Robust HAR with CSI Tokens: Single-Direction Prompt Guidance on a Frozen LLM

Fucheng Miao†, Osamu Takyu†, Lin Shan†, Ou Zhao†,
Tomoaki Ohtsuki‡, Guan Gui⋆, and Fumiyuki Adachi††

†Department of Engineering, Graduate School of Science and Technology, Shinshu University, Japan
‡Department of Information and Computer Science, Keio University, Japan
⋆College of Telecommunications and Information Engineering, NJUPT, Nanjing, China
†† International Research Institute of Disaster Science, Tohoku University, Sendai, Japan

***Abstract*—This paper presents a WiFi Channel State Information (CSI)–based human activity recognition framework that fuses class-aware textual prompts with CSI patch‘ tokens through a Prompt-Query Cross-Modal Attention front end. A lightweight "patch reprogramming" layer aligns CSI tokens to the language-embedding space, after which a frozen GPT-2 performs contextual modeling; only small task-specific heads are trained. To improve representation learning without altering inference, three optional training-only auxiliaries, supervised contrastive learning, masked reconstruction, and temporal order prediction—act as complementary regularizers. The overall objective is optimized as a simple weighted sum without uncertainty weighting. Experiments on two public CSI datasets show consistent gains over representative Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Residual Network (ResNet), and Vision Transformer (ViT) baselines, faster and smoother convergence, robustness under Signal-to-Noise Ratio (SNR) degradation (where SNR refers to the quality of the received WiFi signal in typical indoor multipath environments), and cleaner confusion matrices on hard class pairs. These results suggest that injecting textual priors and aligning CSI tokens before using a frozen LLM is an effective way to create stronger, noise-resilient CSI representations.**

***Index Terms*—WiFi sensing, channel state information (CSI), human activity recognition (HAR), large language models (LLMs), multi-task regularization, Deep Learning, noise robustness.**

I. Introduction

Non-contact human activity recognition (HAR) is valuable for smart homes, healthcare, and security. Compared with vision- and wearable-based approaches, WiFi-based HAR leverages ubiquitous infrastructure, provides through-wall capability, and preserves privacy, making it a practical solution for pervasive sensing and communication applications. [1], [2]. With commodity 802.11 devices exposing Channel State Information (CSI), researchers exploit multipath and Doppler changes induced by human motion to realize device-free recognition; For example, F. Meneghello [3] et al. proposed SHARP, which achieves person- and environment-independent recognition without modifying access points. Nevertheless, challenges remain in both the intrinsic characteristics of CSI, which exhibits substantial distribution shifts across subjects and environments, and the experimental settings, where several benchmarks suffer from split leakage, making reproducible and transferable evaluation more challenging [4], [5]. On the modeling side, deep learning has significantly advanced CSI-HAR. Vision Transformers have emerged as strong baselines, with systematic studies showing competitive representation capability on CSI [6], [7]; meanwhile, lightweight designs such as WiLDAR enable real-time inference on Raspberry-Pi–class hardware, markedly lowering the deployment barrier [8].

Recently, inspired by instruction following and foundation models, large language models (LLMs) have been explored for wireless sensing: by injecting physics-informed prompts and leveraging instruction-tuned reasoning, LLMs can process multi-modal or radio signals under zero-/few-shot conditions, showing early feasibility, e.g., SensorLLM for motion-sensor HAR [10], RadioLLM for cognitive radio with hybrid prompts and token reprogramming [11], WirelessGPT for multi-task learning in wireless communications [12], and Wi-Chat for LLM-powered Wi-Fi sensing [9]. However, how to effectively couple sequential CSI

979-8-3315-7111-5/26/$31.00 ©2026 IEEE

tokens with textual priors and consistently improve the main HAR task without fine-tuning the LLM remains underexplored and lacks systematic validation [13], [14].

To address the aforementioned challenges, we propose a CSI-based HAR framework that incorporates textual priors derived from activity labels and couples them with CSI patches via PQ-CMA. This framework does not require target-user data during training. In summary, our main contributions are:

- A frozen-LLM pipeline that slices CSI into patch tokens, encodes class prompts, and injects semantics through cross-attention before feeding the sequence as inputs_embeds to GPT-2; this avoids LLM fine-tuning, learns user-invariant features, and keeps the backbone unchanged at inference.
- A plug-and-play multi-task regularization strategy with three training-only branches, supervised contrastive learning, masked reconstruction, and temporal order prediction.These branches improve separability, local reconstructability, and temporal consistency while adding zero inference overhead by training only lightweight heads.
- Comprehensive experiments on two public datasets showing consistent gains over CNN/RNN/ResNet/ViT baselines. The model is robus across a wide SNR range with graceful degradation at low SNR. It also exhibits reduced dominant confusions in confusion matrix analysis and favorable parameter and compute efficiency for practical deployment.

