KRONDPBL: SELF-SUPERVISED PATTERN–NOISE DISENTANGLEMENT TOWARDS TRANSFERABLE WIFI CSI REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

WiFi Channel State Information (CSI) has become a promising sensing modality for device-free human activity recognition. However, raw CSI measurements are contaminated by hardware impairments and environmental noise, leading deep neural networks to learn spurious, noise-dependent features that generalize poorly across domains. We propose KronDPBL, a self-supervised representation learning framework that explicitly disentangles recurring signal patterns from segment-specific noise to learn robust and transferable CSI representations. KronDPBL introduces a distribution-guided cross-attention mechanism that separates persistent signal motifs from transient perturbations, together with an asymmetric attention design that retains noise information in the key branch for contrastive similarity estimation while suppressing it in the value branch to synthesize noise-free pattern representations. To exploit the intrinsic structure of CSI measurements, KronDPBL further incorporates Kronecker-sum spatial encoding and a low-rank reconstruction objective, a parameter-efficient two-axis spatial prior into the learned representations. The model is trained with dual contrastive objectives that jointly enforce pattern invariance and noise disentanglement, applied to each corpus under a strict fold-wise protocol, with encoders frozen after pretraining, and producing representations that transfer effectively across sensing tasks. Extensive experiments on six public CSI benchmark datasets demonstrate that a frozen KronDPBL encoder without end-to-end fine-tuning achieves state-of-the-art performance on 11 of 13 downstream tasks, improving accuracy by up to 14.6% over prior methods. Furthermore, KronDPBL consistently exhibits superior robustness under severe signal perturbations, outperforming existing approaches by 5.6%–23.0%.

1 INTRODUCTION

WiFi CSI has emerged as an attractive medium for device-free human sensing. As people move through the environment, their motion perturbs the multipath propagation of the ambient WiFi signal, and the resulting fluctuations in CSI carry rich information about human activities, identities, and behaviors (Pu et al., 2013; Wang et al., 2014; 2015). Compared with camera-based or wearable-based sensing, CSI-based approaches are appealing because they require no line of sight, operate in the dark, do not capture visual imagery, and reuse already-deployed WiFi infrastructure, making them well suited to privacy-sensitive settings such as homes, hospitals, and elderly-care facilities (Liu et al., 2020; Yang et al., 2023). Driven by these advantages, deep learning has been widely adopted to map raw CSI to a variety of sensing tasks, including activity recognition, person identification, people counting, and fall detection (Li et al., 2021; Liang et al., 2026; Liu et al., 2025).

Despite recent advances, existing CSI models remain overwhelmingly task-specific, trained from scratch on isolated datasets and hardware platforms (Guo et al., 2019; Ma et al., 2018; Gu et al., 2022; Zhu et al., 2022; Li et al., 2021; Chen et al., 2023). This paradigm scales poorly: labeled CSI is scarce, rendering supervised networks prone to overfitting room-, device-, and subject-specific artifacts rather than learning transferable features. While cross-domain methods attempt to mitigate environmental shifts (Zhang et al., 2022b;a; Fan et al., 2025; Zhao et al., 2025), self-supervised learning (SSL) faces a unique bottleneck: raw measurements are heavily corrupted by hardware imperfections, carrier frequency offsets, and environmental interference (Halperin et al., 2011; Zeng

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.

et al., 2019), burying task-relevant motion patterns under non-generalizable noise. Existing SSL approaches (masked transformers) treat CSI as generic sequences and reconstruct raw signals (Zhao et al., 2024; 2026; Xiao et al., 2024), failing to explicitly decouple structural dynamics from measurement artifacts while leaving the natural two-dimensional antenna–subcarrier grid unexploited.

Motivated by these observations, we argue that the key to better CSI representation is to *explicitly disentangle* the recurring structural content of the signal from its segment-specific noise, while exploiting the two-dimensional structure of each CSI frame. Our insight is that informative CSI arises from a limited set of recurring motifs, while noise is unstructured and segment-specific. We instantiate this in **KronDPBL**, a self-supervised framework for transferable CSI representation learning. At its core, a CSI Pattern Encoder(CPE) extracts a small set of pattern vectors and noise vectors through cross-attention with learnable pattern and noise distributions, and a CSI Noise Canceller(CNC) then suppresses the noise components via an asymmetric attention mechanism, yielding progressively cleaner features. To inject physically meaningful inductive biases, we exploit Kronecker structure of antenna–subcarrier grid through a parameter-efficient Kronecker spatial encoding and a low-rank Kronecker-factorized reconstruction head. The encoder is pretrained without labels by combining a denoising reconstruction objective with two contrastive objectives structure the pattern and noise latent spaces; the encoder is frozen and paired with downstream task classifiers.

We evaluate **KronDPBL** on six public WiFi CSI datasets spanning diverse hardware platforms, antenna configurations, and four downstream tasks: activity recognition, people counting, person identification, and fall detection. Extensive experiments show that each encoder pretrained once per fold on unlabeled training data and then frozen consistently outperforms strong task-specific baselines and prior self-supervised methods, and transfers robustly across recordings, environments, and devices. In summary, our contributions are:

- We identify noise as central obstacle to transferable CSI representation, and propose **KronDPBL**, a self-supervised framework that explicitly disentangles recurring structural patterns from segment-specific noise via learnable distributions and cross-attention denoising.
- We embed the two-dimensional antenna-subcarrier topology of CSI into the model through a Kronecker spatial encoding and a Kronecker-factorized low-rank reconstruction head, turning the spatial low-rank property of CSI into a self-supervised reconstruction target.
- Extensive experiments show that decoupled representation is robust under signal degradation and is transferred in various real-world tasks, achieving SOTA performance with frozen encoder, establishing a strong benchmark for transferable CSI representation learning.

2 Related Work

2.1 WiFi CSI Sensing

Deep learning now dominates CSI-to-label mapping (Ma et al., 2019; Yang et al., 2023). Approaches have evolved from hand-crafted features and image-style CNNs (Guo et al., 2019; Ma et al., 2018; Wang et al., 2015) to attention-based temporal models (Gu et al., 2022; Zhu et al., 2022; Li et al., 2021; Chen et al., 2023), and have been extended to gait identification, phase-based sensing, untrimmed detection, and generic patch architectures (Liang et al., 2026; Chen et al., 2025; Liu et al., 2025; Dosovitskiy et al., 2021; Nie et al., 2023). Despite strong in-domain accuracy, these models are typically trained from scratch on a single dataset or task and degrade across domains under noise. Traditional signal-level corrections such as phase calibration and CSI-ratio remain platform-specific and cannot remove segment-level environmental noise (Halperin et al., 2011; Zeng et al., 2019).

2.2 Cross-Domain and Self-Supervised CSI Learning

Cross-domain methods reduce environmental shifts via domain-independent motion features, cross-domain class prototypes, multi-source alignment, or prompt-based few-shot adaptation (Zhang et al., 2022b;a; Fan et al., 2025; Zhao et al., 2025). Self-supervised methods instead pretrain on unlabeled CSI, from BERT-style masked recovery to diffusion-augmented contrastive learning (Zhao et al., 2024; 2026; Xiao et al., 2024). Recent designs bypass raw reconstruction: CAPC uses uplink–downlink contrastive coding, CSI-JEPA predicts masked latents, and WiFo applies masked autoencoders to channel prediction (Barahimi et al., 2024; Luo et al., 2026; Liu et al., 2024). Benchmarks

reveal poor generalization across users, devices, and environments (Zhu et al., 2025). Yet raw reconstruction absorbs non-transferable noise, and latent methods lack explicit separation of recurring structure from segment-specific noise, leaving portable denoised representations unresolved.

3 METHODOLOGY

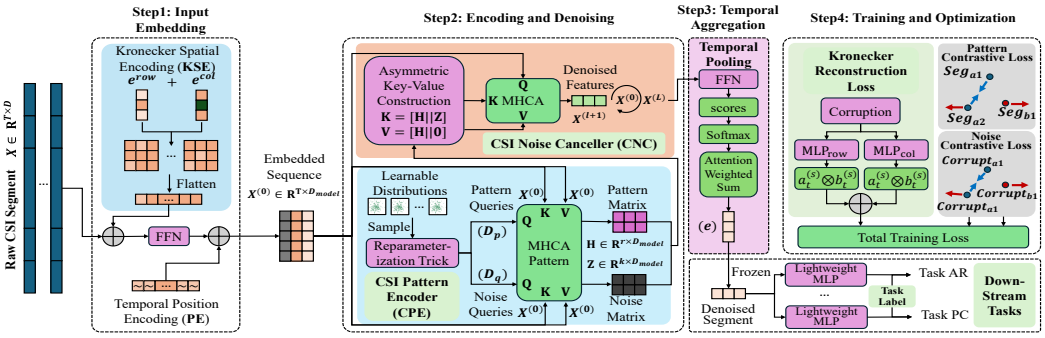

Figure 1: **KronDPBL Overview.** A raw CSI segment is augmented with Kronecker spatial encoding, projected through a feed-forward network, and processed by stacked CPE-CNC layers that extract pattern and noise representations via shared cross-attention with learnable distribution queries. Temporal aggregation produces a segment-level embedding, optimized end-to-end with a joint objective of Kronecker-factorized reconstruction and dual contrastive losses.

We consider CSI measurements collected over T timesteps across an antenna–subcarrier grid, flattened to dimension D. Each frame is organized as a $D_{\text{row}} \times D_{\text{col}}$ matrix with $D = D_{\text{row}} D_{\text{col}}$: a two-dimensional factorization motivated by this physical layout, whose exact shape is a design choice (analyzed in the experiments) since padding and flattening can depart from the raw antenna–subcarrier dimensions. An input segment $\mathbf{X} \in \mathbb{R}^{T \times D}$ captures channel magnitudes. Our goal is to map $\mathbf{X}$ into a compact representation $\mathbf{e} \in \mathbb{R}^{D_{\text{model}}}$ that retains task-relevant motion dynamics while discarding transient hardware and environmental noise.

As shown in Figure 1, KronDPBL operates in four steps. First, CSI segments are embedded with Kronecker spatial and temporal encodings. Second, stacked CPE–CNC layers extract patterns and suppress noise. Third, attention-weighted temporal pooling produces segment-level embeddings. Finally, reconstruction and dual contrastive losses guide self-supervised pretraining. The pretrained encoder is then frozen and paired with lightweight classifiers for downstream tasks.

3.1 INPUT EMBEDDING

The raw CSI segment $\mathbf{X} \in \mathbb{R}^{T \times D}$ is first augmented with a Kronecker spatial encoding that captures the antenna–subcarrier layout, then projected from D to D_{model} dimensions through a feed-forward network (FFN), and finally combined with temporal position encoding. We use Kronecker sum in the spatial encoding and multiplicative in the reconstruction head, a sum of rank-1 outer products.

