional sparse vector in an overcomplete basis, assuming they arise from a few latent factors. CS is widely used in wireless systems for source separation (Donoho, 2006; Candes et al., 2006), direction-of-arrival (DOA) estimation (Yang et al., 2018), and channel estimation (Berger et al., 2010). Classical recovery methods include convex

relaxation (Chen et al., 2001; Tibshirani, 1996), greedy pursuit (Tropp & Gilbert, 2007), and sparse Bayesian learning (Malioutov et al., 2005; Stoica et al., 2011). Recent deep-learning approaches improve reconstruction fidelity while reducing complexity (Machidon & Pejović, 2023). Separately, SAEs (Cunningham et al., 2023; Bricken et al., 2023) enforce sparsity in high-dimensional latent spaces; Rajamanoharan et al. (2024) further decouple component selection from coefficient estimation to avoid shrinking of coefficients. We reinterpret the compressive optimization objective as a pretext task for unsupervised generative modeling, drawing on concepts from convex relaxation and SAEs to incorporate domain-specific knowledge into both architecture and optimization.

Dictionary learning algorithms identify atomic features that sparsely represent underlying data, i.e., the dictionary is learned empirically from the signals themselves. This enables generalization across signal types and often leads to increased sparsity (Elad, 2010). A prominent example is the K-SVD algorithm (Aharon et al., 2006) which iteratively updates the dictionary atoms. SpaRTran can jointly learn the dictionary, increasing flexibility across waveforms and spatio-temporal patterns in cases where the theoretical dictionary may not approximate the signals sufficiently.

3 Sparse Recovery as Self-Supervised Objective

The core of SpaRTran is a novel self-supervised objective using a wireless channel model that is grounded in the well-known Least Absolute Shrinkage and Selection Operator (Tibshirani, 1996). This model uses sparse signal decompositions to characterize the physical propagation process, specifically signal scattering caused by environmental interactions that result in multiple propagation paths arriving at the receiver. The key insight is that the number of dominant scatterers (buildings, reflectors, clusters) is limited, yielding only a few significant interactions. In the following we describe the mathematical formulation of the model and in Section 4 we outline the realization as a pretraining objective, including architecture and training process.

The received signal $y(t)$ can be defined as $y(t) = h(t) * s(t) + w(t)$, where $s(t)$ is the transmitted signal, $h(t)$ is the channel, $w(t)$ is additive white Gaussian noise, and $*$ is the convolution operator. The channel impulse response (CIR) $h(t)$ characterizes the radio transmission channel and can be modeled as

$$h(t) = \sum_{k=0}^{K-1} a_k e^{-i\varphi_k} \delta(t - \tau_k), \tag{1}$$

where τ_k is the signal transmission delay, a_k the magnitude and φ_k the phase of the k-th propagation path of the transmitted signal. δ denotes the Dirac delta function and i denotes the imaginary unit. Eq. 1 describes the superposition of several signals, originating from K far-field sources. In practice, K is assumed to be unknown. The bandwidth-limited discrete channel measurement is modeled as

$$h[m] = \sum_{k=0}^{K-1} a_k \text{sinc}[m - \tau_k W] + w, \tag{2}$$

where W is the bandwidth of the system, a_k are the complex-valued path coefficients, and $m \in \{1, \cdots, M\}$. From this, we derive the sparse channel representation. Assuming a set of N potential signals $\boldsymbol{\psi}_i \in \mathbb{R}^M$ that form a basis, of which only $K \ll N$ effectively contribute to the received signal, we can rewrite Eq. 2 as

$$\boldsymbol{h} = \sum_{i=0}^{N-1} a_i \boldsymbol{\psi}_i + \boldsymbol{w}, \tag{3}$$

where $|a_i| > 0$ if the i-th signal is an active signal component, and $|a_i| = 0$ otherwise. Note that we have replaced the sinc-function with a more generic notation $\boldsymbol{\psi}_i$. Defining the dictionary $\boldsymbol{\Psi} := [\boldsymbol{\psi}_0, \cdots, \boldsymbol{\psi}_{N-1}]$ allows (3) to be expressed more concisely in matrix notation as

$$\boldsymbol{h} = \boldsymbol{\Psi}\boldsymbol{a} + \boldsymbol{w}, \tag{4}$$

where $\boldsymbol{a} = [a_0, \cdots, a_{N-1}]^T$ is the sparse coefficient vector, and $\boldsymbol{\Psi}$ is an $M \times N$ dictionary matrix. Eq. 4 describes an underdetermined system of equations. As there is no unique solution, recovering the sparse channel requires solving the following optimization problem:

$$\min \|\boldsymbol{a}\|_0, \quad \text{s.t. } \|\boldsymbol{\Psi}\boldsymbol{a} - \boldsymbol{h}\|_2 \leq \epsilon, \tag{5}$$

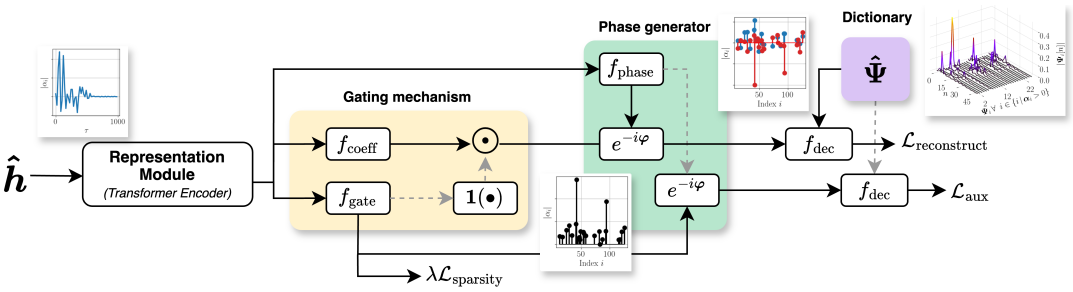


Figure 2: Overview of our unsupervised pretraining method - SpaRTran.

where ϵ denotes the allowed reconstruction error due to noise. Eqs. (4) and (5) together describe the radio channel within the framework of compressed sensing (Donoho, 2006; Candes et al., 2006). Our goal is to find representations that simplify solving (5) and therefore focus on general pattern in the data, matching the sparsity constraints. Consequently, the role of compressed sensing in our framework is not to solve a traditional sparse recovery problem, but rather to provide a principled pretext task for self-supervised representation learning.

We provide theoretical justification for this idea in Section A.1. Our core assumption is that signals can be represented in a reproducing kernel Hilbert space (RKHS) and decomposed into a signal component S and a noise component $\bar{S}$—a prerequisite for any reconstruction objective. The main challenge in moving from theory to practice is that the operator O required to "rotate" h into the improved coordinate frame may be infinite-dimensional, since an RKHS is generally infinite-dimensional. We address this by approximating O with a transformer, which is justified by the universal approximation theorem.

4 SpaRTran Training Pipeline

In general, we consider a set of unlabeled channel measurements $\mathcal{M}$. Our objective is to learn a channel representation $\boldsymbol{z}$ that encodes the environmental characteristics of the radio signal. To this end, we introduce a strong bias into the training process through both model architecture and loss function design. Our approach employs a data-driven encoder (the representation module) that generates the representation $\boldsymbol{z}$ and maps $\boldsymbol{z}$ to a sparse representation $\hat{\boldsymbol{a}}$. This resembles the encoder part of a sparse autoencoder. However, unlike a conventional sparse autoencoder we use a model-driven decoder, based on the sparse channel formulation in Eq. (4), reconstructing the input signal $\hat{\boldsymbol{h}} \sim \mathcal{M}$ from $\hat{\boldsymbol{a}}$. This modeling choice is key for using the sparse signal recovery objective as a pretext task.

In line with existing works (Guler et al., 2025; Salihu et al., 2024; Alikhani et al., 2024; Ott et al., 2024), we use a transformer (TF) architecture (Vaswani et al., 2017) as the backbone of the representation module. We employ a lightweight encoder-only TF with a depth of one and internal latent dimension of $N_{latent} = 512$, using 8 attention heads for the multihead attention mechanism. We interpret the complex values at the m-th timestep as a three-dimensional vector consisting of the real and imaginary values as well as the magnitude $\hat{\boldsymbol{h}}_m = [\mathcal{R}(\hat{h}_m), \mathcal{I}(\hat{h}_m), |\hat{h}_m|]^T$. Similar to the state-of-the-art (Alikhani et al., 2024; Guler et al., 2025), we use non-overlapping patches of $\hat{\boldsymbol{h}}$ to construct our input embeddings $\boldsymbol{e}$. We combine windows of three time steps of the CIR into an input token $\boldsymbol{e}_m = [\hat{\boldsymbol{h}}_{3m-2}, \hat{\boldsymbol{h}}_{3m-1}, \hat{\boldsymbol{h}}_{3m}]^T$, ultimately resulting in 9 dimensions per token. We project each input token into the latent space of dimension N_{latent} via a learned linear transformation to match the internal dimensionality of the TF-encoder. An ablation study comparing different TF configurations can be found in Appendix A.5.

