# Lost in Compression? Matching Signals from Compressed I/Q Data

Dieter Verbruggen and Sofie Pollin
*ESAT - WaveCoRE*
KU Leuven, 3000 Leuven, Belgium
dieter.verbruggen/sofie.pollin@kuleuven.be

Matthias Schäfer
*SeRo Systems GmbH*
Frankfurt am Main, Germany
schaefer@sero-systems.de

Yago Lizarribar and Gérôme Bovet
*armasuisse Science + Technology*
Thun, Switzerland
yago.lizarribarcarrillo/gerome.bovet@armasuisse.ch

***Abstract*—Accurate signal matching across spatially distributed receivers is essential for applications including Time Difference of Arrival (TDOA)-based localization and collaborative sensing. A key challenge arises in data transmission when signal structure and modulation are unknown, often necessitating the transmission of raw I/Q data, which is often infeasible due to bandwidth constraints, as traditional decoding-based methods fail. This paper investigates compression techniques that retain matching capability while significantly reducing data volume. We evaluate classical lossy methods (sample resolution reduction and downsampling) alongside a deep learning approach using a Siamese network to learn compact embeddings. Using real-world Mode S radar signals, we assess matching accuracy and compression ratio. Results show that moderate downsampling improves performance by suppressing noise, and that learned embeddings enable compression ratios two orders of magnitude higher than classical approaches with minimal accuracy loss.**

***Index Terms*—Data compression, Digital signal processing, Deep Learning, Siamese networks, I/Q data**

## I. Introduction

Distributed receiver networks are vital for modern sensing applications, enhancing coverage and accuracy but generating extensive data. A primary hurdle is the efficient processing and transmission of high-bandwidth sensor data, especially when signal modulation and coding are unknown or change dynamically. Traditional payload extraction methods are impractical, and current lossless compression techniques, like those in ElectroSense [1], often can't achieve the significant data reduction needed for deployment in resource-constrained settings. For illustration, with a sample rate of 10 MHz and 32-bit samples, each sensor generates approximately 320 Mbps of data, a rate that frequently surpasses the capabilities of common data links, especially in expansive networks or systems with severe bandwidth limitations.

The central question addressed in this paper is: *How can we produce a compressed representation of unknown signals that minimizes data volume while retaining sufficient information to support accurate signal matching?*

The challenge of packet matching is similar to the well-studied problem of device fingerprinting [2], wherein classifiers attempt to determine whether a signal originates from a specific source based on some abstract representation of the signal (the fingerprint). However, our goal differs: instead of identifying the source, we are interested in matching individual transmissions, independently of the receiver. The core task is to robustly determine, based on abstract (compressed) signal representations, whether two spatially separated sensors have observed the identical transmission event.

The efficient signal matching framework is essential for various applications in collaborative sensing and processing by distributed nodes. Key examples include Time Difference of Arrival (TDOA)-based localization, widely used in aviation for multilateration, satellite constellations [3], high-altitude platforms (UAVs [4], balloons [5]), and tracking IoT devices [6], and distributed modulation classification [7]. This framework is beneficial for collaborative spectrum sensing and network-wide anomaly detection, as all these scenarios require robust matching of identical transmission instances.

To address our central research question, this paper evaluates various methodologies for compressing in-phase quadrature (I/Q) signal data with respect to their suitability for robust signal matching in distributed networks. Our evaluation leverages real-world signal data recorded at 1030 MHz, the uplink frequency employed by secondary surveillance radar and collision avoidance systems in aviation.

We specifically focus on the particular applications which are acutely sensitive to data volume due to inherent constraints on power and data link bandwidth, making them an ideal and representative testbed for the proposed techniques. The selection of this specific dataset and application serves as a concrete example to demonstrate the efficacy and practical implications of our framework.

The main contributions of this work are as follows:

- We analyze how reducing sample resolution and downsampling affect the accuracy of signal matching.
- We propose and evaluate a deep learning-based Siamese network approach using real-world recordings.
- We compare the performance of different methods in terms of data reduction and signal matching accuracy.

## II. Background and Assumptions

### A. System Model

We consider a general matching application where multiple transmitters $\mathrm{Tx}_i$ send packets detected by multiple receivers. Each receiver $\mathrm{Rx}_i$ compresses its received signal and sends this compressed representation $p_i^j$, along with any relevant

This work was supported by armasuisse Science and Technology (S+T), Thun, Switzerland.

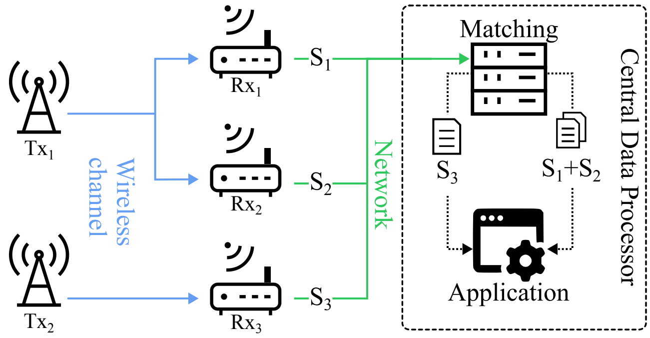


Fig. 1. System model of signal matching.

application-specific data $d_i^j$ like a timestamp for TDOA or a modulation prediction for classification, to a Central Data Processor (CDP). The CDP then processes the received data $S_i^j = (d_i^j, p_i^j)$ by matching the receptions and forwards the processed data to various applications, such as TDOA-based localization or collaborative modulation classification. A summary of the system model is depicted in Figure 1.

