Leveraging Foundation Models for Zero-Shot IoT Sensing

Paper #939

Abstract. Zero-shot learning (ZSL) aims to classify data of unseen classes with the help of semantic information. Foundation models (FMs) trained on web-scale data have shown impressive ZSL capability in natural language processing and visual understanding. However, leveraging FMs' generalized knowledge for zero-shot Internet of Things (IoT) sensing using signals such as mmWave, IMU, and Wi-Fi has not been fully investigated. In this work, we align the IoT data embeddings with the semantic embeddings generated by an FM's text encoder for zero-shot IoT sensing. To utilize the physics principles governing the generation of IoT sensor signals to derive more effective prompts for semantic embedding extraction, we propose to use cross-attention to combine a learnable soft prompt that is optimized automatically on training data and an auxiliary hard prompt that encodes domain knowledge of the IoT sensing task. To address the problem of IoT embeddings biasing to seen classes due to the lack of unseen class data during training, we propose using data augmentation to synthesize unseen class IoT data for fine-tuning the IoT feature extractor and embedding projector. We evaluate our approach on multiple IoT sensing tasks. Experiment results show that our approach achieves superior open-set detection and generalized zero-shot learning performance compared with various baselines. We will open-source our code upon acceptance of this paper.

1 Introduction

With the advancement of edge hardware accelerators, deep learning is increasingly employed for IoT sensing tasks on edge devices, such as deep learning-based Wi-Fi human sensing [28], sound event detection [26], and activity recognition using motion sensor [22]. However, although deep learning models show excellent performance in classifying samples from a set of *seen* classes that are included in the training dataset, identifying and classifying data samples from *unseen* classes using deep models trained under the supervised setting are challenging. To address this, an intuitive solution is to include as many classes as possible during training. However, unlike images, text, and audio, which humans can easily interpret, IoT data often lacks readability and requires costly labeling processes. Thus, IoT datasets usually contain a limited number of classes. For example, most inertial measurement unit (IMU)-based activity recognition datasets contain fewer than 20 activity classes [22]. In comparison, the ImageNet contains 21,814 classes.

Zero-shot learning (ZSL) [16] is a promising learning paradigm to address the aforementioned challenge. ZSL classifies data from unseen classes with the help of semantic information that transfers knowledge from seen classes to unseen ones. Previous studies rely on manually-engineered attributes as semantic information for zero-shot IoT sensing [23, 15], which are labor-intensive to design and difficult to scale to complex datasets. Some works [12, 24, 26, 22] employ learned semantic spaces built from word vectors of class labels or descriptions, which are extracted by word representation models such as Word2Vec [12, 24], BERT [26], and GloVe [22]. However, the word vectors generated by capturing the semantic relationships between words based on their contexts in text corpus may contain task-irrelevant noise, causing a semantic gap between the IoT data and word embeddings. The work in [22] constructs visual semantic space using human activity videos for zero-shot IMU-based human activity recognition, which may raise privacy concerns. In this work, we aim to explore using foundation models, which are considered to have a generalized understanding of the world acquired from diverse and extensive training data, to generate more effective and contextually relevant semantic embeddings for zero-shot IoT sensing.

Foundation models (FMs) are large-scale general deep learning models pre-trained on vast data that serve as the foundation for various downstream tasks [31]. FMs trained on extensive text corpora exhibit remarkable generalizability to a broad spectrum of new tasks, e.g., passing exams [1], code generation [14], and language translation [17]. Large vision-language FMs embed images with language inputs in a joint semantic space using hundreds of millions of image and text pairs, which achieve impressive zero-shot transferability to downstream tasks such as image recognition on unseen datasets [18, 21]. Inspired by this, recent research aligns different audio, depth, infrared, and IMU data with the vision [8] and language [34] modalities, aiming to extend the zero-shot capability of the vision-language FMs to multiple modalities. These multi-modal FMs demonstrate excellent performance in new tasks such as zero-shot classification on unobserved data pairs.

Recent research aligns IoT sensor signals to textual semantic features generated by FMs for zero-shot IoT sensing. For example, the work in [33] jointly aligns FM's textual embeddings with multiple IoT sensor signals, including video, LiDAR, and mmWave in a unified semantic space. It demonstrates FM's ZSL capability in recognizing unseen class IoT data. However, this work is built upon large quantities of multi-modal data samples where all modalities are presented together, which are expensive to acquire and impractical if new modalities are to be added to the semantic space. EdgeFM [27] leverages FMs for zero-shot sensing on resource-limited edge devices. However, EdgeFM only supports the existing modalities of FMs, including video, images, and audio.