Kronecker Spatial Encoding. Standard sinusoidal position encodings capture temporal ordering but are oblivious to the spatial adjacency structure of CSI features. Throughout this paper, the Kronecker (outer) product $\mathbf{a} \otimes \mathbf{b}$ of two vectors denotes the matrix with entries $a_i b_j$, so a large $D_{\text{row}} \times D_{\text{col}}$ matrix can be expressed through two small per-axis factors. We introduce a Kronecker-sum spatial encoding that assigns each position in the $D_{\text{row}} \times D_{\text{col}}$ grid a learnable scalar bias $\mathbf{P}_{i,j} = \mathbf{e}^{\text{row}}_i + \mathbf{e}^{\text{col}}_j$, where $\mathbf{e}^{\text{row}} \in \mathbb{R}^{D_{\text{row}}}$ and $\mathbf{e}^{\text{col}} \in \mathbb{R}^{D_{\text{col}}}$ are learnable vectors. This encoding is the Kronecker sum of the two per-axis vectors, $P = e_{\text{row}} \otimes \mathbf{1}_{D_{\text{col}}} + \mathbf{1}_{D_{\text{row}}} \otimes e_{\text{col}}$, the sum of two rank-1 terms, hence of rank at most two. It reduces spatial parameters from $D_{\text{row}} \times D_{\text{col}}$ to $D_{\text{row}} + D_{\text{col}}$ while retaining an explicit two-axis positional structure. The encoding is reshaped to match the input CSI dimension and added to the raw CSI, providing awareness of the two-dimensional physical layout

Temporal Position Encoding. The projected features are then augmented with a standard sinusoidal position encoding along the temporal axis, where even and odd dimensions use $\sin(\cdot)$ and

$\cos(\cdot)$ of exponentially increasing wavelengths $10000^{2i/D_{\text{model}}}$ with $i \in [0, D_{\text{model}}/2)$, respectively. This encoding is non-learnable and introduces no additional parameters. The resulting embedded sequence $\mathbf{X}^{(0)} \in \mathbb{R}^{T \times D_{\text{model}}}$ serves as the input to the denoising stack.

3.2 Encoding and Denoising

The encoding and denoising pipeline comprises L successive rounds, each with a CPE that extracts structured representations and a CNC that suppresses noise.

Pattern Distribution and Pattern Encoder. A central design principle of our framework is that CSI measurements across different timesteps and segments exhibit a limited set of recurring structural motifs, while noise manifestations are segment-specific and unstructured with frequent occurrence. To operationalize this insight, we maintain r learnable pattern distributions $\{\mathcal{P}_1, \ldots, \mathcal{P}_r\}$ and k learnable noise factor distributions $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_k\}$ in the latent space. Each distribution is parameterized as a diagonal Gaussian with learnable mean and log-variance: $\mathcal{P}_i = \mathcal{N}(\boldsymbol{\mu}_i, \text{diag}(\boldsymbol{\sigma}_i^2))$ and $\mathcal{Q}_j = \mathcal{N}(\boldsymbol{\nu}_j, \text{diag}(\boldsymbol{\tau}_j^2))$, where $\boldsymbol{\mu}_i, \boldsymbol{\nu}_j \in \mathbb{R}^{D_{\text{model}}}$ and $\boldsymbol{\sigma}_i^2, \boldsymbol{\tau}_j^2 \in \mathbb{R}^{D_{\text{model}}}$. At each forward pass, query vectors are sampled via reparameterization: $\mathbf{d}_p^{(i)} = \boldsymbol{\mu}_i + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}_i$ and $\mathbf{d}_q^{(j)} = \boldsymbol{\nu}_j + \boldsymbol{\epsilon} \odot \boldsymbol{\tau}_j$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the shared noise vector. The sampled pattern queries are collected into $\mathbf{D}_p \in \mathbb{R}^{r \times D_{\text{model}}}$ and the noise queries into $\mathbf{D}_q \in \mathbb{R}^{k \times D_{\text{model}}}$. At inference, the means are used directly for deterministic extraction. Given the input $\mathbf{X}^{(l-1)}$ of layer l, where $\mathbf{X}^{(0)}$ is the embedded sequence from the input embedding stage, the CPE extracts pattern and noise representations through multi-head cross-attention with shared projection weights as $\mathbf{H} = \text{MHCA}(\mathbf{D}_p, \mathbf{X}^{(l-1)}, \mathbf{X}^{(l-1)})$ and $\mathbf{Z} = \text{MHCA}(\mathbf{D}_q, \mathbf{X}^{(l-1)}, \mathbf{X}^{(l-1)})$, where $\text{MHCA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ denotes multi-head cross-attention with residual connection, layer normalization, and dropout. $\mathbf{H} \in \mathbb{R}^{r \times D_{\text{model}}}$ aggregates pattern-relevant features and $\mathbf{Z} \in \mathbb{R}^{k \times D_{\text{model}}}$ captures noise-related perturbations. Shared key-value projections encourage complementary attention across pattern and noise pathways.

CSI Noise Canceller. The CNC performs the denoising step by exploiting an asymmetric key-value construction. The concatenation of pattern and noise representations forms the key, providing the attention mechanism with the information needed to compute similarity scores against the current CSI features $\mathbf{X}^{(l-1)}$. The value, however, zeroes out all noise positions: $\mathbf{K} = [\mathbf{H} \| \mathbf{Z}]$ and $\mathbf{V} = [\mathbf{H} \| \mathbf{0}]$, where $[\cdot\|\cdot]$ denotes concatenation along the sequence dimension and $\mathbf{0} \in \mathbb{R}^{k \times D_{\text{model}}}$, yielding the denoised features $\mathbf{X}^{(l)} = \text{MHCA}(\mathbf{X}^{(l-1)}, \mathbf{K}, \mathbf{V})$. The set-to-zero mechanism ensures that any timestep whose features attend to a noise position receives a zero-valued contribution. The attention output, computed as a convex combination of the value vectors, is therefore always a weighted mixture of the pattern representations $\mathbf{H}$ alone. Noise information is structurally prevented from contributing. The residual connection then adds back the original features $\mathbf{X}^{(l-1)}$, preserving signal fidelity while the attention pathway handles denoising. This design cleanly separates denoising from representation: the key, $\mathbf{K}$, encodes *what* each component represents for similarity matching, while the value, $\mathbf{V}$, encodes *how* each component should contribute to the output. Each CPE–CNC pair constitutes one layer, and stacking L such layers enables progressive refinement: the partially denoised features $\mathbf{X}^{(l)}$ output by layer l serve as input to layer $l+1$. With each subsequent layer, the pattern queries $\mathbf{D}_p$ can leverage increasingly clean features to refine their representations, while the noise queries $\mathbf{D}_q$ adapt to capture residual perturbations that survived earlier denoising stages.

3.3 Temporal Aggregation

After the final CNC layer, the features $\mathbf{X}^{(L)}$ are pooled into a segment embedding $\mathbf{e} = \sum_{t=1}^{T} \alpha_t \mathbf{X}_t^{(L)}$ with attention weights $\alpha_t = \exp(\text{FFN}(\mathbf{X}_t^{(L)})) / \sum_{s=1}^{T} \exp(\text{FFN}(\mathbf{X}_s^{(L)}))$, where $\text{FFN}(\cdot)$ is a two-layer FFN mapping each timestep feature to a scalar relevance score. This learned pooling adaptively emphasizes the most informative frames.

3.4 Training and Optimization

The model is trained end-to-end with a joint objective combining Kronecker-factorized reconstruction and dual contrastive losses. Both are motivated by the natural factorization of CSI data into an antenna-subcarrier grid of shape $D_{\text{row}} \times D_{\text{col}}$.

Kronecker Reconstruction Head. $\mathbf{X}^{\text{clean}} \in \mathbb{R}^{T \times D}$ denotes original uncorrupted CSI segment. During training, denoised per-timestep features must be decoded to reconstruct $\mathbf{X}^{\text{clean}}$ from a corrupted input. A conventional approach would apply an unstructured linear projection from D_{model} to D, consuming $D_{\text{model}} \times D$ parameters. Recognizing that CSI measurements exhibit low-rank spatial structure, we instead employ a Kronecker-factorized decoder that reconstructs the $D_{\text{row}} \times D_{\text{col}}$ CSI matrix at each timestep t as a sum of S rank-1 outer products, $\hat{\mathbf{X}}_t = \sum_{s=1}^{S} \mathbf{a}_t^{(s)} \otimes \mathbf{b}_t^{(s)}$, where the row factors $\mathbf{a}_t^{(s)} \in \mathbb{R}^{D_{\text{row}}}$ and column factors $\mathbf{b}_t^{(s)} \in \mathbb{R}^{D_{\text{col}}}$ are predicted from the final-layer denoised feature $\mathbf{X}_t^{(L)}$ — the same per-timestep representations that temporal aggregation pools into $\mathbf{e}$ — through two independent two-layer MLPs, $\mathbf{A}_t = \text{MLP}_{\text{row}}(\mathbf{X}_t^{(L)})$ and $\mathbf{B}_t = \text{MLP}_{\text{col}}(\mathbf{X}_t^{(L)})$, with $\mathbf{A}_t \in \mathbb{R}^{D_{\text{row}} \times S}$ and $\mathbf{B}_t \in \mathbb{R}^{D_{\text{col}} \times S}$. The reconstruction is self-supervised: the loss applies an MSE between $\hat{\mathbf{X}}_t$ and $\mathbf{X}_t^{\text{clean}}$ (reshaped to $D_{\text{row}} \times D_{\text{col}}$), weighted by a binary mask $\mathbf{M}$ that assigns a weight of $w > 1$ to corrupted positions and 1 to clean ones, encouraging the decoder to focus on recovering the damaged regions. Six corruption types are applied: Gaussian noise, random masking, mixed, subcarrier dropout, temporal dropout, and Kronecker-structured perturbation. The reconstruction head is used only during pretraining and is discarded for downstream inference.

Contrastive Learning. We employ two fully unsupervised contrastive objectives to structure the pattern and noise latent spaces. For each training sample, an anchor–positive pair is built from two differently corrupted views of the same segment: the anchor applies one randomly selected corruption type, while the positive is first temporally downsampled to 85% of the timesteps and then corrupted independently. Jointly varying temporal resolution and noise distribution encourages features robust to both sampling-rate mismatch and device-specific impairments. Views of the same segment form a natural positive pair, views from other segments in the batch serve as negatives, and the pairing relies only on segment identity, never on labels.

Pattern Contrastive Loss. The pattern contrastive objective ensures that the segment embedding $\mathbf{e}$ from temporal aggregation is invariant to the specific corruption pattern and temporal downsampling applied to the input. Let $\mathbf{e}_i^a$ and $\mathbf{e}_i^p$ denote the representations of the anchor and positive views derived from the same original CSI segment i through independent corruptions. By construction, they form a natural positive pair since they originate from the same underlying signal. Views derived from different segments in the batch serve as negatives. The loss follows symmetric InfoNCE:

$$\mathcal{L}_{\text{pattern}} = -\frac{1}{2B} \sum_{i=1}^{B} \Big(\log \frac{\exp(\mathbf{e}_i^a \cdot \mathbf{e}_i^p / \tau)}{\sum_{j=1}^{B} \exp(\mathbf{e}_i^a \cdot \mathbf{e}_j^p / \tau)} + \log \frac{\exp(\mathbf{e}_i^p \cdot \mathbf{e}_i^a / \tau)}{\sum_{j=1}^{B} \exp(\mathbf{e}_i^p \cdot \mathbf{e}_j^a / \tau)} \Big), \tag{1}$$

where both representations are ℓ_2-normalized, τ is a temperature hyperparameter, and B is the batch size. This objective pulls together representations of differently corrupted views of the same underlying signal while pushing apart representations of different segments, encouraging the encoder to produce corruption-invariant embeddings.