4.1 Sparse Reconstruction Head

The sparse reconstruction head aims to find a sparse signal decomposition based on the representation $\boldsymbol{z}$. It uses a gating mechanism (inspired by (Rajamanoharan et al., 2024)) and a phase generator. The former promotes the reconstruction to be sparse while the latter converts the real-valued output of the

neural network to the complex-valued coefficients $\hat{\boldsymbol{a}}$. $\hat{\boldsymbol{a}}$ represents the reconstructed signal in terms of an overcomplete dictionary $\boldsymbol{\Psi}$, see Eq. 4. Fig. 2 shows the gating mechanism (yellow), the phase generator (green), and the dictionary (purple). Note that the learned representation $\boldsymbol{z}$ is itself not sparse.

We now discuss the gating mechanism in more detail. Approximating the l_0-norm with the l_1-norm tends to lead to non-optimal reconstruction. This is due to the fact that the sparsity penalty, i.e., the l_1-norm, can be reduced at the cost of reconstruction performance (Wright & Sharkey, 2024). Hence, our strategy for the estimation of $\hat{\boldsymbol{x}}$ follows the work of Rajamanoharan et al. (2024). The idea is to separately handle the selection of active atoms from the dictionary ($\boldsymbol{f}_{\text{gate}}$) and the estimation of the magnitude of the coefficients ($\boldsymbol{f}_{\text{coeff}}$). The encoder output is defined by

$$\hat{\boldsymbol{x}} = \boldsymbol{f}_{\text{coeff}}(\boldsymbol{z}) \odot \mathbf{1}(\underbrace{\boldsymbol{f}_{\text{gate}}(\boldsymbol{z})}_{\boldsymbol{\rho}_{\text{gate}}}), \tag{6}$$

where $\mathbf{1}$ denotes the Heaviside step function, $\odot$ the Hadamard product and $\boldsymbol{\rho}_{\text{gate}}$ is the output of the gating stage before the binarization step. Fig. 2 shows the gating mechanism (yellow). Due to the binarization of the gating values, no gradient flows through this path of the network; see grey arrows in the yellow box of Fig. 2. Thus, an auxiliary loss promotes the detection of active atoms in $\boldsymbol{f}_{\text{gate}}$. The auxiliary loss measures reconstruction fidelity, but instead of $\hat{\boldsymbol{x}}$, it uses $\boldsymbol{\rho}_{\text{gate}}$ to reconstruct the signal. The dictionary should not be updated by the auxiliary reconstruction task. Hence, we prohibit the flow of the gradient accordingly (see grey dashed line in Fig. 2).

We now outline our extensions to the original method. Rajamanoharan et al. (2024) restrict the encoder´s output $\hat{\boldsymbol{x}}$ to real positive numbers. However, this assumption does not hold in our case, as our goal is to estimate complex-valued path coefficients $\hat{\boldsymbol{a}}$. To address this, we interpret the outputs of $\boldsymbol{f}_{\text{coeff}}$ and $\boldsymbol{f}_{\text{gate}}$ as the magnitudes of the complex coefficients. This formulation allows us to suppress negative values via the gating mechanism without violating the underlying physical channel model. In addition, we introduce a third function $\boldsymbol{f}_{\text{phase}}$ that generates the phases of the path coefficients. The final coefficients are then constructed as $\hat{\boldsymbol{a}} = \hat{\boldsymbol{x}} e^{-i\boldsymbol{f}_{\text{phase}}(\boldsymbol{z})}$ and $\boldsymbol{\rho}'_{\text{gate}} = \boldsymbol{\rho}_{\text{gate}} e^{-i\boldsymbol{f}_{\text{phase}}(\boldsymbol{z})}$, where i denotes the imaginary unit. The output of $\boldsymbol{f}_{\text{phase}}$ is constrained to the interval $\pm\pi$ using a scaled tanh activation function. This leads to the following loss function:

$$\mathcal{L} := \underbrace{\|\tilde{\boldsymbol{h}} - \boldsymbol{f}_{\text{dec}}(\hat{\boldsymbol{a}}, \hat{\boldsymbol{\Psi}})\|_2^2}_{\text{reconstruction loss}} + \underbrace{\lambda \|\mathbf{1}(\boldsymbol{\rho}_{\text{gate}})\|_1}_{\text{sparsity penalty}} + \underbrace{\|\tilde{\boldsymbol{h}} - \boldsymbol{f}_{\text{dec}}(\boldsymbol{\rho}'_{\text{gate}}, \hat{\boldsymbol{\Psi}}_{\text{frozen}})\|_2^2}_{\text{auxiliary loss}}, \tag{7}$$

with $f_{dec}(\hat{\boldsymbol{a}}, \hat{\boldsymbol{\Psi}}) = \hat{\boldsymbol{\Psi}}\hat{\boldsymbol{a}}$ (see Fig. 2, purple box). Note that we realize the functions f_{coeff},f_{gate} and f_{phase} as lightweight MLPs using a hidden layer with dimensionality of 256. To enforce non-negativity, Rajamanoharan et al. (2024) employ ReLU functions at the outputs of $\boldsymbol{f}_{\text{gate}}$ and $\boldsymbol{f}_{\text{coeff}}$. We observed that this can lead to a situation where certain dictionary atoms are never activated, i.e., their associated coefficients remain zero, resulting in no gradient updates, a phenomenon akin to the dying ReLU problem. To mitigate this, we use leaky ReLU activations (slope 0.01) instead, ensuring that gradients can still propagate even for inactive units. This introduces the possibility of negative coefficients in the auxiliary path (the reconstruction coefficients are always ≥ 0 due to the binary gate) which corresponds to a 180-degree phase shift.

4.2 Dictionary Composition

There are two ways to obtain the dictionary. First, a theoretically derived fixed dictionary $\boldsymbol{\Psi} \in \mathbb{R}^{M \times N}$ corresponding to the model we outlined in Sec. 3. Second, a learned dictionary $\hat{\boldsymbol{\Psi}}$ that consists of jointly optimized parameters. We first detail the theoretical approach.

Fixed Dictionary. If the bandwidth W of the considered signals is known, it is reasonable to construct $\boldsymbol{\Psi}$ from the theoretical model. In this case, the columns of $\boldsymbol{\Psi}$ contain shifted sinc-functions such that:

$$\Psi_{m,i} = \text{sinc}\left[m - i\frac{\tau_{max} W}{N}\right], \tag{8}$$

where τ_{max} is the last expected arrival of a significant signal component, which can be trivially determined by the number of time-steps M of $\tilde{\boldsymbol{h}}$ and the sampling frequency $1/W$.

Learned Dictionary. If crucial system parameters, such as the signal bandwidth or the receiver's sampling frequency, are unknown, the construction of a dictionary that conforms to the theory is unfeasible. This may occur when using diverse crowd-sourced signals to train a large foundation model. In this case, SpaRTran can treat the dictionary as learnable parameters forming $\hat{\mathbf{\Psi}}$. By normalizing the atomic entries to unit norm, they only determine the direction of the contribution, while $\hat{\boldsymbol{a}}$ provides the amplitude and phase of the complex-valued signal component. However, jointly optimizing an unrestricted dictionary renders problem (5) NP-hard (Tropp, 2004) and the model-aided regularization effect is decreased (see also Sec. 5.4).

4.3 FINETUNING

While SpaRTran learns representations for individual transmitter–receiver links, most tasks exploit correlations across multiple channels. We consider N_r links recorded by an agent traversing the environment. We aggregate these into the channel state $\boldsymbol{H} = [\hat{\boldsymbol{h}}_1 \cdots \hat{\boldsymbol{h}}_{N_r}]$, referred to as the CSI. For finetuning, we compute a representation for each available link using N_r forward passes through our pretrained transformer backbone. We subsequently concatenate them to form a complete representation of the CSI. Our finetuning head is a 1D ResNet-style architecture starting with a 1×1 convolutional block that maps the output dimension N_{latent} of the learned representation to 16 input channels. This is followed by a sequence of $N_{blocks,head}$ residual blocks. The i-th residual block outputs $16 * 2^{i-1}$ channels and uses a kernel size of $5 + 2(i-1)$, so both the channel width and receptive field grow with depth. A global average pooling layer, followed by a fully connected layer, maps the final feature representation to the estimate. We use $N_{blocks,head} = 4$ in our evaluations. An ablation study comparing different head depths can be found in the appendix (Section A.5).