II. System Model and Problem Formulation

A. Task Definition

Given a WiFi CSI sequence $\mathbf{X} \in \mathbb{R}^{T\times N}$, where T denotes the number of time steps and N the number of subcarriers, and a class-wise textual prompt $\texttt{text}$, we aim to learn a prompt-conditioned mapping

$$f_\Theta : (\mathbf{X}, \texttt{text}) \mapsto \hat{\mathbf{y}} \in \mathbb{R}^C, \tag{1}$$

that outputs class logits $\hat{\mathbf{y}}$. The large language model (LLM, GPT-2) remains frozen during both training and inference.

B. Inputs and Intermediate Representations

Given a WiFi-CSI sequence $\mathbf{X} \in \mathbb{R}^{T\times N}$, where T is the number of time steps and N is the number of subcarriers, we first divide the sequence into overlapping patches along the time axis. Using a window length P and stride S, the total number of generated patches is

$$L = \left\lfloor \frac{T-P}{S} \right\rfloor + 1, \tag{2}$$

where L is the number of patches. Each patch has a size of $P \times N$. Each patch is flattened and linearly projected to an embedding of dimension H, producing a sequence

$$\mathbf{V} = \mathrm{PatchEmbed}_{\theta_v}(\mathbf{X}) \in \mathbb{R}^{B\times L\times H}, \tag{3}$$

where B is the batch size, θ_v denotes the parameters of the projection, and $\mathbf{V}$ is the sequence of patch embeddings.

To align CSI embeddings with the input space of the language model, we apply a lightweight linear transformation:

$$\tilde{\mathbf{V}} = \mathrm{Linear}_{\theta_r}(\mathbf{V}) \in \mathbb{R}^{B\times L\times H}, \tag{4}$$

where θ_r are the learnable parameters, and $\tilde{\mathbf{V}}$ denotes the reprogrammed patch embeddings.

In parallel, class-related priors are converted into textual prompts. After tokenization and embedding lookup, the textual sequence is represented as

$$\mathbf{P}_0 = \mathrm{wte}(\texttt{text}) \in \mathbb{R}^{B\times L_p\times H}, \tag{5}$$

where L_p is the number of prompt tokens and wte denotes the frozen word embedding table.

The prompt embeddings $\mathbf{P}_0$ and the reprogrammed CSI embeddings $\tilde{\mathbf{V}}$ are concatenated and fed into a frozen large language model (GPT-2), yielding

$$\mathbf{U} = [\,\mathbf{P}_0;\, \tilde{\mathbf{V}}\,] \in \mathbb{R}^{B\times(L_p+L)\times H}, \quad \mathbf{H} = \text{GPT-2}(\mathbf{U}), \tag{6}$$

where $\mathbf{U}$ is the combined sequence and $\mathbf{H} \in \mathbb{R}^{B\times(L_p+L)\times H}$ is the contextualized hidden representation.

From $\mathbf{H}$, we obtain pooled representations for prompts and CSI patches:

$$\bar{\mathbf{h}}_p = \mathrm{Mean}\big(\mathbf{H}_{[:,:L_p,:]}\big), \quad \bar{\mathbf{h}}_x = \mathrm{Mean}\big(\mathbf{H}_{[:,L_p:,:]}\big), \tag{7}$$

where $\bar{\mathbf{h}}_p \in \mathbb{R}^{B\times H}$ and $\bar{\mathbf{h}}_x \in \mathbb{R}^{B\times H}$ denote the average pooled prompt and patch features, respectively.

Finally, we concatenate them into a joint representation

$$\mathbf{z} = \begin{bmatrix} \bar{\mathbf{h}}_x \\ \bar{\mathbf{h}}_p \end{bmatrix} \in \mathbb{R}^{B\times 2H}, \tag{8}$$

and generate predictions through a lightweight classifier:

$$\hat{\mathbf{y}} = \mathrm{MLP}_{\theta_c}(\mathbf{z}) \in \mathbb{R}^{B\times C}, \qquad \mathcal{L}_{\text{ce}} = \mathrm{CE}(\hat{\mathbf{y}}, \mathbf{y}), \tag{9}$$

where C is the number of classes, $\hat{\mathbf{y}}$ are the predicted logits, $\mathbf{y}$ are the ground-truth labels, and $\mathcal{L}_{\text{ce}}$ is the cross-entropy loss.