Noise Contrastive Loss. The noise contrastive objective shapes the noise representations $\mathbf{Z}$ with two desiderata. First, noise representations extracted from different corrupted views of the same original segment should be similar, reflecting that they capture the same underlying noise characteristics of the recording environment. Second, noise representations should be well-separated from all pattern representations $\mathbf{H}$, preventing pattern information from leaking into the noise pathway. Let $\bar{\mathbf{z}} = \frac{1}{k} \sum_{j=1}^{k} \mathbf{Z}_j$ denote the mean-pooled noise representation. The loss is:

$$\mathcal{L}_{\text{noise}} = \underbrace{-\frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp(\bar{\mathbf{z}}_i^a \cdot \bar{\mathbf{z}}_i^p / \tau)}{\sum_{j=1}^{B} \exp(\bar{\mathbf{z}}_i^a \cdot \bar{\mathbf{z}}_j^p / \tau)}}_{\text{noise compactness}} + \lambda_{\text{sep}} \cdot \underbrace{\frac{1}{B} \sum_{i=1}^{B} \max_{1 \le m \le r} \text{sim}(\bar{\mathbf{z}}_i, \mathbf{h}_i^{(m)})}_{\text{pattern-noise separation}}, \tag{2}$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity and λ_{sep} controls the separation penalty. The first term, an InfoNCE loss on noise representations, encourages noise compactness across different corrupted views of the same underlying signal. The second term penalizes the maximum cosine similarity between pooled noise representation and any pattern representation within the same segment, enforcing that noise and pattern representations occupy distinct regions of the latent space.

Total Loss. Complete training objective combines reconstruction loss with two contrastive losses:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{recon}} \mathcal{L}_{\text{recon}} + \lambda_{\text{pattern}} \mathcal{L}_{\text{pattern}} + \lambda_{\text{noise}} \mathcal{L}_{\text{noise}}, \tag{3}$$

where λ_{recon}, λ_{pattern}, and λ_{noise} are hyperparameters balancing the three objectives. The reconstruction loss $\mathcal{L}_{\text{recon}} = \frac{1}{T}\sum_{t=1}^{T} \|\hat{\mathbf{X}}_t - \mathbf{X}_t^{\text{clean}}\|_2^2$ encourages recovery of the original segment, while $\mathcal{L}_{\text{pattern}}$ and $\mathcal{L}_{\text{noise}}$ structure the representation spaces without external labels. After training, the reconstruction head is discarded, and the KronDPBL encoder is frozen for downstream adaptation with lightweight task-specific classifiers. Hyperparameters are listed in Section 4.

4 EXPERIMENTS

4.1 EXPERIMENT SETTINGS

Datasets and Tasks: We use six public WiFi CSI datasets: CSI-80 Meneghello et al. (2023), WST Yang et al. (2020) (also named as Widar3.0;we use only its WiFi CSI modality), WiGesture Zhao et al. (2024), WiFall and WiCount Zhao et al. (2026), and CSI-Bench Zhu et al. (2025) with diverse hardware platforms (Broadcom BCM4366c0, Intel 5300, Atheros AR9580, HP HealthPod, ESP32), antenna-subcarrier configurations (from 1×52 to 4×242). We evaluate on four downstream tasks that span the canonical problem types in WiFi sensing: **Activity Recognition (AR)** captures motion dynamics, **People Counting (PC)** estimates spatial occupancy, **Person Identification (PI)** extracts fine-grained biometric signatures, and **Fall Detection (FD)** tests safety-critical event detection (Hard tier accuracy). Protocol abbreviations: W = within-environment 5-fold CV, C = cross-recording 4-fold CV, L = cross-location, T = tiered difficulty. Each setting is denoted task-N^p, where N is the number of classes and superscript p the protocol.

Preprocessing: Amplitude-only CSI is used. Z-score normalization is applied per subcarrier within each $T = 100$-frame segment, and each frame is zero-padded to the maximum per-frame dimension of its dataset. No test-time augmentation is used. Pretraining uses unlabeled training splits; downstream evaluation follows cross-recording cross-validation without temporal overlap.

Baselines: We compare KronDPBL against representative methods spanning five categories: (1) **shallow** baselines including MLP on CSI statistics and WiAR Guo et al. (2019) (hand-crafted features + linear classifier); (2) **convolutional** models including SignFi CNN Ma et al. (2018), CLAR Xiao et al. (2024), WiGr Zhang et al. (2022a), WiGRUNT Gu et al. (2022), Wi-ATCN Zhu et al. (2022), and HAR-SAnet Chen et al. (2023); (3) **recurrent** models including WiFiTAD Liu et al. (2025), Widar3.0 Zhang et al. (2022b), and GUARD Liang et al. (2026); (4) **Transformer**-based and **self-supervised** methods including ViT Dosovitskiy et al. (2021), CSI-BERT Zhao et al. (2024) (and its v2 variant Zhao et al. (2026)), THAT Li et al. (2021), and PatchTST Nie et al. (2023); and (5) **prompting** methods including Wi-Prompt Zhao et al. (2025). All baselines use official or best-setting reimplementations; all methods use identical splits and preprocessing, with KronDPBL pretrained only on unlabeled training data.

Hyperparameters: KronDPBL uses a frozen encoder with a 3-layer MLP and baselines are end-to-end. KronDPBL with $L = 2$ CPE-CNC layers, $r = 64$ pattern distributions, $k = 5$ noise factor distributions, $D_{\text{model}} = 512$, 8 attention heads, Kronecker rank $S = 8$; reconstruction grid adapted per dataset (e.g., 64×16 for CSI-80 AR, 4×242 for its PI; 3×30 for WST, 13×4 for Wi*, 8×29 for CSI-Bench); loss weights $\lambda_{\text{recon}} = 0.6$, $\lambda_{\text{pattern}} = 0.3$, $\lambda_{\text{noise}} = 0.1$, $\tau = 0.1$, $\lambda_{\text{sep}} = 0.1$; AdamW with lr 5×10^{-4}, weight decay 10^{-2}, batch 64, cosine annealing for 100 epochs; results averaged over 3 seeds. See Appendix A for datasets, tasks, standard deviations, and parameter analysis.

Shared design and protocol: KronDPBL uses one architecture and one training recipe for all datasets. The model design, hyperparameters, and pretraining objective are unchanged across datasets. Dataset-specific parameters are the input projection, the Kronecker-sum spatial encoding ($D_{\text{row}} + D_{\text{col}}$ scalars), and the reconstruction head, which is discarded after pretraining. The CPE–CNC trunk and temporal aggregation are architecturally identical across datasets, but their weights are learned separately for each dataset. For each evaluation fold, the encoder is pretrained only on the unlabeled training recordings of that fold and then frozen. Test recordings, and any user, location, or device held out by the protocol, never enter pretraining.

4.2 OVERALL PERFORMANCE EVALUATION

Table 1 compares all methods across datasets and tasks. KronDPBL is best or second-best on 12 of 13 settings, winning 11 outright. Pure sequential models are dimension-sensitive: PatchTST jumps

Table 1: Performance comparison across six datasets and four tasks. Best results are boldfaced and second best are underlined.

	CSI-80				WST			WiGesture		WiFall		CSI-Bench	WiCount
Method	AR-8^{W}	AR-4^{C}	PC-9^{W}	PI-11^{W}	AR-6^{L}	AR-10^{L}	PI-13^{L}	AR-6^{C}	PI-8^{C}	AR-5^{C}	FD-2^{C}	FD-2^{T}	PC-4^{C}
WiAR	69.77	65.30	34.92	48.68	32.29	22.86	66.40	52.66	55.00	55.07	73.00	56.16	53.75
SignFi	69.31	72.05	30.28	40.37	75.71	23.89	83.97	41.08	42.08	68.38	85.22	74.79	61.54
WiGr	75.69	73.28	45.34	51.99	82.58	68.32	97.53	63.95	62.76	75.50	86.87	71.78	65.57
WiGRUNT	71.05	66.70	37.67	53.97	84.23	69.43	97.82	34.80	27.89	50.88	73.31	72.88	50.41
Wi-ATCN	69.90	75.41	25.24	41.42	16.67	16.41	23.50	67.20	81.14	71.92	85.96	73.97	68.85
HAR-SAnet	81.92	75.34	57.93	70.74	82.63	65.52	97.41	69.71	81.51	74.51	85.85	72.60	67.78
Widar3.0	72.15	72.91	36.82	67.36	39.64	32.32	80.64	34.38	32.30	70.64	88.04	73.42	69.13
GUARD	70.43	73.92	38.77	55.71	50.47	36.50	90.24	54.02	71.36	64.03	81.25	75.07	67.73
CSI-BERT	52.52	79.22	25.19	27.81	81.51	64.35	97.18	45.36	58.58	75.50	86.87	73.97	65.58
THAT	80.08	75.43	52.01	68.34	46.53	28.93	75.91	68.44	82.62	77.42	87.89	68.49	71.20
Wi-Prompt	55.01	53.11	30.53	30.46	17.00	12.85	20.81	22.36	56.92	55.84	79.11	70.96	**72.16**
PatchTST	66.36	72.57	25.63	31.00	16.85	16.20	24.43	73.87	75.69	69.84	81.82	72.88	64.18
ViT	70.33	70.79	29.09	44.82	84.25	**73.53**	98.20	57.05	68.92	65.23	80.61	73.70	61.89
WiFiTAD	63.24	81.23	29.93	28.22	81.28	36.33	93.64	30.52	12.23	58.64	87.08	57.53	66.87
CLAR	78.02	76.22	56.41	65.48	51.40	50.62	94.72	70.36	56.00	74.24	84.95	73.97	62.96
KronDPBL	**82.92**	**81.77**	**66.20**	**85.35**	**86.81**	70.95	**98.38**	**76.89**	**83.22**	**77.49**	**89.18**	**79.45**	70.71

from 25.63% on CSI-80 PC-9 (4×242) to 64.18% on WiCount PC-4 (1×52), and CSI-BERT drops from 79.22% (AR-4C) to 27.81% (PI-11W) on the same platform, while KronDPBL stays stable and sustains 85.35%. The low-rank Kronecker reconstruction regularizes the latent space to preserve cross-antenna covariance, whereas raw masked recovery reabsorbs device-specific offsets that poison identity features. Per-setting standard deviations over three seeds are reported in Appendix.

The CNC denoiser adds cross-hardware robustness: on CSI-Bench FD-2T (232-dim HP + 64-bin ESP32 CSI), KronDPBL drops only 9.73% and reaches 79.45%, above the best end-to-end GUARD (75.07%), as zeroing noise-projected values suppresses chipset-specific distortions that monolithic reconstruction treats as signal. The Kronecker prior and the denoiser are complementary: one anchors the representation to physical axes, the other strips axis-agnostic perturbations.

4.3 Ablation study

We ablate each design choice on four in-domain tasks spanning diverse hardware and task granularities. Pure MLP (no encoder) isolates the role of learned representations; the component-ablation rows of Table 2 each remove exactly one component, and the two bottom blocks report classical denoising controls. The corruption variants split the six pretraining corruptions into structured (subcarrier and temporal dropout) and unstructured (the remaining four) subsets. Hampel window $w = 7$ and 3×MAD; PCA 95% energy.