5 EVALUATION

We evaluate localization via CSI fingerprinting (Secs. 5.1, 5.2) and codebook selection for beamforming (Sec. 5.3) — both leveraging full CSI complexity. In Sec. 5.4, we test our approach under several ablations and study the effect of varying values for sparsity penalty λ and dictionary size N. A common way to assess representation quality is linear probing, i.e., training a linear layer on top of a frozen representation backbone. However, since our method does not generate a unified representation across all available links (see Section 4.3), this approach is not applicable. Instead, we fine-tune the pretrained backbones together with the downstream task heads.

For our experiments, we compare four SSL baselines (**SWiT** (Salihu et al., 2024), **Masking** (Ott et al., 2024), **LWM** (Alikhani et al., 2024) and **C-WiMAE** (Guler et al., 2025)) and a supervised method (**WiT** (Salihu et al., 2022)), each using a TF backbone. We evaluate all methods on three publicly available datasets: (i) **KUL**, a small controlled environment (Bast et al., 2020), (ii) **FH-IIS**, a larger and more complex environment (Stahlke et al., 2024), and (iii) **DeepMIMO** (Alkhateeb, 2019) for large-scale, diverse urban scenarios used in the codebook selection task. We use the original baseline implementations, as we found that modifying the reported architectures can degrade performance. We attribute this to the fact that the architecture and training objective are usually co-designed rather than independent components. Consequently, altering their architectures—for example, to match parameter counts or head configurations—could introduce confounding factors and place the baselines at an artificial disadvantage relative to our approach. More detailed information on the baselines, training procedure, and the datasets can be found in Appendix A.4.

Downstream Tasks. We use two downstream tasks to evaluate the performance of SpaRTran:
(1) *CSI fingerprinting* maps positions to CSI measurements by exploiting their high spatial correlation. Due to multipath wave propagation, the CSI is typically highly characteristic per position, assuming a wide-sense static environment (i.e., negligible changes in the radio environment between training and inference) (Niitsoo et al., 2019; Stahlke et al., 2022). We use the mean absolute error (MAE) and 90th percentile of the cumulative error (CE90) to determine the positioning accuracy.
(2) *Beamforming* adapts phases/amplitudes across large phased array antennas to perform directed signal transmissions, mitigating high path losses and interference. Codebook selection for beamforming aims to select the optimal beam from a predefined codebook directly from channel measurements. This reduces the channel estimation overhead (Giordani et al., 2019). To assess the performance of this classification task, we use the top-1 accuracy.

5.1 LOCALIZATION IN SMALL DATA REGIME

Table 1: Fingerprinting (FP) performance for SpaRTran and the baseline models for different amounts of labeled training data evaluated on the FH-IIS dataset (MAE / CE90 in meters).

Method	1%	2%	5%	10%	25%	50%	100%
Masking	0.73 / 1.27	0.61 / 1.08	0.50 / 0.89	0.48 / 0.84	0.42 / 0.75	0.44 / 0.76	0.40 / 0.70
LWM	1.17 / 2.10	0.91 / 1.63	0.72 / 1.27	0.63 / 1.11	0.56 / 0.99	0.55 / 0.97	0.58 / 0.99
SWiT	2.33 / 4.24	2.09 / 3.81	1.90 / 3.46	1.79 / 3.29	1.76 / 3.22	1.65 / 3.01	1.51 / 2.75
C-WiMAE	**0.52 / 0.93**	**0.48 / 0.86**	**0.45 / 0.80**	0.43 / 0.76	0.39 / 0.70	0.42 / 0.74	0.39 / 0.69
SpaRTran	0.77 / 1.40	0.58 / 1.04	0.50 / 0.91	**0.41/ 0.74**	**0.34 / 0.62**	**0.30 / 0.56**	**0.30 / 0.55**
WiT (Sup.)	1.27 / 2.32	1.05 / 1.88	0.88 / 1.59	0.89 / 1.61	0.66 / 1.21	0.56 / 1.03	0.49 / 0.90

To evaluate data efficiency, we train the methods on the FH-IIS dataset while using varying amounts of labeled data for finetuning, expressed as a percentage of the finetuning split (see Table 1). SpaRTran achieves the highest accuracy when at least 10% of the labeled samples are used, improving MAE by up to 28% and CE90 by up to 24% over the best baseline Contra-WiMAE, thereby demonstrating the effectiveness of our approach. For finetuning with $\leq 5\%$, however, SpaRTran performs worse than C-WiMAE. This is not surprising, as SpaRTran is pretrained on single links between antennas rather than the full CSI, preventing it from learning inter-channel correlations in an unsupervised manner. This reflects a deliberate design trade-off: by restricting pretraining to single links, we significantly improve generalizability across diverse scenarios (see Section 5.2), at the cost of reduced performance in extremely low-label settings for tasks that depend on cross-link information. Moreover, the compressive pretraining objective trades fine-grained, environment-specific signal patterns for a more general representation (see Sec. A.1), reducing accuracy compared to less general methods that benefit from the fixed environment and system setup.

5.2 LOCALIZATION UNDER DOMAIN SHIFT

Table 2: FP performance across different system setups finetuned on $5\,000$ samples of the KUL dataset (MAE / CE90 in meter).

Method	Pretrain-Set	DIS-LoS	ULA-LoS	URA-LoS	URA-nLoS
Masking:	DIS-LoS	0.093 / 0.158	0.065 / 0.118	0.071 / 0.129	1.176 / 1.626
	ULA-LoS	0.081 / 0.139	0.067 / 0.116	0.071 / 0.127	1.094 / 1.615
	URA-LoS	0.087 / 0.156	0.068 / 0.118	0.073 / 0.131	1.176 / 1.671
	URA-nLoS	0.072 / 0.128	0.067 / 0.120	0.073 / 0.139	1.153 / 1.627
SWiT:	DIS-LoS	0.071 / 0.139	0.069 / 0.138	0.057 / 0.119	0.154 / 0.324
	ULA-LoS	0.076 / 0.146	0.068 / 0.136	0.058 / 0.119	0.140 / 0.303
	URA-LoS	0.070 / 0.138	0.068 / 0.136	0.057 / 0.119	0.156 / 0.327
	URA-nLoS	0.077 / 0.148	0.068 / 0.137	0.059 / 0.119	0.152 / 0.326
LWM:	DIS-LoS	0.083 / 0.144	0.084 / 0.151	0.075 / 0.137	0.244 / 0.454
	ULA-LoS	0.085 / 0.151	0.082 / 0.146	0.079 / 0.147	0.256 / 0.500
	URA-LoS	0.075 / 0.132	0.082 / 0.147	0.065 / 0.122	0.204 / 0.372
	URA-nLoS	0.094 / 0.164	0.092 / 0.163	0.083 / 0.151	0.224 / 0.418
C-WiMAE:	DIS-LoS	**0.053 / 0.093**	0.092 / 0.167	0.125 / 0.259	0.444 / 1.453
	ULA-LoS	0.099 / 0.174	0.057 / **0.097**	0.089 / 0.166	0.283 / 0.755
	URA-LoS	0.083 / 0.152	0.076 / 0.133	0.066 / 0.112	**0.087 / 0.163**
	URA-nLoS	0.110 / 0.206	0.102 / 0.191	0.090 / 0.162	**0.074 / 0.132**
SpaRTran:	DIS-LoS	0.076 / 0.127	**0.055 / 0.101**	**0.052 / 0.090**	**0.096 / 0.191**
	ULA-LoS	**0.070 / 0.123**	**0.055** / 0.102	**0.053 / 0.105**	**0.111 / 0.226**
	URA-LoS	**0.074 / 0.126**	**0.057 / 0.104**	**0.047 / 0.090**	0.099 / 0.199
	URA-nLoS	**0.054 / 0.097**	**0.048 / 0.089**	**0.045 / 0.089**	0.109 / 0.212
WiT (Sup.):		0.074 / 0.135	0.071 / 0.135	0.059 / 0.108	0.196 / 0.387

Table 2 presents the results of wireless localization trained on pairs of scenarios of the KUL dataset, one used for pretraining and the other for finetuning. Overall, SpaRTran achieves the best performance

across the considered system setups, with an average MAE of $0.069\,\mathrm{m}$ and CE90 of $0.129\,\mathrm{m}$. Compared to the supervised WiT baseline, this corresponds to an overall improvement of approximately 31% in MAE and 32% in CE90. In particular, SpaRTran shows strong robustness across different pretraining configurations and achieves the largest improvements for the challenging URA-nLoS setup. While C-WiMAE can achieve very competitive performance when the pretraining and downstream configurations are well aligned, its performance degrades considerably under configuration mismatch. In contrast, SpaRTran maintains consistently low localization errors across the different pretraining sets, indicating improved transferability across antenna and propagation configurations.

