GLCR: GLOBAL-LOCAL COMPLEMENTARY REPRESENTATIONS FOR WI-FI PROBE REQUEST ASSOCIATION UNDER MAC RANDOMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Passive Wi-Fi sensing provides a promising way to support crowd monitoring by counting nearby devices when cameras are limited by occlusion or poor visibility. However, MAC address randomization breaks the connection between Wi-Fi probe requests and devices, which makes device counting unreliable. Existing methods use fingerprints derived from Information Elements (IEs) or rely on wireless measurements and frame metadata that can vary with deployment conditions. To jointly learn complementary representations of IE order and byte-level content without relying on such auxiliary information, we propose Global-Local Complementary Representation (GLCR), a self-supervised learning framework for Wi-Fi probe request association under MAC address randomization. GLCR learns representations from retained IE content and structure. Specifically, we construct a global IE sequence view that preserves IE order, and a local IE subsequence view that focuses on byte-level IE content. For the global view, we devise a hierarchical protocol encoder to capture dependencies at the byte and IE levels. For the local view, we perform salience fingerprint mining on the local subsequence to aggregate contextualized features using learned weights. We further adopt a context-gated refinement to control the local contribution to the global representation. Finally, we cluster the resulting embeddings without a predefined device count. Experiments on three public datasets and our WiDAR dataset show that GLCR outperforms all evaluated baselines, with relative gains of 3.25%–7.86% in clustering accuracy. The code is available at https://anonymous.4open.science/r/GLCR.

1 INTRODUCTION

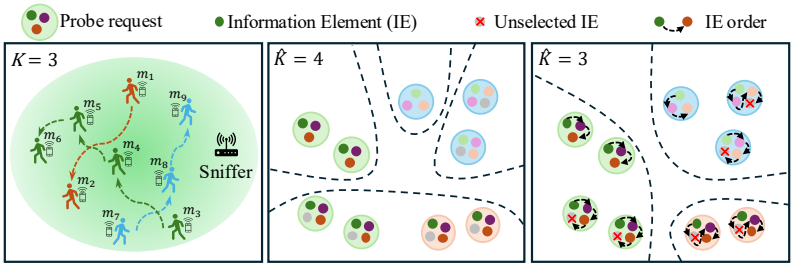
Wi-Fi sensing provides contactless observations of human presence and activity Wang et al. (2026). Applications include indoor occupancy and movement monitoring Depatla et al. (2015); Rösler et al. (2024) and privacy-preserving crowd estimation using multiple Wi-Fi sensors Torkamandi et al. (2024). In crowded places, cameras may not observe all individuals because people can block one another, or they are not in the coverage of the cameras Wang et al. (2025). Passive Wi-Fi sensing instead offers a low-cost, easily deployable method of obtaining non-visual, device-level observations Xi et al. (2014); Hong et al. (2018); Schauer et al. (2014).

A common observation is the probe request, an IEEE 802.11 management frame that mobile devices broadcast while discovering nearby access points Nozari et al. (2025a). Smartphones accounted for 80% of mobile connections worldwide in 2024, a share projected to reach 91% by 2030, motivating the use of Wi-Fi probe requests for crowd analytics GSMA (2025).

Modern mobile operating systems, such as iOS and Android, however randomize the source MAC addresses in probe requests to protect user privacy Fenske et al. (2021); Zúñiga et al. (2025). Consequently, probe request frames from the same device may carry different source MAC addresses. If each observed address is considered as a separate device, the resulting count significantly overestimates the number of devices Vanhoef et al. (2016). Figure 1(a) illustrates the task of associating probe requests from the same device with different randomized source MAC addresses.

A probe request frame contains an ordered sequence of Information Elements (IEs). Existing approaches use IE-derived signatures, selected bytes, or structural templates for probe-request asso-

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



(a) Probe request association (b) Aggregate features (c) GLCR (Ours)

Figure 1: Schematic illustration of Wi-Fi probe request association under MAC address randomization and the motivation for GLCR. **(a)** A passive sniffer captures 9 probe requests from 3 devices ($K = 3$). Each marker represents one probe request labeled by its randomized source MAC address m_j; colored trajectory indicates which requests originate from the same device. The different address labels illustrate address changes. **(b)** An illustrative clustering outcome based on aggregate IE features partitions the 9 requests into 4 clusters ($\hat{K} = 4$); it does not represent a particular prior method.**(c)** The GLCR schematic illustrates the use of IE order and selected byte-level content, with the intended grouping of the 9 requests into 3 device clusters ($\hat{K} = 3$).

ciation Uras et al. (2020); Pintor & Atzori (2022a); Baccichet et al. (2025); Vogel et al. (2026). Figure 1(b) illustrates an aggregate-feature clustering example. Other approaches Uras et al. (2022); Mishra et al. (2024); Tan & Chan (2021); He et al. (2023) incorporate wireless measurements and frame metadata, including RSSI, arrival times, sequence numbers, and channel information. These cues can vary with device state and deployment conditions; for example, indoor propagation affects RSSI measurements Kaemarungsi & Krishnamurthy (2004). We study joint representation learning from ordered IE sequences and selected byte-level content without these auxiliary cues.

Our design uses IE order and byte-level content as complementary inputs for device association: the former captures sequence context, while the latter retains fine-grained variation within IEs. To learn from both, we propose Global-Local Complementary Representation (GLCR), an asymmetric self-supervised framework for Wi-Fi probe request association under MAC address randomization. Specifically, GLCR constructs a global view that preserves IE order and explicit byte-to-IE membership, and a local view that focuses on byte-level content in a selected IE subsequence. A hierarchical encoder explicitly models byte-to-IE membership and cross-IE dependencies in the global view, while a lightweight Transformer contextualizes the selected local sequence and SFM assigns context-dependent weights to its byte tokens. As different features in the local representation may not contribute equally, we use a feature-wise gate to control the local contribution to the global representation. GLCR learns the fused representations without device-identity labels, and the proposed model clusters these without a predefined device count. Figure 1(c) illustrates the intended association of the 9 probe requests with the 3 source devices.

The main contributions of this paper are as follows: **(1)** We develop GLCR, a self-supervised framework that learns complementary global and local representations from IE sequences for probe-request clustering under MAC address randomization. **(2)** We design an asymmetric global-local architecture with complementary IE views. The global view preserves the full ordered IE sequence and byte-to-IE membership, while the local view uses an adaptively selected IE subsequence. Within this architecture, the hierarchical protocol encoder models byte-level and cross-IE dependencies in the global view; the salience fingerprint mining module aggregates contextualized tokens in the local view; and the context-gated refinement module computes a feature-wise gate to control the local contribution to the global representation. **(3)** Experiments on three public datasets and our newly constructed WiDAR dataset show that GLCR outperforms the strongest evaluated baselines, with relative gains of 3.25%–7.86% in clustering accuracy.

2 Related Work

Probe Request Association. Early deployments demonstrated that passively observed Wi-Fi traffic can support smartphone tracking and visitor monitoring at mass events Musa & Eriksson (2012); Bonné et al. (2013). Content-based association methods construct signatures from Element IDs, IE lengths and values, selected bits, or structural templates Uras et al. (2020); Pintor & Atzori (2022a);

Baccichet et al. (2025); Vogel et al. (2026). Prior fingerprints have already combined IE order and field values Vanhoef et al. (2016). Other approaches incorporate sequence numbers, timing, RSSI, or multi-channel probing patterns Uras et al. (2022); Pérez-Hernández et al. (2024); Baccichet et al. (2024); Mishra et al. (2024); Matte et al. (2016). These cues provide additional association evidence but may depend on device behavior or deployment conditions. Espresso and Cappuccino combine such cues to estimate pairwise frame correlations and use minimum-cost flow for multi-frame association Tan & Chan (2021); He et al. (2023); Cappuccino forms self-supervised training pairs from frames carrying globally administered MAC addresses He et al. (2023). Using only retained IE content and structure, GLCR combines hierarchical IE encoding, salience-weighted local aggregation, and context-gated fusion in an asymmetric representation-learning architecture, followed by clustering without a predefined device count.

Self-Supervised Learning. Self-supervised methods use reconstruction Vincent et al. (2008); He et al. (2022), contrastive learning Chen et al. (2020); He et al. (2020), non-contrastive learning or self-distillation without explicit negative pairs Grill et al. (2020); Chen & He (2021); Caron et al. (2021), and redundancy or variance regularization to prevent collapse Zbontar et al. (2021); Bardes et al. (2022). GLCR combines three reconstruction losses with IE compactness and variance regularization to learn probe request representations without device-identity labels.

3 Problem Statement

A passive sniffer collects Wi-Fi probe request frames during an observation window T. Each frame is one input instance represented by its ordered sequence of Information Elements (IEs). Under MAC address randomization, probe request frames from the same device may carry different source MAC addresses. Therefore, source-address matching alone is insufficient for device association.

Let $\mathcal{X} = \{X_i\}_{i=1}^{N}$ denote the N captured frames broadcast by K devices. Each frame X_i originates from exactly one of the K devices, but its device label and K are unknown during representation learning and clustering. We formulate probe request association as clustering, where frames assigned to the same non-noise cluster are inferred to originate from the same device. Given $\mathcal{X}$, the task is to assign each frame either to noise or to one of $\hat{K}$ inferred device clusters, where $\hat{K}$ is determined by the clustering algorithm rather than specified in advance. The goal is to assign frames from the same device to the same cluster and frames from different devices to different clusters.