Reconstruction is the single most critical component. Its removal causes the largest drops: 13.7% on AR-4^{C}, 15.1% on PI-11^{W}, and 9.5% on FD-2^{T}. **Pretraining is essential when labeled data is scarce.** The drop is largest on AR-4^{C} (15.1%), the task with the fewest labeled samples, and mildest on the larger FD-2^{T} (5.5%): SSL acts as a regularizer whose value grows as labels shrink. **Kronecker PE bridges single-antenna gaps.** Its removal drops FD-2^{T} by 12.3% but only 2.1–4.9% on multi-antenna CSI-80 and WST, injecting spatial priors precisely when the physical array provides none. **Full corruption diversity is necessary.** Either subset alone degrades all tasks by 2.1–8.2%; each corruption family covers a distinct failure mode. **Classical filters neither help nor substitute the learned denoiser.** As preprocessing or as a CNC replacement, Hampel and PCA never match the unfiltered full model. They are at best redundant on CSI-80 and can task-dependently degrade others (Hampel up to 17.8% on FD-2^{T}; PCA up to 14.2% on WST), insensitive to the Hampel window. As substitutes, they fall short on every task (1.4–28.4%) and in half of task×filter cases are worse than removing denoising entirely (Hampel collapses PI; PCA collapses WST gesture recognition). The trade-off is unresolvable: classical filters target structures that overlap with task-relevant motion, whereas the learned denoiser preserves reconstructible structure. No substitute is uniformly competitive; the full model on unfiltered input remains best.

Table 2: Ablation study results (accuracy, %).

Setting	CSI-80 AR-4^{C}	CSI-80 PI-11^{W}	WST AR-6^{L}	CSI-Bench FD-2^{T}
Full KronDPBL	81.77	85.35	86.81	79.45
Pure MLP	$25.23_{\downarrow 56.5}$	$84.34_{\downarrow 1.0}$	$73.25_{\downarrow 13.6}$	$65.75_{\downarrow 13.7}$
w/o Pretraining	$66.69_{\downarrow 15.1}$	$75.32_{\downarrow 10.0}$	$79.65_{\downarrow 7.2}$	$73.97_{\downarrow 5.5}$
w/o Kronecker PE	$76.88_{\downarrow 4.9}$	$81.27_{\downarrow 4.1}$	$84.73_{\downarrow 2.1}$	$67.12_{\downarrow 12.3}$
w/o Denoising	$78.86_{\downarrow 2.9}$	$79.41_{\downarrow 5.9}$	$81.45_{\downarrow 5.4}$	$69.86_{\downarrow 9.6}$
w/o Contrastive	$75.18_{\downarrow 6.6}$	$76.27_{\downarrow 9.1}$	$83.92_{\downarrow 2.9}$	$70.37_{\downarrow 9.1}$
w/o Reconstruction	$68.12_{\downarrow 13.7}$	$70.23_{\downarrow 15.1}$	$80.39_{\downarrow 6.4}$	$69.91_{\downarrow 9.5}$
Deterministic	$79.46_{\downarrow 2.3}$	$75.79_{\downarrow 9.6}$	$86.47_{\downarrow 0.3}$	$69.44_{\downarrow 10.0}$
Structured only	$76.14_{\downarrow 5.6}$	$80.06_{\downarrow 5.3}$	$84.67_{\downarrow 2.1}$	$73.97_{\downarrow 5.5}$
Unstructured only	$78.21_{\downarrow 3.6}$	$81.43_{\downarrow 3.9}$	$82.74_{\downarrow 4.1}$	$71.23_{\downarrow 8.2}$
Full + Hampel	$81.52_{\downarrow 0.3}$	$85.07_{\downarrow 0.3}$	$85.53_{\downarrow 1.3}$	$73.97_{\downarrow 5.5}$
Full + PCA	$80.32_{\downarrow 1.5}$	$83.80_{\downarrow 1.6}$	$72.57_{\downarrow 14.2}$	$73.97_{\downarrow 5.5}$
Replace w/ Hampel	$75.84_{\downarrow 5.9}$	$63.30_{\downarrow 22.1}$	$85.20_{\downarrow 1.6}$	$76.71_{\downarrow 2.7}$
Replace w/ PCA	$79.85_{\downarrow 1.9}$	$77.18_{\downarrow 8.2}$	$58.42_{\downarrow 28.4}$	$78.08_{\downarrow 1.4}$

Table 3: Transfer and sensitivity analyses.

(a) Cross-hardware transfer, CB means CSI-Bench.

	KronDPBL	WiFiTAD	THAT	WiGr	CSI-BERT
WiFall → CB	50.68	49.32	48.87	47.95	50.33
CB → WiFall	**78.88**	61.75	50.00	50.00	50.00

(b) Effect of pattern rank r.

Task	$r=4$	$r=8$	$r=16$	$r=32$	$r=64$	$r=96$
AR-4^{C}	68.98	69.96	69.66	75.47	81.77	70.02
PI-11^{W}	63.32	68.34	69.61	69.03	85.35	66.92
AR-6^{L}	58.86	57.97	57.71	87.14	86.81	76.88
FD-2^{T}	60.27	67.12	63.01	73.97	79.45	67.57

(c) Effect of Kronecker grid factorization.

Task	1	2	3	4
AR-4^{C}	4×256: 79.73	8×128: 82.13	16×64: 81.77	32×32: 82.07
PI-11^{W}	4×242: 85.35	8×121: 84.22	11×88: 83.92	22×44: 85.65
FD-2^{T}	2×116: 78.08	4×58: 78.77	8×29: 79.45	–
AR-6^{L}	3×30: 86.81	5×18: 87.53	6×15: 86.54	9×10: 87.53

(d) Effect of positional encoding design.

Task	Kronecker PE	Learned PE	Sinusoidal PE
AR-4^{C}	81.77	76.67	25.05
PI-11^{W}	85.35	75.42	60.87
AR-6^{L}	86.81	68.27	57.33
FD-2^{T}	79.45	67.12	69.86

4.4 Generalization and Transfer

We evaluate generalization at two levels: *same-dataset*, where we hold out dimension (user, environment, or device) during pretraining and test on the held-out slice, measuring domain invariance; and *cross-dataset*, where we train on one dataset and transfer to another dataset for the same task, testing joint invariance to user, environment, and device.

Same-Dataset Generalization. KronDPBL is trained from scratch on the remaining unlabeled data, frozen, and paired with an MLP head for each hold-out split, while baselines are trained end-to-end on the identical labeled split. Key comparisons are visualized in the grouped bar charts in Figure 2.

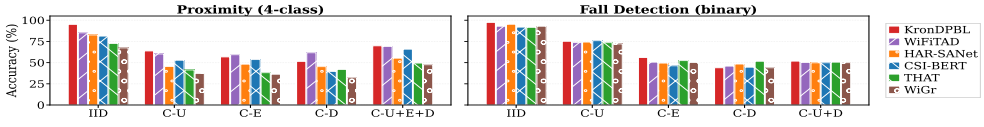

Figure 2: Generalization Analysis Result.

We evaluate 3 datasets. **CSI-Bench Multitask** (HP, 232-dim CSI) has 7 users ×7 environments ×10 devices, supporting cross-user, cross-environment, cross-device, and compound hold-outs on 5-class HAR and 4-class Proximity. **CSI-Bench FallDetection** raises difficulty with 17 users ×6 environments ×2 radically different devices (232-dim HP and 64-bin ESP32 CSI). **CSI-80** (BCM4366c0, 4×242) adds PI-11 with Person×Environment and AR-8 with Environment hold-outs.

Cross-Dataset Transfer: Fall Detection. We test cross-hardware transfer on fall detection, available on both WiFall (Atheros, 52 subcarriers, 928 recordings, one environment) and CSI-Bench (232-dim HP + 64-bin ESP32 CSI, 4,690 recordings, 17 users, 6 environments). Each domain pretrains its own encoder on unlabeled data; the frozen encoder with an MLP head trained on the source domain is evaluated on the target test set. WiFall recordings are zero-padded from 52 to 232 subcar-

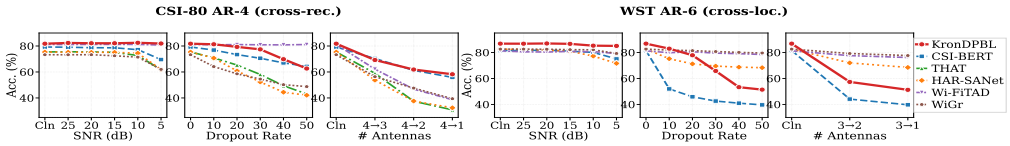

Figure 3: Robustness Analysis Result.

riers. Each direction uses its own in-domain self-supervised encoder, with the target domain entirely held out during pretraining and no cross-dataset checkpoint reused. Transfer is strongly asymmetric (Table 3a). **CSI-Bench → WiFall**: KronDPBL reaches 78.88%, 17.1% above WiFiTAD and 28.9% above chance, while THAT, WiGr, and CSI-BERT stay at chance. **WiFall → CSI-Bench**: all methods fail at random. The asymmetry tracks pretraining diversity: CSI-Bench provides enough variation to learn transferable features, but WiFall's single-environment-hardware corpus does not.

4.5 Robustness Analysis

Input Perturbations. Three perturbation types are applied to the test split: AWGN at SNR $\in \{25, 20, 15, 10, 5\}$ dB, random subcarrier dropout at 10%–50%, and antenna dropout ($4, 3, 2, 1$ for CSI-80; $3, 2, 1$ for WST). All methods use frozen backbones with native heads and no retraining on CSI-80 AR-4^{C} or WST AR-6^{L}. **AWGN immunity is near-complete.** KronDPBL stays virtually flat from clean to 5 dB on both tasks, whereas CSI-BERT loses 9.7% and THAT 13.7%. CPE estimates noise via learnable Z vectors, and CNC filters it out by pretraining corruptions. **Subcarrier dropout resilience follows corruption coverage.** At 20% dropout on AR-4^{C}, KronDPBL loses only 2.5%, versus 10.1% for THAT and 14.6% for HAR-SAnet; at 50%, it retains 62.65%, over 20% above both. WiFiTAD stays flat because its Conv1d embedding mixes all subcarriers into shared channels; this architectural resilience also explains its edge on low-dimensional WST. **Antenna dropout distinguishes learned from architectural resilience.** KronDPBL is best on AR-4^{C}, retaining 61.88% with 2 antennas and 58.24% with 1; factored Kronecker PE absorbs partial antenna loss. On WST, WiFiTAD's channel mixing dominates instead. Robustness follows corruption coverage and is strongest on multi-antenna arrays where Kronecker structure matters most.

4.6 Parameter Analysis

All variants use a frozen encoder with an MLP classifier. **Pattern rank r shows a sweet spot at 64 (Table 3b).** Performance rises monotonically from $r = 4$ to $r = 64$ (AR-4^{C}: from 68.98% to 81.77%), then drops sharply at $r = 96$: too few patterns underfit CSI diversity, too many destabilize the CPE cross-attention. **Kronecker grid factorization has minimal impact on accuracy (Table 3c).** Decompositions are reported as $D_{\text{row}} \times D_{\text{col}}$, underlined entries mark the default, and results of equivalent $(a, b)/(b, a)$ pairs are equal. Across all four tasks, varying $(D_{\text{row}}, D_{\text{col}})$ shifts accuracy by at most 2.4%(AR-4^{C}: 79.73–82.13%). The Kronecker prior is largely insensitive to decomposition granularity. Where the data has a genuine antenna×subcarrier layout the grid matches it exactly (CSI-80 PI: 4×242; WST: 3×30). For single-antenna inputs the factorization is arbitrary by design, and Table 3c shows accuracy varies by at most 2.4% across factorizations, benefits from separable low-rank structure, not from an exact physical-topology embedding. **Kronecker PE dominates alternative encodings (Table 3d).** It outperforms learned and sinusoidal PE by 5.1–18.5 and 9.6–56.7 percentage points, respectively. Other parameters sensitivity analysis is deferred to Appendix B.

