CHALMERS
UNIVERSITY OF TECHNOLOGY

Large Wireless Localization Model (LWLM): A Foundation Model for Positioning in 6G Networks

Citation for the original published paper (version of record):
Pan, G., Huang, K., Chen, H. et al (2026). Large Wireless Localization Model (LWLM): A Foundation Model for Positioning in 6G Networks. IEEE Transactions on Wireless Communications, In Press. http://dx.doi.org/10.1109/TWC.2026.3718678

(article starts on next page)

Large Wireless Localization Model (LWLM): A Foundation Model for Positioning in 6G Networks

Guangjin Pan, *Member, IEEE*, Kaixuan Huang, *Graduate Student Member, IEEE*, Hui Chen, *Member, IEEE*, Shunqing Zhang, *Senior Member, IEEE*, Christian Häger, *Member, IEEE*, and Henk Wymeersch, *Fellow, IEEE*

***Abstract*—Accurate and robust localization is a critical enabler for emerging 5G and 6G applications, including autonomous driving, extended reality (XR), and smart manufacturing. While data-driven approaches have shown promise, most existing models require large amounts of labeled data and struggle to generalize across deployment scenarios and wireless configurations. To address these limitations, we propose a foundation-model-based solution tailored for wireless localization. We first analyze how different self-supervised learning (SSL) tasks acquire general-purpose and task-specific semantic features based on information bottleneck (IB) theory. Building on this foundation, we design a pretraining methodology for the proposed Large Wireless Localization Model (LWLM). Specifically, we propose an SSL framework that jointly optimizes three complementary objectives: 1) spatial-frequency masked channel modeling (SF-MCM), 2) domain-transformation invariance (DTI), and 3) position-invariant contrastive learning (PICL). These objectives jointly capture the underlying semantics of wireless channel from multiple perspectives. We further design lightweight decoders for key downstream tasks, including time-of-arrival (ToA) estimation, angle-of-arrival (AoA) estimation, single base station (BS) localization, and multiple BS localization. Comprehensive experimental results confirm that LWLM consistently surpasses both model-based and supervised learning baselines across all localization tasks. In particular, LWLM achieves 26.0%–87.5% improvement over transformer models without pretraining, and exhibits strong generalization under label-limited fine-tuning, unseen BS configurations and cross-dataset evaluations, confirming its potential as a foundation model for wireless localization.**

***Index Terms*—Wireless localization, foundation model, self-supervised learning, 6G networks, transformer.**

Received 22 October 2025; revised 19 March 2026 and 21 May 2026; accepted 22 July 2026. Date of publication 8 August 2026; date of current version 10 August 2026. This work was supported in part by the Chalmers AI Research Center Consortium (CHAIR), in part by the Smart Networks and Services (SNS) Joint Undertaking (JU) Project 6G-DISAC under European Union's Horizon Europe Research and Innovation Program under Grant 101139130, in part by Swedish Foundation for Strategic Research (SSF) funded by Software Artificial Intelligence for Communications (SAICOM) under Grant FUS21-0004, in part by the Chalmers Areas of Advance in Information and Communication Technology (ICT) and Transport, and in part by the Computational Resources were provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) funded by Swedish Research Council under Grant 2022-06725. The work of Kaixuan Huang and Shunqing Zhang was supported in part by the National Key Research and Development Program of China under Grant 2022YFB2902304 and in part by the National Natural Science Foundation of China under Grant 62571307. The work of Hui Chen was supported by EU's Horizon Europe Research and Innovation Program under the Marie Skłodowska-Curie under Grant 101201808. The work of Christian Häger was supported by Swedish Research Council under Grant 2020-04718. The associate editor coordinating the review of this article and approving it for publication was C. Han. *(Corresponding authors: Hui Chen; Shunqing Zhang.)*

Guangjin Pan, Christian Häger, and Henk Wymeersch are with the Department of Electrical Engineering, Chalmers University of Technology, 41296 Gothenburg, Sweden (e-mail: guangjin.pan@chalmers.se; christian.haeger@chalmers.se; henkw@chalmers.se).

Kaixuan Huang and Shunqing Zhang are with the School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China (e-mail: xuan1999@shu.edu.cn; shunqing@shu.edu.cn).

Hui Chen is with the Department of Electrical Engineering, Chalmers University of Technology, 41296 Gothenburg, Sweden, also with the Department of Electronic and Electrical Engineering, University College London, WC1E 7JE London, U.K., and also with the Department of Electrical Engineering, Uppsala University, 751 05 Uppsala, Sweden (e-mail: hui.chen@chalmers.se).

Digital Object Identifier 10.1109/TWC.2026.3718678

I. Introduction

As WIRELESS systems evolve beyond basic communication functions, precise and reliable localization has become an essential capability, playing a crucial role in fifth generation (5G) and upcoming sixth generation (6G) networks [1]. With location information, a wide array of applications can be supported, including autonomous vehicles, uncrewed aerial vehicle (UAV) coordination, and extended reality (XR) [2]. Accurate location information also enhances system-level functionalities such as mobility management [3], resource scheduling [4], and interference mitigation [5]. Wireless localization is usually addressed using model-based approaches, including time-of-arrival (ToA), angle-of-arrival (AoA), and received signal strength (RSS) measurements [6]. Classical algorithms such as multiple signal classification (MUSIC) and orthogonal matching pursuit (OMP) can estimate distance and angle between the user equipment (UE) and the base station (BS) for localization [7]. However, these methods degrade in multipath-rich and non-line-of-sight (NLoS) scenarios, especially under limited infrastructure. To overcome this, artificial intelligence (AI)-based localization [8] has emerged as a promising technique, allowing AI models to learn direct mappings from channel state information (CSI) to position without the need for explicit channel modeling.

Data-driven supervised learning methods train models to map CSI to user locations using large-scale labeled datasets. To enhance the model's fitting ability and accuracy, various neural network architectures have been studied, including multilayer perceptrons (MLPs) [9], convolutional neural networks (CNNs) [10], [11], and long short-term memory (LSTM) networks [12], [13]. With rapid advances in AI techniques, transformers, which have revolutionized fields such as natural language processing (NLP) and computer vision (CV), are also showing their strong potential in localization tasks [14], [15], [16], [17]. Nevertheless, most of these supervised learning models are task-specific and require a large amount of labeled

data for each new deployment scenario. Their heavy sample dependence leads to high deployment overhead and poor generalization across diverse environments and configurations.

To reduce the dependency on labeled data, semi-supervised and unsupervised learning approaches have been proposed. Some studies adopt autoencoders or generative adversarial networks (GANs) to reconstruct or simulate CSI without relying on ground-truth positions. For instance, iPos-5G [18] uses an autoencoder to extract CSI features for indoor localization. In [19], the authors leverage variational autoencoder (VAE)-based data augmentation and joint classification-reconstruction networks for domain-adaptive localization. The authors in [20] further propose a semi-crowdsourced radio map construction framework using GAN-based augmentation to reduce annotation effort. Other works apply transfer learning and domain adaptation to reduce distribution shifts between source and target domains [21]. While these methods alleviate labeling costs to some extent, they typically do not learn robust, location-dependent, and transferable representations. In particular, they fail to capture high-level semantic features, like spatial relationships or environment-dependent channel patterns, which can be exploited to support a wide range of channel-related tasks and improve their performance. As a result, their performance often deteriorates in heterogeneous scenarios, such as varying bandwidths and pilot configurations.

In recent years, the foundation model paradigm [22] using Large AI Models (LAMs) has achieved transformative success in NLP, CV, and audio processing. These models are pre-trained on large-scale unlabeled datasets using self-supervised learning (SSL) [23] to acquire general-purpose, transferable representations, as exemplified by BERT [24], generative pre-trained Transformer (GPT) [25], and SpectralGPT [26]. Such models serve as universal encoders and have shown significant improvements in semantic understanding, domain transferability, and fine-tuning efficiency [27], [28]. Based on this paradigm, wireless communications can also benefit from the foundation model paradigm to overcome the problems of scarce labeled data and insufficient generalization capabilities for scenarios, environments, and configurations [29], [30], [31], [32], [33]. A well-designed wireless foundation model can serve as a universal semantic encoder, capturing physical propagation characteristics that generalize across deployments, hardware, and tasks. By formulating appropriate SSL tasks over large-scale unlabeled CSI datasets, SSL enables neural networks to discover the latent semantics of wireless channels. Combined with transformer and well-designed SSL tasks [23], such models have the potential to serve as wireless foundation models, and can be fine-tuned for a wide range of downstream tasks.

SSL algorithms mainly include generative-based and contrastive-based approaches [23]. Building on these techniques, several recent works have explored wireless foundation models for general channel-related tasks. Based on a generative-based approach, LWM [29] applies masked reconstruction in the spatial-frequency domain for downstream tasks such as channel estimation and beamforming. WirelessGPT [30] extends masked modeling to three-dimensional (3D) channel representations across spatial, time, and frequency, achieving superior performance in channel estimation, prediction, human activity recognition, and environment reconstruction. WiFo [31] unifies frequency- and time-domain channel prediction under a masked SSL formulation and exhibits strong zero-shot generalization. Based on a contrastive-based approach, the authors in [34] employ data augmentation techniques (e.g., random subcarrier selection and random subcarrier flipping) to construct positive samples, enabling the learning of invariant representations of wireless channels. In addition, LLM4CP [32] and LLM4WM [33] directly fine-tune large language models (LLMs) for wireless channel tasks, transferring language representations to channel representations. Although these works demonstrate the feasibility and effectiveness of foundation models in wireless tasks and may perform well in localization scenarios, none of them is specifically designed for wireless localization. Moreover, contrastive learning has also been employed to capture the correlations between channel samples for low-dimensional channel charting [35], [36], [37], [38]. Although such charting captures channel-sample similarities, it fails to provide generalizable representations for diverse downstream tasks.

In addition, some prior works have proposed SSL-based pretraining frameworks for localization. For example, CrowdBERT [39] applies a masking strategy over RSS fingerprints from multiple access points and fine-tunes the model with limited labeled data. In [40], a signal-guided masked autoencoder uses antenna-domain masking and channel attention to reconstruct channel impulse responses (CIRs) for multi-BS localization. In [41], the paper reconstructs masked CIRs to extract environment-specific semantics, enabling accurate single-BS localization. However, these methods [39], [40], [41] target specific localization tasks, lacking generalization across tasks (e.g., AoA/ToA estimation, single/multi-BS localization) and different BS configurations. Moreover, these methods typically adopt a single SSL pretraining objective, limiting the diversity of the learned channel semantics.

Although prior works have applied SSL to localization and other wireless tasks with promising results, they generally lack a principled theoretical analysis to explain why SSL is effective in the wireless domain and typically rely on a single type of SSL objective, leading to insufficiently rich feature representations. In this paper, we propose a *Large Wireless Localization Model (LWLM)*,[1] a self-supervised foundation model tailored for wireless localization. LWLM is designed to address the key challenges in generalization, adaptability, and data efficiency across diverse localization tasks and deployment configurations.[2] The main contributions are as follows:

- **IB-motivated design principles for hybrid SSL**: Motivated by the information bottleneck (IB) framework, we analyze how different SSL objectives contribute complementary properties. Specifically, generative-based SSL suppresses noise and redundancy to yield general-purpose representations (Theorem 1), while contrastive-based SSL captures task-relevant semantics by maximizing a lower bound on mutual information (Theorem 2). While the connection between IB and SSL has been reviewed

[1]The code is available at: https://github.com/guangjinpan/LWLM

[2]Similar to [42], the term "large" in this work refers to modeling scope rather than parameter count, as LWLM is pretrained on large-scale heterogeneous channel data and supports diverse configurations, tasks, and scenarios.

in prior work [43], our contribution lies in leveraging these principles constructively: the analysis directly informs why spatial-frequency masked channel modeling (SF-MCM) and domain-transformation invariance (DTI) are combined for general-purpose feature learning, and why position-invariant contrastive learning (PICL) is introduced to inject localization-specific semantics.

- **Hybrid self-supervised learning framework for LWLM:** Based on the conclusions from the IB framework, we propose a novel hybrid SSL pretraining framework for wireless localization tasks that jointly optimizes three objectives: (i) SF-MCM to capture local and global correlations across antennas and subcarriers (ii) DTI for enforcing consistency across spatial-frequency and angle-delay domains, and (iii) PICL for capturing robust, location-dependent semantics invariant to BS configurations. While SF-MCM builds upon existing masked modeling techniques, DTI and PICL are newly introduced. This joint learning framework enables the model to acquire diverse, configuration-invariant, and location-aware representations from unlabeled CSI.
- **Task-adaptive decoders for downstream localization tasks:** We develop lightweight task-specific decoders for key downstream localization tasks, including ToA estimation, AoA estimation, single-BS localization, and multi-BS fusion. To support an arbitrary number of BSs, the multi-BS decoder extends single-BS models with an attention-based fusion module that adaptively aggregates the location estimates of each BS.

Experiments show that LWLM consistently outperforms model-based and supervised-learning baselines across all tasks. Specifically, compared with the transformer without pretraining, LWLM achieves improvements of 26.0%, 34.7%, 87.5%, and 32.0% in ToA estimation, AoA estimation, single-BS localization, and multi-BS localization tasks, respectively. Moreover, LWLM demonstrates strong performance in label-limited scenarios, generalizes well to unseen deployment configurations and cross-dataset evaluations, and remains compatible with pilot-based CSI measurements for practical communication systems. In addition, we provide comprehensive analyses on cross-domain generalization, computational complexity, fine-tuning strategies, and pretraining loss weight sensitivity.

II. System Model

Considering an uplink multiple-input multiple-output orthogonal frequency-division multiplexing (MIMO-OFDM) system, a generic UE is equipped with a single omnidirectional antenna, while a generic BS is equipped with a uniform linear array (ULA) consisting of N_{ant} antennas. The total system bandwidth, denoted by B_{bw}, is uniformly divided into N_{subc} orthogonal subcarriers, each with bandwidth $\Delta_f = B_{\text{bw}}/N_{\text{subc}}$.