5.3 BEAMFORMING

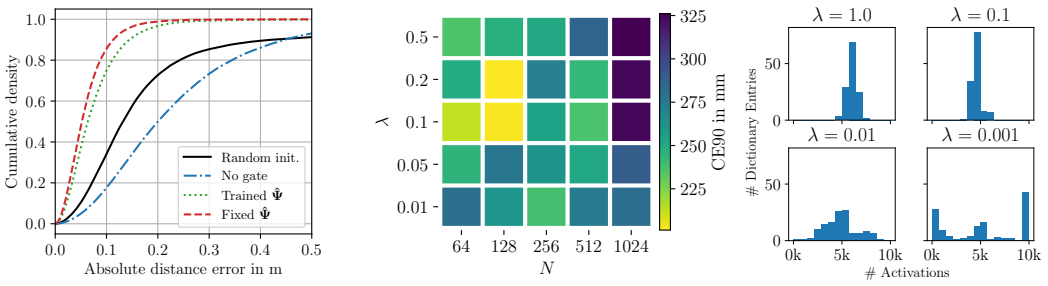
Table 3: Top-1 accuracy (%) for codebook selection in the beamforming task. Finetuning was performed across task complexities defined by codebook sizes (16, 32, 64, 128) and varying proportions of labeled training data (1%, 2%, 5%, 10%, 25%, 50%, 100%). Results for Contra-WiMAE are taken from Guler et al. (2025), all others are obtained by reproducing the results (see Section A.4.1).

Method	Codebook Size	1%	2%	5%	10%	25%	50%	100%
Masking	16	50.7	59.6	68.5	72.4	76.8	78.4	80.6
	32	31.6	40.6	53.5	67.6	74.9	78.1	81.2
	64	15.9	23.4	33.2	43.6	59.1	64.8	65.0
	128	9.4	10.1	13.0	20.8	33.0	40.4	42.1
SWiT	16	**56.7**	61.3	73.5	78.6	81.3	83.7	84.3
	32	40.0	50.6	71.1	76.4	77.2	80.0	81.5
	64	30.1	36.7	51.6	60.1	69.5	70.2	68.0
	128	18.6	22.6	25.3	36.5	44.6	47.6	41.2
LWM	16	40.3	51.0	68.7	72.5	77.4	79.1	81.2
	32	21.0	28.4	51.9	67.3	76.3	82.0	82.4
	64	6.8	12.5	38.7	61.3	72.2	67.8	69.4
	128	4.0	4.3	10.8	17.5	51.8	44.0	57.7
C-WiMAE	16	49.2	55.9	67.7	72.1	72.2	77.4	79.0
	32	35.9	41.4	50.6	63.5	68.4	68.3	75.3
	64	24.7	29.0	31.3	40.1	48.3	56.1	64.5
	128	14.0	16.2	18.5	23.7	29.0	35.4	39.3
SpaRTran	16	56.0	**71.6**	**82.8**	**84.6**	**89.4**	**90.0**	**92.8**
	32	38.7	**73.6**	**87.6**	**88.9**	**90.4**	**92.9**	**93.7**
	64	**32.3**	**54.2**	**74.4**	**84.2**	**86.6**	**89.5**	**89.0**
	128	**23.8**	**25.7**	**39.9**	**62.9**	**70.5**	**74.5**	**71.8**
WiT (Sup.)	16	56.4	67.3	74.3	80.1	82.3	84.3	85.8
	32	**44.3**	69.1	70.2	79.4	82.1	82.9	84.4
	64	31.3	49.0	52.3	60.4	73.3	75.5	77.6
	128	15.9	24.2	21.7	34.4	48.4	48.8	49.8

Table 3 shows the results for the beamforming downstream task, i.e., selecting the best beam ID in a predefined codebook to steer antenna gain in a specific direction to serve a mobile device. Again, SpaRTran outperforms the compared methods in most settings. Notably, in one of the most challenging settings—a codebook size of 128 with fine-tuning on only 10% of the labeled training data—SpaRTran increases top-1 accuracy by 26 percentage points, up to 62.9%, relative to SWiT. Again, with a very small amount of labeled data (1%), SpaRTran's advantage diminishes due to the pretraining on single transmission links rather than full CSI. It is noteworthy that in the general picture none of the self-supervised baselines significantly outperforms the purely supervised method WiT. As the test environments were unseen during pretraining, this could indicate a low degree of generalization, occurring when the model overly adapts to the pretraining distribution. The model-induced bias in SpaRTran appears to make it less susceptible to this effect, improving its generalization capabilities.

5.4 ABLATIONS

Figure 3a shows localization accuracy for the following ablations: no pretraining (randomly initialized backbone; black), no sparsity-inducing gating during pretraining resembling a conventional



(a) (b) (c)

Figure 3: (a) shows the cumulative density of the wireless localization error under ablations, (b) shows the localization accuracy depending on the dictionary size N and the sparsity coefficient λ, and (c) shows the distribution of the number of activations on 10000 data points.

autoencoder with a large dimensional latent (blue), and two variants of our pretraining procedure, one using a learned dictionary (green) and the other using a fixed dictionary based on the theoretical channel model (red). Both regular pretraining cases achieve a very high accuracy of CE90 ≤ 0.145. The learned dictionary incurs a minor CE90 degradation of $0.003\,\mathrm{m}$ versus the fixed dictionary, so it remains competitive and is suitable when system configurations are unknown. SpaRTran reduces CE90 by 66.5% relative to a randomly initialized backbone, demonstrating its effectiveness. Removing the sparsity-inducing gating worsens performance relative to the random initialization, underscoring the critical role of the sparsity assumption.

The parameter λ balances a tradeoff between improved generalization from stronger sparsity and avoiding a systematic underestimation of the nonzero magnitudes (shrinkage) when λ is too large (Wright & Sharkey, 2024). The dictionary size N governs a second tradeoff: smaller dictionaries yield less coherent basis functions, simplifying identification of contributing basis functions, while larger dictionaries improve reconstruction fidelity (Donoho & Huo, 2001).

Figure 3c shows the distribution of number of activations per atom of a learned dictionary with $N = 128$ dependent on λ. In general, higher values of λ (strong sparsity penalty) lead to a less spread out histogram, i.e., the atoms are activated with equal frequency, indicating an effective diversity of the learned dictionary. Note that when λ is very small ($\lambda = 0.001$), some atoms dominate the representation, being activated constantly while others are stalled, an effect akin to mode collapse.

6 Conclusion

We presented SpaRTran, a hybrid model- and data-driven unsupervised method for learning task-agnostic radio channel representations. Our unsupervised objective aims to find a signal transformation that preconditions a sparse signal decomposition, i.e., reduces data ambiguity. This induces a strong inductive bias towards simpler, more general solutions. We use a gated SAE that integrates a channel model inspired by compressed sensing, reflecting the sparse model in the architectural design. We presented results that show an improvement in generalization and overall performance over state-of-the-art approaches, achieving up to 32% reduction in positioning error and a 26-percentage-point increase in the top-1 accuracy on codebook selection for beamforming.

Unlike existing methods, SpaRTran operates on individual radio links rather than full CSI matrices which decouples the model from specific system configurations, taking a step towards large generic radio foundation models. Although this deliberate design choice prohibits the extraction of cross-link patterns during pretraining, reducing the performance at very low data regimes ($\leq 5\%$ of samples), it achieves the best performance from moderate data availability onward and demonstrates excellent generalization capabilities.

AI USE STATEMENT

In this work, we used generative AI tools for creating and editing software code/ implementing the method, assisting with translation, sourcing/searching for information, editing the research paper to improve readability, identifying relevant literature, supporting qualitative and thematic data analysis and designing or providing feedback on research methodology or experiments. We have not used generative AI tools for generating synthetic data sets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, cleaning and reformatting datasets and interpreting results. We have reviewed all AI-assisted work. We verified and tested LLM-generated code for correctness, checked generated text snippets by at least two authors and verified proposed literature. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide sufficient details to reproduce our results, including the model architecture (Section 4), training procedure (Sections 3 and 4), hyperparameters (Section A.5), datasets (Section A.4.2), baselines (Section A.4.1) and evaluation protocols (Section 5) in the main text and appendices. We also provide detailed theoretical considerations and proofs in the appendix (Sections A.1 and A.2). All experiments are conducted with fixed random seeds, and we report results over multiple independent runs for a selected experiment (Section A.6).

