PilotWiMAE: Wireless Channel Pilots Are All You Need

Anonymous Authors[1]

Abstract

Wireless channel foundation models assume access to fully observed channels, an assumption that fails in deployment. We introduce PilotWiMAE, a self-supervised framework whose encoder ingests noisy pilot observations directly and whose attention factorizes along the axis separating temporal from joint space-frequency processing, an inductive bias grounded in the physics of the problem. Pilot input shrinks the observation space by up to two orders of magnitude, while the factorized design yields more robust representations by exploiting separable channel structure and allowing a 99% pretraining mask ratio. Pilot-only processing also removes the unrealistic assumption of full-CSI availability while incurring lower latency. We pair patch-normalized reconstruction, which captures small-scale fading structure, with an auxiliary scale loss that recovers the large-scale fading features, and use an AWGN curriculum to match pilot noise at pretraining and deployment. Pretrained solely on 3.5 GHz and evaluated at 28 GHz across in-distribution and out-of-distribution settings, PilotWiMAE's cross-frequency beam selection and channel characterization beat supervised baselines despite operating on a significantly smaller observation space.

1. Introduction

Recent channel foundation models have made substantial progress in learning transferable representations of wireless channels by pretraining and evaluating fully observed channels generated by stochastic or ray-tracing simulators (Wang et al., 2026; Yang et al., 2026; Guler et al., 2026; Alikhani et al., 2026; Jiang et al., 2025; Alikhani et al., 2025; Liu et al., 2025a;b; 2024). Some of these works add i.i.d. additive white Gaussian noise (AWGN) to fully observed channels as a concession to realism (Wang et al., 2026; Yang et al., 2026; Guler et al., 2026; Liu et al., 2025a; 2024), while others omit noise evaluation altogether. Neither protocol reflects how channel state information (CSI) errors arise in practice. In a real receiver, the channel is estimated from pilots, and only the error at pilot resource elements is i.i.d. AWGN (Edfors et al., 1998). The error at non-pilot resource elements, which make up the vast majority of the grid, depends on the interpolation method, the channel's delay-Doppler structure, the pilot density, the SNR at the pilots, and the pilot design, and does not admit a simple i.i.d. model (Coleri et al., 2002). Evaluation under fully observed or i.i.d.-perturbed channels therefore can only characterize how well a model learns the channel structure, but leaves open how it behaves in a system where such channels are never available. Given the known sensitivity of learned methods to noise and distribution shift (Hendrycks & Dietterich, 2019; Taori et al., 2020), this gap is worth closing.

The second gap concerns the cost of deployment. Foundation models for wireless have largely inherited transformer architectures (Vaswani et al., 2017; Dosovitskiy et al., 2021) and training recipes (Devlin et al., 2019; He et al., 2022) from vision and language, where parameter count and sequence length do not face a hard runtime ceiling and performance predictably improves with additional data and computation (Kaplan et al., 2020; Hoffmann et al., 2022; Zhai et al., 2022). In wireless systems, tasks such as precoding, scheduling, and decoding must be completed within slot-level timing budgets of the order of a millisecond or less (3GPP, 2026). However, latency, memory footprint, and power are rarely reported in the literature on channel foundation models. When reported, the figures often exceed what the pipeline can afford and are measured on high-end GPUs, masking the true cost of deployment.

We address both gaps with two co-equal design principles. First, we pursue *robustness by design* by operating directly on sparse, noisy pilot observations, removing an explicit channel estimator from the critical path to prevent error propagation at realistic low SNR, matching deployment observables, eliminating the full-CSI assumption of prior channel foundation models, and integrating cleanly with existing pilot-based protocols. When full channel recovery is needed, it can be posed as a downstream task on the same learned representations, and it is naturally aligned with our reconstruction-based pretraining objective. Sec-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

ond, we enforce *wireless specificity by design* by factorizing attention along the axis separating temporal from joint space-frequency processing, an inductive bias grounded in the wide-sense stationary uncorrelated scattering (WSSUS) model (Bello, 1963) and its MIMO extension (Matz & Hlawatsch, 2011), where temporal and spectro-spatial correlations arise from distinct physical mechanisms. The same principle motivates our pretraining objective, which pairs patch-normalized reconstruction for small-scale fading with an auxiliary scale loss that recovers large-scale fading statistics. Overall, pilot input shrinks the observation space by up to two orders of magnitude, while the factorized design exploits separable channel structure to support an aggressive 99% pretraining mask ratio. Together, they yield sub-millisecond per-sample inference latency on a single mid-range GPU (Appendix E) and representations that stay reliable in the noisy, partially observed regime where decisions are actually made.

Contributions. We introduce PilotWiMAE, a self-supervised, foundation-model-style framework for wireless channel representation that (i) operates natively on noisy pilot observations for decision-making tasks, thereby bypassing channel estimation and eliminating the need for the perfect-CSI assumption, (ii) factorizes attention along the WSSUS-motivated axis to learn robust, high-performing representations, (iii) combines patch-normalized reconstruction with an auxiliary scale loss to recover large-scale fading structure, and (iv) is pretrained under an AWGN curriculum matched to deployment noise in pilot observations.

Figure 1 provides a high-level overview of the PilotWiMAE pipeline.

1.1. Related work

The vast majority of existing wireless channel foundation models follow pretraining and evaluation protocols built around fully observed channels, with either no corruption or i.i.d. AWGN added as a post-hoc perturbation. Alikhani et al. (2025) and Jiang et al. (2025), for instance, both pretrain and evaluate on fully observed, noise-free channels. The former targets beam selection and line-of-sight (LoS) classification, and the latter extends to positioning as well. Liu et al. (2025a) takes a different path, pretraining a masked autoencoder on channels corrupted with i.i.d. AWGN at 20 dB SNR and evaluating channel prediction under the same condition. Similarly, Alikhani et al. (2026) assesses channel prediction on fully observed noise-free channels, even though the encoder itself was pretrained with AWGN and other augmentations. Yang et al. (2026) also trains on noise-free channels, introducing i.i.d. AWGN for a subset of downstream tasks, while Liu et al. (2024) maintains a consistent SNR range across both pretraining and evaluation on fully observed channels. Guler et al. (2026) pretrains under both reconstructive and contrastive objectives on AWGN-perturbed channels and evaluates under the same noise protocol in channel estimation, beam selection, and LoS classification. Liu et al. (2025b), by contrast, considers fully observed noise-free channels throughout. However, a common limitation persists in all of these works. None of them considers the case where the model input is a sparse pilot observation, the only form of CSI actually available at a real receiver. The only work that acknowledges this limitation is Wang et al. (2026), where the authors propose a framework combining channel estimation and feature extraction into an end-to-end training pipeline.