A. Channel Model

Let $\boldsymbol{p}^{\text{bs}} \in \mathbb{R}^{d_p}$ and $\boldsymbol{p}^{\text{ue}} \in \mathbb{R}^{d_p}$ denote the spatial coordinates of the BS and UE, respectively, where $d_p = 2$ indicates a two-dimensional localization system. The CSI vector at the k-th subcarrier, denoted by $\boldsymbol{h}_k \in \mathbb{C}^{N_{\text{ant}}}$, characterizes the uplink channel from the UE to the BS and is modeled via the channel frequency response (CFR) as $\boldsymbol{h}_k = \sum_{l=1}^{L} \alpha_l \boldsymbol{a}^{\text{bs}}(\theta_l) e^{-j2\pi k \Delta_f \tau_l} + \boldsymbol{n}_k$, where L is the number of multipath components (MPCs), $\alpha_l \in \mathbb{C}$ is the complex gain of the l-th MPC, τ_l denotes its propagation delay, and θ_l is the angle of arrival (AoA) at the BS. $\boldsymbol{n}_k \in \mathbb{C}^{N_{\text{ant}}}$ represents the additive white Gaussian noise (AWGN) vector. The ULA array response vector $\boldsymbol{a}^{\text{bs}}(\cdot) \in \mathbb{C}^{N_{\text{ant}}}$ for a given AoA θ is formulated as $\boldsymbol{a}^{\text{bs}}(\theta) = \left[1, e^{-j\frac{2\pi d}{\lambda}\sin(\theta)}, \ldots, e^{-j\frac{2\pi d}{\lambda}(N_{\text{ant}}-1)\sin(\theta)}\right]^{\top}$, where λ and d denote the carrier wavelength and the antenna spacing, respectively. Aggregating CFRs from all subcarriers, the CFR matrix is $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_{N_{\text{subc}}}] \in \mathbb{C}^{N_{\text{ant}} \times N_{\text{subc}}}$.

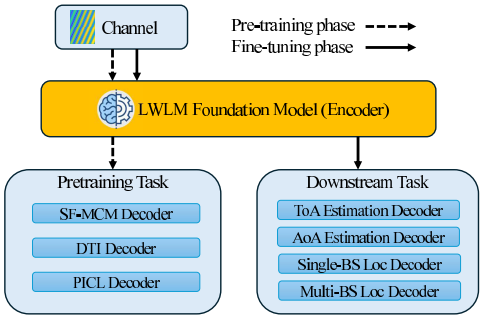


Fig. 1. Schematic diagram of the LWLM algorithm framework. First, the LWLM foundation model is pretrained using the pretraining task, and then fine-tuned on specific localization tasks.

B. Problem Formulation

Learning robust, generalizable representations from channel measurements remains a key challenge in AI-driven wireless localization [8]. To tackle this, we propose a general-purpose channel representation framework based on SSL, named LWLM (see Fig. 1). Specifically, LWLM is a transformer-based encoder designed to learn universal and transferable channel representations from a large-scale unlabeled channel dataset, denoted as $\mathcal{D}^{\text{pre}} = \{\boldsymbol{H}_s | s = 1, \ldots, N^{\text{pre}}\}$, where N^{pre} is the total number of pretraining samples. The LWLM encoder $\mathcal{F}_{\boldsymbol{\Phi}}^{\text{LWLM}}(\cdot)$, parameterized by $\boldsymbol{\Phi}$, transforms input CFR sample s into a latent representation: $\boldsymbol{o}_s = \mathcal{F}_{\boldsymbol{\Phi}}^{\text{LWLM}}(\boldsymbol{H}_s)$, where $\boldsymbol{o}_s$ is the channel semantics, which includes the essential propagation properties of the wireless channel in a task-independent manner. To train the LWLM encoder $\mathcal{F}_{\boldsymbol{\Phi}}^{\text{LWLM}}(\cdot)$ and obtain general-purpose representations $\boldsymbol{o}_s$ beneficial for localization-related tasks, we design hybrid SSL tasks for the pretraining stage. Let $\mathcal{K}_{\text{s-task}} = 1, 2, \ldots, K_{\text{s-task}}$ denote the set of $K_{\text{s-task}}$ SSL pretraining tasks. Each pretraining task $k_{\text{s-task}}$ has an associated decoder $\mathcal{F}_{\boldsymbol{\Theta}_{k_{\text{s-task}}}}^{\text{s-task}}$, and the overall optimization objective for the pretraining is formulated as

$$\min_{\boldsymbol{\Phi}, \{\boldsymbol{\Theta}_{k_{\text{s-task}}}\}_{k_{\text{s-task}} \in \mathcal{K}_{\text{s-task}}}} \sum_{k_{\text{s-task}} \in \mathcal{K}_{\text{s-task}}} \alpha_{k_{\text{s-task}}} \mathbb{E}_{\boldsymbol{H}_s \sim \mathcal{D}^{\text{pre}}} \left[\mathcal{L}_{k_{\text{s-task}}} \left(\mathcal{F}_{\boldsymbol{\Theta}_{k_{\text{s-task}}}}^{\text{s-task}}(\boldsymbol{o}_s), \boldsymbol{H}_s \right) \right], \quad \text{s.t. } \boldsymbol{o}_s = \mathcal{F}_{\boldsymbol{\Phi}}^{\text{LWLM}}(\boldsymbol{H}_s), \tag{1}$$

where $\mathcal{L}_{k_{\text{s-task}}}(\cdot, \cdot)$ is the loss function associated with SSL task $k_{\text{s-task}}$, and $\alpha_{k_{\text{s-task}}} \geq 0$ is its corresponding weight.

After the pretraining, the pretrained LWLM encoder is adapted to multiple downstream localization tasks through lightweight, task-specific decoders. Let $\mathcal{K}_{\text{d-task}} = \{1, 2, \ldots, K_{\text{d-task}}\}$ denote the set of downstream tasks. For each localization-related task $k_{\text{d-task}} \in \mathcal{K}_{\text{d-task}}$, we define a decoder $\mathcal{F}^{\text{d-task}}_{\boldsymbol{\Theta}_{k_{\text{d-task}}}}$ with parameters $\boldsymbol{\Theta}_{k_{\text{d-task}}}$. The model is fine-tuned using a small labeled dataset $\mathcal{D}^{\text{d-task}}_{k_{\text{d-task}}} = \{(\boldsymbol{H}_s, \boldsymbol{y}_s^{k_{\text{d-task}}}) | s = 1, 2, \ldots, N^{\text{d-task}}_{k_{\text{d-task}}}\}$, where $\boldsymbol{y}_s^{k_{\text{d-task}}}$ denotes the task-specific label (e.g., ToA, AoA, 2D coordinates with $d_p = 2$), and $N^{\text{d-task}}_{k_{\text{d-task}}}$ is the number of labeled samples for task $k_{\text{d-task}}$. The fine-tuning objective for each downstream task $k_{\text{d-task}}$ can be expressed as

$$\begin{aligned} \min_{\boldsymbol{\Phi}, \boldsymbol{\Theta}_{k_{\text{d-task}}}} \quad & \mathbb{E}_{(\boldsymbol{H}_s, \boldsymbol{y}_s^{k_{\text{d-task}}}) \sim \mathcal{D}^{\text{d-task}}_{k_{\text{d-task}}}} \left[\mathcal{L}_{k_{\text{d-task}}} \left(\mathcal{F}^{\text{d-task}}_{\boldsymbol{\Theta}_{k_{\text{d-task}}}} (\boldsymbol{o}_s), \boldsymbol{y}_s^{k_{\text{d-task}}} \right) \right], \\ \text{s.t.} \quad & \boldsymbol{o}_s = \mathcal{F}^{\text{LWLM}}_{\boldsymbol{\Phi}} (\boldsymbol{H}_s), \end{aligned} \tag{2}$$

where $\mathcal{L}_{k_{\text{down}}}(\cdot, \cdot)$ is the task-specific loss function for task k_{down}. Notably, during the fine-tuning phase, the parameters $\boldsymbol{\Phi}$ of the LWLM encoder can be either frozen or updated, depending on factors such as the similarity between the downstream and pretraining tasks, the amount of labeled data, and the complexity of the downstream decoder. In this work, since the downstream task differs significantly from the pretraining objective and we adopt a lightweight decoder, we fine-tune the encoder during downstream training.

III. Information Bottleneck Analysis for SSL

The connection between SSL and IB theory has been systematically reviewed in [43]. While that work establishes the general relationship between SSL and the IB principle, it does not provide explicit guidance on how to combine different types of SSL objectives for a specific downstream application. To fill this gap, we adopt the IB framework [44] as an analytical tool to derive design principles for our hybrid SSL pretraining strategy. The IB objective aims to optimize the encoder and decoder by minimizing

$$\mathcal{L}_{\text{IB}}(\boldsymbol{\Phi}, \boldsymbol{\Theta}) = I(O; H) - \beta I(O; Y), \tag{3}$$

where H, O, and Y denote the random variables corresponding to the input channel matrix, the learned intermediate representation, and the downstream task labels, respectively. Their realizations are denoted by $\boldsymbol{H}$, $\boldsymbol{o}$, and $\boldsymbol{y}$. The weighting factor $\beta > 0$ controls the trade-off between compression and prediction, and $\boldsymbol{\Phi}$ and $\boldsymbol{\Theta}$ are the trainable parameters of the encoder and decoder, respectively.

To minimize the IB objective, the first term encourages the encoder to discard task-irrelevant noise and redundant information from the input, while the second term encourages the decoder to retain task-relevant features from the learned representation O. Although this supervised formulation provides an ideal framework for learning meaningful semantic representations, directly optimizing it is often impractical due to limited labeled data and generalization concerns. In the context of foundation models, we expect the learned representation O to encode semantics that are transferable across multiple downstream localization tasks. This requires SSL objectives that not only suppress irrelevant channel information (e.g., noise) and preserve general-purpose semantic content, but also capture task-relevant semantics. Based on this principle, we analyze two types of SSL tasks within the IB framework: *generative-based SSL* and *contrastive-based SSL*. In the following, we show that generative-based SSL promotes generalization by reconstructing underlying channel structures (Theorem 1), while contrastive-based SSL tailors the representation to be more discriminative for localization tasks (Theorem 2).

A. Generative-Based SSL Within IB Framework

As mentioned above, to enable the learned representation O to discard semantically irrelevant information (e.g., channel estimation noise) and prevent overfitting while ensuring generalization across diverse downstream tasks, our goal is to minimize the mutual information $I(O; H)$. However, directly minimizing $I(O; H)$ may lead to results where O and H become completely independent, thus losing useful semantic information. To address this issue, we introduce generative-based SSL tasks to preserve the channel semantics. Specifically, we consider a set of generative-based SSL tasks involving channel transformations $\{T_{k_g}(H)\}_{k_g=1}^{K_g}$ (e.g., masked modeling, domain transformation), where K_g denotes the number of generative-based SSL tasks used during the pretraining stage. For each SSL task k_g, the corresponding IB objective is formulated as

$$\mathcal{L}_{\text{IB-G}}(\boldsymbol{\Phi}, \boldsymbol{\Theta}) = I(O; H) - \beta_{k_g} I(O; T_{k_g}(H)), \tag{4}$$

where β_{k_g} is the weighting factor, and is generally greater than 1, as indicated by later derivations. Based on this, we present the following theorem (proof in Appendix A):

Theorem 1: For the transformation $T_{k_g}(H)$, let $\overline{T}_{k_g}(H)$ be defined so that the map $H \mapsto (T_{k_g}(H), \overline{T}_{k_g}(H))$ is invertible. Define the reconstruction loss for $T_{k_g}(H)$ and residual reconstruction loss for $\overline{T}_{k_g}(H)$ for the generative-based SSL task k_g as

$$\begin{aligned} \mathcal{L}_{k_g}(\boldsymbol{\Phi}, \boldsymbol{\Theta}_{k_g}) &= -\mathbb{E}_{p(H) q_{\boldsymbol{\Phi}}(O|H)} \left[\ln p_{\boldsymbol{\Theta}_{k_g}} \left(T_{k_g}(H) \,|\, O \right) \right] \\ \overline{\mathcal{L}}_{k_g}(\boldsymbol{\Phi}, \overline{\boldsymbol{\Theta}}_{k_g}) &= -\mathbb{E}_{p(H) q_{\boldsymbol{\Phi}}(O|H)} \left[\ln p_{\overline{\boldsymbol{\Theta}}_{k_g}} \left(\overline{T}_{k_g}(H) | T_{k_g}(H), O \right) \right] \end{aligned}$$

where $\boldsymbol{\Phi}$ are the parameters of the encoder, $\boldsymbol{\Theta}_{k_g}$ are the parameters of the decoder for task-relevant output T_{k_g}, and $\overline{\boldsymbol{\Theta}}_{k_g}$ are the parameters of the decoder for the residual component $\overline{T}_{k_g}$. Then, the IB objective $\mathcal{L}_{\text{IB-G}}(\boldsymbol{\Phi}, \boldsymbol{\Theta})$ can be approximated as

$$\mathcal{L}_{\text{IB-G}}(\boldsymbol{\Phi}, \boldsymbol{\Theta}) \approx (\beta_{k_g} - 1) \mathcal{L}_{k_g}(\boldsymbol{\Phi}, \boldsymbol{\Theta}_{k_g}) - \overline{\mathcal{L}}_{k_g}(\boldsymbol{\Phi}, \overline{\boldsymbol{\Theta}}_{k_g}) + C.$$

$(\beta_{k_g} - 1) > 0$ denotes the trade-off between preserving task-relevant information and discarding residual information, and C is a constant independent of parameters $\boldsymbol{\Phi}$, $\boldsymbol{\Theta}_{k_g}$ and $\overline{\boldsymbol{\Theta}}_{k_g}$. Based on Theorem 1, the first term $\mathcal{L}_{k_g}(\boldsymbol{\Phi}, \boldsymbol{\Theta}_{k_g})$ encourages to learn a better representation O to accurately reconstruct $T_{k_g}(H)$, while the negative second term $\overline{\mathcal{L}}_{k_g}(\boldsymbol{\Phi}, \overline{\boldsymbol{\Theta}}_{k_g})$ encourages O to suppress the reconstruction of $\overline{T}_{k_g}(H)$, hence discarding residual information. This highlights a trade-off in the representation O to preserve important semantics from $T_{k_g}(H)$ and discard irrelevant or noisy components. It is worth noting that, in the actual training process, $\overline{T}_{k_g}(H)$ is typically

not directly accessible. As a result, low-dimensional representations of O, along with dropout and normalization layers, are often employed to help the neural network implicitly discard residual information. Furthermore, we generally set $(\beta_{k_g}-1)>0$ to encourage the AI model to learn the important channel semantics while reconstructing $T_{k_g}(H)$. Based on Theorem 1, we can infer that incorporating a richer set of generative-based SSL tasks $\{T_{k_g}(H)\}$ can further guide the representation O to minimize shared redundant information across these tasks (such as noise). This, in turn, will result in O containing richer and more meaningful semantics of the channel. However, because these tasks do not incorporate downstream-specific signals, the resulting representations may lack optimal task specificity. In the next section, we explain how contrastive-based SSL can introduce targeted downstream semantics to further enhance task-specific performance.