5 Conclusion

With one architecture, one recipe, and per-corpus input adapters of negligible size, KronDPBL is pretrained on unlabeled CSI under a strict fold-wise protocol and then frozen; lightweight classifiers on these frozen encoders achieve state-of-the-art accuracy on 11 of 13 settings across six public datasets, and frozen encoders transfer across heterogeneous hardware on task-matched fall detection with 78.88% on CSI-Bench→WiFall versus 61.75% for the strongest baseline, WiFiTAD. Scaling to a single encoder pretrained on pooled heterogeneous corpora is left to future work.

AI USE STATEMENT

In this work, we have not used generative AI tools for research ideation, methodology design, data analysis, code generation, or experimental design. We used generative AI tools solely for language polishing of the manuscript, including grammar correction and phrasing improvements. All AI-assisted edits were reviewed and verified by the authors, and no AI-generated content was incorporated without manual inspection. We take full responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

No new human data has been collected, the use of public datasets and their licenses has been employed, identities have been anonymized as only anonymous category labels, and there is a risk of abuse of the PI technology. All datasets used in our experiments are publicly available and were used in accordance with their respective licenses and terms of use. We do not foresee any direct harmful applications or negative societal impacts arising from this work beyond those generally associated with advances in machine learning. The authors declare no conflicts of interest that could influence the results or interpretation of this study.

REPRODUCIBILITY STATEMENT

We have taken several measures to ensure the reproducibility of our work. The experimental setup, including model architectures, hyperparameters, training details, and evaluation protocols, is described in detail. All datasets used in our experiments are publicly available and can be accessed through the references cited in the paper; the data processing steps are described in Appendix. An anonymized implementation of our method, together with scripts to reproduce the main results, is provided in the supplementary materials.

REFERENCES

B. Barahimi, H. Tabassum, M. Omer, and O. Waqar. Context-Aware Predictive Coding: A Representation Learning Framework for WiFi Sensing, 2024.

Xingcan Chen, Chenglin Li, Chengpeng Jiang, Wei Meng, and Wendong Xiao. WiPhase: A Human Activity Recognition Approach by Fusing of Reconstructed WiFi CSI Phase Features. *IEEE Transactions on Mobile Computing*, 24(1):394–408, 2025.

Zhe Chen, Chao Cai, Tianyue Zheng, Jun Luo, Jie Xiong, and Xin Wang. RF-Based Human Activity Recognition Using Signal Adapted Convolutional Neural Network. *IEEE Transactions on Mobile Computing*, 22(1):487–499, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Tianqi Fan, Sen Qiu, Wei Gong, and Yuguang Fang. Multi-Source Domain Generalization for CSI-Based Human Activity Recognition. *IEEE Transactions on Mobile Computing*, 24(10):11034–11048, 2025.

Yu Gu, Xiang Zhang, Yantong Wang, Meng Wang, Huan Yan, Yusheng Ji, Zhi Liu, Jianhua Li, and Mianxiong Dong. WiGRUNT: WiFi-Enabled Gesture Recognition Using Dual-Attention Network. *IEEE Transactions on Human-Machine Systems*, 52(4):736–746, 2022.

Linlin Guo, Lei Wang, Chuang Lin, Jialin Liu, Bingxian Lu, Jian Fang, Zhonghao Liu, Zeyang Shan, Jingwen Yang, and Silu Guo. WiAR: A Public Dataset for WiFi-Based Activity Recognition. *IEEE Access*, 7:154935–154945, 2019.

Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Tool Release: Gathering 802.11n Traces with Channel State Information. *ACM SIGCOMM Computer Communication Review*, 41 (1):53, 2011.

Bing Li, Wei Cui, Wei Wang, Le Zhang, Zhenghua Chen, and Min Wu. Two-Stream Convolution Augmented Transformer for Human Activity Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pp. 286–293, 2021.

Ying Liang, Wenjie Wu, Haobo Li, Lijun Cui, Jianguo Ju, and Pengfei Xu. GUARD: A Unified Open-Set and Closed-Set Gait Recognition Framework via Feature Reconstruction on Wi-Fi CSI. *IEEE Transactions on Information Forensics and Security*, 21:2520–2533, 2026.

B. Liu, S. Gao, X. Liu, X. Cheng, and L. Yang. WiFo: Wireless Foundation Model for Channel Prediction, 2024.

Jian Liu, Hongbo Liu, Yingying Chen, Yan Wang, and Chen Wang. Wireless Sensing for Human Activity: A Survey. *IEEE Communications Surveys & Tutorials*, 22(3):1629–1645, 2020.

Zhendong Liu, Le Zhang, Bing Li, Yingjie Zhou, Zhenghua Chen, and Ce Zhu. WiFi CSI Based Temporal Activity Detection via Dual Pyramid Network. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2025.

Xuanhao Luo, Zhizhen Li, and Yuchen Liu. CSI-JEPA: Towards Foundation Representations for Ubiquitous Sensing with Minimal Supervision, 2026.

Yongsen Ma, Gang Zhou, Shuangquan Wang, Hongyang Zhao, and Woosub Jung. SignFi: Sign Language Recognition Using WiFi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 2(1):23:1–23:21, 2018.

Yongsen Ma, Gang Zhou, and Shuangquan Wang. WiFi Sensing with Channel State Information: A Survey. *ACM Computing Surveys*, 52(3):1–36, 2019.

Francesca Meneghello, Nicoló Dal Fabbro, Domenico Garlisi, Ilenia Tinnirello, and Michele Rossi. A CSI Dataset for Wireless Human Sensing on 80 MHz Wi-Fi Channels. *IEEE Communications Magazine*, 61(9):146–152, 2023.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A Time Series is Worth 64 Words: Long-Term Forecasting with Transformers. In *International Conference on Learning Representations (ICLR)*, 2023.

Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. Whole-Home Gesture Recognition Using Wireless Signals. In *Proceedings of the 19th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 27–38, 2013.

Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. Understanding and Modeling of WiFi Signal Based Human Activity Recognition. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 65–76, 2015.

Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. E-eyes: Device-Free Location-Oriented Activity Identification Using Fine-Grained WiFi Signatures. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp. 617–628, 2014.

Chunjing Xiao, Yanhui Han, Wei Yang, Yane Hou, Fangzhan Shi, and Kevin Chetty. Diffusion Model-Based Contrastive Learning for Human Activity Recognition. *IEEE Internet of Things Journal*, 11(20):33525–33536, 2024.

Jianfei Yang, Xinyan Chen, Han Zou, Chris Xiaoxuan Lu, Dazhuo Wang, Sumei Sun, and Lihua Xie. SenseFi: A Library and Benchmark on Deep-Learning-Empowered WiFi Human Sensing. *Patterns*, 4(3):100703, 2023.

Zheng Yang, Yi Zhang, Guidong Zhang, Yue Zheng, and Guoxuan Chi. Widar 3.0: Wifi-based activity recognition dataset, 2020. URL `https://dx.doi.org/10.21227/7znf-qp86`.

Youwei Zeng, Dan Wu, Jie Xiong, Enze Yi, Ruiyang Gao, and Daqing Zhang. FarSense: Pushing the Range Limit of WiFi-Based Respiration Sensing with CSI Ratio of Two Antennas. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, 3(3):121:1–121:26, 2019.

Xie Zhang, Chengpei Tang, Kang Yin, and Qingqian Ni. WiFi-Based Cross-Domain Gesture Recognition via Modified Prototypical Networks. *IEEE Internet of Things Journal*, 9(11):8584–8596, 2022a.

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

Yunming Zhao, Wei Gong, Minghui Liwang, Li Li, Baoxian Zhang, and Cheng Li. A Generalizable Prompt-Based Prototypical Framework for CSI-Based Few-Shot and Cross-Domain Activity Recognition. *IEEE Transactions on Mobile Computing*, 24(12):13125–13141, 2025.

Zijian Zhao, Tingwei Chen, Fanyi Meng, Hang Li, Xiaoyang Li, and Guangxu Zhu. Finding the Missing Data: A BERT-Inspired Approach Against Package Loss in Wireless Sensing. In *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 1–6, 2024.

Zijian Zhao, Fanyi Meng, Zhonghao Lyu, Hang Li, Xiaoyang Li, and Guangxu Zhu. CSI-BERT2: A BERT-Inspired Framework for Efficient CSI Prediction and Classification in Wireless Communication and Sensing. *IEEE Transactions on Mobile Computing*, 25(5):7241–7257, 2026.

Aichun Zhu, Zhonghua Tang, Zixuan Wang, Yue Zhou, Shichao Chen, Fangqiang Hu, and Yifeng Li. Wi-ATCN: Attentional Temporal Convolutional Network for Human Action Prediction Using WiFi Channel State Information. *IEEE Journal of Selected Topics in Signal Processing*, 16(4): 804–816, 2022.

Guozhen Zhu, Yuqian Hu, Weihang Gao, Wei-Hsiang Wang, Beibei Wang, and K. J. Ray Liu. CSI-Bench: A Large-Scale In-the-Wild Dataset for Multi-task WiFi Sensing. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025.

A Dataset and Task Details

Table 4 summarizes the six datasets used in our evaluation. We use CSI amplitude only and apply per-segment z-score normalization. Each dataset uses recording-level split protocols to ensure no temporal leakage between training and test sets. Table 5 summarizes the checkpoint used for each setting under the strict fold-wise pretraining protocol. For all within-dataset and within-domain settings, the encoder for an evaluation fold is pretrained only on that fold's unlabeled training recordings; test recordings, subjects, locations, and devices are excluded from pretraining. For the two cross-hardware transfer settings, the entire target domain is held out during both pretraining and head training.

Table 4: Summary of datasets used in evaluation. D = input dimensionality after flattening (antennas $\times$ subcarriers). CSI-80 activity protocols retain all 256 FFT bins (4×256, $D{=}1024$); person identification retains the 242 active subcarriers (4×242, $D{=}968$).

Dataset	Hardware	Freq.	Ant.×Subc.	D	Classes	Samples
CSI-80	Broadcom BCM4366c0	5.21 GHz	4×256/242	1024/968	4–11	248 rec. (16 for pretraining)
Widar3.0	Intel 5300	5 GHz	3×30	90	6–13	258K instances
WiGesture	ESP32-S3	2.4 GHz	1×52	52	6–8	47 rec./2,430 seg.
WiFall	ESP32-S3	2.4 GHz	1×52	52	2–5	928 rec.
WiCount	ESP32-S3	2.4 GHz	1×52	52	4	15 rec.
CSI-Bench	HP HealthPod/ESP32	2.4 GHz	1×232/64	232/64	2–7	6.7K / 55.7K

A.1 CSI-80

The CSI-80 dataset (Meneghello et al., 2023) was collected using a Broadcom BCM4366c0 802.11ac chipset at 5.21 GHz with 80 MHz bandwidth, producing 4 antennas $\times$ 256 FFT bins (242 active subcarriers — 234 data and 8 pilot — after discarding guard and DC bins). Activity-recognition settings retain all 256 bins ($D{=}1024$ after flattening), while person-identification settings retain the 242 active subcarriers ($D{=}968$). Each complex CSI frame is sampled at approximately 167 Hz. The full dataset contains 248 recordings totaling 7.26M frames across 7 environments (Bedroom, Living room, Kitchen, Laboratory, Office, Anechoic chamber, Meeting room).