REFERENCES

M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, November 2006. ISSN 1941-0476. doi: 10.1109/TSP.2006.881199. URL https://ieeexplore.ieee.org/abstract/document/1710377. Conference Name: IEEE Transactions on Signal Processing.

Sadjad Alikhani, Gouranga Charan, and Ahmed Alkhateeb. Large Wireless Model (LWM): A Foundation Model for Wireless Channels, November 2024. URL http://arxiv.org/abs/2411.08872. arXiv:2411.08872 [cs].

A. Alkhateeb. DeepMIMO: A Generic Deep Learning Dataset for Millimeter Wave and Massive MIMO Applications. In *Proc. of Information theory and Applications Workshop (ITA)*, pp. 1–8, San Diego, CA, February 2019.

Sibren De Bast, Andrea P. Guevara, and Sofie Pollin. CSI-based Positioning in Massive MIMO systems using Convolutional Neural Networks. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1–5, May 2020. doi: 10.1109/VTC2020-Spring48590.2020.9129126. URL https://ieeexplore.ieee.org/abstract/document/9129126. ISSN: 2577-2465.

Christian R. Berger, Zhaohui Wang, Jianzhong Huang, and Shengli Zhou. Application of compressive sensing to sparse channel estimation. *IEEE Communications Magazine*, 48(11):164–174, November 2010. ISSN 1558-1896. doi: 10.1109/MCOM.2010.5621984. URL https://ieeexplore.ieee.org/document/5621984/?arnumber=5621984. Conference Name: IEEE Communications Magazine.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

E.J. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52

(2):489–509, February 2006. ISSN 1557-9654. doi: 10.1109/TIT.2005.862083. URL https://ieeexplore.ieee.org/document/1580791. Conference Name: IEEE Transactions on Information Theory.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. pp. 9650–9660, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Caron_Emerging_Properties_in_Self-Supervised_Vision_Transformers_ICCV_2021_paper.

Ferhat Ozgur Catak, Murat Kuzlu, and Umit Cali. BERT4MIMO: A Foundation Model using BERT Architecture for Massive MIMO Channel State Information Prediction, January 2025. URL http://arxiv.org/abs/2501.01802. arXiv:2501.01802 [cs].

Nick Chater and Paul Vitányi. Simplicity: a unifying principle in cognitive science? *Trends in Cognitive Sciences*, 7(1):19–22, January 2003. ISSN 1364-6613, 1879-307X. doi: 10.1016/S1364-6613(02)00005-0. URL https://www.cell.com/trends/cognitive-sciences/abstract/S1364-6613(02)00005-0.

Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Review*, 43(1):129–159, January 2001. ISSN 0036-1445. doi: 10.1137/S003614450037906X. URL https://epubs.siam.org/doi/abs/10.1137/S003614450037906X. Publisher: Society for Industrial and Applied Mathematics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, July 2020. URL http://proceedings.mlr.press/v119/chen20j/chen20j.pdf.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse Autoencoders Find Highly Interpretable Features in Language Models, October 2023. URL http://arxiv.org/abs/2309.08600. arXiv:2309.08600 [cs].

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

D.L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006. ISSN 0018-9448. doi: 10.1109/TIT.2006.871582. URL http://ieeexplore.ieee.org/document/1614066/.

D.L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory*, 47(7):2845–2862, November 2001. ISSN 00189448. doi: 10.1109/18.959265. URL http://ieeexplore.ieee.org/document/959265/.

Michael Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer New York, New York, NY, 2010. ISBN 978-1-4419-7010-7 978-1-4419-7011-4. doi: 10.1007/978-1-4419-7011-4. URL https://link.springer.com/10.1007/978-1-4419-7011-4.

Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. Why Does Unsupervised Pre-training Help Deep Learning? In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 201–208. JMLR Workshop and Conference Proceedings, March 2010. URL https://proceedings.mlr.press/v9/erhan10a.html. ISSN: 1938-7228.

Marco Giordani, Michele Polese, Arnab Roy, Douglas Castor, and Michele Zorzi. A Tutorial on Beam Management for 3GPP NR at mmWave Frequencies. *IEEE Communications Surveys & Tutorials*, 21(1):173–196, 2019. ISSN 1553-877X. doi: 10.1109/COMST.2018.2869411. URL https://ieeexplore.ieee.org/abstract/document/8458146.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, and others. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf.

Berkay Guler, Giovanni Geraci, and Hamid Jafarkhani. A Multi-Task Foundation Model for Wireless Channel Representation Using Contrastive and Masked Autoencoder Learning, May 2025. URL http://arxiv.org/abs/2505.09160. arXiv:2505.09160 [cs].

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/papers/He_Momentum_Contrast_for_Unsupervised_Visual_Representation_Learning_CVPR_2020_paper.pdf.

Jason M. Klusowski and Jonathan W. Siegel. Sharp Convergence Rates for Matching Pursuit. *IEEE Transactions on Information Theory*, 71(7):5556–5569, July 2025. ISSN 1557-9654. doi: 10.1109/TIT.2025.3564227. URL https://ieeexplore.ieee.org/document/10976388.

Wen Liu, Mingjie Jia, Zhongliang Deng, and Changyan Qin. MHSA-EC: An Indoor Localization Algorithm Fusing the Multi-Head Self-Attention Mechanism and Effective CSI. *Entropy*, 24(5):599, May 2022. ISSN 1099-4300. doi: 10.3390/e24050599. URL https://www.mdpi.com/1099-4300/24/5/599. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

Ke Ma, Zhaocheng Wang, Wenqiang Tian, Sheng Chen, and Lajos Hanzo. Deep Learning for mmWave Beam-Management: State-of-the-Art, Opportunities and Challenges. *IEEE Wireless Communications*, 30(4):108–114, August 2023. ISSN 1558-0687. doi: 10.1109/MWC.018.2100713. URL https://ieeexplore.ieee.org/abstract/document/9846951.

Alina L. Machidon and Veljko Pejović. Deep learning for compressive sensing: a ubiquitous systems perspective. *Artificial Intelligence Review*, 56(4):3619–3658, April 2023. ISSN 1573-7462. doi: 10.1007/s10462-022-10259-5. URL https://doi.org/10.1007/s10462-022-10259-5.

D. Malioutov, M. Cetin, and A.S. Willsky. A sparse signal reconstruction perspective for source localization with sensor arrays. *IEEE Transactions on Signal Processing*, 53(8):3010–3022, August 2005. ISSN 1941-0476. doi: 10.1109/TSP.2005.850882. URL https://ieeexplore.ieee.org/document/1468495. Conference Name: IEEE Transactions on Signal Processing.

Arne Niitsoo, Thorsten Edelhäußer, Ernst Eberlein, Niels Hadaschik, and Christopher Mutschler. A deep learning approach to position estimation from channel impulse responses. *Sensors*, 19(5):1064, 2019. Publisher: MDPI.

Jonathan Ott, Jonas Pirkl, Maximilian Stahlke, Tobias Feigl, and Christopher Mutschler. Radio Foundation Models: Pre-training Transformers for 5G-based Indoor Localization. In *2024 14th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–6, October 2024. doi: 10.1109/IPIN62893.2024.10786154. URL https://ieeexplore.ieee.org/document/10786154. ISSN: 2471-917X.

Timothy J. O'Shea, Tamoghna Roy, and Nathan West. Approximating the Void: Learning Stochastic Channel Models from Observation with Variational Generative Adversarial Networks. In *2019 International Conference on Computing, Networking and Communications (ICNC)*, pp. 681–686, February 2019. doi: 10.1109/ICCNC.2019.8685573. URL https://ieeexplore.ieee.org/document/8685573/. ISSN: 2325-2626.

Guangjin Pan, Kaixuan Huang, Hui Chen, Shunqing Zhang, Christian Häger, and Henk Wymeersch. Large Wireless Localization Model (LWLM): A Foundation Model for Positioning in 6G Networks, May 2025. URL http://arxiv.org/abs/2505.10134. arXiv:2505.10134 [eess].

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. 2018. URL https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving Dictionary Learning with Gated Sparse Autoencoders, April 2024. URL http://arxiv.org/abs/2404.16014. arXiv:2404.16014 [cs] version: 2.