This work aims to leverage FMs' generalized knowledge for zero-shot IoT sensing based on mmWave, IMU, and Wi-Fi signals by aligning the IoT data embeddings with the semantic embeddings generated by an FM's text encoder. However, connecting IoT sensor signals with semantic embeddings for effective ZSL is non-trivial. First, IoT sensor signals typically follow certain physics principles, which are strong supervision for effective prompt engineering to gen-

erate robust semantic embeddings. To address this, we employ cross-attention to combine a learnable soft prompt that is optimized automatically using training data and an auxiliary hard prompt that encodes domain knowledge. Second, given that the training only involves seen class IoT data, the ZSL model is easily biased to seen classes. To address the bias problem, we propose using data augmentation to synthesize unseen class IoT data for fine-tuning our IoT feature extractor and embedding projector. Our approach works as follows. We apply prompt engineering on class labels and use an FM's text encoder to extract their semantic embeddings as class prototype representations. Meanwhile, we use an IoT feature extractor to extract features from IoT sensor signals followed by an IoT embedding projector to project the features to the semantic space. During model training, we use contrastive learning to align the class prototypes and IoT embeddings. During zero-shot classification, we conduct open-set detection to identify data of unseen classes and use FM to do zero-shot learning. We evaluate our approach on multiple datasets including MM-Fi (mmWave, Wi-Fi), USC-HAD (IMU), and PAMAP2 (IMU). Our approach achieves superior performance in open-set detection and generalized zero-shot learning compared with various baselines. This paper's contributions are summarized as follows.

- To leverage the domain knowledge for zero-shot IoT sensing, we propose using cross-attention to combine a learnable soft prompt and an auxiliary hard prompt for effective prompt engineering.
- To eliminate the problem of unseen class IoT embeddings biasing to seen class embeddings, we employ data augmentation and open-set detection for generalized zero-shot IoT sensing.
- We evaluate our approach on multiple IoT datasets with IMU, mmWave, and Wi-Fi data. The results demonstrate that our approach outperforms various baselines in both open-set detection and generalized zero-shot learning.

2 Background and Related Work

Foundation Models (FMs) are general deep learning models that are pre-trained on massive amount of data to support various downstream tasks such as chatbot [17, 1] and image recognition [18]. FMs are extensively studied in natural language processing and computer vision [31]. For example, ChatGPT is fine-tuned for conversational tasks from the generative pre-trained transformer-based language foundation models, e.g., GPT-3.5 [3] and GPT-4 [1]. CLIP [18] is a vision-language foundation model that trains an image encoder and a text encoder jointly aiming to predict the correct image-text pairs. CLIP achieves zero-shot transferability to unseen image recognition tasks after training on 400 million image-text pairs. More recently, FMs are applied to other modalities, including audio, depth, IMU, and infrared [34, 8]. These multi-modal FMs use transformer-based encoders to extract embeddings of different modalities. Then, a joint embedding space is learned via contrastive learning that aligns the embeddings of different modalities with the embedding of a "binding" modality, i.e., vision or language. The learned joint embeddings can be used for various tasks such as cross-modal retrieval, cross-modal generation, and composing modalities with arithmetic. The multi-modal FMs trained on different cross-modal data pairs, e.g., (image, text) and (image audio), can implicitly associate unobserved data pairs, e.g., (audio, text), which is defined as *emergent zero-shot* classification. Different from the existing works that focus on FMs' zero-shot transferability on unseen datasets [18, 21] and unobserved data pairs [8, 34], our work aims to investigate the zero-shot capability of FM characterized by the performance of generalizing to unseen object categories in classification tasks, which represents a more practical scenario in IoT sensing tasks.

Zero-Shot Learning (ZSL) aims to classify data of unseen classes with the help of semantic information containing knowledge about both seen and unseen classes [16]. Traditional ZSL methods focus on classifying data into unseen classes. A more realistic setting is the generalized zero-shot learning (GZSL) that classifies data samples of seen and unseen classes simultaneously. GZSL methods can be categorized as *embedding-based* and *generative-based*. Embedding-based GSZL [2, 11] learns a projection function from data feature space to the semantic space. The goal is to map the data embeddings belonging to the same class to the ground-truth label in the semantic space. The embedding-based GZSL is easy to implement but is usually biased towards seen classes due to a lack of unseen class data features during training. Generative-based GZSL [25, 5] trains a model to generate synthetic features of unseen class data based on features of seen class data and semantic information of both seen and unseen classes. The generated features of unseen class data can be used to perform supervised learning, where a model is trained to classify data samples of both seen and unseen classes. The generative-based GZSL alleviates the biasing problem via synthesizing features of unseen classes. However, the generative models are unstable in training and susceptible to model collapse issue.