Table 5: Per-setting checkpoint map under the strict fold-wise pretraining protocol. For every evaluation fold, one encoder is pretrained on that fold's unlabeled training recordings only; test recordings, subjects, locations, and devices never enter pretraining. For cross-hardware transfer (last two rows), the entire target domain is held out during both pretraining and head training.

Setting	Pretraining data	Input dim	Grid	Test fold in pretraining
CSI-80 AR-8^{W} / AR-4^{C} / PC-9^{W} / PI-11^{W}	fold-wise unlabeled training recordings	1024 / 968	64×16 / 4×242	No
Widar3.0 AR-6^{L} / AR-10^{L} / PI-13^{L}	fold-wise unlabeled training recordings	90	3×30	No
WiGesture AR-6^{C} / PI-8^{C}	fold-wise unlabeled training recordings	52	4×13	No
WiFall AR-5^{C} / FD-2^{C}	fold-wise unlabeled training recordings	52	13×4	No
WiCount PC-4^{C}	fold-wise unlabeled training recordings	52	13×4	No
CSI-Bench FD-2^{T}	FD train split only	232	8×29	No
CB→WiFall	CSI-Bench FD train split	232	8×29	Target domain held out
WiFall→CB	WiFall (zero-padded to 232)	232	8×29	Target domain held out

CSI-80 supports three downstream tasks:

- **Activity Recognition (AR-8)**: 8 classes — Empty, Walking, Running, Jumping, Sitting still, Standing, Sit-down/Stand-up, Arm exercises.
- **Person Identification (PI-11)**: 11 subjects (p00, p03, p05–p13) across 4 environment settings.
- **People Counting (PC-9)**: 9 occupancy levels (0–8 persons) across 4 environments.

For self-supervised pretraining, we use a preprocessed subset of 16 recordings (4 classes $\times$ 4 recordings: Empty, Walking, Running, Sitting) under a 4-class activity recognition protocol with non-overlapping segments of $T = 100$ frames (AR-4^{C}). The Kronecker reconstruction grid is task-dependent: activity recognition keeps all 256 FFT bins per antenna ($4\times256 = 1024$) refolded into a 64×16 grid, while person identification retains the 242 active subcarriers on the physical 4×242 grid, preserving subcarrier–position correspondence. Downstream evaluation follows strict cross-recording 4-fold cross-validation where each fold holds out one recording per class, ensuring zero temporal overlap between training and test sets. For person identification, we use within-environment cross-validation with recording-level splits across all 11 subjects (PI-11^{W}). All experiments use per-segment z-score normalization with recording-level splits. Pretraining is strictly fold-wise: for every evaluation fold, only that fold's unlabeled training recordings are used; test recordings, subjects, locations, and devices never enter pretraining.

A.2 WIDAR3.0

The Widar3.0 dataset (Yang et al., 2020; Zhang et al., 2022b) was collected with an Intel 5300 NIC (3 antennas, 30 OFDM subcarriers, $D = 90$) at 5 GHz. It contains approximately 258,000 CSI gesture instances from 17 users across 75 domains and 20 distinct indoor locations, totaling 8,620 minutes. We use only its WiFi CSI modality and do not use the other two modalities. The dataset contains 10 gesture classes; the AR-6 protocols use six of them (Push, Pull, Wave, Clap, Slide, and Circle), while AR-10^{L} uses all ten. We evaluate three protocols:

- **AR-6^{L}** (cross-location gesture): Train on locations 1–4 (users 1–3, 62,520 frames), test on location 5 (15,630 frames).
- **AR-10^{L}** (cross-location gesture): 10-class classification over all ten gesture classes. Train on locations 1–4 (users 1–3, 70,560 samples), test on location 5 (users 1–3, 17,640 samples).

- **PI-13^{L}** (cross-location user identification): 13-way classification across locations (train 105,720 / test 26,430 frames).

All Widar3.0 experiments fix the sequence length at $T = 100$ frames.

A.3 WiGesture, WiFall, and WiCount

These three datasets were collected with an ESP32-S3 device (single antenna) at 2.4 GHz, producing 52-subcarrier CSI ($D = 52$) at approximately 83–84 Hz (Zhao et al., 2024; 2026). All three are pretrained fold-wise on their own recordings and evaluated under 4-fold cross-recording cross-validation.

- **WiGesture** (Zhao et al., 2024): 47 recordings from 8 subjects performing 6 gestures — Applause, Circle (clockwise), Front-and-after, Left-and-right, Up-and-down, Wave (right). Split into $T = 100$ non-overlapping segments (2,430 segments total) and evaluated on gesture classification (AR-6^{C}) and subject identification (PI-8^{C}).
- **WiFall** (Zhao et al., 2026): 928 recordings from 10 subjects performing 5 actions — Fall, Jump, Sit, Stand, Walk. Contains 464 fall and 464 non-fall recordings (balanced binary detection, FD-2^{C}) plus 5-class action recognition (AR-5^{C}).
- **WiCount** (Zhao et al., 2026): 15 recordings from 4 subjects with occupancy levels 0–3 persons. Evaluated as 4-class counting (PC-4^{C}).

A.4 CSI-Bench

CSI-Bench (Zhu et al., 2025) is a comprehensive benchmark covering 7 WiFi CSI tasks from heterogeneous IoT devices. We focus on two task categories:

- **Fall Detection (FD-2^{T})**: 6,700 samples from two devices — HP HealthPod (232 subcarriers, 6,206 samples) and ESP32 (64 subcarriers, 494 samples). Seventeen users perform activities across 6 environments. The test set is partitioned into Easy (IID), Medium, and Hard (cross-device) difficulty tiers. The standard train/val/test split is 4,690/1,005/1,005 samples. We report FD-2^{T} results on the Hard tier (ESP32, 73 samples) as the primary metric, reflecting cross-device generalization. KronDPBL achieves Easy 98.25%, Medium 82.35%, Hard 79.45% (frozen CSI-Bench-pretrained encoder, pretrained on the FD train split only, 3-layer MLP classifier). For reference, the best end-to-end baseline GUARD achieves Hard 75.07% and WiAR achieves Hard 56.16%.
- **Multitask**: 55,684 samples across 7 users and 6 environments, supporting three tasks: Human Activity Recognition (HAR, 5 classes: Jumping, Running, Seated-breathing, Walking, Waving-hand), Person Identification (PI, 7 classes: users U01–U06, U22), and Proximity Recognition (Prox, 4 classes: 0.5 m, 1.5 m, 2.5 m, 3.5 m). Each task is evaluated under IID, cross-device, cross-environment, and cross-user protocols. The train/val/test split is 21,473/4,601/4,602.

A.5 Run-to-run Variability

All results are averaged over three independent runs with different random seeds. For cross-recording (^{C}) protocols, each fold score is first averaged over the three seeds, and the reported standard deviation is computed across folds; for all other protocols, the standard deviation is computed directly over the three seed-level scores. Table 7 reports mean$\pm$std for all thirteen evaluation settings of the main results (Table **??**) and for the two cross-hardware transfer settings (Table **??**); Table 8 reports the corresponding standard deviations for the denoising-control ablation (Table 2).

Two observations follow. First, on the majority of winning settings the margin over the strongest baseline exceeds the corresponding standard deviation (PC-9^{W}, PI-11^{W}, AR-6^{L}, AR-6^{C}, and FD-2^{T}), so these improvements are robust to seed and fold variation. Second, on AR-4^{C}, PI-13^{L}, PI-8^{C}, AR-5^{C}, and FD-2^{C} the margin is within one standard deviation; the main text phrases these as modest gains or statistical ties accordingly.

Table 6: Per-dataset task definitions and evaluation protocols. ^{C} = cross-recording, ^{W} = within-environment, ^{L} = cross-location, ^{T} = tiered difficulty (FD-2 Easy/Medium/Hard); in multitask rows, ^{T} marks the cross-device transfer protocol.

Dataset	Task	#Cls	Protocol	Train/Test Samples
CSI-80	AR-8^{W}	8	Within-environment CV	$\sim$12 / 4 recordings
CSI-80	AR-4^{C}	4	4-fold recording CV	$\sim$12 / 4 recordings
CSI-80	PC-9^{W}	9	Within-environment CV	$\sim$12 / 4 recordings
CSI-80	PI-11^{W}	11	Within-environment CV	4 env.$\times$11 subj.
Widar3.0	AR-6^{L}	6	Cross-location	62,520 / 15,630 frames
Widar3.0	PI-13^{L}	13	Cross-location	105,720 / 26,430 frames
Widar3.0	AR-10^{L}	10	Cross-location	70,560 / 17,640 samples
WiGesture	AR-6^{C}	6	4-fold recording CV	$\sim$1,822 / 608 segments
WiGesture	PI-8^{C}	8	4-fold recording CV	$\sim$1,822 / 608 segments
WiFall	AR-5^{C}	5	4-fold recording CV	$\sim$696 / 232 recordings
WiFall	FD-2^{C}	2	4-fold recording CV	$\sim$696 / 232 recordings
WiCount	PC-4^{C}	4	4-fold recording CV	$\sim$11 / 4 recordings
CSI-Bench	FD-2^{T}	2	IID + cross-device	4,690 / 1,005 / 1,005
CSI-Bench	HAR^{T}	5	IID + cross-dev./env./user	21,473 / 4,601 / 4,602

Table 7: Run-to-run variability of KronDPBL across all evaluation settings. Means and standard deviations over three seeds. For cross-recording (^{C}) protocols, each fold score is first averaged over three seeds; the reported std is across folds. †On WiFall AR-5^{C} the strongest baseline (THAT, 77.42$\pm$1.64 under the same protocol) is within one standard deviation; we regard this setting as a statistical tie.

Setting	Accuracy (%)
CSI-80 AR-8^{W}	82.92$\pm$0.70
CSI-80 AR-4^{C}	81.77$\pm$1.41
CSI-80 PC-9^{W}	66.20$\pm$1.74
CSI-80 PI-11^{W}	85.35$\pm$1.21
Widar3.0 AR-6^{L}	86.81$\pm$0.23
Widar3.0 AR-10^{L}	70.95$\pm$1.56
Widar3.0 PI-13^{L}	98.38$\pm$1.55
WiGesture AR-6^{C}	76.89$\pm$0.29
WiGesture PI-8^{C}	83.22$\pm$0.68
WiFall AR-5^{C}	77.49$\pm$1.18†
WiFall FD-2^{C}	89.18$\pm$1.22
WiCount PC-4^{C}	70.71$\pm$1.35
CSI-Bench FD-2^{T}	79.45$\pm$0.33
CB$\rightarrow$WiFall	78.88$\pm$0.17
WiFall$\rightarrow$CB	50.68$\pm$0.32

Table 8: Standard deviations (three seeds; same convention as Table 7) for the denoising-control ablation rows of the main-text Table 2; the corresponding means are reported in Table 2. The Full row repeats the main-result stds from Table 7 for reference.