On the processing side, attention is typically applied generically over all tokens without regard to the physical correlation structure of the channel. The WSSUS separability that we exploit provides a principled axis along which attention can be decomposed. Factorized space-time (FST) attention itself is not new in representation learning, as it has been used in video to exploit the separability of spatial appearance and temporal motion (Bertasius et al., 2021; Arnab et al., 2021). What is new here is the physically grounded analog of that separability in wireless channels and its pairing with a pilot-only input interface so that both the input and the computation align with the physics of the deployed system.

2. Problem formulation

We consider a base station operating over T consecutive time slots, each with N_{sc} subcarriers and a uniform planar array of $N_{\mathrm{h}} \times N_{\mathrm{v}}$ antennas. The full channel tensor $\mathbf{H} \in \mathbb{C}^{T \times N_{\mathrm{h}} N_{\mathrm{v}} \times N_{\mathrm{sc}}}$ is never observed. Instead, the receiver obtains a noisy observation at a sparse set of pilot resource elements indexed by $\mathcal{P}$,

$$\hat{\mathbf{H}}_{\mathcal{P}} = \mathbf{H}_{\mathcal{P}} + \mathbf{N}_{\mathcal{P}}(\mathrm{SNR}), \tag{1}$$

where $\mathbf{N}_{\mathcal{P}}(\mathrm{SNR})$ is i.i.d. circularly-symmetric complex Gaussian noise whose variance is a known function of the pilot SNR alone (Edfors et al., 1998).

More precisely, pilots are sent at selected OFDM symbol indices $\mathcal{T}_{\mathrm{p}} \subset \{0, \ldots, T-1\}$ and subcarrier indices $\mathcal{F}_{\mathrm{p}} \subset \{0, \ldots, N_{\mathrm{sc}} - 1\}$, and at each pilot $(t, f) \in \mathcal{T}_{\mathrm{p}} \times \mathcal{F}_{\mathrm{p}}$ all $N_{\mathrm{h}} N_{\mathrm{v}}$ antenna entries are observed. Our method is agnostic to the specific layout of $(\mathcal{T}_{\mathrm{p}}, \mathcal{F}_{\mathrm{p}})$. The exact indices used in our experiments are listed in Appendix C.

Existing channel foundation models ignore this constraint and instead assume a noisy observation of the full channel

$$\hat{\mathbf{H}} = \mathbf{H} + \mathbf{E}(\boldsymbol{\theta}), \tag{2}$$

where the residual $\mathbf{E}(\boldsymbol{\theta})$ is generally correlated across resource elements and its statistics depend on the estimator

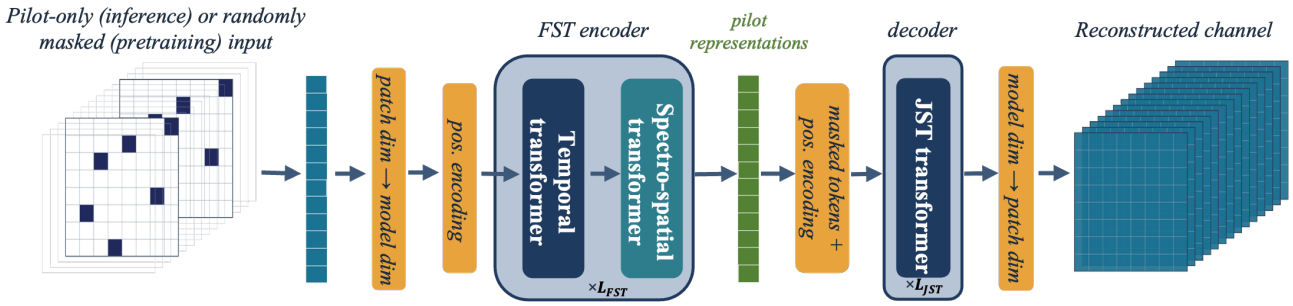

Figure 1. PilotWiMAE architecture. Pilot patches are processed by the FST encoder. The resulting pilot representations are decoded by a JST transformer that maps tokens back to patch dimension to reconstruct the channel.

and on channel parameters that are scenario-specific and unavailable in general. Existing works replace $\mathbf{E}(\boldsymbol{\theta})$ with an i.i.d. circularly-symmetric complex Gaussian surrogate of matched SNR, a convenient but generally non-physical approximation of the true residual.

One straightforward alternative is to prepend a channel estimator to the encoder, recovering $\hat{\mathbf{H}}$ before producing representations. This either requires end-to-end training of an additional block or forces the encoder to be robust to the artifacts of a specific estimator, both of which complicate the design without addressing the root issue. Instead, we feed $\hat{\mathbf{H}}_{\mathcal{P}}$ directly into the encoder.

$$\hat{\mathbf{H}}_{\mathcal{P}} \longrightarrow \text{encoder} \longrightarrow \text{representations} \longrightarrow \text{decision}. \tag{3}$$

The hypothesis is that a sufficiently structured encoder can recover global channel information from sparse input in the representation space, without explicit reconstruction. This does not exclude channel reconstruction or channel estimation. In fact, channel estimation is naturally aligned with our reconstruction-based pretraining objective. We therefore emphasize beam selection and channel characterization in this paper, since they provide a stronger demonstration that the self-supervised representation transfers to downstream decision tasks without task-specific fine-tuning.

3. Our approach

We introduce PilotWiMAE, a self-supervised framework that realizes the two design principles of Section 1 through five concrete choices: a pilot-native input interface, an FST encoder motivated by WSSUS separability, an aggressive masking regime that their combination enables, a pretraining objective that pairs patch-normalized reconstruction with an auxiliary scale loss to capture both small-scale and large-scale fading structure, and a noise-robust pretraining curriculum that matches deployment conditions.

3.1. Pilot-native input interface

The encoder operates directly on the sparse and noisy pilot tensor $\hat{\mathbf{H}}_{\mathcal{P}}$. No channel estimator precedes the encoder, and no reconstruction is required for downstream decisions such as beam selection or channel characterization. During pretraining, the encoder sees sparse inputs generated by structured random masking under AWGN. During inference, it sees sparse inputs induced by the fixed pilot pattern under AWGN. The input tensor is roughly two orders of magnitude smaller than the full channel grid, which the factorized attention turns into a direct latency gain.