B. Contrastive-Based SSL Within IB Framework

To extract task-dependent semantics tailored for downstream localization tasks, our goal is to let channel representation O include more location-related semantics during pretraining. Since ground-truth labels $\boldsymbol{y}$ are typically unavailable during pretraining, we employ a contrastive learning method to indirectly obtain localization-related semantics. For localization, measurements from the same location under varying wireless conditions form positive pairs, while those from different locations serve as negative pairs. To illustrate that contrastive-based SSL can extract task-related semantics, we first give the following theorem (proof in Appendix B):

Theorem 2: Let $\mathcal{N}_{\text{bat}}=\{(\boldsymbol{H}_s,\boldsymbol{y}_s)\}_{s=1}^{N_{\text{bat}}}\subset\mathcal{D}^{\text{pre}}$ be a batch of samples drawn independently from the joint distribution $p(H,Y)$, where N_{bat} denotes the batch size. For each sample s with target label $\boldsymbol{y}_s$, we additionally sample another channel realization $\boldsymbol{H}_{s^+}\sim p(H\mid Y=\boldsymbol{y}_s)$, which serves as a positive sample. The corresponding representations are obtained by $\boldsymbol{o}_s\sim q_{\boldsymbol{\Phi}}(\boldsymbol{o}|\boldsymbol{H}_s)$, and $\boldsymbol{o}_{s^+}\sim q_{\boldsymbol{\Phi}}(\boldsymbol{o}|\boldsymbol{H}_{s^+})$. Constructing the InfoNCE contrastive loss:

$$\mathcal{L}_{\text{InfoNCE}}=-\frac{1}{N_{\text{bat}}}\sum_{s=1}^{N_{\text{bat}}}\left[\ln\frac{\exp\big(\text{sim}(\boldsymbol{o}_s,\boldsymbol{o}_{s^+})/\tau\big)}{\sum_{j=1}^{N_{\text{bat}}}\exp\big(\text{sim}(\boldsymbol{o}_s,\boldsymbol{o}_j)/\tau\big)}\right], \tag{5}$$

where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity, and $\tau>0$ is a temperature parameter, the contrastive objective provides the following lower bound for $I(O;Y)$:

$$I(O;Y)\geq I(O;O^+)\geq\log(N_{\text{bat}})-\mathcal{L}_{\text{InfoNCE}}, \tag{6}$$

where the constant term is independent of $\boldsymbol{\Phi}$.

Constructing the InfoNCE contrastive loss:

$$L_{\text{InfoNCE}}=-\frac{1}{N_{\text{bat}}}\sum_{s=1}^{N_{\text{bat}}}\left[\ln\frac{\exp\big(\text{sim}(\boldsymbol{o}_s,\boldsymbol{o}_{s^+})/\tau\big)}{\sum\exp\big(\text{sim}(\boldsymbol{o}_s,\boldsymbol{o}_j)/\tau\big)}\right], \tag{7}$$

where $\text{sim}(\cdot,\cdot)$ denotes cosine similarity, and $\tau>0$ is a temperature parameter, the contrastive objective provides the following lower bound for $I(O;Y)$:

$$I(O;Y)\geq I(O;O^+)\geq\log(N_{\text{bat}})-L_{\text{InfoNCE}}, \tag{8}$$

where the constant term is independent of $\boldsymbol{\Phi}$.

According to Theorem 2, minimizing the InfoNCE loss in contrastive-based SSL is equivalent to maximizing a lower bound of $I(O;Y)$, thereby indirectly minimizing the original IB cost $I(H;O)-\beta I(O;Y)$. Therefore, contrastive-based SSL can capture the semantic representation of task dependencies. While this paper is primarily discussed in the context of localization, the underlying principle can be extended to other channel-related tasks by identifying meaningful invariances. Consequently, the proposed framework can be effectively adapted to a broad range of wireless applications, such as sensing and channel extrapolation.

In summary, the above analysis yields two design principles for our hybrid SSL framework. First, from Theorem 1, incorporating multiple generative-based objectives encourages the representation to suppress shared redundancy across tasks (e.g., channel noise and hardware impairments) while preserving richer channel semantics. This motivates combining SF-MCM, which exploits spatial-frequency correlations inherent in MIMO-OFDM channels, with DTI, which enforces cross-domain consistency between the spatial-frequency and angle-delay representations. Second, from Theorem 2, contrastive-based SSL introduces task-relevant structure by maximizing a lower bound on the mutual information between the representation and downstream labels. This motivates the design of PICL, which uses co-located channel observations under different BS configurations as positive pairs, thereby injecting localization-specific invariance into the representation. These two principles jointly inform the hybrid pretraining framework presented in Sec. IV.

IV. Self-Supervised Pretraining Framework

Based on the above analysis, we develop a hybrid SSL framework tailored for wireless localization tasks, where SF-MCM and DTI are generative-based SSL methods, while PICL is a contrastive-based SSL method. Unlike purely data-driven SSL frameworks in CV or NLP, the proposed LWLM leverages not only large-scale data but also explicitly integrates domain knowledge of wireless propagation. Each SSL objective is physically motivated: SF-MCM exploits the inherent spatial–frequency correlations of MIMO-OFDM channels, DTI leverages the information equivalence between different physical domains (i.e., spatial–frequency and angle–delay domains), and PICL captures the geometric invariance of propagation paths under varying BS configurations. This design combines knowledge extracted from data with that derived from wireless channel models, enabling LWLM to effectively reuse and internalize physical channel priors for more interpretable and generalizable representations. The schematic diagram of the pretraining algorithm is shown in Fig. 2. It is worth noting that SF-MCM is adapted from existing masked modeling works [29], [30], while DTI and PICL are newly introduced.

A. SF-MCM-Based SSL

SF-MCM is a generative SSL method that captures channel semantics by randomly masking spatial-frequency channel data and reconstructing it. The main objective of this SSL

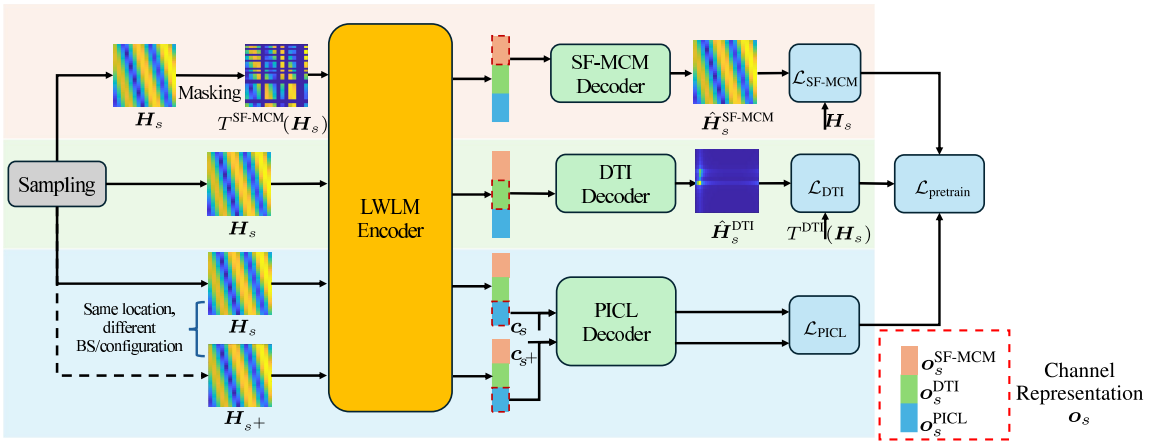


Fig. 2. Schematic diagram of the pretraining algorithm. The channel semantic representation $\boldsymbol{o}_s$ based on the hybrid SSL method is the concatenation of the three representations $\boldsymbol{o}^{\text{SF-MCM}_s}$, $\boldsymbol{o}_s^{\text{DTI}}$, $\boldsymbol{o}_s^{\text{PICL}}$. At each training step, we compute a weighted sum of three separate losses as the final pretraining loss.

method is to enable the neural network to generate the masked portions, thereby learning the correlations across subcarriers and antennas.

Given a CFR sample $\boldsymbol{H}_s$, we randomly select subsets of antennas and subcarriers to be masked, denoted as $\mathcal{N}_{\text{ant}}^M \subseteq \{1,\ldots,N_{\text{ant}}\}$ and $\mathcal{N}_{\text{subc}}^M \subseteq \{1,\ldots,N_{\text{subc}}\}$, with $|\mathcal{N}_{\text{ant}}^M| = \hat{N}_{\text{ant}}^M$ and $|\mathcal{N}_{\text{subc}}^M| = \hat{N}_{\text{subc}}^M$, respectively. We define a binary masking matrix $\boldsymbol{M} \in \{0,1\}^{N_{\text{ant}} \times N_{\text{subc}}}$ as

$$[M]_{i,j} = \begin{cases} 0, & i \in \mathcal{N}_{\text{ant}}^M \text{ or } j \in \mathcal{N}_{\text{subc}}^M \\ 1, & \text{otherwise} \end{cases}, \tag{9}$$

where $[M]_{i,j} = 0$ indicates that the corresponding spatial-frequency entry (i,j) is masked (i.e., removed and requires reconstruction), while $[M]_{i,j} = 1$ denotes an unmasked entry. The masked channel, denoted as $T^{\text{SF-MCM}}(\boldsymbol{H}_s)$, is given by

$$T^{\text{SF-MCM}}(\boldsymbol{H}_s) = \boldsymbol{H}_s \odot \boldsymbol{M}, \tag{10}$$

where $\odot$ denotes element-wise multiplication. The masked channel $T^{\text{SF-MCM}}(\boldsymbol{H}_s)$ serves as the input to the LWLM encoder to obtain a latent representation, i.e., $\boldsymbol{o}_s^{\text{SF-MCM}} = \mathcal{F}_{\boldsymbol{\Phi}}^{\text{LWLM}}(T^{\text{SF-MCM}}(\boldsymbol{H}_s))$, where $\boldsymbol{o}_s^{\text{SF-MCM}}$ represents the latent embedding learned from the SF-MCM pretraining objective.

Subsequently, the latent representation is decoded via a decoder $\mathcal{F}_{\boldsymbol{\Theta}_{\text{SF-MCM}}}^{\text{SF-MCM}}(\cdot)$ to reconstruct the original CFR, i.e., $\hat{\boldsymbol{H}}_s^{\text{SF-MCM}} = \mathcal{F}_{\boldsymbol{\Theta}_{\text{SF-MCM}}}^{\text{SF-MCM}}(\boldsymbol{o}_s^{\text{SF-MCM}})$, where $\hat{\boldsymbol{H}}_s^{\text{SF-MCM}}$ is the reconstructed channel by SF-MCM decoder. As stated in Theorem 1, minimizing the reconstruction error helps capture richer channel semantics. Therefore, we adopt the mean squared error (MSE) loss as the reconstruction objective:

$$\mathcal{L}_{\text{SF-MCM}} = \frac{1}{N_{\text{bat}}} \sum_{\boldsymbol{H}_s \in \mathcal{N}_{\text{bat}}} \left\| (\hat{\boldsymbol{H}}_s^{\text{SF-MCM}} - \boldsymbol{H}_s) \odot (\boldsymbol{1}_M - \boldsymbol{M}) \right\|_F^2, \tag{11}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\boldsymbol{1}_M$ denotes the all-ones matrix of the same size as $\boldsymbol{M}$. $\boldsymbol{1}_M - \boldsymbol{M}$ restricts the loss to masked positions, forcing the encoder to infer them from the unmasked context.

B. DTI-Based SSL

DTI learns general-purpose features across different physically meaningful channel domains by learning their transformations. This essentially enables LWLM to acquire domain transformation knowledge commonly used in the wireless field, thereby obtaining cross-domain consistent representations that preserve essential channel semantics. In this work, we consider the spatial-frequency and angle-delay domains, though the approach is extendable to others. Given a spatial-frequency domain channel representation $\boldsymbol{H}_s$, its corresponding angle-delay domain representation $T^{\text{DTI}}(\boldsymbol{H}_s)$ can be obtained by two-dimensional discrete Fourier transforms (2D-DFT). The transformation process can be expressed as $T^{\text{DTI}}(\boldsymbol{H}_s) = \boldsymbol{W}_\theta^{\mathsf{H}} \boldsymbol{H}_s \boldsymbol{W}_\tau^*$, where $(\cdot)^{\mathsf{H}}$ denotes conjugate transpose and $(\cdot)^*$ denotes the complex conjugation. The unitary DFT matrices $\boldsymbol{W}_\theta \in \mathbb{C}^{N_{\text{ant}} \times N_{\text{ant}}}$ and $\boldsymbol{W}_\tau \in \mathbb{C}^{N_{\text{subc}} \times N_{\text{subc}}}$ map the spatial domain to the angle domain and the frequency domain to the delay domain, respectively. Each element of the DFT matrix can be expressed as $[\boldsymbol{W}_\theta]_{i_1,i_2} = \frac{1}{\sqrt{N_{\text{ant}}}} \exp\left(-j2\pi \frac{i_1 i_2}{N_{\text{ant}}}\right)$, and $[\boldsymbol{W}_\tau]_{i_1,i_2} = \frac{1}{\sqrt{N_{\text{subc}}}} \exp\left(-j2\pi \frac{i_1 i_2}{N_{\text{subc}}}\right)$. To learn transformation-invariant features, the LWLM encoder first maps the input CFR into a latent representation: $\boldsymbol{o}_s^{\text{DTI}} = \mathcal{F}_{\boldsymbol{\Phi}}^{\text{LWLM}}(\boldsymbol{H}_s)$. Then, the DTI decoder, i.e. $\mathcal{F}_{\boldsymbol{\Theta}_{\text{DTI}}}^{\text{DTI}}(\cdot)$ with parameters $\boldsymbol{\Theta}_{\text{DTI}}$, decodes $\boldsymbol{o}_s^{\text{DTI}}$ to the angle-delay domain, i.e., $\hat{\boldsymbol{H}}_s^{\text{DTI}} = \mathcal{F}_{\boldsymbol{\Theta}_{\text{DTI}}}^{\text{DTI}}(\boldsymbol{o}_s^{\text{DTI}})$, where $\hat{\boldsymbol{H}}_s^{\text{DTI}}$ denotes the output of the DTI decoder.

Following Theorem 1, our target is to minimize the reconstruction loss. Considering the sparsity of the channel in the angle-delay domain, the reconstruction loss $\mathcal{L}_{\text{DTI}}$ is defined via angle-delay dissimilarity [35]:

$$\mathcal{L}_{\text{DTI}} = \frac{1}{N_{\text{bat}}} \sum_{\boldsymbol{H}_s \in \mathcal{N}_{\text{bat}}} \left[1 - \frac{\left| \langle T^{\text{DTI}}(\boldsymbol{H}_s), \hat{\boldsymbol{H}}_s^{\text{DTI}} \rangle \right|}{\|T^{\text{DTI}}(\boldsymbol{H}_s)\| \cdot \|\hat{\boldsymbol{H}}_s^{\text{DTI}}\|} \right], \tag{12}$$

where $\langle \cdot, \cdot \rangle$ denotes the Frobenius inner product.