Row	AR-4^{C}	PI-11^{W}	AR-6^{L}	FD-2^{T}
Full (Table **??**)	1.41	1.21	0.23	0.33
Full + Hampel	0.93	0.26	1.78	2.02
Full + PCA	0.59	0.80	0.43	1.45
Replace Hampel	0.63	0.72	2.31	0.98
Replace PCA	1.18	1.35	1.19	2.36
w/o-D	2.67	1.76	0.21	0.93

Variability remains modest across all ablation rows (at most ± 2.67), and never reverses the ranking between the Full model and any denoising-control variant, supporting the robustness of the conclusions drawn from Table 2.

B Hyperparameter Sensitivity

This section reports sensitivity for three hyperparameters not covered in the main text: CPE–CNC depth L, noise factor count k, and downstream MLP depth. All experiments use a frozen KronDPBL encoder. Default values are $L = 2$, $k = 5$, and 3-layer MLP. Entries in the sweep tables of this section are single-run evaluations around the default; run-to-run variability of the default configuration itself is reported in Table 7.

B.1 CPE–CNC Depth L

The number of CPE–CNC blocks controls encoder depth. Table 9 reports accuracy for $L \in \{1, 2, 3, 4\}$. Accuracy increases sharply from L=1 to L=2 on all tasks (AR-4^{C}: $+5.39$ points, AR-6^{L}: $+13.5$ points, FD-2^{T}: $+13.3$ points), indicating that a second CPE–CNC round helps on every task. The difference between L=2 and L=3 is small (within 1.2 percentage points across all four tasks), with AR-4^{C} and PI-11^{W} peaking at L=3 and AR-6^{L} and FD-2^{T} peaking at L=2. At L=4, all tasks degrade by 1.1–5.0 points relative to their peaks (e.g., FD-2^{T} drops from 79.45% to 74.44%). We select $L = 2$ as the default, offering competitive accuracy with fewer CPE–CNC layers.

Table 9: Effect of CPE–CNC depth L on accuracy.

Task	$L = 1$	$L = 2$	$L = 3$	$L = 4$
AR-4^{C}	76.38	81.77	82.10	81.03
PI-11^{W}	80.46	85.35	85.42	83.32
AR-6^{L}	73.29	86.81	85.63	84.75
FD-2^{T}	66.15	79.45	78.55	74.44

B.2 Noise Factor Count k

The number of noise factor distributions in the CPE module controls the granularity of noise modeling. Table 10 reports accuracy for $k \in \{2, 3, 5, 10\}$. The default k=5 provides sufficient capacity to capture common noise modes without over-parameterization.

Table 10: Effect of noise factor count k on accuracy.

Task	$k = 2$	$k = 3$	$k = 5$	$k = 10$
AR-4^{C}	55.01	80.65	81.77	77.57
PI-11^{W}	62.14	82.16	85.35	82.39
AR-6^{L}	56.56	85.70	86.81	86.01
FD-2^{T}	53.59	79.71	79.45	77.30

First, k=2 is severely insufficient. With only two noise factor distributions, accuracy collapses on all four tasks (AR-4^{C}: 55.01%, PI-11^{W}: 62.14%, AR-6^{L}: 56.56%, FD-2^{T}: 53.59%). We conjecture that two noise modes cannot span the diversity of real-world CSI corruptions (thermal noise, packet loss, inter-subcarrier interference, device-specific artifacts), causing the CPE to misroute structured signal into the Z vectors, where it is zeroed out by the CNC.

Second, k=5 is the overall optimum. Three of four tasks peak at k=5, and FD-2^{T} is effectively tied (k=3: 79.71$\pm$0.67% vs. k=5: 79.45$\pm$0.36%, a 0.26-point gap well within one standard deviation). The largest gains over k=3 appear on PI-11^{W} ($+3.2$ points), AR-6^{L} ($+1.1$ points), and AR-4^{C} ($+1.1$ points) — tasks where hardware- and location-specific noise signatures must be separated from

identity- and motion-relevant patterns. Additional noise factors provide finer-grained decomposition capacity that directly benefits these discrimination tasks.

Third, k=10 **degrades every task.** Relative to each task's peak, accuracy drops by 4.2, 3.0, 0.8, and 2.4 points on AR-4^{C}, PI-11^{W}, AR-6^{L}, and FD-2^{T}, though values remain well above the k=2 floor. We conjecture that overspecifying the noise factor count introduces spurious distributions that compete with genuine patterns in the CPE cross-attention, diluting the quality of the extracted $\mathbf{H}$ vectors without capturing any additional noise structure. We select k=5 as the default, offering the best accuracy on three tasks and a statistical tie on the fourth.

B.3 Downstream MLP Depth

In the frozen-encoder regime, the downstream classifier is a lightweight MLP. Table 11 evaluates the number of hidden layers. The default 3-layer setting corresponds to the Full KronDPBL setting.

Table 11: Effect of MLP depth with frozen KronDPBL.

Task	Linear (0)	1-layer	2-layer	3-layer	4-layer
AR-4^{C}	67.88	74.80	78.65	81.77	77.84
PI-11^{W}	11.82	42.82	86.31	85.35	84.26
AR-6^{L}	22.55	65.30	77.58	86.81	87.19
FD-2^{T}	49.32	52.05	60.27	79.45	63.01

Optimal MLP depth varies with task complexity. AR-4^{C} and FD-2^{T} peak at 3 layers (81.77%, 79.45%), while PI-11^{W} peaks at 2 (86.31%) and AR-6^{L} improves through 4 layers (87.19%). Deeper classifiers can overfit (PI-11^{W} declines from 86.31% to 84.26% at 4 layers). A 3-layer MLP is used as the default for all experiments.

C Baseline Selection and Tuning

C.1 Selection Criteria

We compare KronDPBL against 15 baselines spanning five categories, following the grouping in the main text:

Shallow: WiAR (Guo et al., 2019) uses hand-crafted features with a linear classifier (∼2.8K parameters).

Convolutional: SignFi (Ma et al., 2018) employs a shallow CNN–MLP hybrid (∼527K). CLAR (Xiao et al., 2024) combines diffusion-model-based data augmentation with contrastive learning; our baseline trains its convolutional encoder end-to-end (∼14.9M). WiGr (Zhang et al., 2022a) uses 1D ResNet with multi-scale kernels (∼0.88M). WiGRUNT (Gu et al., 2022) extends ResNet-18 with a spatial-temporal dual-attention module (∼11.2M). Wi-ATCN (Zhu et al., 2022) combines convolutional layers with an attentional temporal convolutional network. HAR-SAnet (Chen et al., 2023) employs efficient depthwise separable convolutions with channel attention (∼44K).

Recurrent: WiFiTAD (Liu et al., 2025) employs a TSSE attention mechanism for temporal-spatial feature extraction (∼33.8M). Widar3.0 (Zhang et al., 2022b) combines Conv2D with GRU temporal modeling (∼100K). GUARD (Liang et al., 2026) performs feature reconstruction with global temporal self-attention (∼83K).

Transformer-based: THAT (Li et al., 2021) uses a lightweight transformer encoder with temporal self-attention (∼108K). CSI-BERT (Zhao et al., 2024) adapts BERT-style masked reconstruction for CSI (∼953K); we also evaluate its v2 variant (Zhao et al., 2026). WiFiTAD is listed under Recurrent above. ViT (Dosovitskiy et al., 2021) adapts a standard vision transformer with patch embedding (∼564K). PatchTST (Nie et al., 2023) uses channel-independent patching with transformer encoders (∼905K).

Prompting: Wi-Prompt (Zhao et al., 2025) is a prompt-based prototypical framework that freezes a pretrained backbone and learns only prompt tokens and prototype heads per task (∼70K trainable).

The 15 baselines span four orders of magnitude in parameter count (2.8K to 33.8M), covering the full spectrum of model capacities from hand-crafted features to large pretrained backbones.

Table 12: Baseline trainable parameter counts on CSI-80 AR-4^{C} (4-class, input dimension 1024), sorted by trainable parameters. KronDPBL uses a frozen encoder (7.11M pretrained) plus a lightweight MLP classifier head (512→256→128→64→4, 172,740 trainable parameters).

Model	Type	Trainable Params
WiFiTAD	Recurrent	33,800,452
CLAR	Convolutional	14,856,006
WiGRUNT	Convolutional	11,213,498
CSI-BERT	Transformer-based	952,910
PatchTST	Transformer-based	904,582
WiGr	Convolutional	874,502
ViT	Transformer-based	564,486
SignFi	Convolutional	526,758
KronDPBL	DPBL	172,740 (+ 7.11M frozen)
THAT	Transformer-based	108,446
Widar3.0	Recurrent	100,390
GUARD	Recurrent	83,422
Wi-Prompt	Prompting	69,840
HAR-SAnet	Convolutional	44,038
Wi-ATCN	Convolutional	—
WiAR	Shallow	2,838

KronDPBL's frozen-encoder protocol decouples representation learning from downstream classification: only 173K parameters (2.4% of the 7.28M total) are trained per task, while the 7.11M-parameter encoder requires only a per-fold pretraining run (∼25 minutes, Table 17) and is then frozen and shared by all downstream evaluations of that fold. In contrast, all baselines except CLAR and Wi-Prompt in Table 12 train their full parameter budget from scratch on every downstream setting. The smallest baseline (WiAR, 2.8K) is purely hand-crafted and lacks learned representations; the largest (WiFiTAD, 33.8M) requires 196× more trainable parameters per task than KronDPBL.

Table 13: KronDPBL encoder parameter breakdown on CSI-80 AR-4^{C} under the default configuration (D=1024, D_{model}=512, r=64, k=5, L=2, S=8). *Percentage of the 7.28M inference total.

Component	Parameters	% of Encoder
CSI Embedding (MLP, 1024 to 512)	1,837,056	25.9%
Temporal Positional Encoding (sinusoidal)	0	0%
Temporal Encoding MLP	787,968	11.1%
Kronecker Positional Encoding (64+16)	80	<0.1%
CPE Layer ×2 (PatternDist + CrossAttn)	2,244,608	31.6%
CNC Layer ×2 (CrossAttn)	2,103,296	29.6%
Other (LayerNorm, aggregation FFN)	132,609	1.9%
Encoder total (frozen)		7,105,617
Kronecker Recon Head (rank-8; discarded after pretraining)	853,632	–
MLP Classifier Head (512 to 256 to 128 to 64 to C)	172,740	2.4%*
Inference total (encoder + head)		7,278,357

This frozen-encoder design is the primary source of KronDPBL's practical efficiency advantage.

C.2 Hyperparameter Tuning

All baselines are tuned with a standardized protocol. For each baseline, we perform a grid search over: learning rate $\in \{10^{-2}, 10^{-3}, 10^{-4}\}$, weight decay $\in \{10^{-2}, 10^{-3}, 10^{-4}\}$, and batch size $\in \{32, 64\}$. We report the configuration achieving the best average validation accuracy across folds. For pretrained models (CLAR, Wi-Prompt), only the downstream head and adapter parameters are trained. All baselines use the AdamW optimizer with cosine annealing learning rate schedule (minimum LR = 10^{-6}), gradient clipping at 3.0, and early stopping with patience of 15 epochs on validation loss.