4 Methodology

Overview. Given an unlabeled frame collection $\mathcal{X}$, GLCR maps the ordered IE sequence of each frame X_i to a fused embedding $\mathbf{z}_c$. As shown in Figure 2, **(a)** *Global-Local View Construction (GLVC)* forms a global view from the full preprocessed byte sequence $\mathbf{S}_g$ in original IE order and its aligned byte-to-IE assignments $\mathbf{C}$, and a local view from the byte sequence $\mathbf{S}_l$ of selected IEs. *Iterative Adaptive Pruning (IAP)* selects Element IDs for $\mathbf{S}_l$ based on the number of distinct Value-field lengths per ID across $\mathcal{X}$. Asymmetric branches encode the two views. In the global branch, **(b)** *Hierarchical Protocol Encoder (HPE)* uses *IE Hierarchical Aggregation (IEHA)* to aggregate byte features by Element ID and model cross-IE dependencies, producing $\mathbf{z}_g$. In the local branch, a lightweight Transformer Vaswani et al. (2017) contextualizes the tokens in $\mathbf{S}_l$, and **(c)** *Salience Fingerprint Mining (SFM)* aggregates them with learned weights into $\mathbf{z}_l$. **(d)** *Context-Gated Refinement (CGR)* computes a feature-wise gate from both embeddings to control the contribution of $\mathbf{z}_l$ to $\mathbf{z}_g$, producing $\mathbf{z}_c$. Learned uncertainty weights balance three reconstruction losses, an IE compactness loss, and variance regularization. At inference, the clustering algorithm assigns each $\mathbf{z}_c$ to a device cluster or noise without a predefined device count. The encoder is trained only on the unlabeled WiDAR training set and remains frozen during evaluation. For each target collection, IAP selection and DBSCAN parameter selection use only its unlabeled data.

4.1 Global-Local View Construction

As shown in Figure 3, we retain probe request frames with locally administered unicast source MAC addresses, identified by $b_1 = 1$ and $b_0 = 0$. To exclude potentially privacy-sensitive Service Set Identifier (SSID) values and channel-dependent Distribution System (DS) Parameter Set values, we remove their Value bytes and set their Length fields to zero while retaining their Element IDs

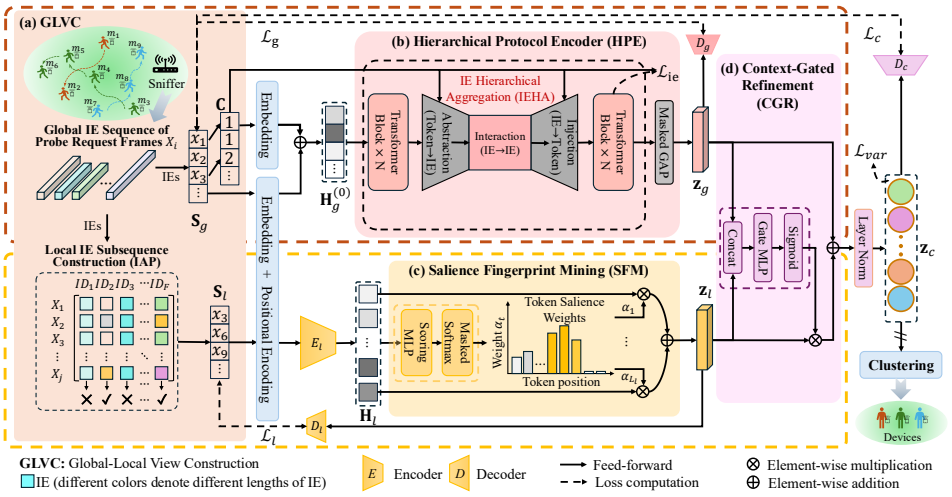


Figure 2: Overview of GLCR. **(a) GLVC** constructs a global IE sequence view and a local IE subsequence view from each probe request frame X_i; IAP selects the Element IDs used to form the local view according to the number of distinct Value-field lengths observed across the frame collection. The global view is represented by $(\mathbf{S}_g, \mathbf{C})$, and the local view by $\mathbf{S}_l$. **(b) HPE** models byte-level and cross-IE dependencies in the global view to produce $\mathbf{z}_g$. A lightweight Transformer contextualizes the local view, and **(c) SFM** aggregates its tokens into $\mathbf{z}_l$ using learned weights. **(d) CGR** controls the local contribution to the global representation through feature-wise gating, producing $\mathbf{z}_c$. The fused embeddings are subsequently clustered without a predefined device count. [Best viewed in color]

and positions in the IE sequence. Each IE is serialized in TLV format, and the IEs are concatenated in their original order.

For each preprocessed frame X_i, GLVC constructs a global view containing the complete serialized IE sequence and a local view containing the IEs whose Element IDs are selected by IAP, in their original relative order. IAP performs this selection based on the number of distinct Value-field lengths observed for each Element ID across the frame collection. The global view is represented by the byte sequence $\mathbf{S}_g$ and its aligned byte-to-IE assignments $\mathbf{C}$, whereas the local view is represented by the byte sequence $\mathbf{S}_l$.

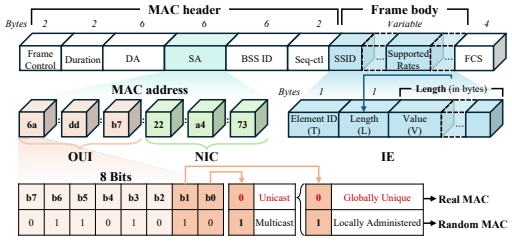


Figure 3: Structure of a Wi-Fi probe request frame and the source MAC address bits used for filtering. The source MAC address (SA) is expanded to show b_0 and b_1 in its first octet. An SA with $b_0 = 0$ and $b_1 = 1$ is a locally administered unicast address, whereas an SA with $b_0 = 0$ and $b_1 = 0$ is a universally administered unicast address, which is typically factory assigned. Each IE in the frame body is encoded in Type-Length-Value (TLV) format.

Global IE Sequence View. The global view retains all preprocessed IEs in their original order. After TLV serialization and batch padding to a common length L, let $\mathbf{S}_g = (x_t)_{t=1}^{L}$ denote the byte sequence, $\mathbf{C} = (c_t)_{t=1}^{L}$ the aligned byte-to-IE assignments, and $\mathbf{M} = (m_t)_{t=1}^{L}$ the padding mask. At each valid position, x_t is the serialized byte and c_t is the Element ID of the IE containing that byte. All bytes belonging to the same IE, including its Element ID, Length, and Value bytes, are assigned the same c_t. We set $m_t = 0$ for valid bytes and $m_t = 1$ for padding. Each valid byte token is embedded as $\mathbf{h}_{g,t}^{(0)} = \mathrm{Emb}_{\mathrm{byte}}(x_t) + \mathrm{Emb}_{\mathrm{ie}}(c_t) + \mathbf{p}_t$, where $\mathrm{Emb}_{\mathrm{byte}}$ and $\mathrm{Emb}_{\mathrm{ie}}$ denote the byte and Element ID embeddings, respectively, and $\mathbf{p}_t$ is the absolute positional encoding at byte position t. The stacked embeddings $\mathbf{H}_g^{(0)} = [\mathbf{h}_{g,1}^{(0)}, \ldots, \mathbf{h}_{g,L}^{(0)}]$, the assignments $\mathbf{C}$, and the mask $\mathbf{M}$ are supplied to HPE.

Local IE Subsequence View. For the current unlabeled frame collection $\mathcal{X} = \{X_i\}_{i=1}^{N}$, the local view retains TLVs whose Element IDs are selected based on collection-level Value-field length

statistics. For each Element ID k, let $\mathcal{L}_k$ denote the set of Value-field lengths observed across $\mathcal{X}$, with absence encoded as zero. We define its length cardinality as $C(k) = |\mathcal{L}_k|$.

Inspired by the iterative IE-length screening of Uras et al. Uras et al. (2020), *Iterative Adaptive Pruning (IAP)* progressively raises the minimum length cardinality required for an Element ID to remain a candidate. It selects the reduced candidate set at the first step where the decrease in candidate-set size exceeds an adaptive margin. For an integer threshold $\tau \geq 1$, define $\mathcal{Q}_\tau = \{k \mid C(k) > \tau\}$, $\Delta_\tau = |\mathcal{Q}_\tau| - |\mathcal{Q}_{\tau+1}|$, and $\beta_\tau = \max(1, \lfloor \beta_0 - \delta(\tau - 1) \rfloor)$. If $\mathcal{Q}_1 = \emptyset$, IAP returns $\mathcal{K}_{\text{local}} = \emptyset$. Otherwise, starting from $\tau = 1$, IAP returns $\mathcal{K}_{\text{local}} = \mathcal{Q}_\tau$ if $\mathcal{Q}_{\tau+1}$ is empty. If $\mathcal{Q}_{\tau+1}$ is nonempty and $\Delta_\tau > \beta_\tau$, it returns $\mathcal{K}_{\text{local}} = \mathcal{Q}_{\tau+1}$; otherwise, it increments τ and continues. We use $\beta_0 = 3$ and $\delta = 1$ in all experiments. Detailed pseudocode is provided in the supplementary material.

IAP is applied once to the training collection and once separately to each target collection using only the corresponding unlabeled frames. The selected Element IDs remain fixed while that collection is encoded, and the model parameters are not updated during target-collection selection. For each frame, the TLVs whose Element IDs belong to $\mathcal{K}_{\text{local}}$ are concatenated in their original relative order to form the local subsequence. If $\mathcal{K}_{\text{local}} = \emptyset$ or a frame contains none of the selected Element IDs, its global TLV sequence is used as the local input.