Before tokenization, each sample is power-normalized using a dataset-level reference power P_{ref} computed on the pretraining split: $\bar{\mathbf{H}} = \mathbf{H}/\sqrt{P_{\mathrm{ref}}}$. Let the complex input tensor be $\bar{\mathbf{H}} \in \mathbb{C}^{T\times S\times F}$ and the 3D patch size be $(p_{\mathrm{t}}, p_{\mathrm{s}}, p_{\mathrm{f}})$. Defining $(n_{\mathrm{t}}, n_{\mathrm{s}}, n_{\mathrm{f}}) = (T/p_{\mathrm{t}}, S/p_{\mathrm{s}}, F/p_{\mathrm{f}})$, tokenization yields $P = n_{\mathrm{t}}n_{\mathrm{s}}n_{\mathrm{f}}$ patches, with $N_{\mathrm{sf}} = n_{\mathrm{s}}n_{\mathrm{f}}$ spectro-spatial tokens per slot. The real and imaginary parts are split and concatenated within each patch. Therefore, each raw patch vector has dimension $D_{\mathrm{p}} = 2p_{\mathrm{t}}p_{\mathrm{s}}p_{\mathrm{f}}$ before linear projection to the model dimension d. For positional encoding, we use sinusoidal-concatenative 3D embeddings: $d = d_{\mathrm{t}} + d_{\mathrm{s}} + d_{\mathrm{f}}$ with separate sinusoidal tables for time, space, and frequency, concatenated and added with a learnable scale α_{pe}. In each transformer block, the feed-forward width is set as $d_{\mathrm{ff}} = \kappa_{\mathrm{ff}}d$.

3.2. WSSUS-motivated factorized attention

Under the classical WSSUS model (Bello, 1963) and its MIMO extension (Matz & Hlawatsch, 2011), the temporal correlation is governed by the Doppler spectrum, a function of the environment mobility, while the spectro-spatial correlation is governed by the joint angular-delay power spectrum, a function of the scattering geometry. Writing the channel autocorrelation over a time lag Δt, a frequency lag Δf, and a spatial lag Δs, the WSSUS assumption and the standard separability of Doppler from angle-delay dispersion in MIMO-WSSUS channels (Matz & Hlawatsch, 2011) yield

$$R_{\mathrm{H}}(\Delta t, \Delta f, \Delta s) \;\approx\; R_{\mathrm{t}}(\Delta t)\, R_{\mathrm{sf}}(\Delta f, \Delta s). \qquad (4)$$

This approximation says that temporal and space-frequency correlations are driven by weakly coupled physical mechanisms, so a representation learner that respects this structure need not model cross-domain correlations that carry little signal. One might ask whether to separate space from frequency as well, but in wideband MIMO propagation they remain coupled through the jointly angle- and delay-dependent scattering structure (Matz & Hlawatsch, 2011), so imposing three-way separability would discard the real physical structure that the encoder should learn.

This prior maps to an attention factorization (Bertasius et al., 2021; Arnab et al., 2021). Let $\mathbf{Z}^{(\ell)} \in \mathbb{R}^{T \times N_{\mathrm{sf}} \times d}$ be the token tensor at Layer ℓ, where T is the number of temporal tokens, N_{sf} is the number of spectro-spatial tokens per slot, and d is the embedding dimension. Our FST encoder applies, within each layer, temporal attention across slots followed by spectro-spatial attention within each slot.

$$\mathbf{Z}^{(\ell+\frac{1}{2})}_{:,s,:} = \mathrm{Attn}_{\mathrm{t}}\left(\mathbf{Z}^{(\ell)}_{:,s,:}\right), \quad s = 1, \ldots, N_{\mathrm{sf}}, \qquad (5)$$

$$\mathbf{Z}^{(\ell+1)}_{t,:,:} = \mathrm{Attn}_{\mathrm{sf}}\left(\mathbf{Z}^{(\ell+\frac{1}{2})}_{t,:,:}\right), \quad t = 1, \ldots, T, \qquad (6)$$

where residual connections and feedforward sublayers are omitted for the sake of clarity. Cross-slot information is exchanged only by temporal attention at fixed spectro-spatial indices, while spectro-spatial information is exchanged only by within-slot attention at fixed time indices, mirroring Eq. (4). Because factorization is an inductive bias rather than a sparsification of the attention matrix, it does not degrade expressivity on structures that WSSUS captures. Our ablations against a joint space-time (JST) baseline of matched parameter count show consistent gains on both in-distribution and out-of-distribution beam selection and channel characterization.

Relative to joint attention over all TN_{sf} tokens, which scales as $\mathcal{O}((TN_{\mathrm{sf}})^2 d)$, factorized attention scales as $\mathcal{O}(TN_{\mathrm{sf}}(T + N_{\mathrm{sf}})\, d)$ per layer. Related discussion and measured training/inference costs are presented in Appendix E.

3.3. Aggressive masking enabled by factorization

Factorized attention and separability permit an unusually aggressive pretraining masking regime. Figure 6 (Appendix B) illustrates the two-step decomposition that this masking strategy is designed to exploit. We apply a structured random mask that retains only T_{k} out of T time slots and, across the retained slots, keeps a common fraction $\rho_{\mathrm{k}} \in (0, 1)$ of the spectro-spatial token positions (the same positions in every retained slot rather than resampled per slot). Therefore, the overall fraction of visible tokens is $(T_{\mathrm{k}}/T)\, \rho_{\mathrm{k}}$ and the overall mask ratio is $1 - (T_{\mathrm{k}}/T)\, \rho_{\mathrm{k}}$. The mask structure matches the factorization. The temporal block mixes information across the T_{k} kept slots at each fixed spectro-spatial position, while the spectro-spatial block mixes across the visible positions within each kept slot, so that the decoder receives informed tokens at every visible location. Randomizing the mask across pretraining examples keeps the encoder agnostic to any specific pilot pattern, so that at inference the same pretrained model can ingest whatever fixed pilot configuration the receiver uses.

The same masking budget applied to a JST encoder does not converge in our experiments. We attribute this to a mismatch between JST's attention pattern and the WSSUS correlation structure. Under 99% masking, most surviving token pairs span both axes of dispersion, across which tokens are generally weakly correlated, leaving JST with little useful signal to learn from. Factorization avoids this by construction, since temporal attention only sees same-position pairs and spectro-spatial attention only sees same-slot pairs, both of which the physics predicts to be well-correlated.