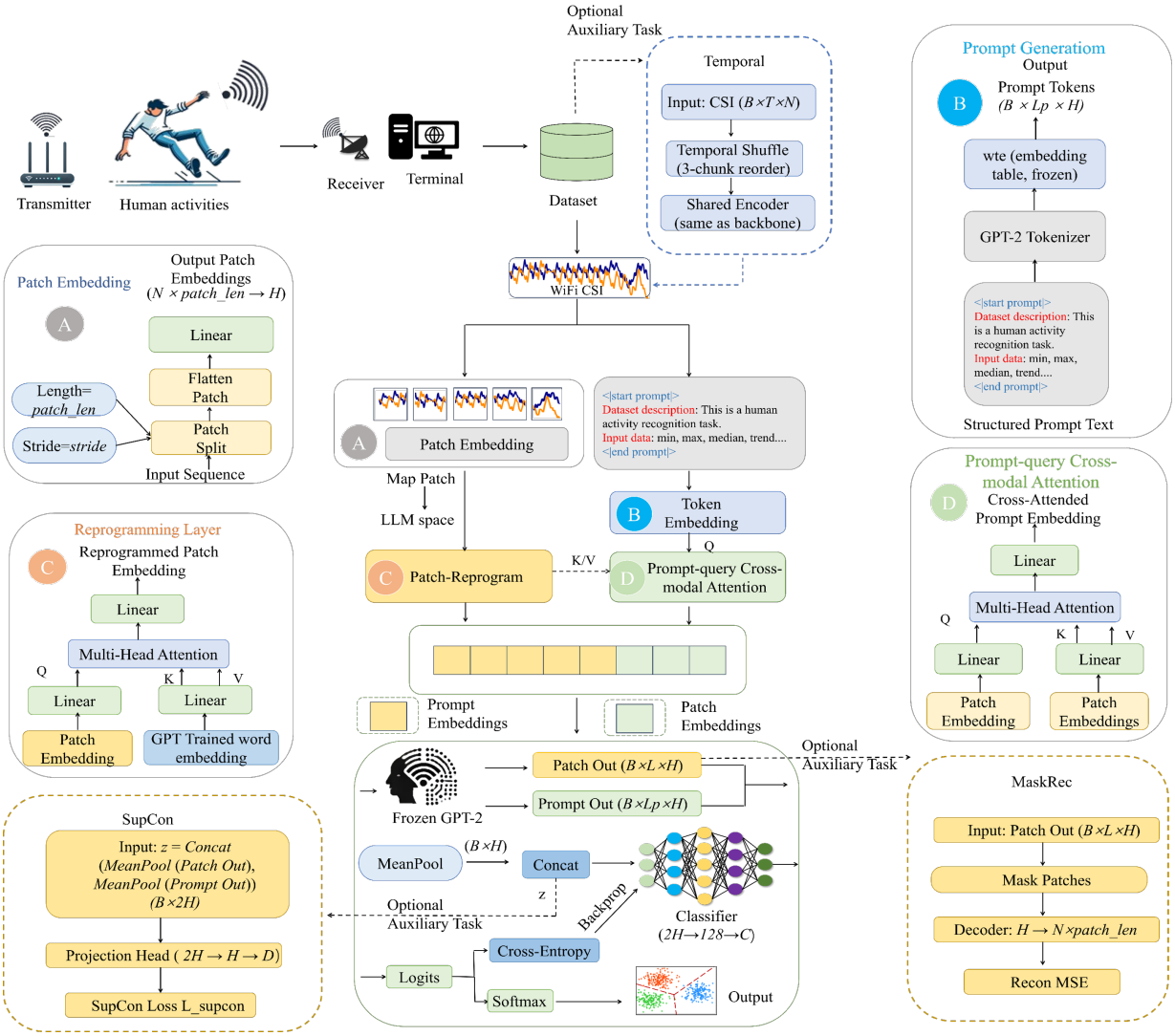


Fig. 1: Overview of the proposed Wi-Fi CSI–HAR pipeline. CSI patches interact with class prompts via cross-attention and feed a frozen GPT-2; SupCon/MaskRec/Temporal are optional training losses. At inference, only the backbone and a light MLP head are used.