Zero-Shot IoT Sensing. Some works use hand-crafted attributes such as the movement of body or related objects and environment to construct semantic information for zero-shot IoT sensing [23, 15], which is labour-intensive and less scalable to large complex datasets. To circumvent manual attribute engineering processes, some studies utilize word vectors, which are numerical representations of words in a continuous vector space extracted by word representation models such as Word2Vec [12, 24], BERT[26], and GloVe [22], to construct semantic space. The word representation models are trained on text corpus to encode semantic relationships between words as vectors. However, these vectors may include task-irrelevant noise and may not directly suit the specific IoT sensing task. The work in [22] proposes to construct visual semantic space using videos of human activities for IMU-based zero-shot human activity recognition, which is shown to outperform the word vector semantic space. However, collecting videos of human raises privacy concerns. A recent work [33] jointly aligns multiple IoT data embeddings, including video, LiDAR, and mmWave, with text embeddings extracted from a vision-language FM, CLIP [18], for human activity recognition. With the unified semantic space, not only actions of seen classes can be identified but also the actions of unseen classes can be recognized by the closet textual embedding in the semantic space. However, this approach requires joint training on a self-collected multi-modal aligned dataset, which has limited usability in reality if additional sensor modalities are to be added to the system. EdgeFM [27] is an edge-cloud cooperative system that achieves zero-shot recognition capability on resource-limited edge devices by leveraging FMs on the cloud for selective knowledge query. However, the zero-shot capability is only demonstrated on the existing modalities of FMs, including video, images, and audio. To this end, the potential of leveraging FMs' generalized knowledge for zero-shot sensing using IoT signals such as mmWave, IMU, and Wi-Fi, which are not covered by the supported modalities of existing FMs, is still under-explored.

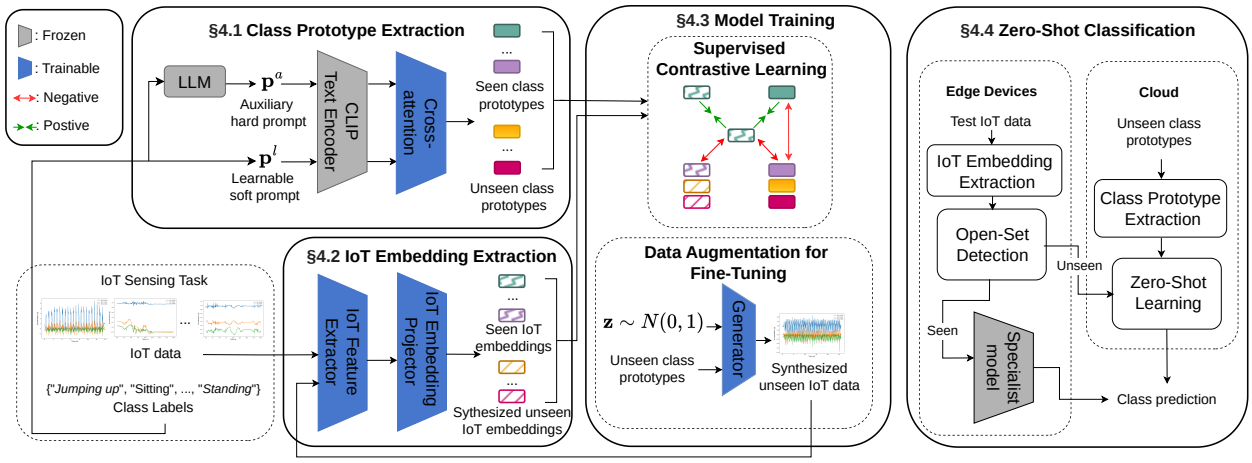

Figure 1. Approach overview. In §4.1, we employ cross-attention to combine the learnable soft prompt and auxiliary hard prompt to generate class prototypes. In §4.2, we use a feature extractor followed by an embedding projector to generate IoT embeddings. During model training in §4.3, we use supervised contrastive learning to align the class prototypes and IoT embeddings. We then use data augmentation to synthesize unseen class data for fine-tuning the IoT feature extractor and embedding projector. During zero-shot classification in §4.4, we first extract the IoT embeddings of input data for open-set detection. Then, the samples detected as seen class will be classified by the specialist model on edge devices. The samples detected as unseen will be uploaded to the cloud for zero-shot learning by the foundation model.

3 Problem Formulation

We target a deep learning-powered IoT sensing task enabled by an edge-cloud cooperative system that contains the following components.

- **Edge Devices** host a small-scale *specialist* deep neural network (DNN) $f(\cdot)$, which can classify a limited set of seen classes $\mathcal{S} = \{c_i^s\}_{i=1}^{N_s}$. The $f(\cdot)$ is trained under supervised setting using a seen train set $D^s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{N_{train}} \in \mathcal{X} \times \mathcal{S}$, where $\mathbf{x}_i^s \in \mathbb{R}^d$ is the raw IoT data, y_i^s is the ground-truth label, and $\mathcal{X}$ denotes the IoT data space. The input test data may include not only samples from known seen classes but also samples from novel unseen classes, denoted by $D^{test} = \{\mathbf{x}_i^{\text{test}}\}_{i=1}^{N_{test}} \in \mathcal{X}$.
- **Cloud Server** runs a large *foundation* model (FM) $\Phi(\cdot)$, which possesses general knowledge learned from web-scale training data and has the potential of zero-shot classification on unseen class data. The cloud maintains a list of interested unseen classes outside the set of seen classes S, denoted by $\mathcal{U} = \{c_i^u\}_{i=1}^{N_u}$, where $\mathcal{S} \cap \mathcal{U} = \emptyset$. Note that $\mathcal{U}$ can be specified by users or include the commonly seen classes in the IoT sensing task.