After batch padding to a common length L_l, let $\mathbf{S}_l = (x_t^l)_{t=1}^{L_l}$ denote the local byte sequence and $\mathbf{M}^l = (m_t^l)_{t=1}^{L_l}$ its padding mask, where $m_t^l = 0$ for valid bytes and $m_t^l = 1$ for padding. Each valid byte token is embedded as $\mathbf{h}_{l,t}^{(0)} = \text{Emb}_{\text{byte}}(x_t^l) + \mathbf{p}_t^l$, where $\mathbf{p}_t^l$ is the absolute positional encoding at position t in the local subsequence. The stacked embeddings $\mathbf{H}_l^{(0)} = [\mathbf{h}_{l,1}^{(0)}, \ldots, \mathbf{h}_{l,L_l}^{(0)}]$ and mask $\mathbf{M}^l$ are supplied to the lightweight Transformer encoder before SFM.

4.2 Hierarchical Protocol Encoder

HPE takes the global input embeddings $\mathbf{H}_g^{(0)}$, the byte-to-IE assignments $\mathbf{C}$, and the padding mask $\mathbf{M}$. Although self-attention models interactions between byte tokens, it does not explicitly aggregate them according to IE membership. HPE therefore inserts *IE Hierarchical Aggregation (IEHA)* between the second and third Transformer blocks. As shown in Figure 4, IEHA abstracts byte tokens into Element-ID group representations, models interactions among these groups, and injects the resulting context back into the token sequence.

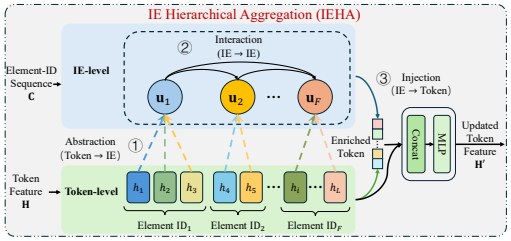


Figure 4: IEHA aggregates byte tokens by Element ID, models interactions among the resulting Element-ID group representations, and injects the refined IE context back into the corresponding token positions.

The first two Transformer blocks produce $\mathbf{H}_g = \mathcal{T}_{\text{pre}}(\mathbf{H}_g^{(0)}, \mathbf{M})$, where $\mathbf{H}_g = (\mathbf{h}_{g,t})_{t=1}^{L}$.

(1) Token-to-IE abstraction. Let $\mathcal{I} = \{t \mid m_t = 0\}$ be the valid byte positions, $\mathcal{F} = \{c_t \mid t \in \mathcal{I}\}$ the observed Element IDs, and $\mathcal{S}_k = \{t \in \mathcal{I} \mid c_t = k\}$. For each $k \in \mathcal{F}$, IEHA computes the centroid $\bar{\mathbf{u}}_k = |\mathcal{S}_k|^{-1} \sum_{t \in \mathcal{S}_k} \mathbf{h}_{g,t}$.

IEHA groups tokens by Element ID, not by IE occurrence. Thus, repeated IEs with the same Element ID within a frame share one group-level representation. Their tokens remain at separate positions in the serialized byte sequence and are contextualized using positional encodings before aggregation. However, these occurrences are not represented as separate IE-level nodes.

(2) Cross-IE interaction. Each centroid is projected as $\mathbf{u}_k = \mathbf{W}_{\text{ie}} \bar{\mathbf{u}}_k + \mathbf{b}_{\text{ie}}$. The projected centroids form a matrix $\mathbf{U}$ whose rows are indexed by Element ID; rows corresponding to absent Element IDs are zero-filled and masked by $\mathbf{M}_{\text{ie}}$. A lightweight Transformer computes $\widetilde{\mathbf{U}} = \mathcal{T}_{\text{ie}}(\mathbf{U}, \mathbf{M}_{\text{ie}})$. Each row $\widetilde{\mathbf{u}}_k$ therefore incorporates context from the other IEs observed in the frame.

(3) IE-to-token injection. For each valid position t, an MLP projection injects the corresponding IE context as $\bar{\mathbf{h}}_{g,t} = \phi([\mathbf{h}_{g,t} \parallel \widetilde{\mathbf{u}}_{c_t}])$, where $\phi : \mathbb{R}^{2d} \to \mathbb{R}^d$ and $[\cdot \parallel \cdot]$ denotes concatenation. Let $\bar{\mathbf{H}}_g$ denote the length-L padded sequence of updated valid-token representations. The final Transformer block produces $\mathbf{H}'_g = \mathcal{T}_{\text{post}}(\bar{\mathbf{H}}_g, \mathbf{M}) \in \mathbb{R}^{L \times d}$, where $\mathbf{M}$ masks the padded positions.

We regularize the final token representations with an IE compactness loss. For each observed Element-ID group (i,k) in mini-batch $\mathcal{B}$, let $\mathcal{S}_{i,k}$ contain its token positions and define $\mathbf{u}'_{i,k} = |\mathcal{S}_{i,k}|^{-1} \sum_{t\in\mathcal{S}_{i,k}} \mathbf{h}'_{g,i,t}$. Let $\mathcal{G}$ contain the groups with $|\mathcal{S}_{i,k}| > 1$. The compactness loss is

$$\mathcal{L}_{\text{ie}} = \frac{1}{|\mathcal{G}|} \sum_{(i,k)\in\mathcal{G}} \frac{1}{d|\mathcal{S}_{i,k}|} \sum_{t\in\mathcal{S}_{i,k}} \left\|\mathbf{h}'_{g,i,t} - \mathbf{u}'_{i,k}\right\|_2^2. \tag{1}$$

This loss reduces the dispersion of token representations assigned to the same Element ID without using device-identity labels.

Finally, padding-masked mean pooling produces $\mathbf{z}_g = L_v^{-1} \sum_{t=1}^{L} (1-m_t)\mathbf{h}'_{g,t} \in \mathbb{R}^d$, where $L_v = \sum_{t=1}^{L}(1-m_t)$.

4.3 Salience Fingerprint Mining

IAP selects the Element IDs used in the local view from collection-level Value-field length statistics, but it does not distinguish the contributions of individual tokens within each frame. SFM therefore learns frame-specific weights over the contextualized local tokens.

The local Transformer encoder E_l produces $\mathbf{H}_l = E_l(\mathbf{H}_l^{(0)}, \mathbf{M}^l) = (\mathbf{h}_{l,t})_{t=1}^{L_l} \in \mathbb{R}^{L_l\times d}$. For each token position t, SFM computes the salience score $e_t = \mathbf{w}_2^\top \tanh(\mathbf{W}_1\mathbf{h}_{l,t} + \mathbf{b}_1) + b_2$, where $\mathbf{W}_1 \in \mathbb{R}^{\frac{d}{2}\times d}$, $\mathbf{w}_2, \mathbf{b}_1 \in \mathbb{R}^{\frac{d}{2}}$, and $b_2 \in \mathbb{R}$ are learnable parameters. Because $\mathbf{h}_{l,t}$ is contextualized, e_t depends on the byte value, its position, and the surrounding local content.

Masked softmax and weighted aggregation produce the local embedding:

$$\alpha_t = \frac{\exp(e_t)(1-m_t^l)}{\sum_{j=1}^{L_l}\exp(e_j)(1-m_j^l)}, \qquad \mathbf{z}_l = \sum_{t=1}^{L_l} \alpha_t \mathbf{h}_{l,t}. \tag{2}$$

Here, α_t is the normalized salience weight of token t. Padded positions receive zero weight, and the weights over valid positions sum to one.

4.4 Context-Gated Refinement

Directly adding $\mathbf{z}_g$ and $\mathbf{z}_l$ assigns a fixed coefficient of one to every dimension of the local embedding. Because local information may not be equally useful across feature dimensions, CGR computes a feature-wise gate $\mathbf{g} \in (0,1)^d$ from both embeddings:

$$\mathbf{g} = \sigma(\mathbf{W}_2^c \tanh(\mathbf{W}_1^c[\mathbf{z}_g \parallel \mathbf{z}_l] + \mathbf{b}_1^c) + \mathbf{b}_2^c). \tag{3}$$

Here, $[\cdot \parallel \cdot]$ denotes concatenation, σ is the sigmoid function, and $\mathbf{W}_1^c \in \mathbb{R}^{d_h\times 2d}$, $\mathbf{W}_2^c \in \mathbb{R}^{d\times d_h}$, $\mathbf{b}_1^c \in \mathbb{R}^{d_h}$, and $\mathbf{b}_2^c \in \mathbb{R}^d$ are learnable parameters.

Using element-wise multiplication $\odot$, the gated local contribution and the fused embedding are $\widetilde{\mathbf{z}}_l = \mathbf{g} \odot \mathbf{z}_l$ and $\mathbf{z}_c = \text{LayerNorm}(\mathbf{z}_g + \widetilde{\mathbf{z}}_l)$, respectively. This residual formulation provides an ungated path for the global embedding, while the gate controls the contribution of the local embedding.