In this work, we focus on challenging scenarios with high signal rates and limited signal knowledge. Including both civil and military contexts, such as localizing drones using unknown control protocols in crowded ISM bands, or identifying unknown and unlawful transmitters by spectrum regulators. Furthermore, we assume constrained data link capacities between receivers and the CDP. Hence, our objective is to transmit *minimal* signal information while still enabling accurate matching at the CDP.

Finally, we note two special conditions in the considered system model that allow for trade-offs in the applied methods. First, decoding of signals at the CDP is not required for application purposes. This allows for the use of lossy compression of the raw signal data. Second, the system can tolerate a certain level of false matches, but missed matches (false negatives) must be minimized. This will be reflected in the choice of the target performance metric in the subsequent experimental analysis.

### B. In-Phase and Quadrature Data

In-phase and quadrature (I/Q) data is a common representation of received baseband signals, typically produced by software-defined radios. These I/Q samples are the output of analog-to-digital converters (ADCs) that digitize the down-converted signal, either in baseband or at an intermediate frequency. Each sample consists of an in-phase (I) and a quadrature (Q) component, usually represented as signed integers of fixed bit width. This bit width, referred to as the ADC resolution, affects the receiver's sensitivity and dynamic range. Typical commercial off-the-shelf (COTS) ADCs provide resolutions ranging from 8 to 16 bits.

The ADC's sample rate determines both the bandwidth limit of the receiver (according to the Nyquist-Shannon sampling theorem) and the resulting data volume. In our setup, we use a 12-bit ADC per I and Q channel, sampling at 12 Msamples per second. This yields a raw data rate of 288 Mbit/s per receiver. However, due to byte alignment in practical computer and network systems, 16 bits must be used to store each 12-bit value, increasing the actual data rate to 384 Mbit/s.

For the remainder of the paper, we make the following additional assumptions:

- Transmissions are of short duration, as typically observed in digital communication systems that use small packets. Long-duration transmissions such as analog voice or continuous wave jamming are out of scope.
- Each transmission is detected and timestamped at the receiver, for example, using a squelch filter or more advanced signal detection methods.

As a result, we only collect I/Q samples corresponding to actual transmissions. This alone provides a substantial reduction in data volume, which depends on the channel occupancy. For instance, if the channel is active 25% of the time, the resulting data volume is reduced to approximately 25% of the continuous I/Q stream.

### C. Correlation-based Signal Similarity

As part of the signal matching step at the CDP using traditional signal matching, we assess whether two receptions $S_i$ and $S_k$ correspond to the same transmission based on the similarity of their I/Q samples. Let $S_i[n]$ and $S_k[n]$ denote the $n$-th complex I/Q sample extracted from two receptions of a signal $S$. To ensure comparability, both sequences are first normalized to a fixed length $N$ through truncation or zero-padding. We then account for any phase offsets to ensure global phase differences don't affect the similarity score.

We use the complex-valued version of the *cosine similarity*, adapted to I/Q signals given by:

$$\mathrm{corr}(S_i, S_k) = \frac{\left|\sum_{n=1}^{N} S_i[n] \cdot S_k'[n]\right|}{\|S_i\| \cdot \|S_k'\|}. \tag{1}$$

Here, $S_k'$ represents the phase-aligned $S_k$ $\|\cdot\|$ is the Euclidean norm of the complex vector. The result lies in the interval $[0, 1]$, where a value close to 1 indicates high similarity (identical up to phase and scaling), and values near 0 indicate dissimilar or approximately orthogonal signals.

If $\mathrm{corr}(S_i, S_k) > T$ for a predefined threshold $T$, the CDP considers the two receptions as matching. This correlation-based similarity check is applied during the Signal Matching step of the CDP processing pipeline. Note that this thresholding rule introduces a trade-off between false positives and false negatives, the impact of which is explored in the evaluation section.

### D. Deep Learning-based Compression and Matching using a Siamese Network

The deep learning approach leverages a compression model and the Siamese network architecture [8] to generate an efficient signal representation for matching. In this architecture, as shown in Figure 2, each receiver employs an identical deep learning model to process detected radio signals. This model generates a compressed, lower-dimensional embedding that encapsulates the most essential features of the transmission.

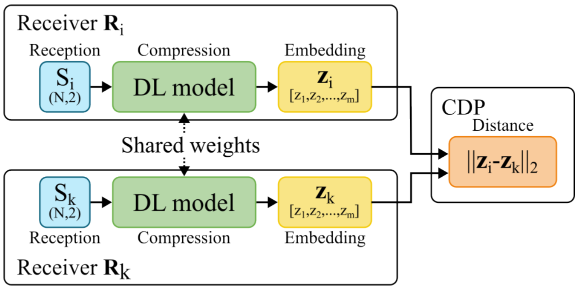


Fig. 2. Siamese network architecture

The use of a Siamese network, with its shared weights across identical branches (one conceptually residing at each receiver), ensures that the resulting embeddings from different receptions of the same signal reside in a common, comparable feature space. This allows for direct similarity assessment at the CDP. The resulting embedding vector, $\mathbf{z}$ (containing real-valued elements), represents a reduced data volume compared to the raw I/Q samples and is transmitted to the CDP.