C. PICL-Based SSL

PICL leverages location invariance across different BSs or configurations using contrastive learning. Unlike SF-MCM and DTI, PICL relies on co-location information (e.g., multi-BS reception of the same uplink transmission or UE identifiers) to construct positive pairs, while still being free of explicit

position labels. By treating pairs of CSI observations from the same location as positive samples, the encoder learns to associate these views with similar latent representations. As shown in Theorem 2, contrastive-based PICL enhances the expressiveness of representations for task-specific semantics, making them more suitable for localization.

Given a CFR sample $\boldsymbol{H}_s$ and its corresponding positive sample $\boldsymbol{H}_{s^+}$, both of which are captured at the same location but under different BS deployments or configuration settings, these paired samples are used to guide the contrastive learning process in PICL.[3] This reflects a realistic acquisition process and does not rely on position labels during training. The LWLM encoder generates latent representations as follows:

$$\boldsymbol{o}_s^{\text{PICL}} = \mathcal{F}_{\boldsymbol{\Phi}}^{\text{LWLM}}(\boldsymbol{H}_s), \quad \boldsymbol{o}_{s^+}^{\text{PICL}} = \mathcal{F}_{\boldsymbol{\Phi}}^{\text{LWLM}}(\boldsymbol{H}_{s^+}). \tag{13}$$

Latent representations $\boldsymbol{o}_s^{\text{PICL}}$ and $\boldsymbol{o}_{s^+}^{\text{PICL}}$ are then processed by PICL decoder $\mathcal{F}_{\boldsymbol{\Theta}_{\text{PICL}}}^{\text{PICL}}(\cdot)$, incorporating the associated BS configuration, to produce configuration-invariant representations:

$$\tilde{\boldsymbol{o}}_s^{\text{PICL}} = \mathcal{F}_{\boldsymbol{\Theta}_{\text{PICL}}}^{\text{PICL}}(\boldsymbol{o}_s^{\text{PICL}}, \boldsymbol{c}_s), \ \tilde{\boldsymbol{o}}_{s^+}^{\text{PICL}} = \mathcal{F}_{\boldsymbol{\Theta}_{\text{PICL}}}^{\text{PICL}}(\boldsymbol{o}_{s^+}^{\text{PICL}}, \boldsymbol{c}_{s^+}), \tag{14}$$

where $\boldsymbol{c}_s = [\boldsymbol{p}_s^{\text{bs}}, B\text{bw}, s$ includes the BS location and bandwidth configuration.[4]

During training, as described in Theorem 2, we employ the NT-Xent loss [45], which is a special form of the InfoNCE loss [46]. Specifically, two mini-batches of channel samples, $\mathcal{N}_1^{\text{bat}}$ and $\mathcal{N}_2^{\text{bat}}$, are sampled from pretraining datasets $\mathcal{D}^{\text{pre}}$, each containing N_{bat} samples. The s-th sample in $\mathcal{N}_1^{\text{bat}}$ and the s^+-th sample in $\mathcal{N}_2^{\text{bat}}$ form a positive pair, while all others $(2N_{\text{bat}}-2)$ samples are treated as negatives. Let $\tilde{\mathcal{N}}^{\text{bat}} = \mathcal{N}_1^{\text{bat}} \cup \mathcal{N}_2^{\text{bat}}$ denote the full batch. The PICL loss is defined as

$$\mathcal{L}_{\text{PICL}} = \frac{1}{2N_{\text{bat}}} \sum_{s \in \tilde{\mathcal{N}}^{\text{bat}}} \sum_{s^+ \in \tilde{\mathcal{N}}^{\text{bat}}} \mathbb{1}_{s,s^+} \ell_{s,s^+}, \tag{15}$$

where $\mathbb{1}_{s,s^+} = 1$ if s and s^+ form a positive pair, and $\mathbb{1}_{s,s^+} = 0$ otherwise. For each positive pair (s, s^+), the contrastive loss ℓ_{s,s^+} is given by

$$\ell_{s,s^+} = -\log \frac{\exp\left(\text{sim}(\tilde{\boldsymbol{o}}_s^{\text{PICL}}, \tilde{\boldsymbol{o}}_{s^+}^{\text{PICL}})/\tau\right)}{\sum_{s^\star \in \tilde{\mathcal{N}}^{\text{bat}}, s^\star \neq s} \exp\left(\text{sim}(\tilde{\boldsymbol{o}}_s^{\text{PICL}}, \tilde{\boldsymbol{o}}_{s^\star}^{\text{PICL}})/\tau\right)}. \tag{16}$$

where $\text{sim}(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a}^\top \boldsymbol{b}}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|}$ denotes cosine similarity and τ is a temperature hyperparameter.

D. LWLM Architecture and Training Details

In this subsection, we provide a detailed description of the implementation and pretraining procedure of the proposed LWLM. Specifically, we describe the architecture of the encoder and SSL-specific decoders.

[3]In practical networks, paired CSI samples can be obtained from multi-BS measurements collected for positioning tasks, e.g., UTDOA-based localization in 5G systems [8]. A UE transmission received by multiple BSs within a short time window naturally forms multi-view observations at approximately the same location. Strict synchronization is not required as long as the UE displacement during the measurement interval is small.

[4]In this paper, we consider only bandwidth variation, but this formulation can be easily extended to incorporate other configuration parameters such as frequency band, transmit power, etc.

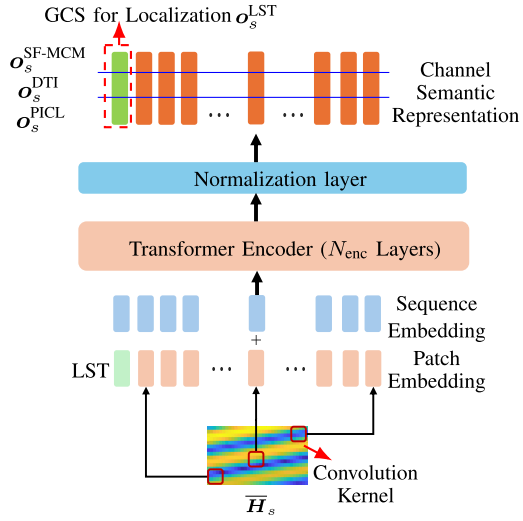


Fig. 3. Schematic diagram of LWLM encoder architecture.

1) Model Input: As mentioned above, the input to the model is the complex CFR matrix $\boldsymbol{H}_s$. To facilitate real-valued neural network processing, we represent the complex CSI using its magnitude and phase components, which form a bijective transformation of the original complex signal.[5] This amplitude–phase representation preserves full information while enabling stable training with standard real-valued networks. Specifically, we rewrite it as

$$\overline{\boldsymbol{H}}_s = [|\boldsymbol{H}_s|, \angle \boldsymbol{H}_s] \in \mathbb{R}^{2 \times N_{\text{ant}} \times N_{\text{subc}}}. \tag{17}$$

The input $\overline{\boldsymbol{H}}_s$ is a 3D tensor of shape $(2, N_{\text{ant}}, N_{\text{subc}})$, with the first dimension corresponding to the amplitude and phase, respectively.

2) Channel Embedding: As shown in Fig. 3, before feeding into the transformer encoder, we perform channel tokenization to convert the 3D channel input $\overline{\boldsymbol{H}}_s$ into a sequence of embeddings. Inspired by the tokens-to-token vision transformer (T2T-ViT) framework [47], we apply a 2D CNN layer with N_{embed} convolution kernels of size $K_{\text{cnn}} \times K_{\text{cnn}}$ and stride S_{cnn} along both dimensions. This operation produces N_{patch} tokens, each represented by an N_{embed}-dimensional vector. The total number of patches N_{patch} is computed as $N_{\text{patch}} = \left\lfloor \frac{N_{\text{ant}} - K_{\text{cnn}}}{S_{\text{cnn}}} + 1 \right\rfloor \times \left\lfloor \frac{N_{\text{subc}} - K_{\text{cnn}}}{S_{\text{cnn}}} + 1 \right\rfloor$.

Let $\boldsymbol{x}_{s,u}^{\text{embed}}$ denote the u-th token for sample s. To guide the model in learning global channel semantics (GCSs) for localization, inspired by BERT's [CLS] token [24], we introduce a learnable embedding called the localization semantic token (LST). LST is denoted by $\boldsymbol{x}_{l,0}^{\text{embed}}$, and is placed at the beginning of the token sequence. As a trainable embedding, LST $\boldsymbol{x}_{l,0}^{\text{embed}}$ has no actual semantics before being input into the transformer encoder. This means that LST will not be biased towards its own local semantics like other tokens. After passing through multiple layers of transformers, it gradually integrates information from all tokens via the multi-head attention mechanism, thereby aggregating GCSs. Additionally,

[5]We adopt real-valued networks for stable and efficient training, consistent with most existing works [10], [31], [40]. Exploring complex-valued neural networks is left for future work.

since transformers are permutation-invariant, we incorporate sequence embeddings[6] to embed sequence-specific information. Consequently, the embedding sequence input $\boldsymbol{X}_s^{\text{embed}} \in \mathbb{R}^{(N_{\text{patch}}+1)\times N_{\text{embed}}}$ to the transformer is defined as

$$\boldsymbol{X}_s^{\text{embed}} = [\boldsymbol{x}_{l,0}^{\text{embed}}, \boldsymbol{x}_{l,1}^{\text{embed}}, \ldots, \boldsymbol{x}_{l,N_{\text{patch}}}^{\text{embed}}] + \boldsymbol{E}^{\text{seq}}, \tag{18}$$

where $\boldsymbol{E}^{\text{seq}} \in \mathbb{R}^{(N_{\text{patch}}+1)\times N_{\text{embed}}}$ is the sequence embedding matrix [48], with entries defined as

$$[\boldsymbol{E}^{\text{seq}}]_{seq,2i} = \sin\left(\frac{seq}{10000^{2i/N_{\text{embed}}}}\right), \tag{19}$$

$$[\boldsymbol{E}^{\text{seq}}]_{seq,2i+1} = \cos\left(\frac{seq}{10000^{2i/N_{\text{embed}}}}\right), \tag{20}$$

for sequence index $seq \in \{0, 1, \ldots, N_{\text{patch}}\}$ and embedding dimension index i. This tokenization strategy transforms the 3D channel structure into a sequential format, enabling the transformer to capture both local and global dependencies across antennas and subcarriers.

3) LWLM Encoder Architecture: As illustrated in Fig. 3, the LWLM encoder $\mathcal{F}_{\boldsymbol{\Phi}}^{\text{LWLM}}(\cdot)$ consists of N_{enc} transformer encoder layers followed by a normalization layer. Each transformer encoder layer includes multi-head self-attention modules and feed-forward neural networks to model long-range dependencies and semantic interactions across tokenized representations. Details of the transformer encoder layer can be found in [48]. The final normalization layer helps stabilize feature distributions and facilitates efficient training convergence.

The transformer encoder maps each N_{embed}-dimensional token to a semantic representation of dimension N_{latent}, resulting in an output matrix $\boldsymbol{o}_s \in \mathbb{R}^{(N_{\text{patch}}+1)\times N_{\text{latent}}}$. We further partition the latent output $\boldsymbol{o}_s$ into three distinct parts to support the individual SSL objectives described in Sections IV-A, IV-B, and IV-C. The entire channel representation can be expressed as $\boldsymbol{o}_s = [\boldsymbol{o}_s^{\text{SF-MCM}}, \boldsymbol{o}_s^{\text{DTI}}, \boldsymbol{o}_s^{\text{PICL}}]$, where $\boldsymbol{o}_s^{\text{SF-MCM}} \in \mathbb{R}^{(N_{\text{patch}}+1)\times N_{\text{SF-MCM}}}$, $\boldsymbol{o}_s^{\text{DTI}} \in \mathbb{R}^{(N_{\text{patch}}+1)\times N_{\text{DTI}}}$, and $\boldsymbol{o}_s^{\text{PICL}} \in \mathbb{R}^{(N_{\text{patch}}+1)\times N_{\text{PICL}}}$ are the feature segments sent to their respective decoders. Therefore, the total dimension of N_{latent} is given by $N_{\text{SF-MCM}} + N_{\text{DTI}} + N_{\text{PICL}} = N_{\text{latent}}$.

In particular, the first row of $\boldsymbol{o}_s$, corresponding to the LST, is denoted by $\boldsymbol{o}_s^{\text{LST}}$, which serves as the GCS information. This vector captures high-level, location-relevant features and is used as input for downstream localization decoders. Detailed use of $\boldsymbol{o}_s^{\text{LST}}$ for localization tasks will be introduced in Sec. V.

4) Pretraining Decoder Architecture: Fig. 4 illustrates the pretraining decoder architectures. The SF-MCM and DTI decoders have identical transformer-decoder-based structures with different parameters. Details of the transformer decoder can be found in [48]. Given semantic representations $\boldsymbol{o}_s^{\text{SF-MCM}}$ or $\boldsymbol{o}_s^{\text{DTI}}$, the transformer decoder with N_{dec} layers reconstructs the channel representation. After decoding, the LST token is discarded, and the remaining tokens of shape $(N_{\text{bat}}, N_{\text{patch}}, N_{\text{embed}})$ are folded [47] to match the input dimensions. The final output is a 4-dimensional tensor $\hat{\boldsymbol{H}}_s^{\text{SF-MCM}}$ or $\hat{\boldsymbol{H}}_s^{\text{DTI}}$ with shape $(N_{\text{bat}}, 2, N_{\text{ant}}, N_{\text{subc}})$, where the second dimension represents the real and imaginary parts of the channel in the spatial-frequency and angle-delay domains for $\hat{\boldsymbol{H}}_s^{\text{SF-MCM}}$ and $\hat{\boldsymbol{H}}_s^{\text{DTI}}$, respectively. Based on $\hat{\boldsymbol{H}}_s^{\text{SF-MCM}}$ and $\hat{\boldsymbol{H}}_s^{\text{DTI}}$, the losses defined in Sec. IV-A and Sec. IV-B can be equivalently computed via a differentiable transformation.

[6]In the context of transformers, this is commonly referred to as position embeddings [48], indicating the sequential order of each patch. However, to avoid confusion with the position information in localization tasks, we term this as sequence embeddings.

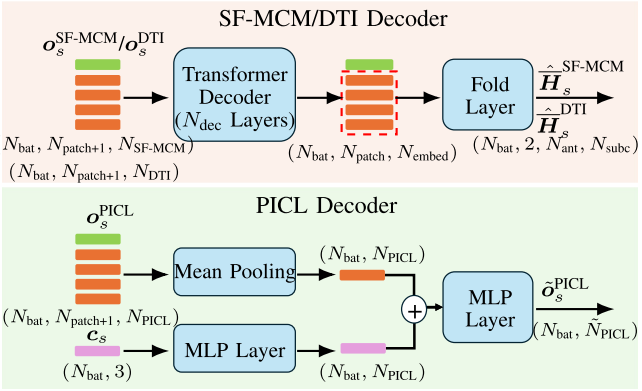


Fig. 4. Schematic diagram of pretraining decoder architecture.