3.4. Patch-normalized reconstruction with an auxiliary scale loss

The power of wireless channel spans an enormous dynamic range across samples. Path loss and shadowing can vary by tens of dB between channels in the same training set, while small-scale multipath fading, the relevant structure for beam selection and many other downstream tasks, occupies a finer amplitude scale. A reconstruction loss computed in raw amplitude would therefore be dominated by high-power patches, and the network would minimize it by fitting large-scale trends while leaving the multipath geometry largely unlearned. We address this by normalizing each patch by its own mean and variance before computing the reconstruction loss. Writing $\mathbf{p}_{b,i} \in \mathbb{R}^{D_{\mathrm{p}}}$ for the i-th patch of sample b (with D_{p} elements per patch) and $\mu_{b,i}$, $\sigma^2_{b,i}$ for its empirical mean and variance, the reconstruction loss over the masked set $\mathcal{M}$ is

$$\mathcal{L}_{\mathrm{recon}} = \frac{1}{|\mathcal{M}|} \sum_{(b,i)\in\mathcal{M}} \left\| \hat{\mathbf{p}}_{b,i} - \frac{\mathbf{p}_{b,i} - \mu_{b,i}\mathbf{1}}{\sqrt{\sigma^2_{b,i} + \epsilon_{\mathrm{r}}}} \right\|_2^2, \qquad (7)$$

where ϵ_{r} is a small stability constant. By cancelling per-patch amplitude, this loss forces the encoder to represent the inter-patch fading structure and correlations rather than the sample-level power that dominates the raw signal. It also removes a trivial shortcut in which the network minimizes the loss by predicting patch means in raw space. The price is that dividing out per-patch amplitude discards the large-scale fading signature (path loss and shadowing) that other downstream tasks rely on. This motivates the auxiliary scale loss that we introduce next.

We recover the signal with an auxiliary scale loss. For each raw patch, we form the target $\mathbf{s}_{b,i} = [\mu_{b,i}, \log(\sigma^2_{b,i} + \epsilon_{\mathrm{s}})]^\top$, representing the variance in the log scale for numerical stability. We ask the model to predict $\mathbf{s}_{b,i}$ from both encoder and decoder token features. The encoder-side predictions $\hat{\mathbf{s}}^{\mathrm{enc}}_{b,i}$ are self-supervised on the visible set $\mathcal{K}$. Therefore, the encoder itself learns to carry large-scale statistics in its latent. The predictions on the decoder-side, $\hat{\mathbf{s}}^{\mathrm{dec}}_{b,i}$, are self-supervised on the masked set $\mathcal{M}$. As a result, the reconstruction path also recovers the scale. The two terms are

$$\begin{aligned} \mathcal{L}_{\mathrm{scale,enc}} &= \frac{1}{|\mathcal{K}|} \sum_{(b,i)\in\mathcal{K}} \left\| \hat{\mathbf{s}}^{\mathrm{enc}}_{b,i} - \mathbf{s}_{b,i} \right\|_2^2, \\ \mathcal{L}_{\mathrm{scale,dec}} &= \frac{1}{|\mathcal{M}|} \sum_{(b,i)\in\mathcal{M}} \left\| \hat{\mathbf{s}}^{\mathrm{dec}}_{b,i} - \mathbf{s}_{b,i} \right\|_2^2, \end{aligned} \tag{8}$$

and the full pretraining objective is

$$\mathcal{L}_{\mathrm{pretrain}} = \mathcal{L}_{\mathrm{recon}} + \lambda_{\mathrm{enc}}\mathcal{L}_{\mathrm{scale,enc}} + \lambda_{\mathrm{dec}}\mathcal{L}_{\mathrm{scale,dec}}. \tag{9}$$

3.5. Noise-robust pretraining curriculum

A noise curriculum corrupts the sparse masked input with AWGN during pretraining while the reconstruction and scale targets are computed against the clean channel. This prepares the encoder for deployment, where it processes sparse pilot-pattern inputs under the corresponding noise regime.

Concretely, let $e \in \{0, \ldots, E-1\}$ index the pretraining epoch, let s_0 denote the initial lower bound for the SNR range in dB, and let $s_{\max}$ denote the upper bound in dB. The lower bound follows a cosine schedule that anneals from s_0 down to 0 dB,

$$s_{\min}(e) = \frac{s_0}{2}\left(1 + \cos\left(\frac{\pi e}{E-1}\right)\right), \tag{10}$$

so that early epochs concentrate on higher-SNR observations and later epochs progressively expose the model to lower SNRs, widening the pretraining distribution as the encoder stabilizes.

We then sample $\mathrm{SNR}_{b,\mathrm{dB}} \sim \mathcal{U}[s_{\min}(e), s_{\max}]$, and the corresponding linear SNR together with the measured per-sample channel power $P_b = \mathrm{mean}(|\mathbf{H}_b|^2)$ sets the noise variance $\sigma_b^2 = P_b/\mathrm{SNR}_{b,\mathrm{lin}}$. Circularly symmetric complex Gaussian noise $\mathbf{N}_b \sim \mathcal{CN}(0, \sigma_b^2)$ is then added to produce the corrupted input $\tilde{\mathbf{H}}_b = \mathbf{H}_b + \mathbf{N}_b$ that the encoder actually consumes, while the reconstruction and scale losses are evaluated against the statistics computed from the clean $\mathbf{H}_b$.

4. Experiments

We pretrain PilotWiMAE on a ray-tracing channel dataset at 3.5 GHz (438,434 training samples), then evaluate transfer without task-specific fine-tuning on held-out test data (54,466 in-distribution (ID) samples and 17,466 out-of-distribution (OOD) samples, both at 28 GHz). Our evaluation includes ID and OOD settings, covering frequency mismatch alone (ID, 3.5 to 28 GHz on pretraining cities) and combined frequency-plus-city mismatch (OOD, 3.5 to 28 GHz on a held-out city), and reports performance on cross-frequency beam selection and channel characterization. The details of the protocol are provided in Appendix C.

4.1. Dataset

We create a ray-tracing channel dataset using our in-house generation pipeline, built on Sionna, across urban deployment scenarios. Each scenario uses six base stations and includes both LoS and NLoS links, with channel tensors generated at 3.5 GHz and 28 GHz under a shared simulation protocol. The pretraining split uses Boston, New York City, San Francisco, and Chicago. For evaluation, we use unseen channels from these same cities for ID testing, and we use Los Angeles as a held-out city for OOD testing. Combined with the 3.5 to 28 GHz carrier shift, this setup isolates frequency-only transfer (ID) from frequency-plus-scene transfer (OOD). Additional details of the generation and split are presented in Appendix A.

4.2. Pretraining

We pretrain the FST encoder at 3.5 GHz using aggressive factorized masking ($T_{\mathrm{k}} = 2$, $\rho_{\mathrm{k}} = 0.1$, about 99% overall masking), model dimension $d = 128$, and a two-phase curriculum over 600 total epochs. The pretraining decoder is always a JST decoder. Phase 2 enables the auxiliary scale objectives and noise-robust corruption. The complete architectural, optimization, and curriculum hyperparameters are provided in Appendix D (Table 3).

4.3. Tasks

We evaluate two downstream tasks under the same cross-band transfer protocol (3.5 to 28 GHz): cross-frequency beam selection and channel characterization (LoS/NLoS classification). Beam selection tests whether the learned representation preserves directional structure across bands, while channel characterization tests whether it preserves propagation-state semantics under frequency-dependent channel statistics. For both tasks, we use frozen pretrained representations and kNN evaluation without task-specific fine-tuning. For beam selection, labels are defined using a DFT codebook. Full task definitions, label construction, and codebook-generation details are provided in Appendix C.