At the CDP, the task of signal matching is performed by evaluating the similarity between the received embedding vectors. Given two embeddings, $\mathbf{z}_i$ and $\mathbf{z}_k$, corresponding to receptions $S_i$ and $S_k$, their similarity is inversely proportional to the Euclidean distance:

$$d(\mathbf{z}_i, \mathbf{z}_k) = \|\mathbf{z}_i - \mathbf{z}_k\| = \sqrt{\sum_{m=1}^{M} (z_{i,m} - z_{k,m})^2} \qquad (2)$$

where $M$ is the dimensionality of the embedding vector. A smaller Euclidean distance between two embeddings suggests a higher degree of similarity between the original received signals, indicating a greater likelihood that they originate from the same transmission.

Similar to the thresholding applied to the correlation score in the correlation-based method, a predefined threshold $T$ will be applied to the Euclidean distance. If $d(\mathbf{z}_i, \mathbf{z}_k) < T$, the CDP considers the two receptions as a match. This threshold $T$ on the embedding distance will also introduce a trade-off between false positives and false negatives in the matching process, which will be analyzed in the evaluation section.

### *E. Evaluation Methodology and Experimental Setup*

To evaluate our approach using real-world signal data, we focus on the practical application of matching transmissions of secondary surveillance radar Mode S systems, which can result in a localisation application. Mode S radars emit interrogation signals directed at aircraft transponders using Differential Binary Phase-Shift Keying (DBPSK) modulation. Each interrogation signal has a duration of either 19.75 µs or 33.75 µs, corresponding to different Mode S uplink formats.

For this work, signal data was collected using a GRX3X receiver developed by SeRo Systems[1]. This receiver is capable of detecting and demodulating Mode S interrogations and provides a GPS-synchronized timestamp with nanosecond resolution for each detection. In addition to detected signal metadata, the GRX3X also outputs a continuous stream of time-synchronized I/Q samples at a rate of 12 MHz and an ADC resolution of 12 bits.

Recordings were conducted at a site approximately 2 km line-of-sight from Frankfurt International Airport. Due to the high traffic density and the presence of ground-based Mode S interrogators within direct view of the receiver, the environment represents a particularly challenging scenario with very high signal rates and transmissions that are often repeated within short time intervals.

*1) Signal Preparation and Ground Truth:* For each detected Mode S signal, we extract the corresponding block of I/Q samples from the raw data stream, beginning with the preamble pulses and ending after the DBPSK modulation block. To enable comparison across both short and long interrogations, all signals are normalized to a fixed window size of 512 samples through zero-padding or truncation, as described in subsection II-C. We note that this window size accommodates both short interrogations (237 samples) and long ones (405 samples).

In our evaluations throughout the paper, two signals are considered a true match in the CDP if the Hamming distance between their decoded symbol sequences is zero, i.e., they carry the exact same bit sequence. This strict criterion defines our ground truth for evaluating matching performance: only signals with identical payloads should be matched.

*2) Data Collection and Evaluation Methodology:* For our evaluation, we collected three datasets, each comprising 5,000 individual signal detections. Two datasets were used for training and validation of the Siamese neural network, while the third was reserved for evaluation. To ensure independence between sets, we ensured that the sets are disjoint, i.e., no signal in one dataset has a Hamming distance of zero to any signal in another set.

To assess the accuracy of our signal matching method, we performed pairwise comparisons between all 5,000 signals in the evaluation dataset, resulting in 12,497,500 unique signal pairs. For each pair, we applied the proposed compression techniques to the signal data and then tested whether they would still be matched correctly. The decoded bit sequences from the GRX3X receiver served as ground truth for determining true and false matches.

Using this ground truth, we compute standard evaluation metrics including precision, recall, and the F1 score, which will be discussed in the following section.

*3) Performance Metrics:* The goal of our approach is to minimize the data volume collected at the CDP while preserving high matching accuracy. Consequently, the two primary performance metrics considered are the *compression ratio* and the *matching accuracy*.

Matching accuracy is evaluated in terms of *precision*, *recall*, and their harmonic mean, the *F1 score*. A precision of 1 indicates no false matches, whereas a recall of 1 implies that all true matches were correctly detected. Given the trade-off between false matches and missed matches introduced by lossy

[1] https://sero-systems.de

compression, the F1 score serves as a balanced measure of matching performance.

We note that the overall *accuracy*, defined as the ratio of correctly classified pairs to all evaluated pairs, is not informative in this context, as non-matches dominate the dataset (99.7%). Due to the very high true negative rate, the accuracy metric is largely determined by the abundance of true negatives and therefore does not meaningfully reflect the system's ability to detect true matches.