The primary goal is to effectively (1) detect data sample of unseen classes $\mathbf{x}^u$ from D^{test} fed to the specialist DNN $f(\cdot)$ on the local edge devices and then (2) leverage the cloud's FM $\Phi(\cdot)$ to perform zero-shot classification by assigning correct label $y^u \in U$ for the detected data of unseen classes. Note that an alternative way is to upload all the data to the cloud's FM for classification. However, in §5.5, we will demonstrate that having the detection step alleviates the GZSL biasing problem. Such a cooperative system is common in IoT applications such as healthcare monitoring, autonomous driving, and AR/VR gaming. To achieve the goal, given an incoming IoT data sample, we first extract its IoT embedding and conduct open-set detection to determine whether the sample belongs to a seen class or unseen class, both on the edge. If it is detected as a seen class sample, we use the local specialist DNN to give prediction. Otherwise, if the sample is considered as unseen class data, we upload it to the cloud's FM for zero-shot learning.

4 Methodology

The overview of our approach is shown in Fig. 1, which consists of the class prototype extraction, IoT embedding extraction, model training, and zero-shot classification modules introduced in detail as follows.

4.1 Class Prototype Extraction

In ZSL, *class prototypes* encapsulate the essential characteristics of each class in the semantic space. During inference, the similarity between the data embedding and each class prototype is measured to determine the sample's class. In this work, we utilize the text encoder of the vision-language FM, CLIP [18], to extract class prototypes from task-specific hints, namely *prompt*. Prompt can be engineered in the form of *hard prompt*, which is natural language instructions, or *soft prompt*, which is continuous, learnable vector representations. The hard prompt can integrate domain expert knowledge but needs to be manually engineered. The soft prompt can be automatically fine-tuned to adapt to various tasks but is not human-interpretable. To combine the advantages of both, we propose to use cross-attention to fuse the soft and hard prompts to generate effective and comprehensive class prototypes.

Learnable Soft Prompt. The default prompt in CLIP is constructed by plugging the class name into a pre-defined prompt template, i.e., "a photo of {class name}". However, such a fixed prompt is difficult to adapt to downstream tasks. Because CLIP's default prompts tend to gather together in the semantic space, which is unfavorable for data-text alignment [32]. To address this and avoid laborious manual prompt engineering, we learn a soft prompt end-to-end from training data, aiming to align the text embedding with IoT data embedding. We follow the work in [32] and put the class token in the middle of the prompt. For each class c, the learnable soft prompt fed to the pre-trained CLIP's text encoder $\Phi_{\text{text}}(\cdot)$ is represented by $\mathbf{p}^l(c) = \oplus(\mathbf{l}_1, ..., \text{CLIPtokenizer}[c], ..., \mathbf{l}_M)$, where $\oplus$ is the concatenation operation, $\mathbf{l}_i$, $(i = 1 \cdots M)$ denotes the i-th learnable token vector, and c is the class name, e.g., "walking forward".

The learnable prompt is optimized over the training data using the loss defined shortly in Eq. 1. The extracted learnable text embedding $\mathbf{t}^l(c) = \Phi_{\text{text}}(\mathbf{p}^l(c))$ has the same dimension as the IoT data embedding. The learned prompt token vectors $\mathbf{l}_i$, $(i = 1 \cdots M)$ are shared for all classes, which are task-specific.

Auxiliary Hard Prompt. The learnable soft prompt provides task-specific context by aligning the text embedding with the IoT data embedding in the semantic space. Meanwhile, IoT data is usually characterized by certain physics principles, which can be leveraged as a strong supervision for prompt crafting. For example, Fig. 2 shows that the data samples of two classes in the USC-HAD [30] dataset exhibit different patterns, which can be utilized to easily distinguish the data of the two classes. To leverage the physics principles governing the generation of the IoT sensor signals, we further use a hard prompt to give auxiliary class-specific information for constructing semantic embeddings. To automate the process, we use a state-of-the-art large language model (LLM), GPT-3.5 [3], to generate class-conditional descriptive text and fine-tune the text manually. For an IoT sensing task, we first feed the list of all classes to the LLM. Then, for each class c, we query the LLM: "What are the important attributes and features to distinguish class c from all the other classes?". We then tokenized the answer to derive the auxiliary hard prompt $\mathbf{p}^a(c)$, which will be fed to CLIP's text encoder to derive the auxiliary text embedding $\mathbf{t}^a(c) = \Phi_{\text{text}}(\mathbf{p}^a(c))$, which has the same dimension as the IoT data embeddings. Fig. 2 illustrates some example answers generated by GPT.