Artan Salihu, Stefan Schwarz, and Markus Rupp. Attention Aided CSI Wireless Localization. In *2022 IEEE 23rd International Workshop on Signal Processing Advances in Wireless Communication (SPAWC)*, pp. 1–5, July 2022. doi: 10.1109/SPAWC51304.2022.9833994. URL https://ieeexplore.ieee.org/abstract/document/9833994. ISSN: 1948-3252.

Artan Salihu, Markus Rupp, and Stefan Schwarz. Self-Supervised and Invariant Representations for Wireless Localization. *IEEE Transactions on Wireless Communications*, 23(8):8281–8296, August 2024. ISSN 1558-2248. doi: 10.1109/TWC.2023.3348203. URL https://ieeexplore.ieee.org/abstract/document/10382964. Conference Name: IEEE Transactions on Wireless Communications.

Nir Shlezinger and Yonina C. Eldar. Model-Based Deep Learning, June 2023. URL http://arxiv.org/abs/2306.04469. arXiv:2306.04469 [eess].

Osvaldo Simeone. A Very Brief Introduction to Machine Learning With Applications to Communication Systems. *IEEE Transactions on Cognitive Communications and Networking*, 4 (4):648–664, December 2018. ISSN 2332-7731. doi: 10.1109/TCCN.2018.2881442. URL https://ieeexplore.ieee.org/document/8542764/.

Maximilian Stahlke, Tobias Feigl, Mario H. Castañeda García, Richard A. Stirling-Gallacher, Jochen Seitz, and Christopher Mutschler. Transfer Learning to adapt 5G AI-based Fingerprint Localization across Environments. In *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pp. 1–5, 2022. doi: 10.1109/VTC2022-Spring54318.2022.9860906. 10 citations (Crossref) [2024-06-13].

Maximilian Stahlke, Tobias Feigl, Sebastian Kram, Bjoern M. Eskofier, and Christopher Mutschler. Uncertainty-Based Fingerprinting Model Monitoring for Radio Localization. *IEEE Journal of Indoor and Seamless Positioning and Navigation*, 2:166–176, 2024. ISSN 2832-7322. doi: 10.1109/JISPIN.2024.3398568. URL https://ieeexplore.ieee.org/document/10526425/?arnumber=10526425. Conference Name: IEEE Journal of Indoor and Seamless Positioning and Navigation.

Petre Stoica, Prabhu Babu, and Jian Li. SPICE: A Sparse Covariance-Based Estimation Method for Array Processing. *IEEE Transactions on Signal Processing*, 59(2):629–638, February 2011. ISSN 1941-0476. doi: 10.1109/TSP.2010.2090525. URL https://ieeexplore.ieee.org/abstract/document/5617289. Conference Name: IEEE Transactions on Signal Processing.

Vladimir Temlyakov. *Greedy Approximation*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2011.

Robert Tibshirani. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, January 1996. ISSN 0035-9246. doi: 10.1111/j.2517-6161.1996.tb02080.x. URL https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

J.A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, October 2004. ISSN 1557-9654. doi: 10.1109/TIT.2004.834793. URL https://ieeexplore.ieee.org/document/1337101/?arnumber=1337101&tag=1. Conference Name: IEEE Transactions on Information Theory.

Joel A. Tropp and Anna C. Gilbert. Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, December 2007. ISSN 1557-9654. doi: 10.1109/TIT.2007.909108. URL https://ieeexplore.ieee.org/abstract/document/4385788. Conference Name: IEEE Transactions on Information Theory.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013. ISBN 1-4757-3264-3.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Benjamin Wright and Lee Sharkey. Addressing Feature Suppression in SAEs. February 2024. URL https://www.alignmentforum.org/posts/3JuSjTZyMzaSeTxKk/addressing-feature-suppression-in-saes.

Tingting Yang, Ping Zhang, Mengfan Zheng, Yuxuan Shi, Liwen Jing, Jianbo Huang, and Nan Li. WirelessGPT: A Generative Pre-trained Multi-task Learning Framework for Wireless Communication, February 2025. URL http://arxiv.org/abs/2502.06877. arXiv:2502.06877 [cs].

Zai Yang, Jian Li, Petre Stoica, and Lihua Xie. Sparse methods for direction-of-arrival estimation. In *Academic Press Library in Signal Processing, Volume 7*, pp. 509–581. Elsevier, 2018.

Bowen Zhang, Houssem Sifaou, and Geoffrey Ye Li. CSI-Fingerprinting Indoor Localization via Attention-Augmented Residual Convolutional Neural Network. *IEEE Transactions on Wireless Communications*, 22(8):5583–5597, August 2023. ISSN 1558-2248. doi: 10.1109/TWC.2023.3235449. URL https://ieeexplore.ieee.org/abstract/document/10018917. Conference Name: IEEE Transactions on Wireless Communications.

Chaoyun Zhang, Paul Patras, and Hamed Haddadi. Deep Learning in Mobile and Wireless Networking: A Survey. *IEEE Communications Surveys & Tutorials*, 21(3):2224–2287, 2019. ISSN 1553-877X. doi: 10.1109/COMST.2019.2904897. URL https://ieeexplore.ieee.org/abstract/document/8666641.

A Appendix

A.1 Theoretical Analysis

We treat the incoming signal as a time-dependent function $h(t)$ to be expressed in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ with basis $\Psi = \{\psi_i\}_{i=0}^{N}$. For the sake of theoretical analysis, we treat the transformer encoder as an invertible operator $O : \mathcal{H} \to \mathcal{H}$ mapping from measured functions $h \in \mathcal{H}$ to latent representations $\tilde{h} = O[h]$. Our idea is to model $\tilde{h}$ rather than h itself to reach a more compact representation. We first characterize the error of a signal transformed by an invertible operator O. Note that all proofs for the theorems can be found in the appendix.

Theorem A.1. *Let $\mathcal{H}$ be a reproducing kernel Hilbert space, equipped with a basis $\{\psi_i\}_{i=0}^{N}$.*

For any $h \in \mathcal{H}$ let the best n-term approximator be

$$\sigma_n(h) = \min_{|I| \leq n} \|h - \sum_{i \in I} a_i \psi_i\|_{\mathcal{H}}.$$

Also define the 1-atomic norm as

$$\|h\|_{A_1(H)} = \inf\{\sum_i \|a_i\| : h = \sum_i a_i \psi_i\}.$$

Assume that there exists an exact recovery condition (ERC) (Tropp, 2004) such that the $O(n^{1/2})$ rate is optimal (Klusowski & Siegel, 2025)[1]).

Then there exists a constant $C > 0$ such that for every $h \in \mathcal{H}$ we have

$$\sigma_n(h) \leq C \frac{\|h\|_{A_1(\mathcal{H})}}{\sqrt{n}}.$$

Let $O : \mathcal{H} \to \mathcal{H}$ be any invertible bounded linear operator with the standard operator norm $\|O\| = \sup_{\|h\|_{\mathcal{H}}=1} \|O[h]\|_{\mathcal{H}}$.

Define the O-atomic norm of $f \in \mathcal{H}$ as

$$\|h\|_{A_1^O} = \|O[h]\|_{A_1(\mathcal{H})}.$$

Then there exists an n-term representation g_n in $\mathcal{H}$ such that

$$\|O[h] - g_n\|_{\mathcal{H}} \leq C \frac{\|h\|_{A_1^O}}{\sqrt{n}}$$

or, in the original space: $g_n = O[\tilde{h}_n]$

$$\|h - \tilde{h}_n\|_{\mathcal{H}} \leq \|O^{-1}\| C \frac{\|h\|_{A_1^O}}{\sqrt{n}}.$$

The result in Theorem A.1 implies that if one tries to bound K functions simultaneously,

$$\max_{0 \leq i \leq K} \|h_i - h_{i,n}\| \leq \|O^{-1}\| C \frac{\max_{0 \leq i \leq K} \|h_i\|_{A_1^O}}{\sqrt{n}}$$

preconditioning with O improves upon the direct estimation if there exists a common substructure that allows us to reduce the coefficients in $\|h_i\|_{A_1^O}$ without overly affecting $\|O^{-1}\|$.

This is not unexpected: This is exactly what happens in signal smoothing where O flattens singularities, or in differential preconditioning for PDEs. We provide an example of an analytically derived operator in Section A.3. Instead of using prior knowledge, we decide to learn O with the option to fit it jointly with the dictionary Ψ on a dataset of $\{h_0, \ldots, h_K\}$ such that the operator preconditions the entire set.