To measure the compression ratio, we use the full, uncompressed I/Q data of the detected transmissions as the baseline. This reflects the maximum amount of signal information available in our system. Given the 12 MHz sample rate, 12-bit ADC resolution (packed into 16-bit integers), and two channels (I and Q), the uncompressed data volume $V(S)$ for a signal $S$ of duration $\tau(S)$ is

$$V(S) = \tau(S) \times 12 \text{ MHz} \times 2 \times 16 \text{ bit} = \tau(S) \times 384 \text{ Mbit/s}$$

Let $\tilde{V}(S)$ denote the size of the compressed representation of the detected signal $S$. The compression ratio $R$ for a given method is then determined by calculating the ratio between the total volume of the uncompressed data to the total volume of the compressed data over all $N$ detected signals $S_i$ in the evaluation dataset:

$$R = \frac{\sum_{i=1}^{N} V(S_i)}{\sum_{i=1}^{N} \tilde{V}(S_i)} \tag{3}$$

Based on these metrics, we evaluate the impact of different compression methods and compression levels on matching accuracy and data reduction. The results, including precision, recall, F1 score, and compression ratios for each method, are presented and discussed in the following sections.

## III. Compression based on I/Q Sample Rate and Sample Resolution Reduction

To reduce the data volume without interpreting the signal content, we first consider two lossy compression strategies: reducing sample resolution and reducing sample rate.

### A. Methodology

*1) Reducing the Sample Resolution:* Reducing the number of bits per sample is effective only when the resulting size aligns with full bytes. Given our 12-bit ADC resolution (stored at 16 bits), we consider reductions to 8 bits and 4 bits per sample. Reducing the resolution to 8 bits halves the data volume, yielding $R = 2$. Further reducing to 4 bits results in another halving as both I and Q can then be packed into a single byte.

The bit-selection strategy for the reduction depends on the expected signal strength:

- **Strong signals:** Drop the least significant bits (LSBs), which are likely to represent noise. This is equivalent to performing an integer division by a power of two.
- **Weak signals:** Drop the most significant bits (MSBs), assuming they are unused (i.e. zeroes). Care must be taken to avoid signal loss or clipping.

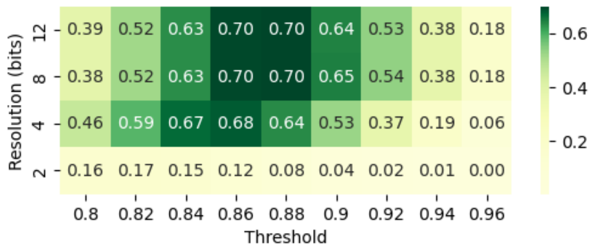

Fig. 3. F1 score by sample resolution and threshold

We dynamically select the bits to be dropped *per detected signal* based on actual bits used in I and Q. To determine which bits can be safely discarded, we first estimate the number of bits actually needed to represent the signal's dynamic range. This is done by evaluating the maximum value present in the I and Q components of each detected signal. If the current bit usage exceeds the target resolution, we discard the excess LSBs by shifting the values to the right. This effectively preserves the most significant structure of the signal while compressing its representation. If the signal already fits within the target bit range, no transformation is applied.

*2) Reducing the Sample Rate:* Sample rate reduction is achieved via downsampling. This involves applying a low-pass filter to avoid aliasing, followed by retaining every $n$-th sample. In our evaluation, we use an order 8 Chebyshev type I filter for the downconversion. Downsampling by a factor of 2 reduces the sample rate from 12 Msamples/s to 6 Msamples/s and the data volume by 50%. Hence, the relationship between the downsampling factor and the compression ratio is direct: a downsampling factor of 2 yields a compression ratio $R = 2$. However, this also cuts the signal bandwidth in half, potentially discarding frequency components valuable for signal matching.

### B. Impact on Accuracy of Correlation-based Matching

In this section, we evaluate the impact of a reduction in signal rate (downsampling) and lowering the signal resolution on the matching performance. We note that, although we consider them separately in our analysis, both downsampling and sample resolution reduction can be applied together. In that case, the overall compression ratio $R$ is the product of the individual compression ratios. For example, reducing the resolution to 8 bits ($R_1 = 2$) and downsampling by a factor of 4 ($R_2 = 4$) results in an overall compression ratio of $R = R_1 \times R_2 = 8$.

*1) Sample Resolution:* We evaluated the effects of sample resolution reduction on matching accuracy. Starting from the original 12-bit ADC resolution, we reduced the sample resolution to 8, 4, and 2 bits and computed the F1 scores across different similarity thresholds. The results are shown in Figure 3.

At full 12-bit resolution, the maximum F1 score of 0.71 is achieved at a similarity threshold of 0.87. This threshold provides the best balance between precision and recall, yielding a recall of 66% and a precision of 75.8%. Thus, approximately 34% of true matches are missed, while 24.2% of the reported matches are false positives.

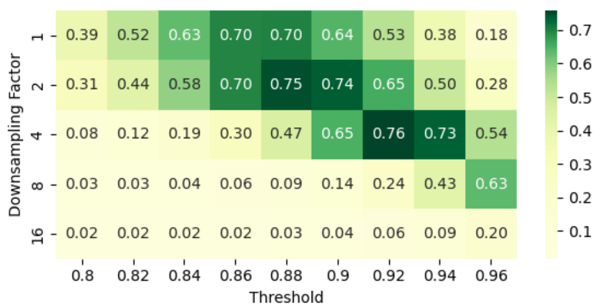


Fig. 4. F1 score by downsampling factor and threshold

We note that this result establishes a baseline for matching accuracy under our specific scenario, including hardware configuration, signal modulation and formats, and channel characteristics. It represents the achievable performance when matching is based solely on signal similarity, without leveraging any prior knowledge of the signal structure, such as preambles, error correction codes, or modulation details.