Training Objectives. For $b \in \{g,l,c\}$, let $\mathbf{S}_b = (x_{b,t})_{t=1}^{T_b}$ and $\mathbf{M}_b = (m_{b,t})_{t=1}^{T_b}$ denote the reconstruction target and its padding mask. We use $(\mathbf{S}_g, \mathbf{M})$ as the target for the global and fused embeddings, and $(\mathbf{S}_l, \mathbf{M}^l)$ for the local embedding. The padding-aware reconstruction loss is

$$\mathcal{L}_b = \frac{\sum_{t=1}^{T_b}(1-m_{b,t})\ell_{\text{ce}}\left(\mathrm{D}_b(\mathbf{z}_b, t), x_{b,t}\right)}{\sum_{t=1}^{T_b}(1-m_{b,t})}. \tag{4}$$

Here, $\mathrm{D}_b(\mathbf{z}_b, t) \in \mathbb{R}^{|\mathcal{V}|}$ denotes the logits over the byte vocabulary $\mathcal{V}$ at position t, and ℓ_{ce} denotes cross-entropy. The three decoders have the same architecture but independent parameters and are discarded after training. Using the global sequence as the fused target encourages $\mathbf{z}_c$ to retain global content after local refinement.

In addition to the IE compactness loss $\mathcal{L}_{\text{ie}}$ in Eq. 1, we apply variance regularization to discourage collapse of the fused embeddings. For a mini-batch of size B, let $\mathbf{Z}_c = [\mathbf{z}_c^{(1)}, \ldots, \mathbf{z}_c^{(B)}]^\top \in \mathbb{R}^{B\times d}$. Following VICReg Bardes et al. (2022), we define

$$\mathcal{L}_{\text{var}} = \frac{1}{d}\sum_{j=1}^{d} \max\left(0, \gamma - \sqrt{\text{Var}(\mathbf{Z}_c^{(:,j)}) + \epsilon}\right), \tag{5}$$

where the variance is computed across the mini-batch, γ is the target standard deviation, and ϵ ensures numerical stability.

Because the five losses have different numerical scales, we combine them using a bounded variant of homoscedastic uncertainty weighting Kendall et al. (2018). Let $\mathcal{J} = \{g, l, c, \text{ie}, \text{var}\}$ index the losses. For each $r \in \mathcal{J}$, let s_r be a learnable log-variance parameter and define $\bar{s}_r = \min(s_{\max}, \max(s_{\min}, s_r))$. The joint objective is

$$\mathcal{L}_{\text{total}} = \sum_{r\in\mathcal{J}} \left[\frac{1}{2}\exp(-\bar{s}_r)\mathcal{L}_r + \frac{1}{2}\bar{s}_r\right]. \tag{6}$$

We use $\gamma = 1$, $\epsilon = 10^{-4}$, $s_{\min} = -2$, and $s_{\max} = 5$. The network parameters and $\{s_r\}_{r\in\mathcal{J}}$ are optimized jointly.

Clustering. At inference, DBSCAN Ester et al. (1996) clusters $\{\mathbf{z}_c^{(i)}\}_{i=1}^N$ without taking the device count as input. Its parameters $(\varepsilon, \text{minPts})$ are selected without device-identity labels using the internal-index ranking described in *Implementation Details*. Frame embeddings that do not satisfy the density criterion are treated as noise, which we denote by label 0. The remaining clusters define the inferred device groups.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate GLCR on four datasets: three public datasets (**WPR22** Pintor & Atzori (2022b), **WPR24** Pintor et al. (2025), and **NEST** Rusca et al. (2023)) and our **WiDAR** dataset.

Table 1: Dataset statistics.

Dataset	Probe request frames	Modes	Ground truth	Number of devices
WPR22	69,701	4	✓	22
WPR24	26,314	4	✓	18
NEST	4,111	-	✓	14
WiDAR	126,506	-	✓	50

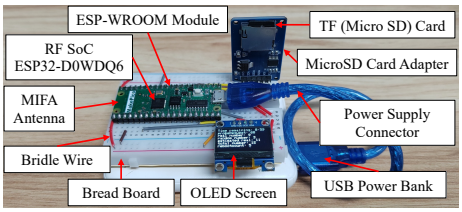


Figure 5: Wi-Fi sniffing device.

We constructed WiDAR to complement the public datasets with a controlled, device-disjoint evaluation on a larger and more diverse device set. Table 1 summarizes their statistics. We collected WiDAR in controlled environments using a custom passive Wi-Fi capture platform (Figure 5). The sniffer hopped among channels in the 2.4 GHz band with a dwell time of 100 ms per channel and recorded probe request frames. WiDAR contains 126,506 probe request frames from 50 devices across eight vendors, including smartphones and personal computers. We split WiDAR by device into disjoint training and test sets, each containing 25 devices. Ground-truth device labels were used only to construct this split and compute evaluation metrics, not for representation learning or DBSCAN parameter selection. Probe requests are management frames and contain no upper-layer traffic payloads. GLCR uses only the retained IE content and structure as model inputs.

Baselines. We compare GLCR with six baselines grouped by their primary mechanisms: (i) IE-based clustering: IE-Clustering Uras et al. (2020) and IE-RF Pintor & Atzori (2022a); (ii) multi-cue association: CrowdWatch Nozari et al. (2025b), TC-Hybrid Uras et al. (2022), and TSC Baccichet et al. (2024); and (iii) deep pairwise association: Cappuccino He et al. (2023).

Implementation Details. We implement GLCR in PyTorch 2.9.0 and run all experiments on a single NVIDIA RTX PRO 6000 GPU. We train GLCR with AdamW Loshchilov & Hutter (2019) using a learning rate of 1×10^{-3} and weight decay of 1×10^{-4}. GLCR has 28.66 K parameters, 16.74 M FLOPs, and 11.53 ms single-sample inference latency. We select DBSCAN parameters without device-identity labels from a finite grid with $\varepsilon \in [0.1, 2.0]$ and $\text{minPts} \in \{2, \ldots, 10\}$. We choose the setting with the lowest rank sum across the Silhouette Rousseeuw (1987) and Calinski–Harabasz Caliński & Harabasz (1974) scores, for which higher values are preferred, and the Davies–

Bouldin score Davies & Bouldin (1979), for which lower values are preferred. Baselines use recommended settings; those using DBSCAN follow the same label-free selection procedure.

5.2 Experimental Results

Task Settings. We evaluate all methods on the same probe-request frame collections from WPR22, WPR24, NEST, and the device-disjoint WiDAR test set. We train GLCR only on the unlabeled WiDAR training set and apply the frozen encoder to WPR22, WPR24, NEST, and the WiDAR test set without retraining. GLCR learns frame representations without device-identity labels and clusters them using DBSCAN without a predefined device count. Ground-truth device identities are used only to construct the WiDAR split and compute the evaluation metrics, not to train GLCR or select the DBSCAN parameters. For each evaluated collection, IAP uses only the unlabeled frames to select the Element IDs used to construct the local view, and the selected set remains fixed during encoding and clustering. We evaluate clustering performance using Accuracy (ACC), computed after optimal label matching Kuhn (1955), Normalized Mutual Information (NMI) Vinh et al. (2010), and Adjusted Rand Index (ARI) Hubert & Arabie (1985) as percentages. Unless otherwise specified, these scores are averaged over 10 runs. The GLCR results in the observation-window study instead use the window-level averaging protocol described below.

Table 2: Clustering results (%) on three public datasets and the WiDAR dataset. **Bold** indicates the best results, while underline marks the second-best results.

Methods	WPR22			WPR24			NEST			WiDAR		
	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑
CrowdWatch Nozari et al. (2025b)	60.32	61.03	60.46	50.99	54.02	35.69	66.54	81.23	61.55	48.38	54.61	29.88
IE-Clustering Uras et al. (2020)	70.48	80.10	77.78	72.35	87.73	73.34	79.92	86.59	75.21	71.79	86.14	68.45
TC-Hybrid Uras et al. (2022)	67.55	81.55	63.55	65.29	85.74	64.46	82.48	85.68	77.16	71.36	88.02	72.56
TSC Baccichet et al. (2024)	76.00	79.95	75.18	74.74	90.73	73.47	74.07	86.40	73.15	69.89	89.48	71.50
IE-RF Pintor & Atzori (2022a)	59.41	77.28	43.86	79.22	91.37	83.69	75.92	88.46	71.07	78.74	90.07	79.87
Cappuccino He et al. (2023)	64.72	76.34	58.85	77.06	84.37	72.86	73.22	83.67	70.31	70.74	84.87	67.59
GLCR (Ours)	**81.97**	**86.13**	**80.35**	**83.43**	**93.49**	**87.97**	**86.61**	**92.96**	**84.07**	**81.30**	**92.26**	**83.36**
Relative Improvement	+7.86	+5.62	+3.30	+5.31	+2.32	+5.11	+5.01	+5.09	+8.96	+3.25	+2.43	+4.37

Overall Evaluation. Table 2 compares GLCR with six baselines on three public datasets and WiDAR. GLCR achieves the highest ACC, NMI, and ARI on all four datasets, with relative improvements ranging from 2.32% to 8.96% over the strongest baseline in each dataset–metric setting. ACC, NMI, and ARI exceed 80% on all four datasets. These results show that GLCR can learn effective representations from IE content and structure without using wireless measurements or frame metadata as model inputs.

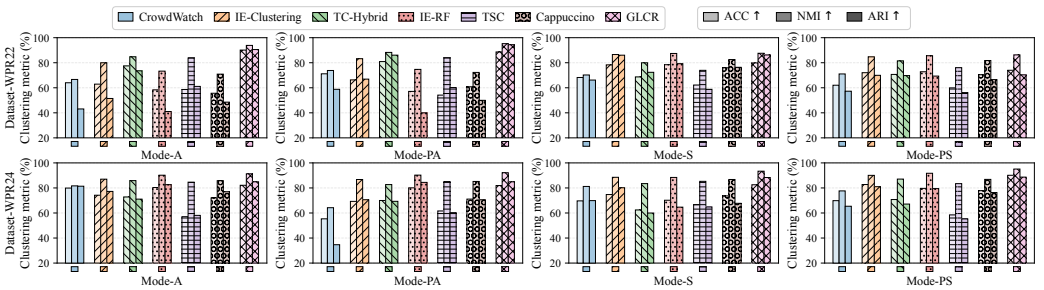


Figure 6: Study of different device operating modes on probe-request association.