C. Plug-and-Play Regularized Objective

We study training under a frozen-LLM backbone. Given input $\mathbf{X}$ with label y, the backbone produces $\mathbf{z}$, and only lightweight heads (parameters Θ) are optimized. We adopt a plug-and-play scheme with optional branches indexed by $\mathcal{A} = \{\text{sup}, \text{rec}, \text{temp}\}$; each branch is controlled by a switch $s_k \in \{0, 1\}$ and weighted by $\alpha_k \geq 0$. Switches may be on during training and are off at inference so the runtime stays the same as a plain classifier.

The goal is to learn a parameter-efficient model that improves class separability, preserves local structure, and respects temporal order, while keeping the backbone frozen and the inference path unchanged. SupCon is used to enhance discriminative margins in the embedding space; MaskRec encourages recovery of masked local tokens to retain fine-grained cues; Temporal probes sequence order with simple perturbations to promote temporal coherence.

This subsection only states the problem and design intent. Concrete losses, architectures, and training details are deferred to the next section. We emphasize modularity and head-only adaptation, enabling easy ablations and clean integration with existing pipelines and deployment constraints in practice.

III. Our Proposed Method

A. Overview of the Proposed Method

We propose a PQ-CMA framework with a frozen GPT-2 backbone for WiFi-CSI based HAR. The overall pipeline is illustrated in Fig. 1. CSI is first segmented into patch tokens (Block A). Class names and domain priors are composed into a structured prompt and embedded by the frozen word embedding table of GPT-2 (Block B). We then inject semantics by attending with the prompt as queries over CSI patches (Block D). The prompt-enhanced sequence, concatenated with the reprogrammed patches (Block C), is fed to GPT-2 as `inputs_embeds` for contextual modeling. Mean-pooled representations from the prompt and the patch segments are concatenated and classified by a lightweight MLP head. During training, we optionally enable three auxiliary branches, SupCon, MaskRec, and Temporal—which act as plug-in regularizers. At inference, only the backbone is executed, adding no runtime overhead [15]. This design highlights our key contributions of a frozen-LLM pipeline, plug-and-play regularization, and robustness under varying SNR conditions.

B. Block A: Patch Embedding

Let $\mathbf{X} \in \mathbb{R}^{B\times T\times N}$ denote a mini-batch of CSI samples with T time steps and N subcarriers. A sliding window of length P and stride S partitions the sequence along time into

$$L = \left\lfloor \frac{T-P}{S} \right\rfloor + 1 \tag{10}$$

patches (zero-pad if necessary). Each $P \times N$ patch is flattened and linearly projected to width H, yielding

$$\mathbf{V} = \mathrm{PatchEmbed}_{\theta_v}(\mathbf{X}) \in \mathbb{R}^{B\times L\times H}, \tag{11}$$

with optional LayerNorm/Dropout consistent with implementation.

C. Block B: Prompt Generation

Class names and concise priors are filled into a fixed template to form a structured prompt `text`. After the GPT-2 tokenizer and the frozen word-embedding table wte, we obtain

$$\mathbf{P}_0 = \mathrm{wte}(\texttt{text}) \in \mathbb{R}^{B\times L_p\times H}, \tag{12}$$

where L_p is the number of prompt tokens. No Transformer parameters are updated at this stage. In the subsequent cross-modal attention module (Block D), $\mathbf{P}_0$ serves as the **query (Q)**, while CSI patches provide the key and value.

D. Block C: Patch Reprogramming (Lightweight Alignment)

To reduce the distribution gap between CSI tokens and LLM word embeddings, we apply a lightweight affine remapping before cross-attention:

$$\tilde{\mathbf{V}} = \mathrm{Linear}_{\theta_r}(\mathbf{V}) \in \mathbb{R}^{B\times L\times H}. \tag{13}$$

This module is parameter-efficient yet crucial for matching the LLM embedding space.This linear adapter follows the reprogramming perspective of aligning non-linguistic sequences to a frozen LLM embedding space, as in Time-LLM for time-series forecasting [15].