Reducing the resolution from 12 bits to 8 bits has almost no effect on matching performance. The maximum F1 score remains at 0.71, achieved at the same threshold of 0.87. In more than 40% of the signals required no bit shift, and an additional 40% required the removal of only a single least significant bit. This indicates that most received signals were weak relative to the receiver's dynamic range, and thus, the most significant bits were unused. Consequently, removing these bits causes negligible information loss. Furthermore, the least significant bits of strong signals predominantly contain noise, so their removal has minimal impact on matching performance.

Reducing the resolution to 4 bits introduces noticeable degradation. The maximum F1 score decreases to 0.67, and the optimal similarity threshold shifts slightly lower, to between 0.84 and 0.87. This degradation is attributed to the loss of significant information as more LSBs are discarded. Further reducing the resolution to 2 bits results in a near-complete loss of usable information. The best achievable F1 score drops sharply to 0.17 at a threshold of 0.82, rendering reliable signal matching infeasible at this extreme compression level.

Overall, the results show that reducing the sample resolution to 8 bits is highly effective, achieving a compression ratio of 2 with negligible loss in matching performance, while more aggressive reductions significantly degrade the system's ability to match signals accurately.

*2) Downsampling:* We evaluated the effects of downsampling on matching accuracy by applying downsampling factors of 2, 4, 8, 16, 32, and 64 to the received signals and computing the resulting F1 scores. The results are shown in Figure 4.

As described in the previous section, the original I/Q data sampled at 12 MHz (downsampling factor of 1) achieves a maximum F1 score of 0.71 at a similarity threshold of 0.87.

Interestingly, moderate downsampling improves the matching performance. Specifically, downsampling by a factor of 2 increases the maximum F1 score to 0.74 at a threshold of 0.90, and further downsampling by a factor of 4 increases the F1 score to 0.76 at a threshold of 0.92. While this appears counter-intuitive at first, since reducing the sample rate usually entails information loss, the improvement can be explained by the characteristics of the signals under consideration. SSR Mode S interrogations occupy a significantly narrower bandwidth than the $\pm 6$ MHz captured at the original 12 MHz sampling rate. The effective bandwidth of the signals is around 4 MHz, meaning the GRX3X receiver oversamples the signals substantially, capturing noise from the adjacent bands. Downsampling effectively acts as a low-pass filter, suppressing out-of-band noise and thereby increasing the similarity of true matches, as reflected by the ability to apply stricter (higher) matching thresholds.

The best performing configuration is achieved at a downsampling factor of 4, corresponding to a sampling rate of 3 MHz, with a similarity threshold of 0.92. Under this configuration, the matching recall increases from 66% to 78%, substantially reducing the fraction of missed true matches from 34% to 22%. The precision slightly decreases from 75.8% to 73%, but the overall improvement in recall leads to a higher F1 score. Further increasing the downsampling factor beyond 4 leads to a significant degradation in matching performance. At high downsampling factors, essential signal characteristics are lost due to the corresponding reduction in effective bandwidth, making reliable matching impossible.

These results demonstrate that, although oversampling can improve timestamp accuracy, for the purpose of similarity-based matching, downsampling to a rate close to the true bandwidth of the underlying signals significantly improves performance by enhancing the signal-to-noise ratio.

## IV. DL-Based Compression with Siamese Networks

The DL-based compression method follows a two-step optimization process. First, a DL model is trained to learn a signal embedding using the siamese network architecture and contrastive loss function, shown in Figure 2. In the second step, this embedding is further compressed by reducing the number of bits required to encode each element in the embedding dimensionality, improving storage and transmission efficiency while preserving essential information.

### A. Methodology

*1) Model Architecure:* The deep learning model considered for the compression of the signal is a ResNet architecture, originally designed by O'Shea et al. [9] for modulation classification using raw (IQ) samples. This architecture has demonstrated its effectiveness in learning robust features directly from time-series data, making it well-suited for our task of signal embedding within a Siamese framework.

The ResNet model is characterized by its use of multiple residual blocks, often organized into several residual stacks. These residual connections facilitate the training of deeper networks by mitigating the vanishing gradient problem, allowing the network to learn complex representations. Following the residual stacks, the network typically includes fully connected layers that aggregate the learned features into a final output.

TABLE I
CONSIDERED DL MODEL PARAMETERS

| Downsample | 1 | 2 | 4 | 8 | 16 |
|---|---|---|---|---|---|
| input size | 256 | 128 | 64 | 32 | 16 |
| Residual stacks | 6 | 5 | 4 | 3 | 2 |
| nr.param | 420k | 406k | 392k | 379k | 365k |
| Mflops | 6.85 | 3.54 | 1.88 | 1.05 | 0.64 |

In our case, the output of these fully connected layers serves as the lower-dimensional embedding of the input signal.