For the PICL decoder, the semantic representation $\boldsymbol{o}_s^{\text{PICL}}$ is first processed by a mean pooling operation over the patch dimension, producing a pooled channel representation of shape $(N_{\text{bat}}, N_{\text{PICL}})$. Simultaneously, the BS configuration vector $\boldsymbol{c}_s$ is passed through a single-layer MLP to generate a configuration embedding of the same shape as the pooled channel representation. These two embeddings are then combined via element-wise addition. The resulting vector is fed into another single-layer MLP, which outputs a final location-dependent embedding $\tilde{\boldsymbol{o}}_s^{\text{PICL}} \in \mathbb{R}^{N_{\text{bat}}\times \tilde{N}_{\text{PICL}}}$, where $\tilde{N}_{\text{PICL}}$ is an output dimension for downstream contrastive learning objectives.

5) Pretraining Process: The pretraining procedure is summarized in Algorithm 1, which outlines the detailed steps for optimizing LWLM under the proposed SSL method. First, a batch of channel samples $\mathcal{N}_1^{\text{bat}}$ is drawn from the unlabeled dataset $\mathcal{D}^{\text{pre}}$. Based on $\mathcal{N}_1^{\text{bat}}$, a second batch $\mathcal{N}_2^{\text{bat}}$ is constructed, where each sample in $\mathcal{N}_2^{\text{bat}}$ serves as a positive counterpart for the corresponding sample in $\mathcal{N}_1^{\text{bat}}$, as discussed in Sec. IV-C. Using $\mathcal{N}_1^{\text{bat}}$, we compute the SF-MCM and DTI losses $\mathcal{L}_{\text{SF-MCM}}$ and $\mathcal{L}_{\text{DTI}}$ as described in Eq. (11) and (12), respectively (see Sections IV-A and IV-B). For the PICL loss $\mathcal{L}_{\text{PICL}}$, both batches $\mathcal{N}_1^{\text{bat}}$ and $\mathcal{N}_2^{\text{bat}}$ are used jointly to compute the loss, as defined in Eq. (15) and (16). For each task-specific loss, we first compute the full semantic representation $\boldsymbol{o}_s$ for every sample s, and extract the corresponding task-specific embeddings $\boldsymbol{o}_s^{\text{SF-MCM}}$, $\boldsymbol{o}_s^{\text{DTI}}$, and $\boldsymbol{o}_s^{\text{PICL}}$ as described in Sec IV-A, IV-B and IV-C, and illustrated in Fig. 3. Finally, the total pretraining objective $\mathcal{L}_{\text{pretrain}}$ is defined as a weighted combination of the three pretraining losses:

$$\mathcal{L}_{\text{pretrain}} = \alpha_{\text{SF-MCM}}\mathcal{L}_{\text{SF-MCM}} + \alpha_{\text{DTI}}\mathcal{L}_{\text{DTI}} + \alpha_{\text{PICL}}\mathcal{L}_{\text{PICL}}, \tag{21}$$

where $\alpha_{\text{SF-MCM}}$, α_{DTI}, α_{PICL} are predefined weighting coefficients that balance the contribution of each loss component.

V. Downstream Localization-Relevant Tasks

We evaluate LWLM on four localization-related downstream tasks: ToA estimation, AoA estimation, single-BS localization,

Algorithm 1 Self-Supervised Pretraining of LWLM

Input: $\mathcal{D}^{\text{pre}}$, N_{bat}, τ, $\alpha_{\text{SF-MCM}}$, α_{DTI}, α_{PICL}
Output: $\mathbf{\Phi}$, $\mathbf{\Theta}_{\text{SF-MCM}}, \mathbf{\Theta}_{\text{DTI}}, \mathbf{\Theta}_{\text{PICL}}$
foreach *training step* **do**
  Sample a batch $\mathcal{N}_1^{\text{bat}} \subseteq \mathcal{D}^{\text{pre}}$;
  Sample another batch $\mathcal{N}_2^{\text{bat}} \subseteq \mathcal{D}^{\text{pre}}$ based on $\mathcal{N}_1^{\text{bat}}$;
  // **SF-MCM Task**
  foreach $\boldsymbol{H}_s \in \mathcal{N}_1^{bat}$ **do**
    Apply random masking to obtain $T^{\text{SF-MCM}}(\boldsymbol{H}_s)$ using (10);
    Generate $\boldsymbol{o}_s$ using $\mathcal{F}_{\mathbf{\Phi}}^{\text{LWLM}}(\cdot)$ and obtain $\boldsymbol{o}_s^{\text{SF-MCM}}$;
    Reconstruct $\hat{\boldsymbol{H}}_s$ using $\mathcal{F}_{\mathbf{\Theta}_{\text{SF-MCM}}}^{\text{SF-MCM}}(\cdot)$;
  Compute SF-MCM loss $\mathcal{L}_{\text{SF-MCM}}$ using (11);
  // **DTI Task**
  foreach $\boldsymbol{H}_s \in \mathcal{N}_1^{bat}$ **do**
    Generate $\boldsymbol{o}_s$ using $\mathcal{F}_{\mathbf{\Phi}}^{\text{LWLM}}(\cdot)$ and obtain $\boldsymbol{o}_s^{\text{DTI}}$;
    Reconstruct $\hat{\boldsymbol{H}}_s^{\text{DTI}}$ using $\mathcal{F}_{\mathbf{\Theta}_{\text{DTI}}}^{\text{DTI}}(\cdot)$;
  Compute DTI loss $\mathcal{L}_{\text{DTI}}$ using (12);
  // **PICL Task**
  foreach $\boldsymbol{H}_s \in \mathcal{N}_1^{bat} \cup \mathcal{N}_2^{bat}$ **do**
    Generate $\boldsymbol{o}_s$ using $\mathcal{F}_{\mathbf{\Phi}}^{\text{LWLM}}(\cdot)$ and obtain $\boldsymbol{o}_s^{\text{PICL}}$;
    Obtain latent representation $\tilde{\boldsymbol{o}}_s^{\text{PICL}}$ using $\mathcal{F}_{\mathbf{\Theta}_{\text{PICL}}}^{\text{PICL}}(\cdot)$;
  Compute PICL loss $\mathcal{L}_{\text{PICL}}$ using (15) and (16);
  Compute total loss $\mathcal{L}_{\text{pretrain}}$;
  Update parameters $\mathbf{\Phi}$, $\mathbf{\Theta}_{\text{SF-MCM}}, \mathbf{\Theta}_{\text{DTI}}, \mathbf{\Theta}_{\text{PICL}}$ by minimizing $\mathcal{L}_{\text{pretrain}}$;

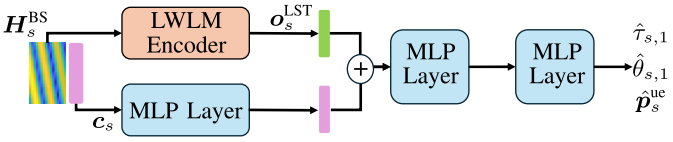


Fig. 5. Schematic diagram of the encoder-decoder architecture for ToA estimation, AoA estimation, and single-BS localization. Blue boxes indicate the decoder components.

and multi-BS localization. Each task employs a lightweight decoder fine-tuned with a small amount of labeled data, with both the pretrained encoder and decoder updated during fine-tuning.

A. ToA Estimation

ToA estimation aims to predict the first-path propagation delay $\tau_{s,1}$ from the BS to the UE for channel $\boldsymbol{H}_s$ in LOS scenarios, which is often used to infer the radial distance between them. We first extract the GCS $\boldsymbol{o}_s^{\text{LST}}$ using the pretrained LWLM encoder $\mathcal{F}_{\mathbf{\Phi}}^{\text{LWLM}}(\cdot)$, as introduced in Sec. IV-D. The ToA is then predicted using a task-specific decoder $\mathcal{F}_{\mathbf{\Theta}_{\text{ToA}}}^{\text{d-task}}$ that takes both the semantic representation $\boldsymbol{o}_s^{\text{LST}} = \mathcal{F}_{\mathbf{\Phi}}^{\text{LWLM}}(\boldsymbol{H}_s)$ and the BS configuration $\boldsymbol{c}_s$ as input:

$$\hat{\tau}_{s,1} = \mathcal{F}_{\mathbf{\Theta}_{\text{ToA}}}^{\text{d-task}}(\boldsymbol{o}_s^{\text{LST}}, \boldsymbol{c}_s). \tag{22}$$

The decoder is implemented by MLPs as shown in Fig. 5. The BS configuration $\boldsymbol{c}_s$ uses a single-layer MLP embedding and adds it to $\boldsymbol{o}_s^{\text{LST}}$, which is then estimated by a two-layer regression MLP. The training objective is to minimize the mean absolute error (MAE):

$$\mathcal{L}_{\text{ToA}} = \frac{1}{N_{\text{bat}}} \sum_{s \in \mathcal{N}_{\text{bat}}} \left| \hat{\tau}_{s,1} - \tau_{s,1} \right|. \tag{23}$$

B. AoA Estimation

Similar to ToA estimation, AoA estimation aims to determine the azimuth angle $\theta_{s,1}$ of the first path from the UE to the BS in LOS scenarios. Based on channel $\boldsymbol{H}_s$, we use the LWLM encoder to extract the GCS $\boldsymbol{o}_s^{\text{LST}} = \mathcal{F}_{\mathbf{\Phi}}^{\text{LWLM}}(\boldsymbol{H}_s)$. The AoA is then estimated by a dedicated decoder $\mathcal{F}_{\mathbf{\Theta}_{\text{AoA}}}^{\text{d-task}}$:

$$\hat{\theta}_{s,1} = \mathcal{F}_{\mathbf{\Theta}_{\text{AoA}}}^{\text{d-task}}(\boldsymbol{o}_s^{\text{LST}}, \boldsymbol{c}_s). \tag{24}$$

The decoder architecture is similar to that of ToA, as shown in Fig. 5. The loss function is also based on MAE:

$$\mathcal{L}_{\text{AoA}} = \frac{1}{N_{\text{bat}}} \sum_{s \in \mathcal{N}_{\text{bat}}} \left| \hat{\theta}_{s,1} - \theta_{s,1} \right|. \tag{25}$$

C. Single-BS Localization

Single-BS localization aims to estimate the UE position based solely on the CSI received from a single BS. Given a CFR sample $\boldsymbol{H}_s$, the LWLM encoder extracts the GCS representation $\boldsymbol{o}_s^{\text{LST}}$. A task-specific decoder $\mathcal{F}_{\mathbf{\Theta}_{\text{SB-Loc}}}^{\text{d-task}}$ then predicts the UE location $\hat{\boldsymbol{p}}_s^{\text{ue}}$:

$$\hat{\boldsymbol{p}}_s^{\text{ue}} = \mathcal{F}_{\mathbf{\Theta}_{\text{SB-Loc}}}^{\text{d-task}}(\boldsymbol{o}_s^{\text{LST}}, \boldsymbol{c}_s), \tag{26}$$

As illustrated in Fig. 5, the decoder shares the same architecture as those used for ToA and AoA estimation. It consists of a configuration embedding module (a single-layer MLP) followed by a two-layer MLP with output dimension d_p. The configuration embedding is first combined with the semantic vector $\boldsymbol{o}_s^{\text{LST}}$ via element-wise addition, and the resulting feature is used for position regression. The training loss is the mean Euclidean distance:

$$\mathcal{L}_{\text{loc}} = \frac{1}{N_{\text{bat}}} \sum_{s \in \mathcal{N}_{\text{bat}}} \left\| \hat{\boldsymbol{p}}_s^{\text{ue}} - \boldsymbol{p}_s^{\text{ue}} \right\|_2. \tag{27}$$

D. Multi-BS Localization

Although single-BS localization performs well in certain cases, it often suffers in multipath-rich or NLoS scenarios. In contrast, multi-BS localization aggregates CSI from multiple BSs to jointly estimate the UE position, thereby improving robustness and accuracy.

Let M denote the number of participating BSs. For a fixed UE location, the set of CFR samples collected from M BSs is denoted as $\{\boldsymbol{H}_{m,s}\}_{m=1}^{M}$. Each channel sample is independently encoded by the LWLM encoder to obtain the corresponding GCS representation $\boldsymbol{o}_{m,s}^{\text{LST}}$.

To maintain scalability with varying numbers of BSs, we extend the single-BS decoder design, as shown in Fig. 5. Specifically, each BS independently estimates a candidate UE position $\hat{\boldsymbol{p}}_{m,s}^{\text{ue}}$ using a dedicated single-BS decoder $\mathcal{F}_{\mathbf{\Theta}_{\text{SB-Loc},m}}^{\text{d-task}}$, which operates on the semantic vector $\boldsymbol{o}_{m,s}^{\text{LST}}$ and its corresponding BS configuration $\boldsymbol{c}_{m,s}$:

$$\hat{\boldsymbol{p}}_{m,s}^{\text{ue}} = \mathcal{F}_{\mathbf{\Theta}_{\text{SB-Loc},m}}^{\text{d-task}}(\boldsymbol{o}_{m,s}^{\text{LST}}, \boldsymbol{c}_{m,s}). \tag{28}$$

An attention-based fusion mechanism is subsequently applied to aggregate position estimates from all BSs. The final UE

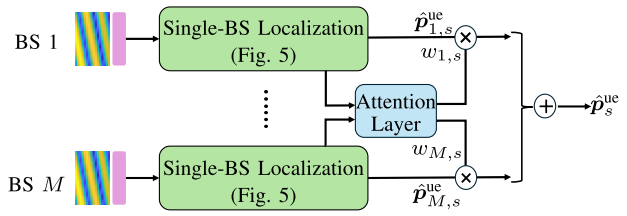


Fig. 6. Schematic diagram of multi-BS localization decoder architecture.

position is computed as a weighted sum of the individual predictions:

$$\hat{\boldsymbol{p}}_s^{\text{ue}} = \sum_{m=1}^{M} w_{m,s} \cdot \hat{\boldsymbol{p}}_{m,s}^{\text{ue}}, \tag{29}$$

where $w_{m,s}$ denotes the attention weight associated with BS m. The attention weight $w_{m,s}$ is derived from the feature vector $\boldsymbol{x}_{m,s}^{\text{SB}}$ extracted from the penultimate layer of the corresponding single-BS localization decoder. We employ a lightweight MLP as the attention layer, denoted as $\text{MLP}_{\text{attn}}(\cdot)$, to compute the attention logits:

$$w_{m,s} = \frac{\exp\left(\text{MLP}_{\text{attn}}(\boldsymbol{x}_{m,s}^{\text{SB}})\right)}{\sum_{m^*=1}^{M} \exp\left(\text{MLP}_{\text{attn}}(\boldsymbol{x}_{m^*,s}^{\text{SB}})\right)}. \tag{30}$$

This attention mechanism enables the model to assign adaptive confidence weights to each BS, effectively leveraging the most reliable information sources in challenging environments.