E. Block D: PQ-CMA and Frozen GPT-2

We inject semantics by attending with the prompt as query and CSI patches as key/value:

$$\begin{aligned} &\mathbf{P}^* = \mathrm{MHA}_{\theta_a}\Big(\mathbf{Q}{=}\mathbf{P}_0,\ \mathbf{K}{=}\tilde{\mathbf{V}},\ \mathbf{V}{=}\tilde{\mathbf{V}}\Big),\\ &\mathbf{P}^* \in \mathbb{R}^{B\times L_p\times H}. \end{aligned} \tag{14}$$

The prompt-enhanced tokens are concatenated with the reprogrammed patches and fed to frozen GPT-2 as `inputs_embeds`:

$$\mathbf{U} = [\,\mathbf{P}^*;\,\tilde{\mathbf{V}}\,] \in \mathbb{R}^{B\times(L_p+L)\times H},\ \mathbf{H} = \mathrm{GPT}{-}2(\mathbf{U}). \tag{15}$$

We split $\mathbf{H}$ into the prompt and patch segments and aggregate them via mean pooling:

$$\begin{aligned} &\bar{\mathbf{h}}_p = \mathrm{Mean}(\mathbf{H}_{[:,:L_p,:]}),\\ &\bar{\mathbf{h}}_x = \mathrm{Mean}(\mathbf{H}_{[:,L_p:,:]}), \end{aligned} \qquad \mathbf{z} = [\,\bar{\mathbf{h}}_x;\,\bar{\mathbf{h}}_p\,] \in \mathbb{R}^{B\times 2H}. \tag{16}$$

In contrast to Block C, this block performs cross-modal attention and contextual modeling.

F. Classifier and Training/Inference Paths

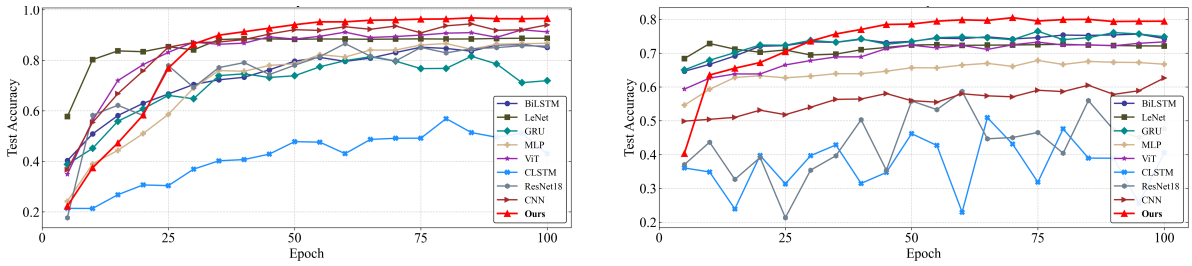
A two-layer MLP produces logits and the main loss:

$$\hat{\mathbf{y}} = \mathrm{MLP}_{\theta_c}(\mathbf{z}) \in \mathbb{R}^{B\times C}, \qquad \mathcal{L}_{\mathrm{ce}} = \mathrm{CE}(\hat{\mathbf{y}}, \mathbf{y}). \tag{17}$$

We explicitly separate the training path (logits $\rightarrow$ cross-entropy) from the inference path (logits $\rightarrow$ softmax/argmax $\rightarrow$ output), which is also reflected in Fig. 1.

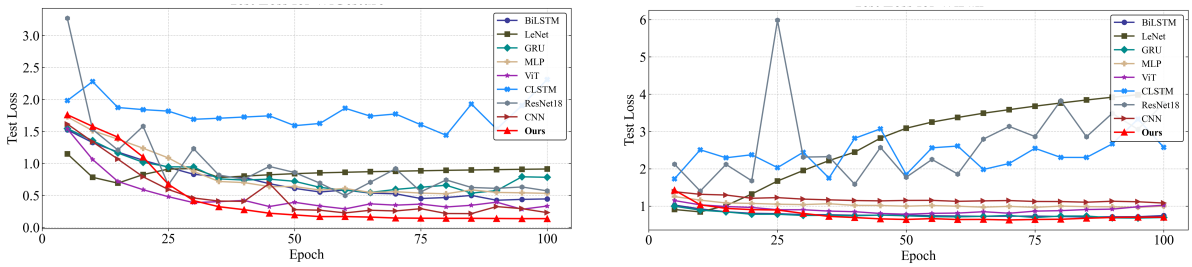
G. Optional Auxiliary Branches (Training Only)

SupCon (left dashed block). Given $\mathbf{z}$, a projection head yields $\mathbf{r} = \mathrm{Proj}_{\theta_p}(\mathbf{z}) \in \mathbb{R}^{B\times D}$, and the multi-