4.4. Results

We compare against supervised and self-supervised baselines under a common kNN protocol (Appendix C). The

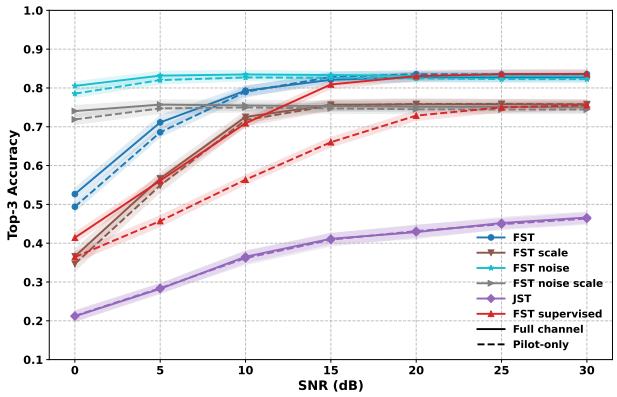

Figure 2. Cross-frequency beam selection (top-3 accuracy) in-distribution at 28 GHz with codebook size $M = 128$.

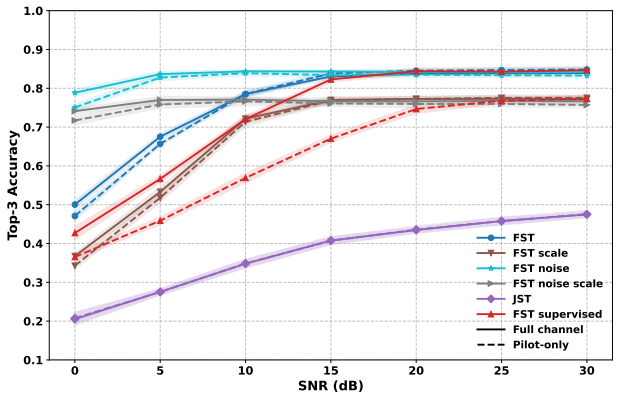

Figure 3. Cross-frequency beam selection (top-3 accuracy) out-of-distribution at 28 GHz with codebook size $M = 128$.

supervised baseline uses a factorized encoder with a linear classifier trained on full-channel task labels with cross-entropy loss. For representation evaluation, we discard the linear classifier and apply kNN to mean-pooled encoder outputs, as done for all models. For self-supervised variants, we report a standard JST encoder and our FST encoder. We also report ablations that progressively add the auxiliary scale loss and noise-robust pretraining to FST (FST, FST+scale, FST+noise, FST+noise+scale). Across all plots, curves show mean performance and shaded regions show $\pm$ one standard deviation over folds.

Across beam selection and channel characterization, the factorized encoder family is consistently more robust than JST under both ID and OOD transfers. Noise-robust pretraining improves stability across SNR, and the auxiliary scale objective is most beneficial for channel characterization, where large-scale fading cues are directly informative. Its effect on beam selection is smaller. Overall, FST+noise+scale provides the best trade-off across tasks while pilot-only inference remains competitive with full-channel inputs despite using a substantially smaller observation space.

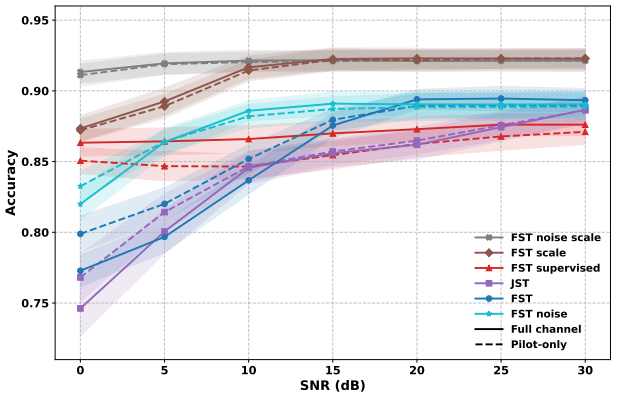

Figure 4. Channel characterization (LoS accuracy) in-distribution at 28 GHz.

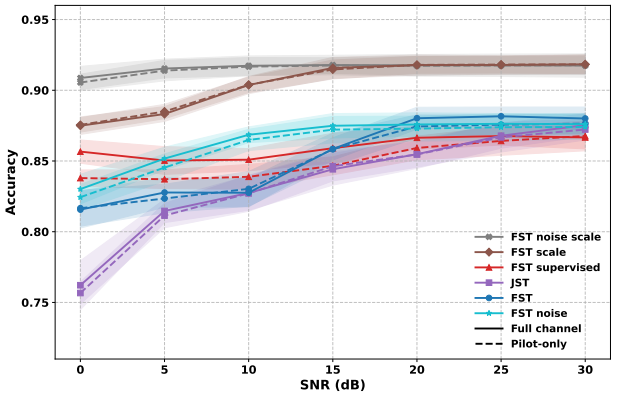

Figure 5. Channel characterization (LoS accuracy) out-of-distribution at 28 GHz.

5. Conclusion

PilotWiMAE demonstrates that self-supervised wireless channel representation learning can be both robust and deployment-aware by design: pilot-native inputs avoid unrealistic full-CSI assumptions, and factorized attention improves transfer under frequency shift, including to a held-out city, while reducing inference burden. With self-supervised pretraining at 3.5 GHz, the learned representations transfer to 28 GHz for beam selection and channel characterization without task-specific fine-tuning, with strong performance under both ID and OOD evaluation.

The ablations show that noise-robust pretraining is key for stability, and that auxiliary scale supervision is particularly useful for channel-state semantics. These results suggest a practical recipe for future wireless representation learning: combine structure-aware encoders with deployment-matched corruption and task-aligned pretraining objectives. Extending this recipe to broader pilot patterns, frequencies, and system-level latency profiling is a natural next step.

References

3GPP. NR; Physical Channels and Modulation. Technical Specification TS 38.211, 3rd Generation Partnership Project (3GPP), March 2026. URL https://www.3gpp.org/dynareport/38211.htm. V19.3.0.

Alikhani, S., Charan, G., and Alkhateeb, A. LWM: A pre-trained wireless foundation model for universal feature extraction. In *2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, pp. 1–6, May 2025. doi: 10.1109/ICMLCN64995.2025.11140266.

Alikhani, S., Malhotra, A., Hamidi-Rad, S., and Alkhateeb, A. LWM-Temporal: Sparse spatio-temporal attention for wireless channel representation learning, 2026. URL https://arxiv.org/abs/2603.10024.

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. ViViT: A video vision transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6816–6826, October 2021. doi: 10.1109/ICCV48922.2021.00676.