Consequently, the overall multi-BS localization decoder $\mathcal{F}_{\boldsymbol{\Theta}_{\text{MB-Loc}}}$ consists of two key components: (i) Single-BS decoders $\{\mathcal{F}^{\text{d-task}}_{\boldsymbol{\Theta}_{\text{SB-Loc},m}}\}_{m=1}^{M}$ and (ii) an attention-based fusion module implemented via $\text{MLP}_{\text{attn}}(\cdot)$. During fine-tuning, we only need to train the attention layer once, even for different numbers of participating BSs. The output of the attention layer can be normalized by (30) to adapt to different numbers of BSs. In addition, during actual deployment, the localization results and weights of each BS can be transmitted to the fusion center or the user side for aggregation, effectively reducing the data transmission overhead. This modular design offers both flexibility and scalability, making it well-suited for deployment in dynamic wireless environments with varying numbers of BSs. Similar to the single-BS localization, we use Eq. (27) as the loss function.

VI. Experiments

A. Experiment Settings

In this study, we use DeepMIMO [49], a widely adopted ray-tracing-based channel dataset, to evaluate our proposed wireless localization algorithms. Specifically, we select the O1 scenario [49], which models a typical outdoor urban environment with two intersecting streets, as shown in Fig. 7.

For the self-supervised pretraining phase, we consider 4 BSs (i.e., BS 3, 4, 9, and 10). Each BS is equipped with a ULA comprising 32 antennas and utilizes 128 subcarriers. The channel bandwidths used during the pretraining phase are set to 10 MHz, 20 MHz, and 50 MHz. Channel data are collected from the spatial region highlighted by the red rectangle along the horizontal street shown in Fig. 7. For each BS configuration (combination of BS location and bandwidth setting),

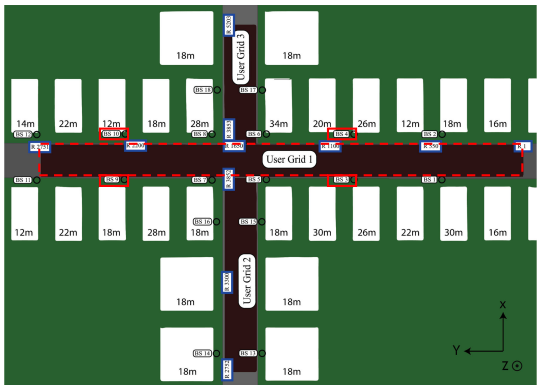


Fig. 7. Schematic diagram of channel simulation scenario O1. The red dashed box is the sampling area of the channel, and the red solid box represents the 4 BSs involved in channel collection.

TABLE I
System and Model Parameters

Parameter	Value
Number of antennas/subcarriers ($N_{\text{ant}}/N_{\text{subc}}$)	32/128
BS bandwidth (B_{bw})	10/20/50 MHz
2D CNN kernel size stride (K_{cnn}) / stride (S_{cnn})	4 / 4
Number of patches (N_{patch})	256
Number of Transformer encoder layers (N_{enc})	4
Number of Transformer decoder layers (N_{dec})	2
Number of masked antennas/subcarriers ($\hat{N}_{\text{ant}}^{M}/\hat{N}_{\text{subc}}^{M}$)	16/64
Number of heads in Transformer encoder/decoder	4
Embedding dimension (N_{embed})	512
Latent dimension (N_{latent})	256
Batch size (N_{bat})	32
SF-MCM latent dimension $\boldsymbol{o}_s^{\text{SF-MCM}}$ ($N_{\text{SF-MCM}}$)	96
DTI latent dimension $\boldsymbol{o}_s^{\text{DTI}}$ (N_{DTI})	96
PICL latent dimension $\boldsymbol{o}_s^{\text{PICL}}$ (N_{PICL})	64
PICL latent dimension for $\tilde{\boldsymbol{o}}_s^{\text{PICL}}$ ($\tilde{N}_{\text{PICL}}$)	32
Pretraining/fine-tuning epochs	200/1000
Learning rate	0.0001
Loss function weight ($\alpha_{\text{SF-MCM}}$, α_{DTI}, α_{PICL})	10, 20, 1
LWLM encoder parameters	5.27 M
ToA/AoA/Single-BS localization decoder parameters	0.07 M

we collect channel measurements from 488,698 unique spatial locations, resulting in a comprehensive pretraining dataset containing $4 \times 3 \times 488,698 = 5,864,376$ channel samples. In the fine-tuning stage, unless otherwise specified, we randomly select channel samples from $10,000$ locations for training, channel samples from $1,000$ locations for validation, and an additional 10,000 locations for testing the performance of our model. System parameters and model parameters deployment details are shown in Table I. We use 200 epochs for pretraining and 1000 epochs for fine-tuning. Although fine-tuning uses more epochs numerically, the total number of optimization steps during fine-tuning (about 300,000 steps) is nearly one order of magnitude smaller than that of pretraining (about 3,000,000 steps). The larger fine-tuning epoch budget is used to ensure sufficient convergence under limited labeled data and to allow non-pretrained baselines to fully approach their performance limits for fair comparison. In the ToA estimation, AoA estimation, and single BS localization experiments, we take the performance of BS 3 as a representative case study to evaluate and compare the effectiveness of the proposed method. Although the data for both pretraining and fine-tuning are collected from the same scenario, this setting is reasonable

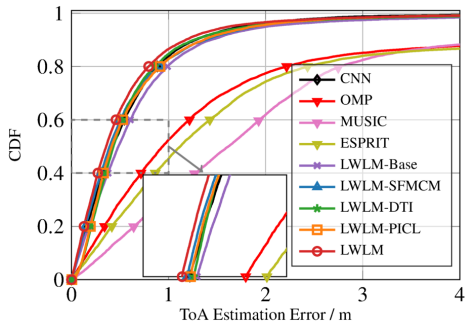


Fig. 8. CDF of ToA estimation error with 10,000 fine-tuning samples.

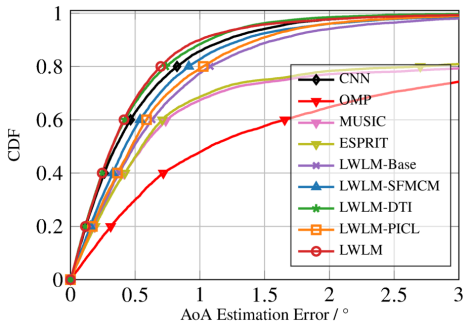


Fig. 9. CDF of AoA estimation error with 10,000 fine-tuning samples.

since the pretraining stage does not require labeled data, and in practical systems, each BS can typically provide sufficient channel data to support pretraining.

We evaluate the performance of the following algorithms:

- **LWLM**: The proposed model trained with our hybrid self-supervised pretraining approach combining SF-MCM, DTI, and PICL.
- **LWLM-SFMCM**: Only the SF-MCM-based pretraining strategy is used as an ablation experiment. LWLM-SFMCM is similar to existing work, i.e., LWM [29].
- **LWLM-DTI**: Only the DTI-based pretraining strategy is used as an ablation experiment.
- **LWLM-PICL**: Only the PICL-based pretraining strategy is used as an ablation experiment.
- **LWLM-Base**: The proposed Transformer-based architecture trained directly on the localization task without any pretraining. The baseline model employs an advanced Transformer structure but does not utilize pretraining for representation extraction.
- **CNN**: Use ResNet-34 [10], with 21.28 M parameters, as the backbone network model architecture for localization.
- **OMP**: A classical model-based localization algorithm employing OMP, which estimates angles and distances between the user and the BS. For multi-BS localization tasks, OMP estimates from multiple BSs are averaged to obtain the final location estimation.
- **MUSIC**: A subspace-based method that estimates spatial parameters using eigenvalue decomposition of the covariance matrix.
- **ESPRIT**: A subspace-based method that exploits rotational invariance properties for parameter estimation.

B. ToA Estimation Results

Fig. 8 presents the cumulative distribution function (CDF) of ToA estimation errors across different methods. Among the single-objective pretraining methods, LWLM-SFMCM, LWLM-DTI, and LWLM-PICL achieve 50% ToA estimation errors of 0.39 m, 0.43 m, and 0.41 m, respectively. The LWLM-Base model without any pretraining yields a 50% error of 0.48 m. Our proposed LWLM, which integrates all three SSL objectives, achieves the best performance with a 50% ToA estimation error of only 0.36 m. This corresponds to a reduction of approximately 9.0%–16.7% in error compared to the single-objective variants, and a 26.0% reduction relative to LWLM-Base. LWLM also outperforms the CNN-based approach, achieving a performance improvement of 16.0%. Moreover, LWLM achieves better performance than the high-resolution model-based algorithms, including OMP, MUSIC, and ESPRIT, further demonstrating its effectiveness for high-precision ToA estimation.

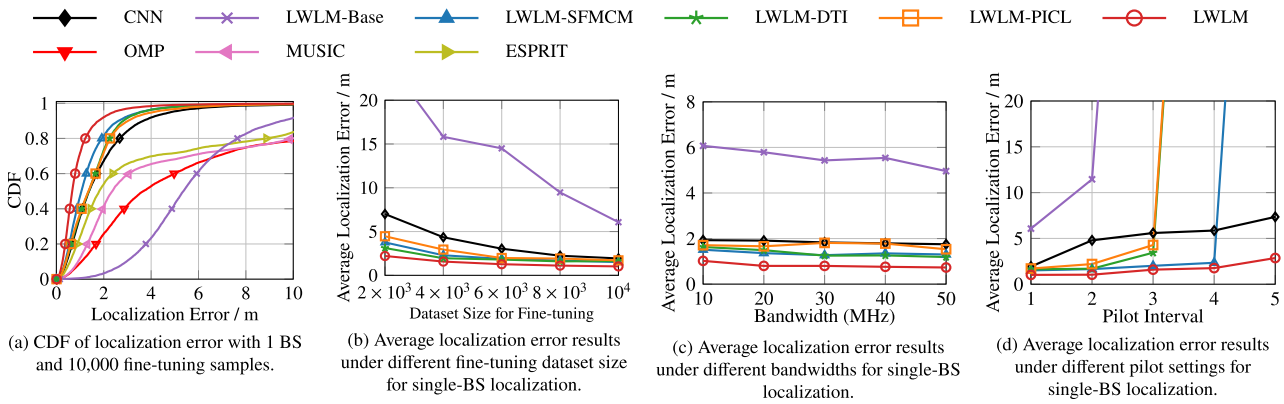
C. AoA Estimation Results

Fig. 9 compares the AoA estimation performance across different models in terms of the CDF of estimation errors. Among all methods, our proposed LWLM achieves the best result, with a 50% AoA error of only 0.32°, indicating its strong capability in capturing angle-related semantic features from wireless channels. The LWLM-Base model without pretraining reaches a 50% error of 0.49°, while the single-objective pretrained models, i.e., LWLM-SFMCM, LWLM-DTI, and LWLM-PICL, show 0.36°, 0.32°, and 0.47°, respectively. LWLM achieves a 34.7% and 25.9% reduction in the median AoA error compared to LWLM-Base and the CNN approach, respectively. Furthermore, LWLM also outperforms the model-based methods (OMP, MUSIC, and ESPRIT). Notably, LWLM-DTI achieves the best performance among the three single-objective SSL methods, as its cross-domain consistency objective enables the encoder to capture features in the angle-delay domain that are directly relevant to AoA estimation.

D. Single-BS Localization Results

Fig. 10 (a) shows the CDF of localization errors. All three single-pretraining methods significantly outperform the non-pretrained models (LWLM-Base and CNN), while the proposed LWLM, integrating all three objectives, further improves localization accuracy. Specifically, the proposed LWLM achieves a median localization error of 0.67 m, reducing the error by approximately 81.8%, 71.6%, and 63.0% compared to the model-based OMP, MUSIC, and ESPRIT algorithms, respectively. Additionally, LWLM achieves 51.4%–87.5% improvement over LWLM-Base and CNN, and further reduces localization error by 36.7%–51.1% compared with single-pretraining methods, demonstrating the effectiveness of the proposed hybrid SSL approach in single-BS localization.

To evaluate generalization under limited supervision during the fine-tuning phase, we compare localization accuracy across



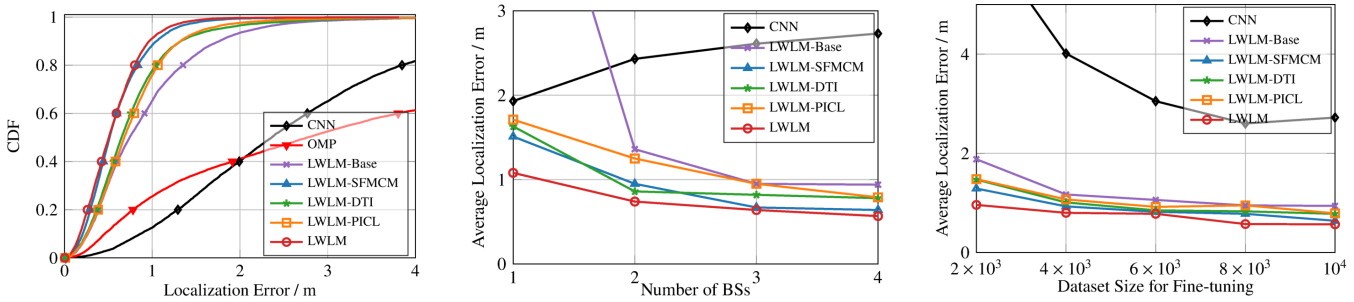
(a) CDF of localization error with 1 BS and 10,000 fine-tuning samples.

(b) Average localization error results under different fine-tuning dataset size for single-BS localization.

(c) Average localization error results under different bandwidths for single-BS localization.

(d) Average localization error results under different pilot settings for single-BS localization.

Fig. 10. Performance evaluation of single-BS localization tasks under different settings.



(a) CDF of localization error with 4 BSs and 10,000 fine-tuning samples.

(b) Average localization error results under different numbers of BSs.

(c) Average localization error results under different fine-tuning dataset sizes for 4-BS localization.

Fig. 11. Performance evaluation of multi-BS localization tasks under different settings.