Specifically, we can construct such an operator as follows:

Theorem A.2. *Let $\{h_i\}_{i=0}^K$ be a dataset of signals, and $\mathcal{H}$ be a Hilbert space with a (possibly infinite) basis $\{\psi\}_{j=0}^N$. Let $R = \max_{0 \leq i \leq K} \sum_{j=1}^N |a_{i,j}|$ and fix a nonempty index set $S \subset \{1, \ldots, N\}$. Define*

$$R_S := \max_{0 \leq i \leq K} \sum_{j \in S} |a_{i,j}| \qquad B := \max_{0 \leq i \leq K} \sum_{j \notin S} |a_{i,j}|.$$

If $B < R_S$ then there exists an operator O such that

$$\max_{0 \leq i \leq K} \|h_i - h_{i,n}| \leq C \frac{\max_{0 \leq i \leq K} \|h_i\|_{A_1^O}}{\sqrt{n}},$$

which is strictly better than the original rate $C \frac{R}{\sqrt{n}}$.

Remark A.3. The assumptions in Theorem A.2 are quite technical, but can be boiled down to "One can sacrifice a more irrelevant subset $\bar{S}$ in favor of modeling a relevant subset S with higher weight." In practice, this is a small assumption for an uninformed selection of the basis.

Of course, the "small index set" assumption for Theorem A.2 is generally restrictive, but we can easily generalize the proof towards a low-rank assumption:

Corollary A.4. *Assume there exists a rank-s subspace S, then the same bounds from Theorem A.2 hold.*

[1]This is equivalent to asserting a generalised Jackson-type inequality over the codebook established by Temlyakov (2011).

Remark A.5. The assumptions in Corollary A.4 may seem strong at first, but in practice boil down to a weaker version of the assumptions necessary for compressed sensing: The real signal lives in a lower-dimensional (function) space than the (noisy) measured signal.

One important consequence of Corollary A.4 is that we can find a smaller basis expansion, i.e., the weighted sum of basis functions in Ψ, by implicitly learning an infinite-dimensional dense rotation and diagonal scaling operator O. While simple finite operations at first appear to be sufficient for this, one has to note that the basis set we work with is generally infinite, meaning an orthogonal projection is not naively parameterizable. We decide to approximate O with a TF encoder network by jointly minimizing the reconstruction error after applying O together with the codebook Ψ. In short, our backbone transforms channel measurements to align with compressed sensing assumptions, biasing outputs toward simpler and more general solutions.

A.2 Proofs

Proof of Theorem A.1. Let

$$u := O[f] \in \mathcal{H}$$

then, by definition of $\|\cdot\|_{A_1(\mathcal{H})}$ we have for all $\varepsilon > 0$ a finite set of coefficients c_i such that

$$u = \sum_{i\in I} c_i\psi, \quad \sum_{i\in I} |c_i| \leq \|u\|_{A_1(\mathcal{H})} + \varepsilon = \|h\|_{A_1^O} + \varepsilon.$$

Applying ERC to u we get

$$\sigma_n(u) = \min_{J\subset[N],|J|\leq n} \left\| u - \sum_{j\in J} a_j\psi_j \right\| \leq C\frac{\|u\|_{A_1(\mathcal{H})}}{\sqrt{n}} \leq C\frac{\|h\|_{A_1^O} + \varepsilon}{\sqrt{n}},$$

by definition, we have a

$$g_n = \sum_{j\in J_n} a_j\psi_j, \quad |J_n| = n$$

such that

$$\|u - g_n\|_{\mathcal{H}} \leq C\frac{\|h\|_{A_1^O} + \varepsilon}{\sqrt{n}}.$$

Finally, set $O^{-1}[g_n] = \tilde{h}_n$ to obtain

$$\|h - \tilde{h}_n\|_{\mathcal{H}} = \|O^{-1}[u - g_n]\|_{\mathcal{H}} \leq \|O^{-1}\|\|[u - g_n]\|_{\mathcal{H}} \leq \|O^{-1}\|C\frac{\|h\|_{A_1^O} + \varepsilon}{\sqrt{n}}.$$

Now let $\varepsilon \to 0$ and we obtain our bound

$$\|h - \tilde{h}_n\|_{\mathcal{H}} \leq \|O^{-1}\|C\frac{\|h\|_{A_1^O}}{\sqrt{n}}.$$

□

Proof of Theorem A.2. Let O be a diagonal operator with weights w_i such that $O[h] = \sum w_i c_i \psi_i$. Let w_i be defined as

$$w_i = \begin{cases} c & j \in S \\ 1 & j \notin S \end{cases}$$

with $c > 1$. Operators with this weighting have inverse norm $\|O^{-1}\| = \max(1/c, 1) = 1$ and

$$\max_{0\leq i\leq K} \|h_i\|_{A_1^O} = \max_{0\leq i\leq K} \sum_{j\in S} |a_{i,j}| + \frac{1}{c}\sum_{j\in S} |a_{i,j}| = B + \frac{1}{c}\max_{0\leq i\leq K}\sum_{j\in S} |a_{i,j}|.$$

Since, by assumption $B < R$ we may choose any c large enough such that the residual

$$\frac{R_S}{c} < R - B.$$

Specifically, choose

$$c > \frac{R_S}{R - B},$$

then

$$\max_{0 \le i \le K} \|h_i\|_{A_1^O} = B + \frac{R_S}{c} < B + (R - B) = R.$$

In short, the resulting rate is (nonasymptotically) better than the original one. □

Proof of Corollary A.4. Let M_{diag} be the advantage gained by a diagonal scaler. First apply an orthonormal transform V projecting onto s axes (via SVD or polar decomposition), then apply theorem A.2. This only adds the burden of back-projecting from V space to the original space, i.e.

$$\max_i \|h_i - h_{i,n}\| \le \|V^{-1}\| C \, M_{\text{diag}} / \sqrt{n},$$

which due to orthonormality yields

$$\max_i \|h_i - h_{i,n}\| \le C \, M_{\text{diag}} / \sqrt{n}.$$

□

A.3 Example Preconditioning Operator

Assume one wants to solve a Poisson differential equation using the Spectral method on the basis of Legendre Polynomials. Let's define the Poisson equation $-\Delta u(x) = f(x)$ for ground truth $f(x) = \frac{1}{\sqrt{1-x^2}}$ over $x \in (-1, 1)$ with $u(-1) = u(1) = 0$. Approximating f with Legendre polynomials has slow decay for spectral methods due to the endpoint singularities. Preconditioning with Green's operator

$$O := \left(-\frac{d^2}{dx^2} \right)^{-1},$$

then

$$O[f] = u$$

is smooth over $(-1, 1)$ and in fact has an analytic interior. Specifically, Legendre coefficients in f decay on the order of n^{-1} due to the singularity, while coefficients in u decay exponentially. For us this would mean

$$\|f\|_{A_1} := \inf \left\{ \sum_i |c_i| \, | f = \sum_i c_i \psi_i \right\},$$

while

$$\|f\|_{A_1^O} = \|O[f]\|_{A_1} = \|u\|_{A_1},$$

by the convergence rate we get

$$\|f\|_{A_1^O} << \|f\|_{A_1}.$$

The norm

$$\|O^{-1}\| = \| - \frac{d^2}{dx^2} \|$$

over $H^2 \cap H_0^1$ - the second-order Sobolev space cut with the first-order subspace which vanishes on the boundary - is a fixed constant.

A.4 Experimental Setup: Baselines & Datasets

A.4.1 Baselines

An important difference between SpaRTran and the baseline methods is the use of single links between antennas instead of full CSI to compute the representations (see Section 4.3). To ensure optimal performance, we finetune all backbone layers along with the downstream task heads rather than freezing them. For both pretraining and fine-tuning, we train each method until convergence, using sufficiently long training schedules to ensure that performance has saturated. This protocol is

Table 4: Comparison of method configurations.

Method	N_{latent}	N_{hidden}	N_{heads}	N_{blocks}	#param backbone	#param head
Masking	512	1024	8	3	8.7 M	4 k
SWiT	384	384	1	1	4.0 M	889 k
LWM	64	256	1	12	1.3 M	17 k
Contra-WiMAE	64	128	8	20	587 K	176 k
SpaRTran (ours)	512	1024	8	1	2.6 M	1.4 M

applied consistently across all methods to avoid differences arising from under-training rather than from the respective pretraining approaches.

SWiT: Salihu et al. (2024) propose a joint embedding-based approach (Grill et al., 2020) called self-supervised wireless transformer (SWiT), that predicts the output of a momentum encoder, given different augmented views of the same input signal. The aim is to learn representations that are invariant to six randomly selected augmentations, diversifying the views.

WiT: Salihu et al. (2022) employ a compact TF model consisting of a single encoder block with single-head attention that is trained end-to-end in a supervised manner.