In our experiments, we consider different input sizes for the ResNet architecture, which directly impacts the number of preceding residual stacks within the network and consequently its computational demands. We also explore a range of embedding dimensionalities, from 4 to 20 with a step of 2, for each considered input size. This allows us to analyze the trade-off between the degree of signal compression and the resulting signal matching accuracy. An overview of the specific model configurations, including input sizes and the corresponding computational complexity in terms of FLOPs, is provided in Table I. Notably, with the exception of the input size and the final embedding dimensionality, all other design parameters of our ResNet architecture, such as the number of filters and kernel sizes within the convolutional layers and residual blocks, are kept consistent with the original design proposed by O'Shea et al. [9].

The detected signals are normalized to a fixed window size of 256 samples through zero-padding or truncation. For signals exceeding 256 samples, we assume that the initial 256 samples provide sufficient matching information to distinguish between different transmissions. For experiments involving smaller input sizes to the ResNet, we apply a downsampling operation to this 256-sample sequence before feeding it into the network. This allows us to investigate the impact of reduced input data on the performance of our signal matching approach.

*2) Model training and threshold determination:* The considered ResNet models are trained based on the Siamese architecture and contrastive learning [10]. In this learning paradigm, we consider pairs of inputs that either match (come from the same transmission, $Y = 0$) or do not match (belong to different transmissions, $Y = 1$).

The loss for each pair of signals is computed using the contrastive loss function. This function is designed to learn embeddings where signals from the same transmission (positive pairs) have a small Euclidean distance between their representations ($\mathbf{z}_1$ and $\mathbf{z}_2$), while signals from different transmissions have a distance greater than a margin $m$.

The contrastive loss $\mathcal{L}$ for a single pair is defined as:

$$\mathcal{L} = (1-Y)\frac{1}{2}(\|\mathbf{z}_1-\mathbf{z}_2\|_2)^2+Y\frac{1}{2}(\max(0,m-\|\mathbf{z}_1-\mathbf{z}_2\|_2))^2,$$

Here, $Y$ is an indicator variable: $Y = 0$ for signals from the same transmission, and $Y = 1$ for signals from different transmissions. $\mathbf{z}_1$ and $\mathbf{z}_2$ are the embeddings produced by the Siamese network for the input pair, and $m$ represents the margin, which we empirically set to 0.7. All models are trained using 40 epochs with early stopping based on the validation dataset, the Adam optimizer, and a learning rate of 0.0005.

*3) Embedding quantization:* To further compress the learned 32-bit floating-point embeddings from the ResNet model, we employ a uniform quantization scheme. This data-driven approach learns a minimum value ($f_{min}$) and a scaling factor for each feature from the training data to optimally cover the range of the embeddings and reduce the number of bits ($n_{bits}$) required for their representation, potentially with some performance degradation.

The quantization parameters for each feature are derived from the training dataset. For a desired number of bits ($n_{bits}$), the minimum ($f_{min}$) and maximum ($f_{max}$) values are determined along each dimension of the embedding space. The scaling factor for each dimension is then calculated as:

$$\text{scale} = \frac{f_{max} - f_{min}}{2^{n_{bits}} - 1}$$

This `scale` defines the step size between the discrete quantization levels. To quantize an embedding, each element of the embedding vector is transformed into a discrete integer level. This transformation involves first subtracting the learned minimum value ($f_{min}$) for that feature, then dividing by the learned scaling factor (`scale`), and finally rounding the result to the nearest integer. The dequantization process reverses this operation: the integer level is multiplied by the `scale`, and then the $f_{min}$ is added back to obtain an approximation of the original continuous value. By deriving $f_{min}$ and `scale` from the training data, the quantization scheme adapts to the distribution of the learned embeddings, aiming to minimize information loss. The number of bits, $n_{bits}$, serves as a hyper-parameter that regulates the balance between the compression rate and the representational fidelity of the embeddings.

### *B. Impact on Accuracy of Deep Learning-Based Matching*

This section analyzes the influence of the downsampling factor, embedding dimensionality, and embedding quantization on the signal matching performance achieved by the proposed deep learning approach.

*1) Downsampling Factor and Embedding Size:* To evaluate the impact of the downsampling factor and the embedding size on matching accuracy, we first established an optimal decision threshold for each model configuration using the validation dataset. This threshold was determined by maximizing the F1 score, which balances precision and recall. This threshold optimization is considered an integral part of the model training process. The resulting F1 scores for various downsampling factors and embedding dimensions are illustrated in Figure 5.

As depicted in Figure 5, the downsampling factor significantly impacts the maximum F1 score, directly correlating with computational cost. Reducing the downsampling factor from 1 (no downsampling) to 2 drops the F1 score by about 0.05, mainly due to decreased precision. Specifically, recall stays around 0.85 for both factors 1 and 2, but precision falls from roughly 0.67 to 0.60 for a factor of 2. Downsampling factors above 2 lead to substantial F1 score degradation, with low recall and precision.