Bello, P. Characterization of randomly time-variant linear channels. *IEEE Transactions on Communications Systems*, 11(4):360–393, December 1963. ISSN 1558-2647. doi: 10.1109/TCOM.1963.1088793.

Bertasius, G., Wang, H., and Torresani, L. Is space-time attention all you need for video understanding? In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 813–824. PMLR, July 2021. URL https://proceedings.mlr.press/v139/bertasius21a.html.

Coleri, S., Ergen, M., Puri, A., and Bahai, A. Channel estimation techniques based on pilot arrangement in OFDM systems. *IEEE Transactions on Broadcasting*, 48(3):223–229, September 2002. ISSN 1557-9611. doi: 10.1109/TBC.2002.804034.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (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/.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Edfors, O., Sandell, M., van de Beek, J.-J., Wilson, S., and Borjesson, P. OFDM channel estimation by singular value decomposition. *IEEE Transactions on Communications*, 46(7):931–939, July 1998. ISSN 1558-0857. doi: 10.1109/26.701321.

Guler, B., Geraci, G., and Jafarkhani, H. A multi-task foundation model for wireless channel representation using contrastive and masked autoencoder learning. *IEEE Journal on Selected Areas in Communications*, 44:4489–4504, 2026. doi: 10.1109/JSAC.2026.3677157.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15979–15988, Los Alamitos, CA, USA, June 2022. IEEE Computer Society. doi: 10.1109/CVPR52688.2022.01553. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01553.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Vinyals, O., Rae, J. W., and Sifre, L. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Jiang, J., Yu, W., Li, Y., Gao, Y., and Xu, S. A MIMO wireless channel foundation model via CIR-CSI consistency. In *2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, pp. 1–6, May 2025. doi: 10.1109/ICMLCN64995.2025.11140262.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Liu, B., Liu, X., Gao, S., Cheng, X., and Yang, L. LLM4CP: Adapting large language models for channel prediction. *Journal of Communications and Information Networks*, 9 (2):113–125, 2024. doi: 10.23919/JCIN.2024.10582829.

Liu, B., Gao, S., Liu, X., Cheng, X., and Yang, L. WiFo: Wireless foundation model for channel prediction. *Science China Information Sciences*, 68(6): 162302, May 2025a. ISSN 1869–1919. doi: 10.1007/s11432-025-4349-0.

Liu, X., Gao, S., Liu, B., Cheng, X., and Yang, L. WiFo-CF: Wireless foundation model for csi feedback, 2025b. URL https://arxiv.org/abs/2508.04068.

Matz, G. and Hlawatsch, F. Chapter 1 - fundamentals of time-varying communication channels. In Hlawatsch, F. and Matz, G. (eds.), *Wireless Communications Over Rapidly Time-Varying Channels*, pp. 1–63. Academic Press, Oxford, 2011. ISBN 978-0-12-374483-8. doi: https://doi.org/10.1016/B978-0-12-374483-8.00001-7. URL https://www.sciencedirect.com/science/article/pii/B9780123744838000017.

Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B., and Schmidt, L. Measuring robustness to natural distribution shifts in image classification. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Wang, Y., Sun, L., Yang, T., Shi, Y., Elkashlan, M., and Tang, X. Filter-and-attend: Wireless channel foundation model with noise-plus-interference suppression structure, 2026. URL https://arxiv.org/abs/2509.15993.

Yang, T., Zhang, P., Zheng, M., Shi, Y., Jing, L., Huang, J., and Li, N. WirelessGPT: A generative foundation model for multi-task integrated sensing and communication. *IEEE Journal on Selected Areas in Communications*, 44:2259–2273, 2026. ISSN 1558-0008. doi: 10.1109/JSAC.2025.3640156.

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. Scaling vision transformers. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1204–1213, 2022. doi: 10.1109/CVPR52688.2022.01179.

A. Dataset details

Table 1 reports the number of channels per city in the pretraining split and in the 28 GHz evaluation splits. We use 10% of the training set to validate the model during pretraining.

Table 1. Per-city channel counts used in this paper.

City	Train (3.5 GHz)	Test ID (28 GHz)	Test OOD (28 GHz)
Boston	152,675	18,423	–
New York City	92,153	8,130	–
San Francisco	94,262	17,755	–
Chicago	99,344	10,158	–
Los Angeles	–	–	17,466
Total	438,434	54,466	17,466

Table 2. Sionna-based dataset generation parameters used across cities. Channel tensors are globally normalized by the dataset-level reference power P_{ref} (Section 3) prior to use.

Parameter	Value
BS sectors / mechanical downtilt	6 sectors / 10°
BS transmit power	43 dBm per array
TX array	4×8, vertical polarization, 0.5λ spacing
TX element pattern	TR 38.901
RX array	1×1, isotropic, vertical polarization
UE height offset	1.5 m above ground
OFDM grid	14 symbols, 32 subcarriers
Subcarrier spacing	30 kHz

B. Factorized space-time attention schematic

Figure 6 renders the FST attention wiring for Section 3 (Eqs. (5)–(6)).

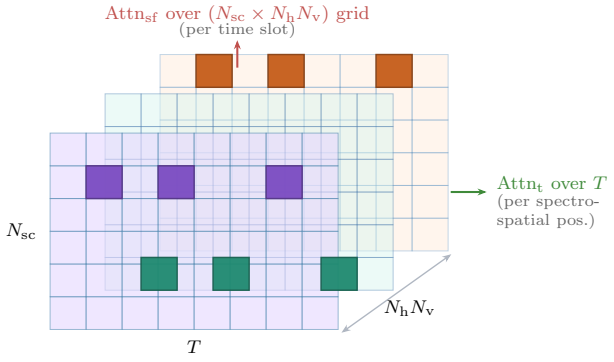

Figure 6. WSSUS-motivated factorized space-time attention. Temporal attention Attn_{t} is applied across OFDM symbols at fixed spectro-spatial positions, followed by spectro-spatial attention Attn_{sf} across subcarriers and antenna dimensions at fixed time indices. Attention is computed only over visible patches, highlighted in color. The figure illustrates a masking pattern with 3 visible time indices out of 8 and 3 visible spatial positions out of 18.

C. Task details

This appendix consolidates the task setup used in Section 4.

C.1. Shared transfer protocol

Both downstream tasks use the same frozen-feature transfer protocol:

- The representation encoder is pretrained at 3.5 GHz and then frozen.
- Evaluation is performed at 28 GHz without task-specific fine-tuning.
- Features are evaluated with a common kNN protocol for all compared methods (10 disjoint folds), using mean-pooled encoder representations.
- In each fold, kNN is fit on 90% of samples and evaluated on the held-out 10%.
- We report mean and standard deviation over cross-validation folds.