Cross-Attention for Combining Prompts. To leverage the advantages of both the learnable soft prompt and auxiliary hard prompt, we combine the text embeddings of the two prompts using the cross-attention [4], which is an attention mechanism for fusing two different sequences. In particular, we set $\mathbf{t}^a$ as the key input, denoted by $\mathbf{K}$, and $\mathbf{t}^l$ as the query and value inputs, denoted by $\mathbf{Q}$ and $\mathbf{V}$, respectively. The idea is to compute the attention weights between the query and key inputs, which embed the useful class-specific context information from $\mathbf{t}^a$, and then use the weights to aggregate the value input $\mathbf{t}^l$. Specifically, $\mathbf{Q} = \rho_{\mathbf{Q}}(\mathbf{t}^l)$, $\mathbf{K} = \rho_{\mathbf{K}}(\mathbf{t}^a)$, and $\mathbf{V} = \rho_{\mathbf{V}}(\mathbf{t}^l)$, where $\rho_m(\cdot), (m \in \{\mathbf{Q}, \mathbf{K}, \mathbf{V}\})$ is a single-layer fully-connected neural network. $\rho_m(\cdot)$ is optimized over the training data on the loss defined shortly in Eq. 1. The attention weights are computed by $\mathbf{A} = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_{\mathbf{K}}}})$, where $d_{\mathbf{K}}$ is the dimension of $\mathbf{K}$. The output embedding, denoted by $\mathbf{t} = \mathbf{A}\mathbf{V}$, is the class prototype. The semantic space is formed by the set of all class prototypes $\mathcal{T} = \{\mathbf{t}(c) \mid c \in \mathcal{S} \cup \mathcal{U}\}$.

4.2 IoT Embedding Extraction

For each input IoT data $\mathbf{x}_i$, we first use a feature extractor $\mu(\cdot)$ to extract its features $\mathbf{h}_i = \mu(\mathbf{x}_i)$. The feature extractor $\mu(\cdot)$ can be a commonly-used encoder like CNN, ResNet, and Transformer, which is decided by the IoT sensing modality. Then, we use an embedding projector $g(\cdot)$ aiming to project the IoT features $\mathbf{h}_i$ into the semantic space and derive the IoT embeddings $\mathbf{e}_i = g(\mathbf{h}_i)$.

4.3 Model Training

We freeze the text encoder of CLIP $\Phi_{\text{text}}(\cdot)$ and conduct model training under the supervised contrastive learning strategy, which trains the models to distinguish between similar (positive) and dissimilar (negative) data sample pairs. This allows us to learn effective representations by maximizing the distance between different classes and minimizing the distance within the same class [10].

Supervised Contrastive Learning. First, we jointly train the learnable soft prompt $\mathbf{p}^l$, $\rho_k(\cdot)$ in the cross-attention module, IoT feature extractor $\mu(\cdot)$, and IoT embedding projector $g(\cdot)$ on the seen train set D^s using a supervised contrastive loss. Within a batch of randomly sampled data $\{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{N_B}$ from D^s, the positive pairs contain (1) two IoT data samples belonging to the same class and (2) an IoT data sample and its class label text. The negative pairs consist of (1) two IoT data samples belonging to different classes; (2) an IoT data sample and a class label other than its own; and (3) two different class labels. The loss pulls together embeddings of positive pairs while pushing away the embeddings of negative pairs. Let $i \in I \equiv \{1 \cdots N_B\}$ be the index of the data in a train batch. Let N_T represent the number of distinct classes in the batch and $j \in J \equiv \{1 \cdots N_T\}$ be the index of distinct classes. We define the supervised contrastive loss as:

$$\begin{aligned}\mathcal{L} &= \sum_{i \in I} \mathcal{L}_i \\ &= \sum_{i \in I} \Bigg(\frac{-1}{|P(i)+1|} \cdot \Bigg(\sum_{p \in P(i)} \mathbf{e}_i \cdot \mathbf{e}_p/\tau + \mathbf{e}_i \cdot \mathbf{t}_j/\tau \Bigg) \\ &\quad + \log \Bigg(\sum_{a \in A(i)} \exp\left(\mathbf{e}_i \cdot \mathbf{e}_a/\tau\right) \\ &\quad + \sum_{n \in N(j)} \left(\exp\left(\mathbf{e}_i \cdot \mathbf{t}_n/\tau\right) + \exp\left(\mathbf{t}_j \cdot \mathbf{t}_n/\tau\right)\right) \Bigg) \Bigg),\end{aligned} \tag{1}$$

where, for each IoT data sample $\mathbf{x}_i$, $\mathbf{e}_i$ is its IoT embedding, $\mathbf{t}_j$ is its corresponding class prototype, $A(i) \equiv I \backslash \{i\}$, $N(j) \equiv J \backslash \{j\}$, $P(i) \equiv \{p \in A(i) : y_p = y_i\}$, and τ is a positive temperature scalar.