Masking: Ott et al. (2024) introduce a predictive objective for learning FP representations, in which masked portions of the input signal are reconstructed. During training, up to 50% of the input fingerprint is removed, forcing the model to learn spatiotemporal correlations between the multipath components (MPCs).

LWM: Alikhani et al. (2024) also use a masking strategy to pretrain the TF model. Here multiple patches of 15 time steps of the input signal are gathered to be presented to the network as a token. They randomly select 15% of patches and, for those, replace 80% with a uniform MASK vector, 10% with random noise, and leave 10% unchanged.

Contra-WiMAE: Guler et al. (2025) combine a masked autoencoder with a contrastive pretraining objective. They generate positive samples by applying additive white Gaussian noise to a CSI sample while treating all other channels in the batch as negative samples. The goal is to encourage the model to learn structural patterns via the masking objective and discriminative features via the contrastive objective. In Table 3 we report the results for Contra-WiMAE from the paper.

A.4.2 DATASETS

Table 5: Comparison of dataset configurations.

Dataset	f_c [GHz]	W [MHz]	N_r	Area
KUL	2.61	20	64	3 m × 3 m
FH-IIS	3.7	100	6	40 m × 30 m
DeepMIMO	3.5	20	32	$>1\,\mathrm{km}^2$

KUL Dataset[2] (Bast et al., 2020): The dataset comprises four antenna configurations: distributed antennas (DIS-LoS), a uniform linear array (ULA-LoS), and a uniform rectangular array under both line-of-sight (LoS) (URA-LoS) and non-line-of-sight (nLoS) conditions (URA-nLoS). Each configuration contains 252,004 CSI samples with recording positions arranged in a grid-pattern with 5 mm spacing. The channels are measured at 20 MHz bandwidth. We split the dataset randomly into 70 % for training, 10 % for validation, and 20 % for testing.

FH-IIS Dataset[3] (Stahlke et al., 2024): This dataset contains CIR fingerprints collected using a 5G-FR1-compatible software-defined radio system (DL-PRS reference signal) with a bandwidth of 100 MHz. The recorded environment resembles an industrial hall featuring tall metal shelves, and a

[2]The dataset is released under the Creative Commons Attribution (CC BY) license. It is available at https://ieee-dataport.org/open-access/ultra-dense-indoor-mamimo-csi-dataset

[3]Use of the dataset was authorized by the original creators under a formal agreement. It is available at https://www.iis.fraunhofer.de/en/ff/lv/dataanalytics/pos/fingerprinting-dataset-for-positioning.html

Table 6: Architectural Hyperparameter Ablation

	N_{latent}	N_{blocks}	$N_{blocks,head}$	MAE	CE90	#param ($\times 10^6$) backbone	head
base	512	1	4	0.093	0.177	2.61	1.40
(a)			3	0.128	0.240		0.26
			2	0.159	0.303		0.05
			5	0.076	0.149		6.65
(b)	256			0.089	0.179	0.83	
	1024			0.211	0.409	9.31	
		2		0.064	0.126	4.71	
		3		0.113	0.223	6.81	

narrow corridor with large walls that introduce signal blockages and complex multipath propagation. The CSI is captured along a random walking trajectory of a person at a sampling rate of 6.6 Hz. Six base-stations are distributed along the perimeter of the localization area. The split sizes are as follows: 566,589 training, 141,639 validation and 593,022 test samples.

DeepMIMO Dataset[4] (Alkhateeb, 2019): This is a synthetic dataset generated by a ray-tracing engine. We configure it to simulate a uniform linear array with 32 antennas at 20 MHz bandwidth, and 3.5 GHz carrier frequency. We split the dataset into a pretraining and finetuning subset including 15 scenarios (*O1, Boston5G, ASU Campus, New York, Los Angeles, Chicago, Houston, Phoenix, Philadelphia, Miami, Dallas, San Francisco, Austin, Columbus, Seattle*) and 6 scenarios (*Denver, Fort Worth, Oklahoma, Indianapolis, Santa Clara, San Diego*) respectively. This yields the following split sizes: pretraining — 540,272 training and 135,075 validation samples; finetuning — 10,385 training, 1,481 validation, and 1,481 test samples.

We normalize the received signal strength of the FH-IIS and KUL datasets using a global normalization factor. This preserves the relative signal strength differences within the channel measurements, a crucial property for the downstream task of wireless localization. In contrast, DeepMIMO is normalized per CSI matrix to reduce sensitivity to outliers from large peaks that occur at short transmitter–receiver distances (signal strength scales roughly with $1/d^2$).

A.5 Hyperparameter Analysis

Table 6 shows changes in localization accuracy in dependency on changes in the hyperparameters from the base model—i.e. architectural variations—and the corresponding number of parameters. In rows (a) we vary the depth of the finetuning head $N_{blocks,head}$ while the other hyperparameters remain fixed (see Sec. 4.3 for head network description). As the representation module handles only single links between antennas the spatial correlations between links in the full CSI must be learned solely by the head. This contrasts with existing works which are less dependent on the expressivity of their head network as the spatial correlations are implicitly learned during pretraining. As a result, they can use lightweight heads for finetuning reducing the additional computational cost (see Table 4). This dependency is clearly reflected in our results as the MAE gets consistently smaller with an increase in $N_{blocks,head}$ from $0.159\,\mathrm{m}$ given $N_{blocks,head} = 2$ to $0.076\,\mathrm{m}$ given $N_{blocks,head} = 5$. It should be noted that the number of parameters in the network increases drastically with larger $N_{blocks,head}$. This is why we choose $N_{blocks,head}$ as the base configuration providing good results with a reasonable amount of parameters for the head network (1.4 million). Rows (b) provide insight into varying configurations of the representation module, i.e., the transformer encoder network. We alter the latent dimension N_{latent} and the depth N_{blocks} of the network. While a smaller latent dimension of 256 has no relevant effect on the localization accuracy, doubling the size of it up to 1024 dimensions significantly increases the MAE by 227% from $0.093\,\mathrm{m}$ to $0.211\,\mathrm{m}$ and the CE90 by 231% from $0.177\,\mathrm{m}$ to $0.409\,\mathrm{m}$. As this comes with a radical increase in parameters up to ≈ 9 million we think that the reduction in accuracy comes from overfitting the training data. The results

[4]The dataset is released under the Attribution-NonCommercial-ShareAlike 4.0 International license. We use Version 3 of the dataset. It is available at https://www.deepmimo.net/versions/legacy#v3

for varying depths N_{blocks} of the transformer encoder again indicate a tradeoff between overfitting and expressivity which is well balanced at $N_{blocks} = 2$. Nevertheless, we use $N_{blocks} = 1$ for our base configuration to reduce the number of parameters making our model more lightweight than Masking and SWiT (see Table 4).

A.6 Results Across Random Seeds

As an additional robustness check, we evaluate the sensitivity to random initialization by reporting the median and variance across five seeded runs for SpaRTran and the strongest baseline, C-WiMAE, on the KUL cross-environment experiment. The results reported in Table 7 show that the performance gains of SpaRTran are consistent across random seeds, providing further evidence that the observed improvements are robust to random initialization.

Table 7: Median and standard deviation in meters of FP performance across five seeded runs for C-WiMAE and SpaRTRan methods on the KUL cross-environment experiment, fine-tuned on 5,000 samples (MAE).

Method	Pretrain-Set	DIS-LoS	ULA-LoS	URA-LoS	URA-nLoS
C-WiMAE:	DIS-LoS	**0.053 ± 0.008**	0.092 ± 0.004	0.125 ± 0.006	0.444 ± 0.017
	ULA-LoS	0.098 ± 0.006	0.089 ± 0.009	0.089 ± 0.005	0.283 ± 0.017
	URA-LoS	0.083 ± 0.004	0.076 ± 0.005	0.066 ± 0.009	**0.087 ± 0.007**
	URA-nLoS	0.110 ± 0.005	0.102 ± 0.003	0.090 ± 0.006	**0.074 ± 0.010**
SpaRTran:	DIS-LoS	0.076 ± 0.013	**0.055 ± 0.001**	**0.052 ± 0.003**	**0.096 ± 0.015**
	ULA-LoS	**0.069 ± 0.015**	**0.055 ± 0.003**	**0.053 ± 0.003**	**0.111 ± 0.022**
	URA-LoS	**0.074 ± 0.005**	**0.057 ± 0.004**	**0.047 ± 0.004**	0.099 ± 0.012
	URA-nLoS	**0.054 ± 0.004**	**0.048 ± 0.010**	**0.045 ± 0.007**	0.109 ± 0.012