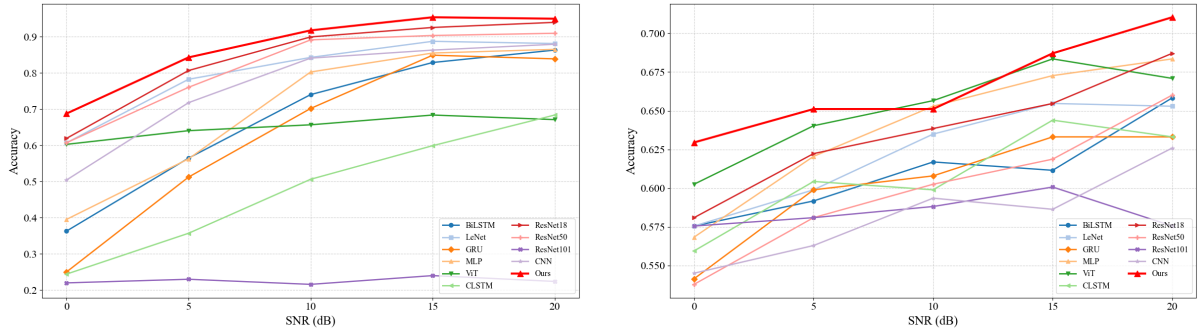
(a) WiGesture (b) WiFall

Fig. 2: Test accuracy vs. epoch on two datasets. Our method converges faster and reaches a higher plateau.



(a) WiGesture (b) WiFall

Fig. 3: Test loss vs. epoch. Lower and smoother curves indicate stable optimization.



(a) WiGesture (b) WiFall

Fig. 4: Accuracy under different SNR levels (0 $\sim$ 20 dB). Our method maintains the best or tied-best across SNRs.

positive supervised contrastive loss with temperature τ is

$$\mathcal{L}_{\text{sup}} = -\frac{1}{B}\sum_{i}\frac{1}{|P(i)|}\sum_{p\in P(i)}\log \frac{\exp\left(\frac{\text{sim}(\mathbf{r}_i,\mathbf{r}_p)}{\tau}\right)}{\sum_{a\neq i}\exp\left(\frac{\text{sim}(\mathbf{r}_i,\mathbf{r}_a)}{\tau}\right)}. \tag{18}$$

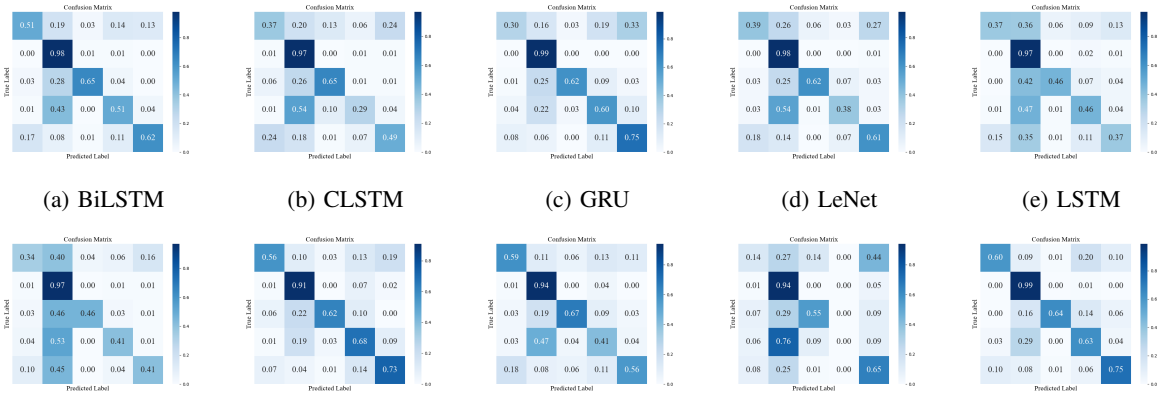
MaskRec (right dashed block). On the patch segment, we randomly mask a subset $\mathcal{M}$, $|\mathcal{M}| = K$. A linear decoder reconstructs the original flattened patch $\tilde{\mathbf{x}}_m \in \mathbb{R}^{N\cdot P}$ from hidden token $\mathbf{h}_m$:

$$\mathcal{L}_{\text{rec}} = \frac{1}{K}\sum_{m\in\mathcal{M}}\left\|\text{Dec}_{\theta_d}(\mathbf{h}_m) - \tilde{\mathbf{x}}_m\right\|_2^2. \tag{19}$$