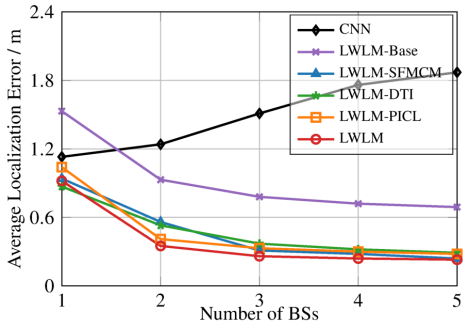


Fig. 12. Average localization error with different numbers of BSs on Sionna datasets.

different fine-tuning sample sizes, as shown in Fig. 10 (b). We observe that the localization error consistently decreases as the amount of training data increases. However, when fine-tuning data is scarce, the advantages of self-supervised pretraining become particularly prominent. Notably, our proposed LWLM model, utilizing a hybrid pretraining strategy, achieves an average localization error of only 2.21 m with only 2,000 fine-tuning samples, significantly outperforming non-pretrained models and those baselines using a single pretraining method. These results confirm the strong generalization capability of LWLM in label-limited localization scenarios.

Fig. 10 (c) illustrates the localization performance under varying bandwidth configurations. As the bandwidth increases, the temporal resolution of the channel improves, resulting in lower localization error across all models. Importantly, our pretraining dataset includes only 10 MHz, 20 MHz, and 50 MHz bandwidths, whereas the 30 MHz and 40 MHz configurations were excluded from pretraining. Despite this, LWLM maintains strong performance across all bandwidth settings. With 30MHz bandwidth, LWLM can achieve an average localization error of 0.80 m, which is 36.0-55.8% lower than other methods based on a single pretraining method. This result demonstrates the robustness of our pretrained model and its strong transferability across unseen deployment conditions.

To simulate practical LTE/5G systems where CSI is available only at pilot positions, we evaluate localization under comb-type pilot configurations. Different from Fig. 10 (c), Fig. 10 (d) shows results at 10 MHz with pilot intervals $N_{\text{pilot}} = \{1, 2, 3, 4, 5\}$, where pilots are inserted every N_{pilot} subcarriers or antennas and other entries are set to zero. Here, $N_{\text{pilot}} = 1$ indicates that pilot signals are present at all positions of the channel matrix. As N_{pilot} increases, performance degrades due to sparser channel information. When $N_{\text{pilot}} = 3$, LWLM-Base suffers significant performance loss. DTI and PICL maintain reasonable accuracy up to $N_{\text{pilot}} = 3$ but degrade rapidly beyond that. SF-MCM learns the correlations across subcarriers and antennas, making it naturally suited for modeling channel features under sparse pilot configurations. It therefore outperforms other single-objective variants but still exhibits significant degradation when $N_{\text{pilot}} > 4$. In contrast, the proposed LWLM achieves the highest robustness, attaining only 2.86 m average error even at $N_{\text{pilot}} = 5$, highlighting the effectiveness of combining masked modeling and invariant feature learning under sparse pilot conditions.

E. Multi-BS Localization Results

Fig. 11 (a) shows the CDF of localization errors with 4 BSs and 10,000 fine-tuning samples. Even without pretraining, LWLM-Base achieves a median error of 0.74 m, significantly outperforming CNN, as the multi-BS setting enables learning position-invariant semantics across BSs, thus facilitating better generalization. With pretraining, performance further improves. Our proposed LWLM achieves the best median error of 0.51 m, corresponding to 1.9%–25.9% improvement over single-objective pretraining methods and 32.0% over LWLM-Base.

Fig. 11 (b) illustrates the impact of the number of BSs on localization performance. As shown in Fig. 11 (b), the localization error of all transformer-based methods consistently decreases as the number of BSs increases. This is expected, as more BSs provide richer spatial information, which helps the model better disambiguate UE positions and improves estimation certainty. In contrast, the CNN method exhibits a performance degradation when the number of BSs increases. This is because the increase in the number of BSs leads to an increase in the input dimension, which exacerbates the overfitting of the CNN architecture. However, CNN is unable to perform effective semantic aggregation, which increases the localization error during testing. Among all methods, the proposed hybrid pretraining model LWLM consistently achieves the best localization accuracy. When using 4 BSs, LWLM achieves an average localization error of 0.57 meters, which is 11.9% to 27.8% lower than all single pretraining algorithms. This highlights the robustness and scalability of the hybrid self-supervised framework in multi-BS scenarios.

Fig. 11 (c) shows the average localization error under different fine-tuning dataset sizes. Increasing the amount of labeled data improves the performance of all models. Our proposed LWLM consistently outperforms all baselines across all data scales. Remarkably, even with only 2,000 fine-tuning samples, LWLM achieves an average localization error of 0.96 m, which represents a relative improvement of 25.6%–48.9% over other baselines based on the transformer. This demonstrates the strong data efficiency and few-shot generalization capability of LWLM in multi-BS localization scenarios.

F. Cross-Dataset Results

To evaluate cross-domain generalization, we test LWLM on the Sionna dataset [50], which features a distinct propagation environment from the DeepMIMO-O1 scenario used for pretraining. We adopt the same system configuration (10 MHz bandwidth, 32 antennas per BS, 128 subcarriers) but with Sionna's ray-tracing channel model in a different 64 m $\times$64 m scenario with 5 BSs at the 4 corners and the center. We use 2000 labeled samples for fine-tuning and 1000 for testing. The results for varying numbers of BSs are shown in Fig. 12. CNN performance degrades as the number of BSs increases due to overfitting caused by the enlarged input dimension. LWLM achieves the best accuracy in almost all cases, except for a slight inferiority to LWLM-DTI in the single-BS scenario. For example, with 2 BSs, LWLM improves accuracy by approximately 0.18–0.93 m over all baselines. These results confirm the strong cross-domain generalization of the proposed hybrid pretraining framework. Although ray-tracing datasets do not fully capture real-world hardware impairments such as phase noise, frequency offset, RF nonlinearity, and calibration mismatch, these effects can be incorporated during pretraining to improve robustness. In addition, domain adaptation strategies, such as fine-tuning with limited real-world labeled data or impairment-aware data augmentation, can help mitigate the simulation-to-reality gap.

TABLE II
MODEL COMPLEXITY AND AVERAGE INFERENCE LATENCY

Algorithm	Params (M)	FLOPs (G)	Inference latency (ms)
OMP	–	–	86
MUSIC	–	–	140
ESPRIT	–	–	63
CNN	21.28	0.59	2.4
LWLM	5.27	2.70	2.5

G. Computational Complexity and Inference Latency

We further analyze the computational complexity and inference latency of the proposed LWLM and compare it with representative baselines. Table II reports parameter count, FLOPs, and average inference latency per sample for the single-BS localization task. The inference latency is measured by dividing the total processing time by the number of samples, with 1000 samples used for testing. For the AI-based methods (CNN and LWLM), inference is performed on an NVIDIA A40 GPU, while for the model-based methods (OMP, MUSIC, and ESPRIT), inference is performed on an Intel Xeon Gold 6226R CPU at 2.90 GHz. Although LWLM has higher theoretical FLOPs than CNN, its practical inference latency remains comparable (about 2.5 ms). This is due to the high parallelizability of Transformer operations on GPUs. In contrast, classical model-based methods such as OMP, MUSIC, and ESPRIT require significantly longer inference time (63–140 ms per sample), as they are typically executed on CPUs and are less amenable to parallelization. The latency of about 2.5 ms per sample is within the typical 6G positioning budget, indicating the feasibility of LWLM for delay-sensitive localization applications.

H. Impact of Fine-Tuning Strategies

We also compare three fine-tuning strategies for LWLM: (i) Frozen (only the decoder is trained), (ii) Gradual unfreezing (encoder frozen for the first 500 epochs, then fully fine-tuned), and (iii) Full fine-tune (all parameters updated throughout). LWLM-Base, trained from scratch without pretraining, is included as a reference. Fig. 13 compares the localization performance of these strategies. With a frozen encoder, LWLM achieves a median localization error of 3.39 m, which outperforms LWLM-Base, confirming that the pretrained representations alone carry substantial localization-relevant semantics. Gradual unfreezing further improves performance to 0.74 m by allowing the encoder to adapt after the decoder has stabilized, but still exhibits approximately 10.5% higher

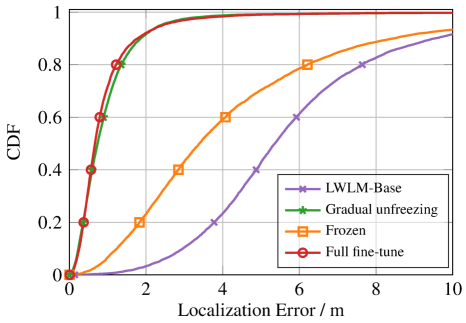


Fig. 13. CDF of single-BS localization error under different fine-tuning strategies.

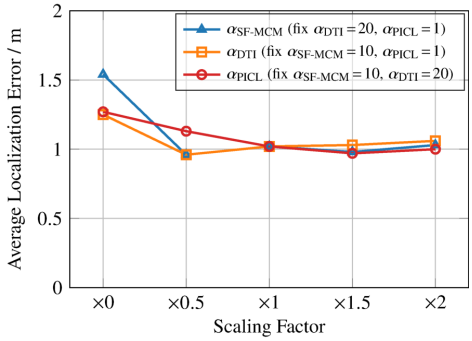


Fig. 14. Average single-BS localization error under different scaling factors of each loss weight.

median error compared to full fine-tuning. Full fine-tuning achieves the best median localization error of 0.67 m, as it allows the encoder to fully adapt to the downstream localization task. These results indicate that the pretrained representations provide a high-quality initialization, but the gap between the self-supervised pretraining objectives and the supervised localization task requires encoder adaptation for optimal performance. In future work, we will explore parameter-efficient fine-tuning techniques such as low-rank adaptation (LoRA) to better balance preserving pretrained knowledge and adapting to downstream tasks.

I. Sensitivity Analysis of Loss Weights

To investigate the sensitivity of LWLM to the loss weight selection, we vary each weight individually while keeping the other two fixed at their default values. Fig. 14 shows the average single-BS localization error as a function of the scaling factor applied to each weight. The scaling factor $\times 0$ corresponds to removing the respective pretraining objective, while $\times 1$ corresponds to the default setting ($\alpha_{\text{SF-MCM}} = 10$, $\alpha_{\text{DTI}} = 20$, $\alpha_{\text{PICL}} = 1$). As shown in Fig. 14, removing any single objective ($\times 0$) consistently degrades performance, with SF-MCM removal causing the largest drop. Within the $\times 0.5$ to $\times 2$ range, all three curves exhibit only marginal variation, confirming that LWLM is robust to moderate weight variations. These results indicate that the default weights are near-optimal under the linear combination framework, and that the hybrid of all three objectives consistently outperforms any pairwise combination. Furthermore, more flexible multi-task balancing strategies, such as Mixture-of-Experts (MoE), may further improve the integration of different SSL objectives and can be explored in future work.

VII. Conclusion

In this paper, we proposed the LWLM, a transformer-based self-supervised foundation model tailored for wireless localization tasks. To overcome the limitations of task-specific models and improve generalization across diverse wireless configurations, we introduced a hybrid SSL framework that integrates SF-MCM, DTI, and PICL. These objectives enable LWLM to learn rich, diverse, and transferable channel semantics from large-scale unlabeled data. We further designed lightweight task-specific decoders for a range of downstream localization tasks, including ToA and AoA estimation, as well as single-BS and multi-BS localization. Extensive experiments demonstrate that LWLM significantly outperforms existing supervised and unsupervised baselines across all tasks. In particular, LWLM achieves high accuracy with minimal labeled data and demonstrates strong robustness to unseen BS configurations and cross-dataset scenarios. Overall, LWLM represents a meaningful step toward foundation models for wireless localization in future 6G networks, providing a reusable pretrained encoder that can be efficiently adapted to a range of localization-related tasks.

Looking ahead, future work could focus on extending LWLM towards a broader localization foundation model. Important directions include considering 3D localization scenarios, incorporating hardware impairments, exploring richer BS configurations, and leveraging multimodal auxiliary information. While ray-tracing channels serve as a necessary and widely accepted first step, extending LWLM to real-world measurements is an important next step. In addition, with respect to AI model training and fine-tuning, it is crucial to investigate broader deployment scenarios while reducing fine-tuning overhead, such as LoRA.

Appendix A
Proof of Theorem 1

For each generative-based SSL task, the IB objective can be expanded as $I(O;H) - \beta_{k_g} I\left(O; T_{k_g}(H)\right) = H(H) - H(H|O) - \beta_{k_g}\left(H(T_{k_g}(H)) - H(T_{k_g}(H)|O)\right) = H(H) - H\left(T_{k_g}(H)|O\right) - H\left(\overline{T}_{k_g}(H)|T_{k_g}(H), O\right) - \beta_{k_g}\left(H(T_{k_g}(H)) - H(T_{k_g}(H)|O)\right) = H(H) - \beta_{k_g} H\left(T_{k_g}(H)\right) + (\beta_{k_g} - 1) H\left(T_{k_g}(H)|O\right) - H\left(\overline{T}_{k_g}(H)|T_{k_g}(H), O\right)$. Here, $H(H) - \beta_{k_g} H\left(T_{k_g}(H)\right) = C$ is a constant independent of the parameters $\mathbf{\Phi}$, $\mathbf{\Theta}_{k_g}$ and $\overline{\mathbf{\Theta}}_{k_g}$.

We assume that $q_{\mathbf{\Phi}}(O|H) \approx q(O|H)$, $p_{\mathbf{\Theta}_{k_g}}(T_{k_g}(H)|O) \approx p(T_{k_g}(H)|O)$, and $p_{\overline{\mathbf{\Theta}}_{k_g}}(\overline{T}_{k_g}(H)|T_{k_g}(H), O) \approx p(\overline{T}_{k_g}(H)|T_{k_g}(H), O)$.[7] Generative model $p_{\mathbf{\Theta}_{k_g}}(T_{k_g}(H)|O)$ and $p_{\overline{\mathbf{\Theta}}_{k_g}}(\overline{T}_{k_g}(H)|T_{k_g}(H), O)$ can also be considered as decoders for $T_{k_g}(H)$ and $\overline{T}_{k_g}(H)$. Then, we have

$$H\left(T_{k_g}(H)|O\right) = -\mathbb{E}_{p(O,H)}\left[\ln p(T_{k_g}(H)\,|\,O)\right]$$

[7]In practice, we assume that the model capacity is large enough for these parameterized distributions to closely match the true ones.

$$\approx -\mathbb{E}_{p(H)q_{\boldsymbol{\Phi}}(O|H)}\Big[\ln p_{\boldsymbol{\Theta}_{k_g}}\big(T_{k_g}(H)|O\big)\Big] = \mathcal{L}_{k_g}(\boldsymbol{\Phi},\boldsymbol{\Theta}_{k_g}), \tag{31}$$

and

$$\begin{aligned} &H\big(\overline{T}_{k_g}(H)|T_{k_g}(H),O\big) \\ &\quad = -\mathbb{E}_{p(O,H)}\big[\ln p\big(\overline{T}_{k_g}(H)|T_{k_g}(H),O\big)\big] \\ &\quad \approx -\mathbb{E}_{p(H)q_{\boldsymbol{\Phi}}(O|H)}\left[\ln p_{\overline{\boldsymbol{\Theta}}_{k_g}}\left(\overline{T}_{k_g}(H)|T_{k_g}(H),O\right)\right] \\ &\quad = \overline{\mathcal{L}}_{k_g}(\boldsymbol{\Phi},\overline{\boldsymbol{\Theta}}_{k_g}). \end{aligned} \tag{32}$$

Based on above, We have $I(O;H)-\beta_{k_g}I\big(O;T_{k_g}(H)\big) \approx (\beta_{k_g}-1)\mathcal{L}_{k_g}(\boldsymbol{\Phi},\boldsymbol{\Theta}_{k_g}) - \overline{\mathcal{L}}_{k_g}(\boldsymbol{\Phi},\overline{\boldsymbol{\Theta}}_{k_g}) + C$, which completes the proof.