Data Augmentation for Fine-Tuning. During the model training on D^s described in the previous paragraph, the IoT feature extractor and embedding projector are only trained on data of seen classes. Consequently, the IoT embeddings of unseen classes are biased to the seen ones, and thus, the data samples of unseen classes may easily be classified as seen ones. To address this bias problem, we propose to train a generative model under the Generative Adversarial Network (GAN) setting to synthesize data samples of unseen classes. The goal is to derive more robust IoT embeddings by fine-tuning the IoT feature extractor and embedding projector using the augmented unseen class data. Given the train set D^s, we learn a conditional generator $G(\cdot)$ that takes as input the class prototype $\mathbf{t}(y)$ and a random Gaussian noise vector $\mathbf{z}$, aiming to output the synthesized IoT data $\tilde{\mathbf{x}} \in \mathcal{X}$ of class y. Note that the class prototype $\mathbf{t}(y)$ is generated by the frozen text branch. To achieve this goal, we modify the loss in [25] and define the data augmentation loss as: $\mathcal{L}_{\text{DA}} = \mathcal{L}_{\text{WGAN}} + \mathcal{L}_{\text{CLS}}$. Specifically, $\mathcal{L}_{\text{WGAN}} = \mathbb{E}[D(\mathbf{x}, \mathbf{t}(y))] - \mathbb{E}[D(\tilde{\mathbf{x}}, \mathbf{t}(y))] - \xi\mathbb{E}\left[\left(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}}, \mathbf{t}(y))\|_2 - 1\right)^2\right]$, where $D(\cdot)$ is the discriminator, $\mathbf{x}$ is the real data, $\tilde{\mathbf{x}} = G(\mathbf{z}, \mathbf{t})$ is the generated data, $\hat{\mathbf{x}} = \alpha\mathbf{x} + (1-\alpha)\tilde{\mathbf{x}}$ with $\alpha \sim U(0,1)$, and ξ is the penalty coefficient. $\mathcal{L}_{\text{CLS}} = -\mathbb{E}[\log \Pr(y \mid \tilde{\mathbf{x}}; \theta)]$ is the classification loss computed by a linear softmax classifier parameterized by θ that is pre-trained on D^s. The generator is trained by optimizing the objective: $\min_G \max_D \mathcal{L}_{\text{DA}}$. The generator $G(\cdot)$ aims to fool the discriminator $D(\cdot)$ by generating IoT data that are considered as real, while the discriminator $D(\cdot)$ aims to distinguish real data from the synthesized one. After $G(\cdot)$ is trained, we use it to generate a synthesized train set of unseen classes $D^{aug} = \{(\tilde{\mathbf{x}}_i^u, y_i^u)\}_{i=1}^{N_{aug}} \in \mathcal{X} \times \mathcal{U}$ and use it to fine-tune the IoT feature extractor and embedding projector using the loss defined in Eq. 1.

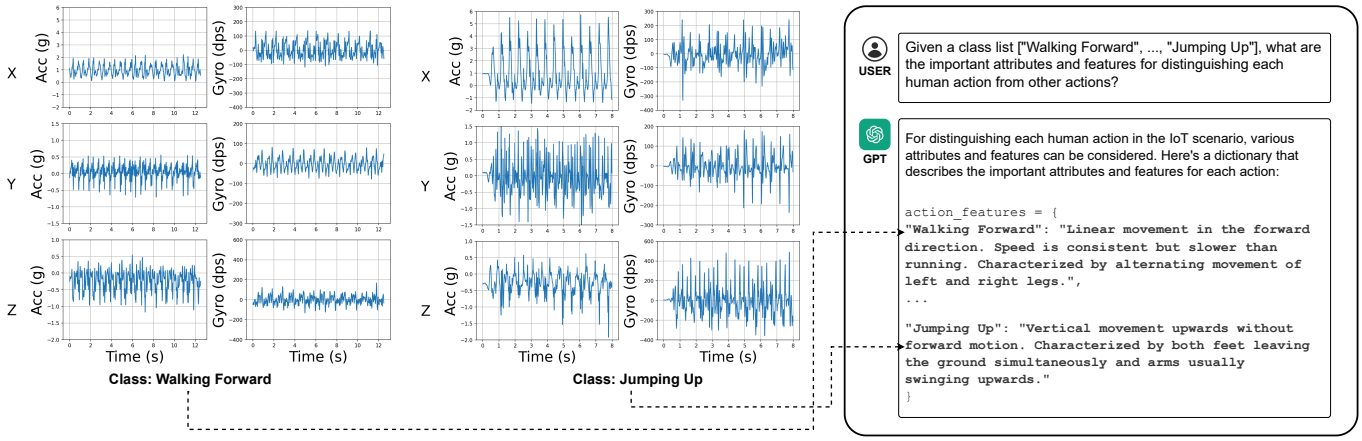

Figure 2. Visualization of two data samples from an IMU activity recognition dataset [30]. X-axis, Y-axis, and Z-axis are aligned with gravity, walking direction, and perpendicular direction to walking direction, respectively. The data sample of class "walking forward" has around zero values in the Y-axis of the accelerometer reading, indicating a constant speed along the walking direction. The data sample of "jumping up" has large positive values in the X-axis of the accelerometer reading, indicating vertical movements upwards. The patterns of the samples are characterized by the generated descriptive text.

4.4 Zero-Shot Classification

As outlined in §3, we decompose the zero-shot classification into two steps. The first step is to identify unseen class data, i.e., *open-set detection*, on the local edge devices. The second step is to conduct *zero-shot learning* using the FM located on the cloud.