Device Mode Study. We evaluate probe-request association under four device operating modes in WPR22 and WPR24. Wi-Fi remains enabled in all modes: screen active with power saving disabled (A), screen active with power saving enabled (PA), standby with power saving disabled (S), and standby with power saving enabled (PS). Figure 6 shows that GLCR achieves the strongest overall performance and maintains comparatively stable ACC, NMI, and ARI across the four modes. Several baselines exhibit larger changes across modes, particularly in ACC and ARI. These results suggest that GLCR is less sensitive to device operating modes on the two evaluated datasets.

Observation-Window T Study. Figure 7 shows how clustering performance on WiDAR changes as the observation window T varies from 0.5 to 60 minutes. For GLCR, we extract ten overlapping

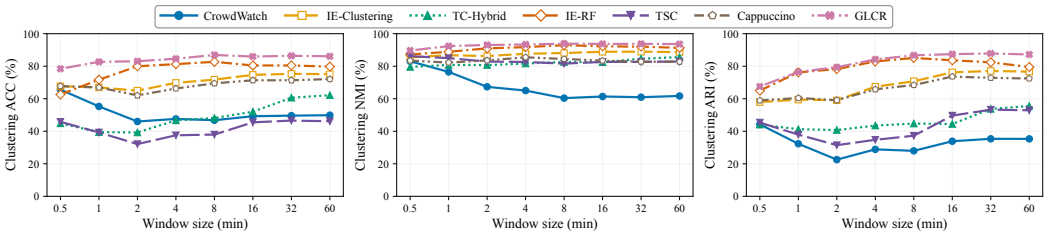


Figure 7: Sensitivity of the time window T on different methods on the dataset WiDAR.

windows of each duration T, starting at the earliest timestamp and advancing by 10 seconds. Using a fixed trained encoder, we cluster each window independently and report the arithmetic mean of the window-level ACC, NMI, and ARI. The evaluated windows include device arrivals and departures and unequal frame counts across devices. GLCR achieves the highest ACC, NMI, and ARI at every evaluated window size. Its performance improves as T increases from 0.5 to 4 minutes and remains comparatively stable for $T \geq 4$ minutes, whereas several baselines exhibit larger changes as T varies. These results show that GLCR maintains strong association performance across the evaluated observation-window lengths without using RSSI or inter-arrival times as model inputs.

Table 3: Component ablation on the proposed method. Clustering results (%) on three public datasets and the WiDAR dataset. **Bold** indicates the best results.

Methods	WPR22			WPR24			NEST			WiDAR		
	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑	ACC ↑	NMI ↑	ARI ↑
w/o IEHA	78.73	84.43	79.74	82.24	92.09	85.31	85.33	91.69	83.66	75.99	90.28	77.83
w/o SFM	78.88	84.62	79.78	79.64	90.97	83.71	86.52	92.40	83.98	81.26	92.22	82.40
w/o CGR	54.67	59.35	56.81	40.81	66.74	32.15	41.14	54.67	27.91	34.60	61.84	24.18
w/o Local branch	76.66	83.22	78.69	66.38	87.31	66.90	69.69	83.50	53.91	81.18	91.95	82.37
w/o Global branch	73.10	81.36	78.87	68.80	87.48	68.85	70.28	82.42	68.13	71.15	87.44	75.56
GLCR	**81.97**	**86.13**	**80.35**	**83.43**	**93.49**	**87.97**	**86.61**	**92.96**	**84.07**	**81.30**	**92.26**	**83.36**

Ablation Study. Table 3 presents the component ablation results on three public datasets and WiDAR. The full model outperforms both single-branch variants (*w/o Local branch* and *w/o Global branch*) on every dataset and metric. On WPR24, removing the global or local branch reduces ACC by 14.63 and 17.05 percentage points, respectively, supporting the complementary roles of full IE-sequence context and selected byte-level IE content. Removing CGR produces the largest decrease in every dataset–metric setting; for example, WPR24 ARI decreases from 87.97% to 32.15%. This result supports the importance of learned cross-view fusion within GLCR. Removing IEHA or SFM also lowers performance for every dataset and metric, supporting the contribution of IEHA to the global branch and SFM to the local branch.

6 Conclusion

We presented GLCR, a self-supervised framework for Wi-Fi probe request association under MAC address randomization. GLCR learns representations without device-identity labels and clusters them without a predefined device count. It combines a global view that preserves IE order and byte-to-IE membership with a local view that focuses on selected byte-level IE content through hierarchical encoding, salience-weighted aggregation, and feature-wise gated refinement. Experiments on three public datasets and WiDAR show that GLCR outperforms all evaluated baselines across ACC, NMI, and ARI and maintains strong performance across the evaluated device modes and observation-window lengths. These results show that IE content and structure can support probe request association without using wireless measurements or frame metadata as model inputs.

AI USE STATEMENT

Generative AI tools were used solely for language polishing and improving the readability of this manuscript.

ETHICAL STATEMENT

This work studies privacy-aware Wi-Fi probe request association for passive crowd monitoring in emergency response and public safety, following responsible AI and privacy-by-design principles while restricting the analysis to anonymous within-collection association: (1) *Data Minimization*: Each retained probe request frame is processed as an individual input instance. The source MAC address is used only to identify locally administered unicast frames and is not included in the model input. Probe requests contain no upper-layer traffic payloads, and the Value fields of the SSID and DS Parameter Set IEs are removed during preprocessing. (2) *Scope Limitation*: Analysis is restricted to anonymous association within each evaluated collection. No cross-session linking, location tracking, or physical identity inference is performed. (3) *Controlled Release*: Data is encrypted offline. Public artifacts exclude raw PCAPs/headers. Only desensitized features are released; quasi-identifiers are secured via salted hashing/HMAC under strict usage policies.

REFERENCES

Giovanni Baccichet, Corrado Innamorati, Alessandro E. C. Redondi, and Matteo Cesana. MAC address de-randomization using multi-channel sniffers and two-stage clustering. In *Proceedings of the IEEE 35th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–6. IEEE, 2024. doi: 10.1109/PIMRC59610.2024.10817186.

Giovanni Baccichet, Fabio Palmese, Alessandro E. C. Redondi, and Matteo Cesana. Compact probe request fingerprinting with asymmetric pairwise boosting. In *2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, pp. 1–6. IEEE, 2025. doi: 10.1109/ICMLCN64995.2025.11140351.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *The Tenth International Conference on Learning Representations*. OpenReview.net, 2022. URL `https://openreview.net/forum?id=xm6YD62D1Ub`.

Bram Bonné, Arno Barzan, Peter Quax, and Wim Lamotte. WiFiPi: Involuntary tracking of visitors at mass events. In *2013 IEEE 14th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–6. IEEE, 2013. doi: 10.1109/WoWMoM.2013.6583443.

Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974. doi: 10.1080/03610927408827101.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9630–9640, 2021. doi: 10.1109/ICCV48922.2021.00951.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pp. 1597–1607, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15745–15753, 2021. doi: 10.1109/CVPR46437.2021.01549.

David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979. doi: 10.1109/TPAMI.1979.4766909.

Saandeep Depatla, Arjun Muralidharan, and Yasamin Mostofi. Occupancy estimation using only WiFi power measurements. *IEEE Journal on Selected Areas in Communications*, 33(7):1381–1393, 2015. doi: 10.1109/JSAC.2015.2430272.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 226–231. AAAI Press, 1996.

Ellis Fenske, Dane Brown, Jeremy Martin, Travis Mayberry, Peter Ryan, and Erik Rye. Three years later: A study of MAC address randomization in mobile devices and when it succeeds. *Proceedings on Privacy Enhancing Technologies*, 2021(3):164–181, 2021. doi: 10.2478/popets-2021-0042.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html.

GSMA. The mobile economy 2025. Technical report, GSM Association, 2025. URL https://www.gsma.com/solutions-and-impact/connectivity-for-good/mobile-economy/the-mobile-economy-2025/. Accessed: 2026-07-19.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020. doi: 10.1109/CVPR42600.2020.00975.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022. doi: 10.1109/CVPR52688.2022.01553.

Tianlang He, Jiajie Tan, and S.-H. Gary Chan. Self-supervised association of Wi-Fi probe requests under MAC address randomization. *IEEE Transactions on Mobile Computing*, 22(12):7044–7056, 2023. doi: 10.1109/TMC.2022.3205924.

Hande Hong, Girisha Durrel De Silva, and Mun Choon Chan. CrowdProbe: Non-invasive crowd monitoring with Wi-Fi probe. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3):1–23, 2018. doi: 10.1145/3264925.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985. doi: 10.1007/BF01908075.

K. Kaemarungsi and P. Krishnamurthy. Properties of indoor received signal strength for WLAN location fingerprinting. In *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pp. 14–23. IEEE, 2004. doi: 10.1109/MOBIQ.2004.1331706.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7482–7491, 2018. doi: 10.1109/CVPR.2018.00781. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Kendall_Multi-Task_Learning_Using_CVPR_2018_paper.html.