Temporal (top dashed block). We create a perturbed sequence $\mathbf{X}'$ via three-chunk reordering, reuse the backbone to get $\mathbf{z}'$, and train a binary head:

$$\hat{\mathbf{o}} = \text{Head}_{\theta_t}(\mathbf{z}'), \qquad \mathcal{L}_{\text{temp}} = \text{CE}(\hat{\mathbf{o}}, \mathbf{o}). \tag{20}$$

All three branches are plug-and-play: they contribute to training when enabled and are disabled during



(a) BiLSTM (b) CLSTM (c) GRU (d) LeNet (e) LSTM

(f) MLP (g) ResNet18 (h) ViT (i) CNN (j) Ours

Fig. 5: Row-normalized confusion matrices on WiFall. Baselines include MLP, LeNet/AlexNet [18], [19], ResNet [20], LSTM/BiLSTM/GRU [22]–[24], CLSTM [25], and ViT [21]. The proposed method shows the cleanest diagonal and mitigates common confusions.

inference.

H. Objective and Optimization

We adopt a simple weighted sum without uncertainty weighting:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{ce}}\mathcal{L}_{\text{ce}} + \lambda_{\text{sup}}\mathcal{L}_{\text{sup}} + \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{temp}}\mathcal{L}_{\text{temp}}. \tag{21}$$

By default $\lambda_{\text{ce}} = 1$, while the auxiliary weights are chosen on a validation set. We optimize $\Theta = \{\theta_v, \theta_r, \theta_a, \theta_c\} \cup \{\theta_p, \theta_d, \theta_t\}$ (when the corresponding branch is enabled). GPT-2 and its word-embedding table remain frozen throughout.

IV. Experimental Results and Analysis

A. Convergence and Generalization

Fig. 2 depicts the evolution of test accuracy on WiGesture [16] and WiFall [17], while Fig. 3 reports the corresponding test loss. Our method converges faster and reaches a higher plateau than CNN/RNN/ResNet/ViT baselines [18]–[25] on WiGesture, while maintaining competitive and stable performance on WiFall. The training loss curves are lower and smoother, suggesting stable optimization. This observation is consistent with freezing GPT-2 and training only lightweight heads, which helps mitigate overfitting and reduce variance.

B. Noise Robustness Across SNR

We evaluate robustness under different Signal-to-Noise Ratio (SNR) levels, where SNR denotes the ratio of received signal power to background noise power in the WiFi channel, reflecting the quality of the wireless propagation environment. Fig. 4 summarizes accuracy under additive noise from 0 to 20 dB. Across all SNR levels, our method outperforms or matches the best competitors. The margin becomes more pronounced at high SNR ($15 \sim 20$ dB), while at low SNR ($0 \sim 5$ dB) our accuracy degrades more gracefully than ViT/ResNet families, suggesting that the prompt-conditioned cross-attention yields noise-resilient CSI representations.

C. Class-wise Behavior: Confusion Matrices

We analyze row-normalized confusion matrices across representative backbones, BiLSTM, LSTM, GRU, CLSTM, LeNet, CNN, ResNet18, and ViT, together with the proposed method. For brevity, Fig. 5 reports WiFall only; the WiGesture matrices are omitted due to space, show the same trend, and are provided in the supplementary material. The proposed method yields the cleanest main diagonal and markedly reduces the common ambiguities between sit/stand and jump/walk, which frequently challenge inertial/CSI systems.

RNN variants tend to spill over from stand to fall or walk, whereas ResNet18/ViT alleviate some errors yet still confuse sit vs. stand. By injecting class-aware prompts and aligning CSI patches before the frozen GPT-2, the prompt–patch cross-attention produces more discriminative representations, improving true positives for critical classes such as fall and suppressing false alarms. The omitted WiGesture matrices show analogous improvements, especially for visually similar motions.

V. Conclusion

We presented a CSI-based HAR framework that couples CSI patches with class-aware prompts via cross-attention on a frozen GPT-2. It outperforms CNN/RNN/ResNet/ViT baselines, converges faster, and is robust under SNR degradation while keeping inference efficient. Future work will present more comprehensive visualizations and explore larger or domain-adapted LLMs for broader generalization.