Open-Set Detection is a binary classification problem to identify whether a data sample belongs to seen or unseen classes. Inspired by the work in [20], we develop a distance-based method for open-set detection. First, based on a train set D^s, we cluster the IoT embeddings of all data samples based on their classes and denote these clusters by $\{E_i^s\}_{i=1}^{N_s}$, where N_s is the number of seen classes. Each class cluster E_i^s, $(i = 1 \cdots N_s)$ consists of a set of IoT embeddings $\{\mathbf{e}_{i,j}\}_{j=1}^{N_i}$, where N_i is the number of data samples in E_i^s. For an input data sample $\mathbf{x}^{\text{test}} \in D^{test}$, which may belong to either seen or unseen classes, we compute the Euclidean distances between its IoT embedding $\mathbf{e}^{\text{test}}$ and the IoT embeddings in each class cluster E_i^s as $d_{i,j} = \|\mathbf{e}^{\text{test}} - \mathbf{e}_{i,j}\|_2, \mathbf{e}_{i,j} \in E_i^s$. We sort $d_{i,j}$ to obtain the k-th smallest distance for each cluster, denoted by $d_i^{(k)}$. We use a simple threshold-based criterion on $d_i^{(k)}$ to determine whether the input sample belongs to seen or unseen classes:

$$Q(\mathbf{x}^{\text{test}}; k) = \sum_{i}^{N_s} \mathbf{1}|d_i^{(k)} \leq \lambda_i|, \tag{2}$$

$$S_{\text{open}}(\mathbf{x}^{\text{test}}) = \begin{cases} \text{Unseen}, Q = 0 \\ \text{Seen}, Q \geq 1 \end{cases}, \tag{3}$$

where $\mathbf{1}|\cdot|$ is the indicator function and λ_i is the class-specific distance threshold that is decided empirically by correctly associating a high fraction of seen class data samples to their corresponding class clusters using a validation set. If the value of Q equals 0, it indicates that the test sample does not belong to any seen class clusters and should be considered as unseen. If the value of $Q \geq 1$, it means that the test sample can be associated with at least one seen class cluster and should be considered as seen.

Zero-Shot Learning. For a detected "unseen" test sample $\mathbf{x}^{\text{det}}$ with IoT embedding $\mathbf{e}^{\text{det}}$, we upload it to the cloud's FM for zero-shot learning. Specifically, we compute the similarity scores, i.e., dot product, between $\mathbf{e}^{\text{det}}$ and all the class prototypes in $\{\mathbf{t}(c_i^u), c_i^u \in U\}$. Then, the class with the highest similarity score is the predicted label $\hat{y}^{\text{det}}$ for $\mathbf{x}^{\text{det}}$:

$$\hat{y}^{\text{det}} = \underset{c_i^u \in U}{\text{argmax}}(\mathbf{e}^{\text{det}} \cdot \mathbf{t}(c_i^u)), \tag{4}$$

5 Evaluation

5.1 Datasets

USC-HAD [30]. The USC Human Activity Dataset is an IMU dataset of 12 different daily activities collected from 14 human subjects. By sampling it with a 1.28-second window and a 50% overlap rate, we obtain 42,708 samples, each consisting of 1.28-second 3-axis accelerometer and 3-axis gyroscope readings. We divide the activities into 9 seen classes and 3 unseen classes.

PAMAP2 [19]. The Physical Activity Monitoring Dataset consists of 12 daily activities by collecting IMU data following a protocol from 9 subjects. We divide the activities into 9 seen classes and 3 unseen classes. We adopt a 1.71-second sliding window with a 10% overlap rate to extract 4,178 samples.

MM-Fi [29]. The MM-Fi dataset is a multi-modal wireless human sensing dataset consisting of 1,080 consecutive sequences with over 320k synchronized frames from five sensing modalities. We adopt the Wi-Fi and filtered mmWave sub-datasets in environment 4 from the MM-Fi. We resample mmWave and Wi-Fi data using 1-second and 0.6-second sliding windows with 10% overlap, respectively, yielding 27,337 mmWave samples and 8,748 Wi-Fi samples. For both mmWave and Wi-Fi, we split the 27 activity classes into 22 seen classes and 5 unseen classes.

We adopt a K-fold evaluation strategy to split each dataset into seen classes and unseen classes. For USC-HAD and PAMAP2, we randomly select 3 unseen classes in each of K=4 folds. For mmWave and Wi-Fi, we randomly select 5 unseen classes in K=5 folds. For the seen class data samples, we divide them into training, validation, and test sets with a ratio of 8:1:1. The validation set is used to tune the parameters like λ_i. The test set has equal number of seen class and unseen class data samples.

5.2 Implementation Details

We use Pytorch to implement our approach. We use Vision Transformer as the IoT Feature Extractor for all modalities. For class prototype extraction, we use GPT-3.5 to generate auxiliary hard prompts. The text encoder is adopted from the frozen CLIP text encoder with ViT-B/16 backbone. The supervised contrastive loss's temperature parameter τ is set to 0.2. For data augmentation, the random Gaussian noise vector $\mathbf{z}$ follows a normal distribution $\mathcal{N} \sim (0, 1)$, and the penalty coefficient ξ is set to 10. During training, the optimization is performed via the Stochastic Gradient Descent with Momentum (SGDM) algorithm. The learning rate is 0.001 and the batch size for training is 64. In open-set detection, the k is set to $0.08 \times N_i$. The threshold λ_i is set to a number that guarantees a large percentage of seen data in validation set can be successfully classified. This percentage is set to 80% for USC-HAD, PAMAP2, MM-Fi (Wi-Fi), and 75% for MM-Fi (mmWave). All results are obtained by calculating the mean and variance on all splits for each dataset.