H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, 1955. doi: 10.1002/nav.3800020109.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Célestin Matte, Mathieu Cunche, Franck Rousseau, and Mathy Vanhoef. Defeating MAC address randomization through timing attacks. In *Proceedings of the 9th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 15–20. ACM, 2016. doi: 10.1145/2939918.2939930.

Abhishek Kumar Mishra, Aline Carneiro Viana, and Nadjib Achir. Bleach: From Wi-Fi probe-request signatures to MAC association. *Ad Hoc Networks*, 164:103623, 2024. doi: 10.1016/j.adhoc.2024.103623.

A. B. M. Musa and Jakob Eriksson. Tracking unmodified smartphones using Wi-Fi monitors. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pp. 281–294. ACM, 2012. doi: 10.1145/2426656.2426685.

Sheida Nozari, Chiara Garibotto, Andrea Sciarrone, Igor Bisio, and Fabio Lavagetto. Analyzing broadcast patterns and randomization techniques in Wi-Fi probe request frames. *IEEE Network*, 39(4):228–234, 2025a. doi: 10.1109/MNET.2024.3479457.

Sheida Nozari, Halar Haleem, Chiara Garibotto, Andrea Sciarrone, Igor Bisio, and Fabio Lavagetto. Crowdwatch: Privacy-preserving monitoring leveraging WiFi multiple access information. *IEEE Internet of Things Journal*, 12(11):15238–15248, 2025b. doi: 10.1109/JIOT.2025.3528226.

Lucia Pintor and Luigi Atzori. Analysis of Wi-Fi probe requests towards information element fingerprinting. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, pp. 3857–3862. IEEE, 2022a. doi: 10.1109/GLOBECOM48099.2022.10001618.

Lucia Pintor and Luigi Atzori. A dataset of labelled device Wi-Fi probe requests for MAC address de-randomization. *Computer Networks*, 205:108783, 2022b. doi: 10.1016/j.comnet.2022.108783.

Lucia Pintor, Alessio Chirigu, and Luigi Atzori. A dataset of labelled device Wi-Fi probe requests for MAC address de-randomization – 2024. Mendeley Data, Version 2, 2025. Dataset, doi: 10.17632/5tvnwhsj2p.2.

Abraham Pérez-Hernández, Maydelis N. Barreras-Martín, Juan A. Becerra, María J. Madero-Ayora, and Pablo Aguilera. De-randomization of MAC addresses using fingerprints and RSSI with ML for Wi-Fi analytics. *IEEE Access*, 12:150857–150868, 2024. doi: 10.1109/ACCESS.2024.3410549.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. doi: 10.1016/0377-0427(87)90125-7.

Riccardo Rusca, Filippo Sansoldo, Claudio Casetti, and Paolo Giaccone. What WiFi probe requests can tell you. In *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, pp. 1086–1091. IEEE, 2023. doi: 10.1109/CCNC51644.2023.10060447.

Sascha Rösler, Anatolij Zubow, and Falko Dressier. Passive device-free indoor movement detection using WiFi time-of-flight information. In *2024 IEEE 49th Conference on Local Computer Networks (LCN)*, pp. 1–9. IEEE, 2024. doi: 10.1109/LCN60385.2024.10639768.

Lorenz Schauer, Martin Werner, and Philipp Marcus. Estimating crowd densities and pedestrian flows using Wi-Fi and bluetooth. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ICST, 2014. doi: 10.4108/icst.mobiquitous.2014.257870.

Jiajie Tan and S-H Gary Chan. Efficient association of Wi-Fi probe requests under MAC address randomization. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pp. 1–10. IEEE, 2021. doi: 10.1109/INFOCOM42981.2021.9488769.

Pegah Torkamandi, Ljubica Kärkkäinen, and Jörg Ott. Privacy-preserving crowd estimation using multiple Wi-Fi sensors. In *IEEE 21st International Conference on Mobile Ad-Hoc and Smart Systems (MASS)*, pp. 314–320, 2024. doi: 10.1109/MASS62177.2024.00049.

Marco Uras, Raimondo Cossu, Enrico Ferrara, Ovidiu Bagdasar, Antonio Liotta, and Luigi Atzori. WiFi probes sniffing: an artificial intelligence based approach for MAC addresses derandomization. In *2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 1–6. IEEE, 2020. doi: 10.1109/CAMAD50429.2020.9209257.

Marco Uras, Enrico Ferrara, Raimondo Cossu, Antonio Liotta, and Luigi Atzori. MAC address derandomization for WiFi device counting: Combining temporal- and content-based fingerprints. *Computer Networks*, 218:109393, 2022. doi: 10.1016/j.comnet.2022.109393.

Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo S. Cardoso, and Frank Piessens. Why MAC address randomization is not enough: An analysis of Wi-Fi network discovery mechanisms. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pp. 413–424. ACM, 2016. doi: 10.1145/2897845.2897883.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008, 2017.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103, 2008. doi: 10.1145/1390156.1390294.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(95):2837–2854, 2010. URL https://www.jmlr.org/papers/v11/vinh10a.html.

Daniel Vogel, Felix Viola, Nicholas Malte Kreimeyer, Daniel Bücheler, Sebastian Böhm, and Michael Meier. I still know you were here: Leveraging probe request templates for identifying Wi-Fi devices at scale. *Computer Communications*, 246:108367, 2026. doi: 10.1016/j.comcom.2025.108367.

Fei Wang, Tingting Zhang, Wei Xi, Han Ding, Ge Wang, Di Zhang, Yuanhao Cui, Fan Liu, Jinsong Han, Jie Xu, and Tony Xiao Han. A survey on Wi-Fi sensing generalizability: Taxonomy, techniques, datasets, and future research prospects. *IEEE Communications Surveys & Tutorials*, 28:5227–5266, 2026. doi: 10.1109/COMST.2026.3670854.

Mingtao Wang, Xin Zhou, and Yuanyuan Chen. A comprehensive survey of crowd density estimation and counting. *IET Image Processing*, 19(1):e13328, 2025. doi: 10.1049/ipr2.13328.

Wei Xi, Jizhong Zhao, Xiang-Yang Li, Kun Zhao, Shaojie Tang, Xue Liu, and Zhiping Jiang. Electronic frog eye: Counting crowd using WiFi. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 361–369. IEEE, 2014. doi: 10.1109/INFOCOM.2014.6847958.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320, 2021.

Juan Carlos Zúñiga, Carlos J. Bernardos, and Amelia Andersdotter. State of affairs for randomized and changing media access control (MAC) addresses. RFC 9724, Internet Engineering Task Force, March 2025. URL https://www.rfc-editor.org/info/rfc9724.

A Iterative Adaptive Pruning Algorithm

The *Global-Local View Construction* subsection of the main paper introduces Iterative Adaptive Pruning (IAP), which selects the Element IDs used to form the local view without identity labels. Algorithm 1 provides the detailed procedure.

For each Element ID k, the collection-level statistic $C(k)$ counts the distinct Value-field lengths observed across the frame collection, with absence encoded as zero. IAP is run once on the training collection before optimization and once on each unlabeled target collection before inference. The selected Element IDs remain fixed while the corresponding frames are encoded and clustered, and the model parameters remain fixed during target-collection selection. The margin β_τ is non-increasing as the candidate threshold τ grows. We use $\beta_0 = 3$ and $\delta = 1$ in all experiments. If $\mathcal{Q}_1 = \emptyset$, IAP returns $\mathcal{K}_{\text{local}} = \emptyset$. GLVC uses a frame's global TLV sequence as its local input whenever $\mathcal{K}_{\text{local}} = \emptyset$ or that frame contains no selected Element ID.

Algorithm 1 Iterative Adaptive Pruning (IAP)

Input: Unlabeled frame collection $\mathcal{X} = \{X_i\}_{i=1}^{N}$
Parameter: Initial sensitivity β_0, Decay rate δ
Output: Selected Element IDs $\mathcal{K}_{\text{local}}$

Compute length cardinality $C(k)$ for every Element ID k in $\mathcal{X}$.
Initialize threshold $\tau \leftarrow 1$.
$\mathcal{Q}_\tau \leftarrow \{k \mid C(k) > \tau\}$.
if $\mathcal{Q}_\tau = \emptyset$ **then**
  return $\mathcal{K}_{\text{local}} \leftarrow \emptyset$.
end if
while $\mathcal{Q}_\tau \neq \emptyset$ **do**
  $\mathcal{Q}_{\tau+1} \leftarrow \{k \mid C(k) > \tau + 1\}$.
  if $\mathcal{Q}_{\tau+1} = \emptyset$ **then**
    return $\mathcal{K}_{\text{local}} \leftarrow \mathcal{Q}_\tau$.
  end if
  $\Delta_\tau \leftarrow |\mathcal{Q}_\tau| - |\mathcal{Q}_{\tau+1}|$.
  $\beta_\tau \leftarrow \max(1, \lfloor \beta_0 - \delta(\tau - 1) \rfloor)$.
  if $\Delta_\tau > \beta_\tau$ **then**
    $\tau^* \leftarrow \tau + 1$.
    return $\mathcal{K}_{\text{local}} \leftarrow \mathcal{Q}_{\tau^*}$.
  end if
  $\tau \leftarrow \tau + 1$.
end while

B Overall Training Algorithm

To facilitate reproducibility, Algorithm 2 summarizes the complete GLCR training and inference pipeline. It integrates GLVC, HPE, SFM, and CGR as described in the corresponding methodology subsections of the main paper. The joint-optimization steps implement the bounded uncertainty weighting defined in the *Training Objectives* paragraph. Following the main paper, let $\mathcal{J} = \{g, l, c, \text{ie}, \text{var}\}$ index the five losses and let $\Omega = [s_{\min}, s_{\max}]$ denote the clipping interval for their learnable log-variance parameters. IAP is applied once to the complete training collection before optimization and once to each complete target collection before inference; the selected Element IDs remain fixed while the corresponding collection is processed. For compactness, $\mathbf{Z}$ denotes the target collection of fused embeddings, and $\text{ConsensusRank}(\mathbf{Z})$ denotes the label-free DBSCAN parameter-selection procedure detailed in the *Consensus Ranking for Clustering* subsection. It returns $(\varepsilon^*, \text{minPts}^*)$.