Fig. 5. F1 score by downsampling factor and number of features

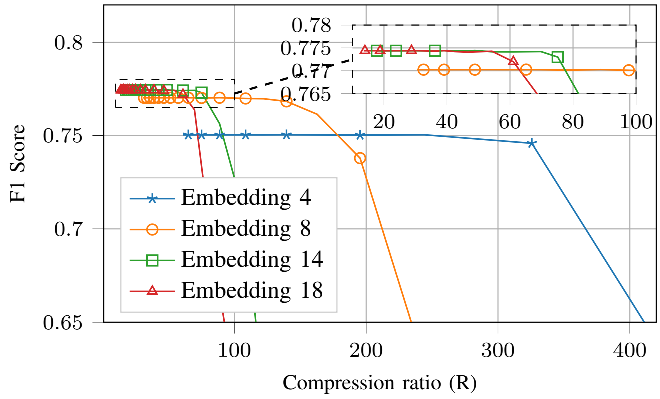


Fig. 6. F1 scores for various compression ratios for different embedding sizes

In contrast, embedding dimensionality has a less pronounced impact on the F1 score, remaining consistent across tested sizes with minor training-related variations. The highest F1 score of 0.77 is achieved for embedding sizes 8, 14, and 18 without downsampling, with corresponding recall values of 0.87, 0.86, and 0.88. For localization, where high recall (detecting potential signal origins) is crucial, the model without downsampling and an embedding size of 8 is optimal. This choice balances identifying potential matches with maintaining a compact embedding size for efficient processing and transmission. Ultimately, the downsampling factor heavily impacts matching performance, primarily due to reduced precision, while embedding size plays a less significant role.

*2) Embedding Quantization:* We investigate the impact of embedding quantization on model performance for embedding sizes of 4, 8, 14, and 18 without downsampling. Recognizing the practical constraints of CPU-based processing, our approach involves quantizing each embedding dimension and subsequently packing the resulting bits into bytes. The compressed data volume in bytes to represent a single compressed signal embedding, denoted as $\tilde{V}(S)$, is determined by the following equation:

$$\tilde{V}(S) = \left\lceil \frac{n_{bits}\ k}{8} \right\rceil \tag{4}$$

where $n_{bits}$ represents the number of bits used to quantize each of the $k$ embedding dimensions. The total compressed data volume for $N$ signals is then $N \cdot \tilde{V}(S)$. In order to calculate the compression ratio $R$ (as defined in Equation 3), we need to obtain the total uncompressed data volume. We approximate the total uncompressed data volume by $N \cdot V_{avg}(S)$, where $V_{avg}(S)$ is the average uncompressed data volume per signal, which is equal to 976 bytes for our evaluation dataset.

As a baseline, unquantized embeddings use 32-bit floating-point numbers (4 bytes/dimension), resulting in initial data volumes of 16, 32, 56, and 72 bytes for embedding sizes 4, 8, 14, and 18, respectively. Their initial compression ratios (relative to 976 bytes uncompressed) are approximately 61, 30.5, 17.5, and 13.5. Figure 6 illustrates the F1 score versus compression ratio for these models. Models with embedding sizes 14 and 18 show the highest initial F1 scores (∼0.774), maintaining this performance until compression ratios exceed 75 and 55, respectively. Beyond these points, F1 score drops significantly due to reduced bits per dimension (around 7 bits).

The embedding size 8 model has a slightly lower initial F1 score (∼0.770) but is more resilient to compression, maintaining higher F1 scores at greater compression ratios than larger embeddings. Performance only declines substantially beyond a compression ratio of approximately 160, corresponding to roughly 6 bits per embedding dimension.

Finally, the embedding size 4 model offers a trade-off: lowest initial F1 score (∼0.75) but stable performance up to a very high compression ratio of around 325. This also corresponds to approximately 6 bits per embedding dimension, compressing each detection to roughly 24 bits (3 bytes).

Overall, our results indicate that all models can be compressed to approximately 6 bits per embedding dimension without substantial F1 score loss, representing a data volume reduction of about 5.3 times compared to the 32-bit floating-point baseline. The highest achievable compression ratio, around 300, is for the model with an embedding size of 4.

## V. Comparison of the Proposed Approaches

The resolution reduction and downsampling approaches offer straightforward lossy compression techniques that reduce data volume either in the amplitude domain (by reducing the number of bits per sample) or in the time/frequency domain (by reducing the number of samples). These methods are agnostic to signal type and do not require prior knowledge of signal structure or modulation. As a result, they are broadly applicable across a wide range of frequencies and signal types. Moreover, their computational complexity is low, making them well-suited for implementation on resource-constrained devices. A further advantage is that the compressed signal retains its original time-domain representation, which may support precise time alignment and e.g. TDOA estimation at the CDP. The main limitation of these methods is that they do not exploit learned or channel-specific features, which limits the achievable matching performance, especially at higher compression rates.

In contrast, the deep learning-based approach employs a Siamese neural network to extract compact embeddings that preserve only the features most relevant for signal matching. These features are learned directly from data, implicitly capturing certain channel-specific characteristics. As a result, this method can offer improved compression-to-accuracy trade-offs. However, because the extracted features are signal-specific, retraining may be required when the system is de-

TABLE II
COMPARISON OF MATCHING PERFORMANCE AND COMPRESSION RATE ACROSS DIFFERENT APPROACHES

| Compression Approach | Compression Ratio | F1 Score | Recall | Precision |
|---|---|---|---|---|
| Downsampling (factor 4) | 4.0 | 0.76 | 0.78 | 0.74 |
| Resolution Reduction (8 bits) | 2.0 | 0.71 | 0.66 | 0.76 |
| Embedding 4, quant. 6 bits | 325.3 | 0.75 | 0.87 | 0.66 |
| Embedding 8, quant. 6 bits | 162.7 | 0.77 | 0.87 | 0.69 |
| Embedding 14, quant. 7 bits | 75.1 | 0.77 | 0.86 | 0.71 |

ployed in new environments with different signal types or propagation conditions. Furthermore, this approach requires sufficient computational resources at the receiver to generate embeddings in real time.