D Efficiency and Parameter Analysis

D.1 Parameter Budget per Component

Table 13 breaks down the KronDPBL encoder parameters by component using the default paper configuration (D=1024, D_{model}=512, r=64, k=5, L=2, S=8). The total encoder is approximately 7.11M parameters, which is frozen after pretraining and shared by all downstream evaluations of the same fold.

The CPE and CNC layers together account for 61.2% of the encoder parameters, reflecting the query, key, value, and output projections of the cross-attention blocks. The CSI embedding (input projection MLP) is the next largest single component at 25.9%. The Kronecker positional encoding contributes only 80 parameters while providing 2D spatial structure. The reconstruction head (853,632 parameters) is discarded after pretraining, reducing the inference-time footprint to the encoder plus the lightweight MLP head.

D.2 Per-Dataset Encoder Configuration

Table 14 compares the encoder size across datasets. All configurations share the same core architecture (r=64, k=5, D_{model}=512, L=2, S=8); only the input dimension D and Kronecker reshape layout differ.

Table 14: KronDPBL encoder configurations per dataset (unified r=64, k=5, D_{model}=512).

Dataset	D	Reshape	Params
CSI-80 AR	1024	64×16	7,105,617
CSI-80 PI	968	4×242	6,965,503
Widar3.0	90	3×30	5,585,952
WiGesture/WiFall/WiCount	52	4×13 / 13×4	5,561,046
CSI-Bench FD-2	232	8×29	5,704,526

The encoder parameter count is largely insensitive to the input dimension D because the dominant components (CPE–CNC cross-attention, PatternDistribution, temporal MLP) depend only on D_{model} and (r, k), not on D. Only the CSI embedding MLP scales with the input dimension ($\sim$0.29M at D=52 to $\sim$1.84M at D=1024), so the total frozen budget stays within 5.56–7.11M across all datasets. As a result, the encoder size varies by less than 30% across datasets.

D.3 Inference Efficiency

Table 15 compares the inference cost of KronDPBL against representative baselines on a standard CSI-80 segment (T=100, D=1024, batch size 1). MACs are computed analytically for a single forward pass. Latency is measured on an NVIDIA RTX 4090 GPU with FP32 precision, averaged over 1,000 forward passes after a 100-iteration warm-up. For KronDPBL, the reconstruction head and contrastive losses are absent during inference; only the frozen encoder and trained MLP classifier are executed.

KronDPBL achieves competitive inference speed ($\sim$3.1 ms per segment). The key efficiency factors are: (1) the CPE–CNC cross-attention operates on $\mathcal{O}((r+k) \cdot T)$ query–key pairs instead of the

Table 15: Inference efficiency comparison on CSI-80 AR-4^{C} (input 100×1024, 4-class output). KronDPBL uses a frozen encoder during inference.

Model	Params (total)	MACs (M)	Latency (ms)	Thru. (seg/s)
KronDPBL (enc+head)	∼7.28M	∼25	∼3.1	∼320
KronDPBL (linear head)	∼7.11M	∼25	∼3.1	∼320
WiFiTAD	33.8M	∼180	∼8.5	∼118
CLAR	14.9M	∼15,500*	∼12.3	∼81
CSI-BERT	7.8M	∼95	∼4.7	∼213
THAT	0.11M	∼12	∼1.1	∼909

*The CLAR entry uses a 2D convolutional feature extractor operating on a spectrogram representation; its high MAC count is not directly comparable to models operating on raw 1D CSI.

$\mathcal{O}(T^2)$ of full self-attention, keeping compute linear in T despite $r{=}64$ queries; (2) the reconstruction head is used only during pretraining and is removed at inference; (3) the frozen encoder output can be pre-computed and cached when segments are evaluated repeatedly. At batch size 1 the measured latency is dominated by fixed kernel-launch and framework overhead rather than compute; at batch size 64 the per-segment cost of KronDPBL drops to 0.064 ms on this setting.

D.4 Inference Cost Across Datasets

Table 16 reports the per-segment inference cost of KronDPBL across the six settings in Table 16. All measurements use FP32 precision, on an NVIDIA RTX 4090 GPU, averaged over 1,000 forward passes after a 100-iteration warm-up; only the frozen encoder and classifier head are executed (the reconstruction head is absent at inference). The encoder dominates inference cost; the MLP classifier head adds negligible overhead ($<$0.2 ms).

Table 16: Per-dataset inference cost for KronDPBL (frozen encoder + classifier head) at $T{=}100$. The Lat. column uses batch size 1; the last column reports per-segment latency at batch size 64, where fixed launch overhead amortizes and per-dataset compute differences become visible.

Dataset	T	D	Enc.+Head Params	Lat. (ms)	Lat. (ms/seg), $b{=}64$
CSI-80 AR-4	100	1024	7,278,357	3.13	0.0639
CSI-80 PI-11	100	968	7,138,698	3.17	0.0639
Widar3.0 AR-6	100	90	5,750,950	3.15	0.0555
WiGesture AR-6	100	52	5,726,044	3.14	0.0551
WiFall FD-2^{C}	100	52	5,725,528	3.11	0.0551
CSI-Bench FD-2^{T}	100	232	5,869,008	3.14	0.0560

At batch size 1, per-segment latency is nearly identical across datasets (3.11–3.17 ms) because it is dominated by fixed kernel-launch overhead, not compute. Latency is likewise flat in the segment length: at the datasets' native segment lengths latency stays within 3.07–3.17 ms, and a T-scan on CSI-80 AR from $T{=}100$ to $T{=}2000$ varies only between 2.94 and 3.04 ms. The compute differences that do exist across datasets are visible once overhead amortizes: at batch size 64 the per-segment cost ranges from 0.055 ms (Wi*, 52-dim) to 0.064 ms (CSI-80, 1024-dim), comfortably supporting real-time deployment.

D.5 Training Efficiency

Table 17 reports the pretraining and per-task head-training cost for KronDPBL. All measurements use a single NVIDIA RTX 4090 GPU.

With early stopping, pretraining stops after 50–80 of the scheduled epochs. Downstream head training is highly efficient: only the MLP classifier head is trained while the encoder remains frozen, requiring only ∼2 minutes per fold.

Table 17: Training cost for KronDPBL pretraining and downstream head training. P stands for pretraining; Head rows report downstream head training with the frozen encoder. Wall Time is the single-run wall-clock time on one RTX 4090 GPU.

Phase	Epochs	Wall Time
P(CSI-80, 16 recordings)	∼60	∼25 min
P(Widar3.0, 271K frames)	∼80	∼45 min
P(CSI-Bench, 4,690 samples)	∼50	∼10 min
Head (per fold)	∼30	∼2 min
Head (all 4 folds)	–	∼8 min

D.6 Head Capacity Ablation on Cross-Dataset Transfer

Table 18 examines the effect of classifier head capacity on the CSI-Bench → WiFall cross-dataset transfer task. The frozen CSI-Bench FD-pretrained KronDPBL encoder is used with varying MLP head sizes. A single linear layer (1,026 parameters) reaches 78.12% on this binary transfer task, within 0.76 percentage points of the mlp3-h512 head, which uses 500× more parameters.

Table 18: Classifier head capacity ablation on CSI-Bench → WiFall transfer (928 test samples). Head format: "mlp$L$-hD" = L hidden layers, D units per layer.

Head	Accuracy (%)	Trainable Params
mlp3-h512	**78.88**	526,338
mlp2-h512	78.77	263,682
mlp3-h256	77.91	198,146
mlp2-h128	73.28	66,050
Linear	78.12	1,026

D.7 Kronecker Structure Efficiency

Table 19 compares the parameter counts of the Kronecker-structured components in KronDPBL against unstructured alternatives.

Table 19: Parameter count comparison between KronDPBL's structured components and equivalent unstructured alternatives (D_{model}=512, 64×16 grid). Pos stands for Positional. Re Head stands for Reconstruction Head; Full denotes a dense 512→512→1024 decoder producing the same output. Ratio is Full divided by Kron.

Component	Kron	Full	Ratio
Pos Encoding (D=1024)	80 (64+16)	1,024	12.8×
Re Head (S=8)	853,632	787,968	0.92×
Re Head (S=16)	1,181,952	787,968	0.67×

The Kronecker-sum positional encoding uses only $D_{\text{row}} + D_{\text{col}} = 80$ parameters versus the $D_{\text{row}} \times D_{\text{col}} = 1024$ of a full spatial bias, a 12.8× reduction that encodes a decoupled antenna-subcarrier prior. The Kronecker-factorized reconstruction head factorizes a 64×16 CSI matrix into a sum of rank-1 outer products $\sum_s \mathbf{a}^{(s)} \otimes \mathbf{b}^{(s)}$, costing a modest parameter overhead relative to an unstructured dense decoder (853,632 vs. 787,968 at S=8) in exchange for a rank-S constraint on each reconstructed frame; the benefit of the factorization is regularization rather than compression.

D.8 Overall Parameter Budget and Inference Cost

Table 20 breaks down the KronDPBL encoder and classifier head sizes across evaluation settings. All use the unified default configuration (r=64, k=5, D_{model}=512). The encoder is frozen after

pretraining; only the MLP head is trained per task. Single-segment inference latency is measured on an NVIDIA RTX 4090 GPU with batch size 1, FP32 precision.

Table 20: KronDPBL parameter budget and per-segment inference latency (T=100, batch size 1).

Task	Encoder (frozen)	Cls. Head	Inf. Lat. (ms)
CSI-80 AR-4	7,105,617	172,740	∼3.1
Widar3.0 AR-6	5,585,952	164,998	∼3.1
WiGesture AR-6	5,561,046	164,998	∼3.1
CB→WiFall, mlp3-h512	5,704,526	526,338	∼3.1
CB→WiFall, linear	5,704,526	1,026	∼3.1

Table 21 compares KronDPBL's total and trainable parameter counts against representative baselines on CSI-80 AR-4^{C}.

Table 21: Parameter and inference cost comparison against baselines on CSI-80 AR-4^{C}. KronDPBL freezes the encoder and only trains the MLP head.

Model	Total Params	Trainable	Inf. Lat. (ms)
KronDPBL (frozen)	7,278,357	172,740	∼3.1
CSI-BERT	∼7,800,000	7,800,000	∼4.7
WiFiTAD	33,800,452	33,800,452	∼8.5
CLAR	14,856,006	14,856,006	∼12.3
WiGRUNT	11,213,498	11,213,498	–

Taken together, Tables 20 and 21 illustrate KronDPBL's asymmetric efficiency profile. The encoder is large enough to capture rich CSI representations through self-supervised pretraining, but the per-task training cost is minimal because only the MLP head is updated (172,740 trainable parameters on CSI-80 AR-4^{C}, 2.4% of the 7.28M total). At batch size 1, inference latency is uniform across datasets (∼3.1 ms) because fixed launch overhead dominates; at batch size 64 the per-segment cost is 0.055–0.064 ms across the six settings in Table 16. KronDPBL thus occupies a favorable point in the accuracy-efficiency Pareto frontier: high accuracy through pretrained representations, low per-task cost through the frozen-encoder protocol, and moderate inference latency compatible with real-time deployment.