C Detailed Network Architecture

Table 4 details the two asymmetric branches of GLCR. The global branch uses the three-block HPE with IEHA, whereas the local branch applies the two-block Transformer encoder E_l followed by SFM.

Algorithm 2 Training and Inference of GLCR

Input: Unlabeled WiDAR training collection, unlabeled target collection $\mathcal{X} = \{X_i\}_{i=1}^{N}$
Parameter: Model parameters, log-variances $\{s_r\}_{r\in\mathcal{J}}$, number of training epochs
Output: Inferred device groups

Initialize the model parameters and $\{s_r\}_{r\in\mathcal{J}}$.
$\mathcal{K}_{\text{local}} \leftarrow \text{IAP}(\text{training collection})$
for each training epoch **do**
  for mini-batch $\mathcal{B}$ from the training collection **do**
    // Step 1: Global-Local View Construction (GLVC)
    $(\mathbf{S}_g, \mathbf{C}, \mathbf{M}), (\mathbf{S}_l, \mathbf{M}^l) \leftarrow \text{GLVC}(\mathcal{B}, \mathcal{K}_{\text{local}})$
    // Step 2: HPE and SFM
    Construct $\mathbf{H}_g^{(0)}$ from $(\mathbf{S}_g, \mathbf{C})$ and $\mathbf{H}_l^{(0)}$ from $\mathbf{S}_l$ using the embedding definitions in the main paper
    $\mathbf{z}_g \leftarrow \text{HPE}(\mathbf{H}_g^{(0)}, \mathbf{C}, \mathbf{M})$
    $\mathbf{H}_l \leftarrow E_l(\mathbf{H}_l^{(0)}, \mathbf{M}^l)$; $\mathbf{z}_l \leftarrow \text{SFM}(\mathbf{H}_l, \mathbf{M}^l)$
    // Step 3: Context-Gated Refinement (CGR)
    $\mathbf{z}_c \leftarrow \text{CGR}(\mathbf{z}_g, \mathbf{z}_l)$
    // Step 4: Joint Optimization
    Compute $\{\mathcal{L}_g, \mathcal{L}_l, \mathcal{L}_c, \mathcal{L}_{\text{ie}}, \mathcal{L}_{\text{var}}\}$
    $\bar{s}_r \leftarrow \min(s_{\max}, \max(s_{\min}, s_r)),\ \forall r \in \mathcal{J}$ {Clip the loss-weighting parameters}
    $\mathcal{L}_{\text{total}} \leftarrow \sum_{r\in\mathcal{J}} \left(\frac{1}{2}\exp(-\bar{s}_r)\mathcal{L}_r + \frac{1}{2}\bar{s}_r\right)$
    Update the model parameters and $\{s_r\}_{r\in\mathcal{J}}$ via backpropagation
  end for
end for
// Inference Phase
$\mathcal{K}_{\text{local}} \leftarrow \text{IAP}(\mathcal{X})$
Generate $\mathbf{Z} = \{\mathbf{z}_c^{(i)}\}_{i=1}^{N}$ from $\mathcal{X}$ using $\mathcal{K}_{\text{local}}$
$(\varepsilon^*, \text{minPts}^*) \leftarrow \text{ConsensusRank}(\mathbf{Z})$
return $\text{DBSCAN}(\mathbf{Z}; \varepsilon^*, \text{minPts}^*)$

Table 4: Detailed architecture configuration of GLCR.

Module	Parameter	Value
Input Embedding	Embedding Dimension (d)	16
	Dropout Rate	0.1
Global Encoder (HPE)	HPE Transformer Blocks	3
	Attention Heads	4
	FFN Hidden Dimension	64
	IEHA Interaction Layers	1
	Activation Function	ReLU
Local Branch	Transformer Blocks in E_l	2
	Salience MLP Hidden Dim	8
	SFM Projection $\mathbf{W}_1$	$16 \to 8$
	SFM Projection $\mathbf{w}_2$	$8 \to 1$
Refinement (CGR)	Fusion Gate MLP	$32 \to 16 \to 16$
	Final Embedding Dim	16

Reconstruction Decoders. In the released implementation, each decoder D_b, $b \in \{g, l, c\}$, contains two Transformer self-attention blocks with dimension $d = 16$, four attention heads, FFN width $4d = 64$, ReLU activation, dropout 0.1, and post-residual layer normalization. The latent vector $\mathbf{z}_b$ is repeated over the target length T_b; a fixed sinusoidal positional encoding is added at each position, followed by dropout. The two blocks process this sequence using the target padding mask, and a position-wise linear projection maps the resulting features to vocabulary logits $\text{D}_b(\mathbf{z}_b, t)$. Decoding is non-autoregressive: no target-byte embeddings, causal mask, or cross-attention are used.

The global and fused decoders reconstruct $\mathbf{S}_g$, whereas the local decoder reconstructs $\mathbf{S}_l$. All three decoders have independent parameters and are removed at inference.

D Implementation Details

D.1 Hyperparameter Settings

Table 5 lists the training hyperparameters. The adaptive loss-weighting parameters s_r use a smaller learning rate and are clipped to $\Omega = [s_{\min}, s_{\max}]$ during optimization.

Table 5: Training hyperparameters.

Category	Hyperparameter	Value
Training	Optimizer	AdamW
	Base Learning Rate	1×10^{-3}
	Weight Decay	1×10^{-4}
	Batch Size	128
	Training Epochs	2,000
	Random Seeds	10 distinct seeds
	Uncertainty LR	5×10^{-5}
Loss Weights	VICReg γ (Target Std)	1.0
	VICReg ϵ	1×10^{-4}
	Log-variance range Ω	$[-2.0, 5.0]$

The global and local byte sequences are not truncated at 512 bytes. Within each mini-batch, they are padded independently to their respective maximum lengths, denoted by L and L_l in the main paper.

D.2 Consensus Ranking for Clustering

To select the DBSCAN parameters $(\varepsilon, \mathrm{minPts})$ without ground-truth labels, we use the consensus ranking strategy described in the *Implementation Details* paragraph of the main paper. Let $\mathcal{E}$ contain 20 equally spaced ε values in $[0.1, 2.0]$, and let $\mathcal{M} = \{2, \ldots, 10\}$ contain the nine tested minPts values. The candidate set $\mathcal{P} = \mathcal{E} \times \mathcal{M}$ therefore contains 180 parameter pairs. For each candidate $p = (\varepsilon, \mathrm{minPts}) \in \mathcal{P}$, we compute the Silhouette coefficient (Sil), Calinski–Harabasz index (CH), and Davies–Bouldin index (DB). Following the main paper, Sil and CH are ranked in descending order and DB is ranked in ascending order, with rank 1 being best:

$$\mathrm{Score}(p) = R_{\mathrm{Sil}}(p) + R_{\mathrm{CH}}(p) + R_{\mathrm{DB}}(p). \tag{7}$$

The selected configuration $p^* = (\varepsilon^*, \mathrm{minPts}^*)$ attains the lowest aggregate rank:

$$p^* \in \arg\min_{p \in \mathcal{P}} \mathrm{Score}(p). \tag{8}$$

The inferred cluster count $\hat{K}$ is the number of non-noise clusters produced by DBSCAN with the selected parameters. This implements the main paper's lowest-rank-sum rule without using device-identity labels.

For internal-index computation, all samples are retained, and DBSCAN noise samples are treated as a single additional group sharing the same label. This group is excluded from the inferred device count $\hat{K}$. This convention applies only to the internal indices used for parameter selection; noise samples remain in the ACC denominator and are counted as incorrect, as specified below.

D.3 Evaluation Metrics

Let $\mathcal{I}_{\mathrm{eval}}$ contain all frame indices retained for evaluation and let $N_{\mathrm{eval}} = |\mathcal{I}_{\mathrm{eval}}|$. Let K and $\hat{K}$ denote the numbers of ground-truth devices and inferred non-noise clusters, respectively. For frame i, let $y_i \in \{1, \ldots, K\}$ denote its ground-truth device label and let $\hat{y}_i \in \{0, 1, \ldots, \hat{K}\}$ denote its

DBSCAN output, where 0 denotes noise. Define the complete label vectors as $\mathbf{y} = (y_i)_{i\in\mathcal{I}_{\text{eval}}}$ and $\hat{\mathbf{y}} = (\hat{y}_i)_{i\in\mathcal{I}_{\text{eval}}}$.

Because $\hat{K}$ need not equal K, ACC is computed using an optimal one-to-one partial matching between the K ground-truth device labels and the $\hat{K}$ non-noise cluster labels. Let n_{kj} denote the number of frames from device k assigned to predicted label $j \in \{0, 1, \ldots, \hat{K}\}$, including the noise label $j = 0$, and let $\mathfrak{M}$ denote the set of valid one-to-one partial matchings π between ground-truth and non-noise labels. Then

$$\text{ACC} = \frac{1}{N_{\text{eval}}} \max_{\pi\in\mathfrak{M}} \sum_{(k,j)\in\pi} n_{kj}. \tag{9}$$

Frames assigned the noise label, together with frames in unmatched clusters, remain in the denominator and are counted as incorrect.