The evaluation includes both in-distribution (ID) and out-of-distribution (OOD) settings, covering frequency mismatch alone (ID) and combined frequency-plus-city mismatch (OOD) under the same split protocol described in Appendix A.

kNN details. We use $k = 20$ with cosine-distance-based weighted voting. Mean-pooled encoder features are passed to kNN without external feature normalization. For cosine distance, L2 normalization is applied internally by the kNN evaluator. For all models and both full-channel and pilot-only inputs, mean pooling over encoded patches yields a $d = 128$ representation.

C.2. Pilot observation pattern used at inference

For pilot-only evaluation, non-pilot entries are masked and only the fixed pilot pattern is retained:

- OFDM-symbol indices: $\{2, 11\}$.
- Subcarrier indices: $\{0, 1, 2, 3, 8, 9, 10, 11, 16, 17, 18, 19, 24, 25, 26, 27\}$.

This corresponds to 32 pilot resource elements out of $14 \times 32 = 448$ total resource elements, i.e., an effective pilot ratio of $32/448 = 0.0714$ (about 7.14%).

Figure 7 visualizes this fixed pattern on the 14×32 OFDM resource grid (OFDM symbol vs. subcarrier): highlighted cells are pilot resource elements observed at all antennas. All other positions are masked for pilot-only evaluation.

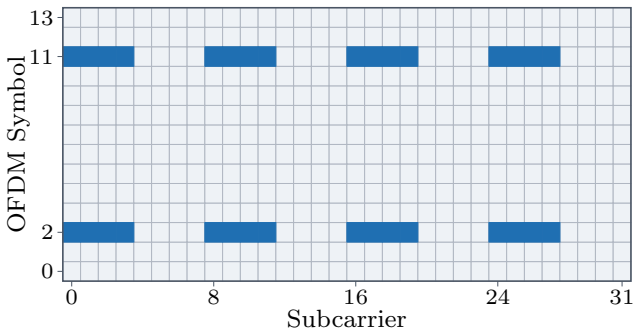

Figure 7. Pilot resource elements (highlighted) on the 14×32 time–frequency grid used at inference. Axes index OFDM symbols $t \in \{0, \ldots, 13\}$ and subcarriers $f \in \{0, \ldots, 31\}$.

C.3. Task 1: Cross-frequency beam selection

Beam selection evaluates whether pretrained representations preserve directional structure across frequency bands.

- Labels are constructed from a DFT beam codebook.

- We consider multiple codebook sizes. Representative plots in the main paper report codebook size $M = 128$.
- Performance is reported as top-k beam-selection accuracy (top-3 in the reported figures).

For a UPA with horizontal and vertical dimensions $(N_{\mathrm{h}}, N_{\mathrm{v}})$, codewords are built as Kronecker products of 1D steering vectors:

$$\mathbf{w}_{m_{\mathrm{h}},m_{\mathrm{v}}} = \mathbf{a}_{\mathrm{v}}(m_{\mathrm{v}}) \otimes \mathbf{a}_{\mathrm{h}}(m_{\mathrm{h}}), \tag{11}$$

with total codebook size $M = O_{\mathrm{h}}N_{\mathrm{h}}\, O_{\mathrm{v}}N_{\mathrm{v}}$ under oversampling factors $(O_{\mathrm{h}}, O_{\mathrm{v}})$. Given channel vectors $\mathbf{h}_{t,f}$, a single beam label per frame is assigned by maximizing average beam gain over subcarriers and slots:

$$m^{\star} = \arg \max_{m \in \{0,\ldots,M-1\}} \frac{1}{T\, N_{\mathrm{sc}}} \sum_{t=0}^{T-1} \sum_{f=0}^{N_{\mathrm{sc}}-1} \left| \mathbf{w}_m^{\mathrm{H}} \mathbf{h}_{t,f} \right|^2, \tag{12}$$

since the angular structure is approximately constant within an OFDM frame. For the reported codebooks, we use $(N_{\mathrm{h}}, N_{\mathrm{v}}) = (8, 4)$ with $(O_{\mathrm{h}}, O_{\mathrm{v}}) = (2, 2)$ for $M = 128$.

C.4. Task 2: Channel characterization

Channel characterization evaluates whether pretrained representations preserve propagation-state semantics under the same 3.5 to 28 GHz transfer.

- The task is binary LoS/NLoS classification.
- Performance is reported as LoS classification accuracy in both ID and OOD settings.

C.5. Compared methods under the common protocol

All methods are evaluated through the same frozen-feature kNN interface:

- **Supervised baseline:** factorized encoder with a linear classifier trained with full-channel task labels (cross-entropy). For kNN evaluation, the linear classifier is discarded and mean-pooled encoder representations are used.
- **Self-supervised baselines:** JST and FST encoders.
- **FST ablations:** FST, FST+scale, FST+noise, and FST+noise+scale.

For beam selection, supervised models are trained separately per codebook size. Across both tasks, the pilot-only supervised curves are produced by masking non-pilot entries only at inference while keeping the supervised training regime unchanged.

D. Pretraining hyperparameters

Table 3 lists the detailed hyperparameters for FST-based pretraining. Tables 4 and 5 list the detailed hyperparameters for JST-based pretraining and FST-based supervised training.

E. Computational complexity

This appendix reports the training and inference cost of PilotWiMAE and the supervised baseline, profiled with the protocol implemented in our profiler. All measurements use a single NVIDIA RTX A4000 GPU with PyTorch and mixed precision. Training cost is measured at training batch size $B_{\mathrm{tr}} = 256$ (Table 6), and inference cost is measured at inference batch size $B_{\mathrm{inf}} = 32$, averaged over 100 timed repeats after warmup (Table 7).

Asymptotic joint vs. factorized attention. Joint self-attention over the full grid of length TN_{sf} has complexity $\mathcal{O}((TN_{\mathrm{sf}})^2 d)$ (Vaswani et al., 2017). The factorized encoder applies temporal attention independently at each fixed spectro-spatial index (N_{sf} tensors of length T), then spectro-spatial attention independently at each time index (T tensors of length N_{sf}). The dominant terms are thus $\mathcal{O}(T^2 N_{\mathrm{sf}}\, d) + \mathcal{O}(N_{\mathrm{sf}}^2 T\, d) = \mathcal{O}(TN_{\mathrm{sf}}(T + N_{\mathrm{sf}})\, d)$. Relative to dense joint attention, the leading pairwise cost tightens by a factor of order $\frac{TN_{\mathrm{sf}}}{T+N_{\mathrm{sf}}}$, which exceeds an order of magnitude for our full-channel token grids and improves as either T or N_{sf} grows. Pilot-only inference further lowers cost by shortening the effective sequences.