Table II summarizes the matching performance and compression ratio achieved by the different methods. The results confirm that the deep learning-based approach achieves much higher compression rates at slightly higher F1 scores, particularly due to strong recall. However, the simplicity and robustness of downsampling and resolution reduction make them attractive for scenarios with strict resource constraints and limited prior knowledge about the signals.

Overall, resolution reduction and downsampling offer simple and efficient solutions suitable for diverse signal types and low-power hardware, whereas the Siamese network-based approach provides higher compression efficiency and recall in environments where training and deployment of neural models are feasible.

## VI. Related Work

Signal compression techniques have been widely studied to reduce data volume while preserving essential information, broadly falling into lossless and lossy categories.

Lossless compression, such as Huffman coding and Lempel-Ziv-Welch (LZW) [11], allows for perfect data reconstruction. While crucial where no information loss is permissible (e.g., Rajendran et al.'s ElectroSense [1]), these methods typically achieve limited compression ratios, especially for continuous-valued signals like IQ data.

Conversely, lossy compression techniques achieve higher compression ratios by selectively discarding information, aiming to minimize the impact on the reconstructed signal. Common lossy methods for radio signals include downsampling, which reduces temporal resolution, and quantization, which decreases amplitude precision [12]. The selection of these parameters directly governs the trade-off between compression rate and signal fidelity. More advanced lossy techniques leverage signal statistics and redundancies. Transform coding methods, like the Discrete Cosine Transform and Discrete Wavelet Transform, decompose the signal to remove less significant components. Zeng et al.'s FlexSpec framework [13], based on the Walsh-Hadamard Transform, is a notable example of transform-based approaches achieving superior compression.

The new paradigm of deep learning and machine learning has shown promising results in image compression, and its application to RF signal compression for reconstruction has been explored in works like [14]. More recent research has investigated task-specific learned compression, where the compressed representation is optimized for downstream tasks such as classification or detection [15]. This paradigm shift towards task-aware compression, potentially enabling reduced communication overhead and improved efficiency in distributed systems, motivates our investigation into learning compressed embeddings specifically designed for efficient signal matching in a collaborative localization scenario.

## References

[1] S. Rajendran *et al.*, "Electrosense: Open and big spectrum data," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 210–217, 2017.

[2] Z. Liao, Y. Gao, S. Yang, and K. Fan, "Based on i/q signal and ai algorithms to identify the unique fingerprint of each mobile phone hardware device," in *2024 8th International Symposium on Computer Science and Intelligent Control (ISCSIC)*, pp. 242–244, 2024.

[3] HawkEye 360 Inc., "RFGeo: Space-Based RF Signal Detection and Geolocation." https://www.he360.com, May 2024. Product Brief, Public Release 0424.

[4] D.-H. Kim, K. Lee, M.-Y. Park, and J. Lim, "Uav-based localization scheme for battlefield environments," in *MILCOM 2013 - 2013 IEEE Military Communications Conference*, pp. 562–567, 2013.

[5] M. Schäfer, Y. Lizarribar, G. Bovet, and D. Verbruggen, "Let's take this upstairs: Localizing ground transmitters with high-altitude balloons," IEEE, 2024.

[6] S. Ghorpade, M. Zennaro, and B. Chaudhari, "Survey of localization for internet of things nodes: Approaches, challenges and open issues," *Future Internet*, vol. 13, no. 8, 2021.

[7] D. Verbruggen, H. Sallouha, and S. Pollin, "Distributed deep learning for modulation classification in 6g cell-free wireless networks," in *2024 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 114–119, 2024.

[8] I. Melekhov, J. Kannala, and E. Rahtu, "Siamese network features for image matching," *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 378–383, 2016.

[9] T. O'Shea, T. Roy, and T. Clancy, "Over the air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. PP, 12 2017.

[10] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 539–546 vol. 1, 2005.

[11] K. Sayood, *Introduction to Data Compression, Fourth Edition*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 4th ed., 2012.

[12] A. Oppenheim, *Discrete-Time Signal Processing*. Pearson education signal processing series, Pearson Education, 1999.

[13] Y. Zeng, R. Calvo-Palomino, D. Giustiniano, G. Bovet, and S. Banerjee, "Adaptive uplink data compression in spectrum crowdsensing systems," *IEEE/ACM Transactions on Networking*, vol. 31, no. 5, pp. 2207–2221, 2023.

[14] A. El Sayed, M. Ruiz, H. Harb, and L. Velasco, "Deep learning-based adaptive compression and anomaly detection for smart b5g use cases operation," *Sensors*, vol. 23, no. 2, 2023.

[15] A. Rodriguez, Y. Kaasaragadda, and S. Kokalj-Filipovic, "Deep-learned compression for radio-frequency signal classification," in *2024 IEEE International Symposium on Information Theory Workshops (ISIT-W)*, pp. 1–6, 2024.