Since $K > 1$, define geometric-mean-normalized NMI as

$$\text{NMI} = \begin{cases} \dfrac{I(\mathbf{y};\hat{\mathbf{y}})}{\sqrt{H(\mathbf{y})H(\hat{\mathbf{y}})}}, & H(\hat{\mathbf{y}}) > 0, \\ 0, & H(\hat{\mathbf{y}}) = 0 \quad \text{(e.g., } \hat{K} = 0\text{)}, \end{cases} \tag{10}$$

Here I and H denote mutual information and entropy, respectively; NMI and ARI use complete label vectors with DBSCAN noise as the predicted label 0.

For ARI, extend the contingency table to include the noise column $j = 0$, with $a_k = \sum_{j=0}^{\hat{K}} n_{kj}$ and $b_j = \sum_{k=1}^{K} n_{kj}$. Define $A = \sum_{k=1}^{K} \binom{a_k}{2}$, $B = \sum_{j=0}^{\hat{K}} \binom{b_j}{2}$, $S_{\text{joint}} = \sum_{k=1}^{K}\sum_{j=0}^{\hat{K}} \binom{n_{kj}}{2}$, and $N_{\text{pair}} = \binom{N_{\text{eval}}}{2}$. Then

$$\text{ARI} = \frac{S_{\text{joint}} - AB/N_{\text{pair}}}{\frac{1}{2}(A+B) - AB/N_{\text{pair}}}. \tag{11}$$

All three metrics are reported as percentages.

E Data Collection Platform Details

To construct the WiDAR dataset, we developed a custom portable sniffing platform (shown in Figure 5 of the main paper). We used the system to passively capture IEEE 802.11 management frames in controlled environments.

Hardware Specifications. The sniffer is built on the Espressif ESP32-D0WDQ6 SoC, which includes dual Xtensa 32-bit LX6 MCUs and an integrated Wi-Fi/Bluetooth module. The RF front end operates at 2.4 GHz, supports IEEE 802.11 b/g/n, and has a nominal receiver sensitivity of -100 dBm.

Data Storage and Operation. The system is equipped with a high-speed Micro SD storage module (128 GB) to log captured frames locally, ensuring data integrity without reliance on real-time network backhaul. The device operates independently and is powered by an external DC supply. It records raw probe requests, RSSI, timestamps, and channel information, but GLCR uses only retained IE content and structure for probe-request association through clustering.

F Experimental Results

Ablation Configurations. For *w/o CGR*, we replace the entire CGR module, including its layer normalization, with fixed averaging: $\mathbf{z}_c = (\mathbf{z}_g + \mathbf{z}_l)/2$. For *w/o SFM*, we retain the local Transformer encoder and replace salience-weighted pooling with mean pooling over non-padding local tokens.

The main configuration (GLCR-Base with $d = 16$ and IEHA between the second and third HPE blocks) was fixed before evaluation on the independent test sets. Additional model scales, embedding dimensions, and IEHA insertion positions were evaluated for sensitivity analysis without changing the main configuration.

Model Scale Study. GLCR-Base corresponds to the configuration used in the main paper. To examine sensitivity to model scale, we additionally compare four variants (*Small*, *Base*, *Large*, and *Huge*) on the three public benchmarks and WiDAR. Table 6 summarizes their architectural parameters and computational overhead, while Figure 8 shows the corresponding clustering performance.

As shown in Table 6, the parameter count, FLOPs, and latency increase across the four configurations. GLCR-Base has 28.66 K parameters, 16.74 M FLOPs, and a single-sample inference latency of 11.53 ms, whereas GLCR-Huge has 149.47 K parameters and 130.92 M FLOPs. The block count in Table 6 refers to the number of HPE Transformer blocks; the local branch follows the configuration in Table 4. Figure 8 shows that GLCR-Base obtains the highest reported scores across the four datasets. For example, WiDAR ACC increases from 65% with Small to 81% with Base and decreases to 65% with Huge. The Large and Huge variants also obtain lower ACC, NMI, and ARI than Base across the four datasets. Because the variants jointly change the HPE block count, embedding dimension, and number of attention heads, this study compares complete model configurations rather than isolating the effect of depth.

Table 6: Configuration details and efficiency profile of GLCR variants. Latency is measured with batch size 1 on a single NVIDIA RTX PRO 6000 GPU (96 GB) with Ubuntu 24.04 OS.

Model Variant	**Architecture**			**Efficiency**		
	HPE Blocks	**Dim**	**Heads**	**Params**	**FLOPs**	**Latency**
GLCR-Small	1	16	2	22.10 K	10.91 M	2.69 ms
GLCR-Base (Ours)	**3**	**16**	**4**	**28.66 K**	**16.74 M**	**11.53 ms**
GLCR-Large	6	32	8	98.21 K	79.01 M	34.29 ms
GLCR-Huge	12	32	8	149.47 K	130.92 M	61.24 ms

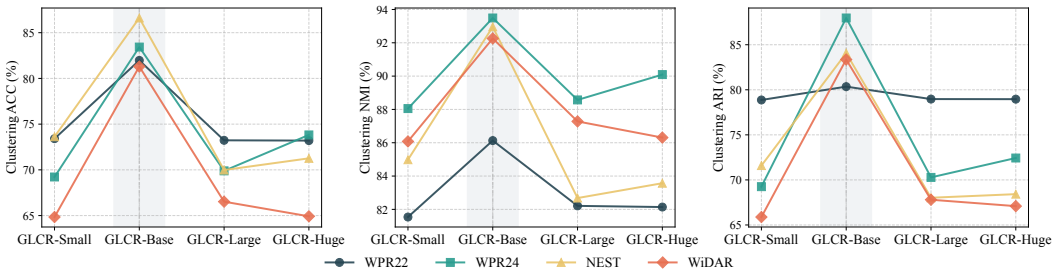


Figure 8: Study of the clustering performance trends of four GLCR variants, Small, Base, Large, and Huge, on three public benchmarks and the WiDAR dataset.

Latent Dimension Study. The representation dimension d determines the size of the fused embedding used for density-based clustering. The main experiments use $d = 16$; to examine sensitivity to representation size, we additionally evaluate $d \in \{8, 16, 32, 64\}$. Figure 9 shows that the reported scores peak at $d = 16$ across the four datasets. At $d = 8$, ACC is 1–22 percentage points lower than its value at $d = 16$, depending on the dataset. Increasing the dimension beyond 16 does not improve the reported scores; on WiDAR, ACC at $d = 64$ is more than 13 percentage points below that at $d = 16$. Higher-dimensional embeddings may make Euclidean neighborhoods less informative for density-based clustering, but this experiment does not isolate the cause of the decline.

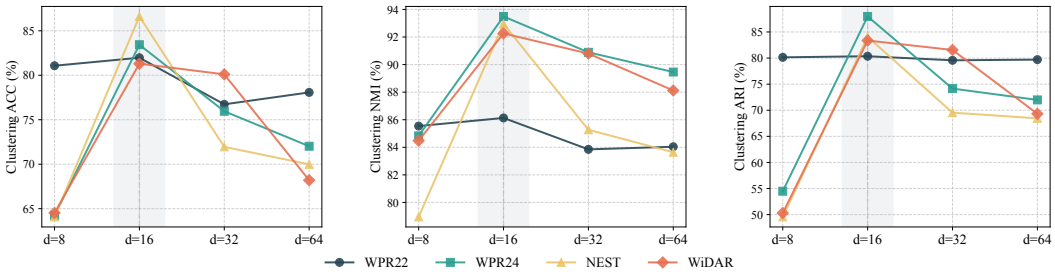


Figure 9: Study of different embedding dimensions d on clustering.

IEHA Insertion Position Study. The main experiments place IEHA between the second and third HPE blocks. To examine sensitivity to insertion position, we additionally evaluate IEHA after the first, second, and third Transformer blocks. Figure 10 shows that inserting IEHA after the second block provides the best overall trade-off across the four datasets. It achieves the highest scores for most dataset–metric combinations, whereas insertion after the third block yields marginally higher NMI and ARI on NEST. Moving IEHA to the end of the third block reduces WiDAR ARI from approximately 83.4% to 70.7%.

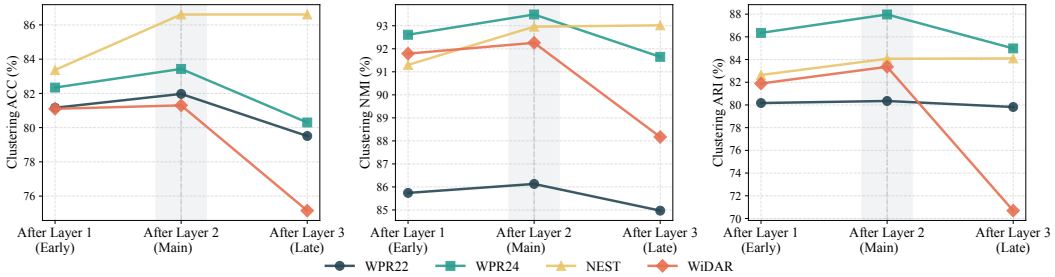


Figure 10: Study of IEHA insertion position.