Table 3. PilotWiMAE detailed pretraining parameters.

Hyperparameter	Value
Encoder type	FST
Input shape	$(14, 32, 32)$
Patch shape	$(1, 4, 4)$
Embedding	Linear
Positional encoding	Sinusoidal concatenative
Positional encoding scale	$\alpha_{\mathrm{pe}} = 0.01$ at the start
Factorized masking	$T_{\mathrm{k}} = 2$, $\rho_{\mathrm{k}} = 0.1$ (keep fractions, about 99% overall masking)
Model dimension	$d = 128$
FFN expansion	$\kappa_{\mathrm{ff}} = 4$
Encoder depth / heads	3 factorized block-pairs / 8 heads
Decoder type	JST decoder
Decoder depth / heads	2 layers / 4 heads
Optimizer	AdamW, betas $(0.9, 0.999)$, weight decay 0.005
Epochs	300 (phase 1) + 300 (phase 2)
LR schedule	Cosine, warmup 10
Phase-1 LR	$\eta_{\mathrm{start}} = 5 \times 10^{-4}$, $\eta_{\mathrm{min}} = 5 \times 10^{-6}$
Phase-2 LR	$\eta_{\mathrm{start}} = 10^{-4}$, $\eta_{\mathrm{min}} = 10^{-6}$
Auxiliary scale losses (phase 2)	$\lambda_{\mathrm{enc}} = \lambda_{\mathrm{dec}} = 0.05$
Noise setup	$s_0 = 40\,\mathrm{dB}$, $s_{\max} = 40\,\mathrm{dB}$
Batch size	512
Precision / clipping	Mixed precision, gradient clipping at 1.0

Per-sample latency. We report inference cost per sample, obtained by dividing the per-batch latency by B_{inf}. This amortized convention matches our profiling pipeline and is appropriate as a throughput-style figure of merit at moderate-to-large batches, where the GPU is well-utilized and per-batch latency scales approximately linearly with B_{inf}. At very small batch sizes, fixed kernel-launch and memory-traffic overhead become non-negligible, so the latency at $B = 1$ can exceed the amortized per-sample latency.

Discussion. Several observations follow from these tables. First, self-supervised pretraining of either the FST or JST encoder is roughly $3\times$ cheaper per epoch than end-to-end supervised training of the same FST backbone, because pretraining processes only the visible token subset (factorized keep set or random keep set at 99% overall masking) while the supervised baseline always sees the full grid. Second, at inference, the FST encoder benefits decisively from pilot-only input: its per-sample latency drops from $0.70\,\mathrm{ms}$ in the full-channel case to $0.15\,\mathrm{ms}$ in the pilot-only case, a $4.6\times$ reduction that is consistent with the reduced sequence length combined with the $\mathcal{O}(TN_{\mathrm{sf}}(T + N_{\mathrm{sf}})d)$ scaling of factorized attention. Third, the JST encoder is faster than FST in the pilot-only regime because, after 99% masking, its sequence length is small enough that the quadratic attention cost is no longer the bottleneck and the simpler block structure dominates. However, JST is the slowest of all configurations on the full grid ($1.83\,\mathrm{ms}$ per sample), where its quadratic attention is exposed.

Table 4. JST pretraining configuration used as the self-supervised baseline.

Parameter	Value
Encoder type	JST
Input shape	$(14, 32, 32)$
Patch shape	$(1, 4, 4)$
Embedding	Linear
Positional encoding	Sinusoidal concatenative
Positional encoding scale	$\alpha_{\mathrm{pe}} = 0.01$ at the start
Masking	Random, mask ratio 0.95
Encoder dimension	$d = 128$
Encoder depth / heads	6 layers / 8 heads
Decoder depth / heads	2 layers / 4 heads
Optimizer	AdamW, $\beta = (0.9, 0.999)$, weight decay 0.005
Epochs	300
Batch size	512
LR schedule	Cosine, warmup 10, $\eta_{\min} = 10^{-5}$
Initial LR	$\eta_{\mathrm{start}} = 10^{-3}$
Loss / patch normalization	MSE with normalized patch loss
Precision / clipping	Mixed precision, gradient clipping at 1.0

Table 5. Supervised baseline configurations (factorized encoder backbone + linear classification head).

Parameter	Beam prediction	LoS/NLoS
Encoder backbone	FST	FST
Input shape	$(14, 32, 32)$	$(14, 32, 32)$
Patch shape	$(1, 4, 4)$	$(1, 4, 4)$
Encoder dimension	$d = 128$	$d = 128$
Encoder depth / heads	3 layers / 8 heads	3 layers / 8 heads
Linear head matrix	$d \times M$	$d \times 2$
Optimizer	AdamW, $\beta = (0.9, 0.999)$, wd 0.05	AdamW, $\beta = (0.9, 0.999)$, wd 0.005
Epochs	200	200
Batch size	256	256
LR schedule	Cosine, warmup 10, $\eta_{\min} = 5 \times 10^{-6}$	Cosine, warmup 10, $\eta_{\min} = 5 \times 10^{-6}$
Initial LR	5×10^{-4}	5×10^{-4}
Loss	Cross-entropy	Cross-entropy
Precision / clipping	Mixed precision, gradient clipping at 1.0	Mixed precision, gradient clipping at 1.0

Table 6. Per-epoch training cost on a single NVIDIA RTX A4000 at training batch size $B_{\mathrm{tr}} = 256$. The PilotWiMAE rows report the self-supervised pretraining cost of the FST and JST encoders under 99% and 95% masking regime, respectively. The supervised row reports the end-to-end training cost of the FST backbone with a task-specific linear head on full channels (cross-entropy). FLOPs are reported in petaFLOPs (P).

Method	Encoder	Trainable params	Train FLOPs / epoch	Time / epoch (s)
PilotWiMAE (self-supervised)	FST	1,594,658	2.017 P	208.01
PilotWiMAE (self-supervised)	JST	1,594,658	2.115 P	204.43
Supervised baseline	FST	1,210,369	2.656 P	653.69

Table 7. Inference cost on a single NVIDIA RTX A4000 at inference batch size $B_{\mathrm{inf}} = 32$, averaged over 100 timed repeats after warmup. "Pilot only" uses the inference pilot pattern of Appendix C, while "Full channel" uses the full $14 \times 32 \times 32$ grid. Per-sample latency is the measured per-batch latency divided by B_{inf}, and standard deviations are likewise scaled by $1/B_{\mathrm{inf}}$.

Encoder	Input	Encoder params	Per-sample latency (ms)
FST	Pilot only	1,594,658	0.153 ± 0.007
JST	Pilot only	1,594,658	0.073 ± 0.000
FST	Full channel	1,594,658	0.698 ± 0.007
JST	Full channel	1,594,658	1.833 ± 0.015