Acknowledgements

This work was supported by JSPS KAKENHI under Grant Number JP24K00882.

References

[1] F. Miao, Y. Huang, Z. Lu, T. Ohtsuki, G. Gui, and H. Sari, "Wi-Fi sensing techniques for human activity recognition: Brief survey, potential challenges, and research directions," *ACM Comput. Surv.*, vol. 57, no. 5, art. 107, pp. 1–30, May 2025.

[2] X. Li, Y. Cui, J. A. Zhang, F. Liu, D. Zhang, and L. Hanzo, "Integrated human activity sensing and communications," *IEEE Commun. Mag.*, vol. 61, no. 5, pp. 90–96, May 2023.

[3] F. Meneghello, D. Garlisi, N. Dal Fabbro, I. Tinnirello, and M. Rossi, "SHARP: Environment and person independent activity recognition with commodity IEEE 802.11 access points," *IEEE Trans. Mobile Comput.*, vol. 22, no. 10, pp. 6160–6175, Oct. 2023.

[4] D. Varga, "Mitigating data leakage in a WiFi CSI benchmark for human activity recognition," *Sensors*, vol. 24, no. 24, Art. 8201, 2024.

[5] N. Rashid, B. U. Demirel, and M. A. Al Faruque, "AHAR: Adaptive CNN for energy-efficient human activity recognition in low-power edge devices," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13041–13051, Aug. 2022.

[6] F. Luo, S. Khan, B. Jiang, and K. Wu, "Vision Transformers for human activity recognition using WiFi channel state information," *IEEE Internet Things J.*, vol. 11, no. 17, pp. 28111-28122, 1 Sept.1, 2024.

[7] Y. Ma, G. Zhou, and S. Wang, "WiFi sensing with channel state information: A survey," *ACM Comput. Surv.*, vol. 52, no. 3, Art. 46, pp. 1–36, May 2020.

[8] F. Deng, E. Jovanov, H. Song, W. Shi, Y. Zhang, and W. Xu, "WiLDAR: WiFi signal-based lightweight deep learning model for human activity recognition," *IEEE Internet Things J.*, vol. 11, no. 2, pp. 2899–2908, Jan. 2024.

[9] H. Zhang, Y. Ren, H. Yuan, J. Zhang, and Y. Shen, "Wi-Chat: Large language model powered Wi-Fi sensing," *arXiv preprint* arXiv:2502.12421, 2025.

[10] Z. Li, S. Deldari, L. Chen, H. Xue, and F. Salim, "SensorLLM: Aligning Large Language Models with Motion Sensors for Human Activity Recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2025.

[11] S. Chen, Y. Zu, Z. Feng, S. Yang, and M. Li, "RadioLLM: Introducing Large Language Model into Cognitive Radio via Hybrid Prompt and Token Reprogrammings," *arXiv preprint* arXiv:2501.17888, 2025.

[12] T. Yang, P. Zhang, M. Zheng, Y. Shi, L. Jing, and J. Huang, "WirelessGPT: A Generative Pre-trained Multi-task Learning Framework for Wireless Communication," *IEEE Netw.*, vol. 39, no. 9, pp. 58–65, Sept. 2025.

[13] Z. He, M. Bouazizi, Y. Yin, G. Gui, and T. Ohtsuki, "Robust cross-scenario Wi-Fi wireless sensing using incremental learning and elastic weight consolidation loss," *IEEE Internet Things J.*, vol. 12, no. 13, pp. 24288–24299, July 2025.

[14] K. Niu, F. Zhang, X. Wang, Q. Lv, H. Luo, and D. Zhang, "Understanding WiFi signal frequency features for position-independent gesture sensing," *IEEE Trans. Mob. Comput.*, vol. 21, no. 11, pp. 4156–4171, Nov. 2022.

[15] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen,"Time-LLM: Time series forecasting by reprogramming large language models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[16] H. Abdelnasser, M. Youssef, and K. A. Harras, "WiGest: A ubiquitous WiFi-based gesture recognition system," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2015, pp. 1472–1480.

[17] Y. Wang, K. Wu, and L. M. Ni, "WiFall: Device-free fall detection by wireless networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 2, pp. 581–594, Feb. 2017.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2012, pp. 1097–1105.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[21] A. Dosovitskiy *et al.*, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021. [Online]. Available: arXiv:2010.11929

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[23] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[24] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[25] X. Shi *et al.*, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2015, pp. 802–810.