APPENDIX B
PROOF OF THEOREM 2

Let O and O^+ denote the random variables corresponding to the encoder representations of two channel observations $(\boldsymbol{H}_s,\boldsymbol{H}_{s^+})$ that share the same label $\boldsymbol{y}_s$. Assuming the encoder operates independently across samples, and both $\boldsymbol{H}_s$ and $\boldsymbol{H}_{s^+}$ are drawn conditionally on the same label, we have the Markov chain of random variables: $O \to Y \to O^+$. By the data processing inequality, we obtain $I(O;Y) \geq I(O;O^+)$. Following the result from [46], the InfoNCE loss L_{InfoNCE} provides a variational lower bound on $I(O;O^+)$: $I(O;O^+) \geq \log(N_{\text{bat}}) - L_{\text{InfoNCE}}$. Combining the two inequalities, we get $I(O;Y) \geq I(O;O^+) \geq \log(N_{\text{bat}}) - L_{\text{InfoNCE}}$, which completes the proof.

REFERENCES

[1] H. Wymeersch et al., "6G positioning and sensing through the lens of sustainability, inclusiveness, and trustworthiness," *IEEE Wireless Commun.*, vol. 32, no. 1, pp. 68–75, Feb. 2025.

[2] L. Italiano, B. Camajori Tedeschini, M. Brambilla, H. Huang, M. Nicoli, and H. Wymeersch, "A tutorial on 5G positioning," *IEEE Commun. Surveys Tuts.*, vol. 27, no. 3, pp. 1488–1535, Jun. 2025.

[3] R. Di Taranto, S. Muppirisetty, R. Raulefs, D. Slock, T. Svensson, and H. Wymeersch, "Location-aware communications for 5G networks: How location information can improve scalability, latency, and robustness of 5G," *IEEE Signal Process. Mag.*, vol. 31, no. 6, pp. 102–112, Nov. 2014.

[4] G. Kwon, Z. Liu, A. Conti, H. Park, and M. Z. Win, "Integrated localization and communication for efficient millimeter wave networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 12, pp. 3925–3941, Dec. 2023.

[5] C. Chen, H. Jiang, and C. Pan, "Location-dependent performance analysis for RIS-aided or interference mitigation assisted large-scale networks," *IEEE Trans. Veh. Technol.*, vol. 74, no. 4, pp. 6844–6849, Apr. 2025.

[6] H. Chen, H. Sarieddeen, T. Ballal, H. Wymeersch, M. Alouini, and T. Y. Al-Naffouri, "A tutorial on terahertz-band localization for 6G communication systems," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 3, pp. 1780–1815, 3rd Quart., 2022.

[7] M. F. Keskin, H. Wymeersch, and V. Koivunen, "MIMO-OFDM joint radar-communications: Is ICI friend or foe?" *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 6, pp. 1393–1408, Nov. 2021.

[8] G. Pan et al., "AI-driven wireless positioning: Fundamentals, standards, state-of-the-art, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 28, pp. 4394–4428, 2026.

[9] J. Gao, D. Wu, F. Yin, Q. Kong, L. Xu, and S. Cui, "MetaLoc: Learning to learn wireless localization," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 12, pp. 3831–3847, Dec. 2023.

[10] C. Wu et al., "Learning to localize: A 3D CNN approach to user positioning in massive MIMO-OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4556–4570, Jul. 2021.

[11] G. Pan, T. Wang, S. Zhang, and S. Xu, "High accurate time-of-arrival estimation with fine-grained feature generation for Internet-of-Things applications," *IEEE Wireless Commun. Lett.*, vol. 9, no. 11, pp. 1980–1984, Nov. 2020.

[12] Z. Chen, Z. Zhang, Z. Xiao, Z. Yang, and R. Jin, "Deep learning-based multi-user positioning in wireless FDMA cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 12, pp. 3848–3862, Dec. 2023.

[13] R. Klus, J. Talvitie, J. Equi, G. Fodor, J. Torsner, and M. Valkama, "Robust NLoS localization in 5G mmWave networks: Data-based methods and performance," *IEEE Trans. Veh. Technol.*, vol. 74, no. 1, pp. 1534–1550, Jan. 2025.

[14] J. Liu, T. Wang, Y. Li, C. Li, Y. Wang, and Y. Shen, "A transformer-based signal denoising network for AoA estimation in NLoS environments," *IEEE Commun. Lett.*, vol. 26, no. 10, pp. 2336–2339, Oct. 2022.

[15] X. Gong, A. Lu, X. Liu, X. Fu, X. Gao, and X.-G. Xia, "Deep learning based fingerprint positioning for multi-cell massive MIMO-OFDM systems," *IEEE Trans. Veh. Technol.*, vol. 73, no. 3, pp. 3832–3849, Mar. 2024.

[16] W. Li, X. Meng, Z. Zhao, Z. Liu, C. Chen, and H. Wang, "LoT: A transformer-based approach based on channel state information for indoor localization," *IEEE Sensors J.*, vol. 23, no. 22, pp. 28205–28219, Nov. 2023.

[17] X. Xu et al., "Swin-loc: Transformer-based CSI fingerprinting indoor localization with MIMO ISAC system," *IEEE Trans. Veh. Technol.*, vol. 73, no. 8, pp. 11664–11679, Aug. 2024.

[18] Y. Ruan et al., "IPos-5G: Indoor positioning via commercial 5G NR CSI," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 8718–8733, May 2023.

[19] X. Chen, H. Li, C. Zhou, X. Liu, D. Wu, and G. Dudek, "Fidora: Robust WiFi-based indoor localization via unsupervised domain adaptation," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9872–9888, Jun. 2022.

[20] S. A. Junoh and J.-Y. Pyun, "Enhancing indoor localization with semi-crowdsourced fingerprinting and GAN-based data augmentation," *IEEE Internet Things J.*, vol. 11, no. 7, pp. 11945–11959, Apr. 2024.

[21] L. Li, X. Guo, M. Zhao, H. Li, and N. Ansari, "TransLoc: A heterogeneous knowledge transfer framework for fingerprint-based indoor localization," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3628–3642, Jun. 2021.

[22] R. Bommasani et al., "On the opportunities and risks of foundation models," 2021, *arXiv:2108.07258*.

[23] J. Gui et al., "A survey on self-supervised learning: Algorithms, applications, and future trends," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 9052–9071, Dec. 2024.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, Hum. Lang. Technol., Minneapolis, MI, USA, vol. 1, Jun. 2019, pp. 4171–4186.

[25] T. B. Brown et al., "Language models are few-shot learners," in *Proc. NIPS*, 2020, pp. 1877–1901.

[26] D. Hong et al., "SpectralGPT: Spectral remote sensing foundation model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5227–5244, Aug. 2024.

[27] N. Fei et al., "Towards artificial general intelligence via a multimodal foundation model," *Nature Commun.*, vol. 13, no. 1, p. 3094, Jun. 2022.

[28] X. He, Y. Chen, L. Huang, D. Hong, and Q. Du, "Foundation model-based multimodal remote sensing data classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024.

[29] S. Alikhani, G. Charan, and A. Alkhateeb, "Large wireless model (LWM): A foundation model for wireless channels," 2024, *arXiv:2411.08872*.

[30] T. Yang et al., "WirelessGPT: A generative pre-trained multi-task learning framework for wireless communication," 2025, *arXiv:2502.06877*.

[31] B. Liu, S. Gao, X. Liu, X. Cheng, and L. Yang, "WiFo: Wireless foundation model for channel prediction," 2024, *arXiv:2412.08908*.

[32] B. Liu, X. Liu, S. Gao, X. Cheng, and L. Yang, "LLM4CP: Adapting large language models for channel prediction," *J. Commun. Inf. Netw.*, vol. 9, no. 2, pp. 113–125, Jun. 2024.

[33] X. Liu, S. Gao, B. Liu, X. Cheng, and L. Yang, "LLM4WM: Adapting LLM for wireless multi-tasking," 2025, *arXiv:2501.12983*.

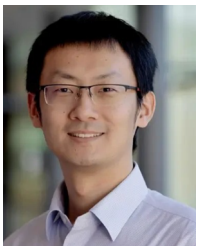
[34] A. Salihu, M. Rupp, and S. Schwarz, "Self-supervised and invariant representations for wireless localization," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 8281–8296, Aug. 2024.

[35] P. Stephan, F. Euchner, and S. T. Brink, "Angle-delay profile-based and timestamp-aided dissimilarity metrics for channel charting," *IEEE Trans. Commun.*, vol. 72, no. 9, pp. 5611–5625, Sep. 2024.

[36] V. Palhares, S. Taner, and C. Studer, "CSI2Vec: Towards a universal CSI feature representation for positioning and channel charting," 2025, *arXiv:2506.05237*.
[37] Y. Zhang et al., "UNILocPro: Unified localization integrating model-based geometry and channel charting," 2025, *arXiv:2510.27394*.
[38] H. Huang et al., "A retrieval-assisted framework for wireless localization," 2026, *arXiv:2603.06158*.
[39] Y. Han, Z. Li, Z. Zhao, and T. Braun, "CrowdBERT: Crowdsourcing indoor positioning via semi-supervised BERT with masking," *IEEE Internet Things J.*, vol. 11, no. 24, pp. 40100–40112, Dec. 2024.
[40] J. Wang, W. Fang, J. Xiao, Y. Zheng, L. Zheng, and F. Liu, "Signal-guided masked autoencoder for wireless positioning with limited labeled samples," *IEEE Trans. Veh. Technol.*, vol. 74, no. 1, pp. 1759–1764, Jan. 2025.
[41] J. Ott, J. Pirkl, M. Stahlke, T. Feigl, and C. Mutschler, "Radio foundation models: Pre-training transformers for 5G-based indoor localization," in *Proc. 14th Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2024, pp. 1–6.
[42] X. Cheng, B. Liu, X. Liu, and X. Cai, "Large wireless foundation models: Stronger over bigger," 2026, *arXiv:2601.10963*.
[43] R. Shwartz Ziv and Y. LeCun, "To compress or not to compress—Self-supervised learning and information theory: A review," *Entropy*, vol. 26, no. 3, p. 252, Mar. 2024.
[44] S. Hu, Z. Lou, X. Yan, and Y. Ye, "A survey on information bottleneck," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5325–5344, Aug. 2024.
[45] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, Jul. 2020, pp. 1597–1607.
[46] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.
[47] L. Yuan et al., "Tokens-to-token ViT: Training vision transformers from scratch on ImageNet," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2021, pp. 558–567.
[48] A. Vaswani et al., "Attention is all you need," in *Proc. NIPS*, vol. 30, 2017, pp. 1–11.
[49] A. Alkhateeb, "DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications," in *Proc. ITA*, San Diego, CA, USA, Feb. 2019, pp. 1–8.
[50] *Sionna*. Accessed: Jul. 31, 2026. [Online]. Available: https://nvlabs.github.io/sionna/

Guangjin Pan (Member, IEEE) received the B.E. and Ph.D. degrees from the School of Communication and Information Engineering, Shanghai University, Shanghai, China, in 2018 and 2024, respectively. He is currently a Post-Doctoral Researcher with the Chalmers University of Technology, Gothenburg, Sweden. He has achieved outstanding results in more than ten competitions related to wireless AI, including two first-place awards. His research interests include radio localization and sensing, goal-oriented semantic communication, and AI-native wireless networks.

Kaixuan Huang (Graduate Student Member, IEEE) received the B.E. and M.S. degrees from the School of Communication and Information Engineering, Shanghai University, in 2020 and 2023, respectively, where he is currently pursuing the Ph.D. degree. His research interests include indoor localization, navigation, advanced signal processing, deep learning, and integrated sensing and communication (ISAC).



Hui Chen (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the King Abdullah University of Science and Technology, Saudi Arabia, in 2021. He is currently an Assistant Professor at Uppsala University, Sweden, and also affiliated with the Chalmers University of Technology, Sweden, as a Staff Scientist, and University College London, U.K., as a Marie Skłodowska-Curie Fellow. His research focuses on 5G/6G localization and sensing.

Shunqing Zhang (Senior Member, IEEE) received the B.S. degree from the Department of Microelectronics, Fudan University, Shanghai, China, in 2005, and the Ph.D. degree from the Department of Electrical and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, in 2009. Since 2017, he has been with the School of Communication and Information Engineering, Shanghai University, Shanghai, China, as a Full Professor.

Christian Häger (Member, IEEE) received the Dipl.-Ing. (M.Sc. equivalent) degree from Ulm University, Germany, in 2011, and the Ph.D. degree from the Chalmers University of Technology, Sweden, in 2016. He was a Post-Doctoral Researcher with the Department of Electrical and Computer Engineering, Duke University, USA, and the Department of Electrical Engineering, Chalmers University of Technology, where he is currently an Associate Professor with the Department of Electrical Engineering. His research interests include the intersection of communication systems, machine learning, and signal processing. He received the Marie Skłodowska-Curie Global Fellowship from European Commission in 2017 and the Starting Grant from Swedish Research Council in 2020.

Henk Wymeersch (Fellow, IEEE) received the Ph.D. degree in electrical engineering/applied sciences from Ghent University, Belgium, in 2005. He is currently a Professor of communication systems with the Department of Electrical Engineering, Chalmers University of Technology, Sweden, and a Distinguished Visiting Professor at Tsinghua University. Prior to joining the Chalmers University of Technology, he was a Post-Doctoral Researcher with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, from 2005 to 2009. His current research interests include the convergence of communication and sensing in a 5G and beyond 5G context. From 2019 to 2021, he was an IEEE Distinguished Lecturer with the Vehicular Technology Society. He is a Senior Editorial Board Member of *IEEE Signal Processing Magazine*. He served as an Associate Editor for IEEE COMMUNICATIONS LETTERS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and IEEE TRANSACTIONS ON COMMUNICATIONS.