5.3 Open-Set Detection Performance

5.3.1 Baselines and Evaluation Metrics

We consider the following open-set detection baselines.

MSP [9] measures the maximum softmax probability generated by a model trained on the seen class data using cross-entropy loss to detect unseen class data. For the MSP baseline, we adopt the Vision Transformer as model architecture.

KNN [20] computes the k-th nearest neighbor distance between an input image feature and the training set for unseen class data detection. The images are augmented, e.g., by adding Gaussian noise, for supervised contrastive learning in KNN. In the KNN baseline, we augment the IoT data also by adding noise and use supervised contrastive learning to extract IoT embeddings.

MCM [13] measures the distance between an input image feature and its closest label embedding, both directly generated by a large vision-language FM, for unseen class data detection. For the MCM baseline, we replace the image features with our IoT embeddings and use the prompt template "The human action of [CLASS]" for text encoding.

For all open-set detection baselines, we set the detection thresholds and parameters using the same strategy as our method.

To evaluate the performance of open-set detection, we employ the weighted precision, recall, and F1 score.

5.3.2 Results

In Table 1, we can see that our approach achieves the best open-set detection performance compared with all baselines on all three modalities' datasets. In detail, our approach outperforms the traditional softmax-based method MSP because the supervised contrastive loss can help our model obtain more distinguishable IoT embeddings than the cross-entropy loss in MSP. The KNN method performs worse than ours. This is because the image augmentation used by KNN for supervised contrastive learning, e.g., adding noise, cannot be directly applied to IoT data. Differently, our approach aligns text embeddings with IoT embeddings using supervised contrastive learning and achieves more generalized IoT embeddings. Our approach performs better than MCM since the MCM only takes hard prompts to generate text embeddings, which is undesirable for aligning IoT embeddings of different tasks with text embeddings.

Dataset	Method	Performance		
		Precision	Recall	F1 score
MM-Fi (mmWave)	MSP	72.1±0.1%	71.9±0.1%	71.8±0.1%
	KNN	68.9±0.0%	68.5±0.1%	68.4±0.1%
	MCM	70.8±0.2%	70.5±0.3%	70.4±0.3%
	Ours	**73.5**±0.1%	**73.2**±0.1%	**73.0**±0.1%
USC-HAD	MSP	69.4±0.3%	68.6±0.4%	67.8±0.6%
	KNN	77.8±0.1%	77.7±0.1%	77.7±0.1%
	MCM	66.8±1.2%	65.7±1.3%	64.1±1.7%
	Ours	**79.2**±0.3%	**78.9**±0.3%	**78.8**±0.3%
PAMAP2	MSP	87.6±0.1%	87.0±0.0%	87.0±0.0%
	KNN	88.7±0.1%	87.7±0.1%	87.6±0.1%
	MCM	81.4±0.3%	81.1±0.2%	81.1±0.2%
	Ours	**89.6**±0.0%	**88.0**±0.0%	**87.9**±0.0%
MM-Fi (Wi-Fi)	MSP	77.2±0.1%	77.0±0.1%	77.0±0.1%
	KNN	58.1±0.1%	56.5±0.1%	54.0±0.1%
	MCM	74.0±0.1%	73.6±0.1%	73.4±0.1%
	Ours	**77.4**±0.0%	**77.3**±0.0%	**77.3**±0.0%

Table 1. Open-set detection performance.

5.4 Zero-Shot Classification Performance

5.4.1 Baselines and Evaluation Metrics

We consider the following baselines for evaluating the GZSL performance of our approach.

ALE [2] measures the compatibility of image features and class label embeddings in the Euclidean space for ZSL.

DCN [11] uses a Deep Calibration Network to map image features and class prototypes to a common embedding space for ZSL.

BERT [7] We replace the frozen CLIP text encoder in our approach with the pre-trained BERT to process the prompt template as a baseline.

f-CLSWGAN [25] uses an attribute conditional feature generating adversarial network to generate CNN features of unseen classes for ZSL.

FREE [5] learns a visual feature generator jointly with a feature refinement module for ZSL.

ALE, DCN, and BERT are embedding-based methods, while f-CLSWGAN and FREE are generative-based methods. We replace the image features with IoT embeddings in all the above methods as baselines.

We evaluate the performance of GZSL using the following metrics. We measure the percentage of correctly classified seen and unseen class data samples, i.e., seen class accuracy $\text{ACC}_{\mathcal{S}}$ and unseen class accuracy $\text{ACC}_{\mathcal{U}}$, respectively. Note that these accuracies are the weighted average across all seen/unseen classes. We also compute the harmonic mean [16], which is a conventional metric to measure the inherent biasness of a GZSL method with respect to the seen classes:

$$\text{ACC}_H = \frac{2 \times \text{ACC}_{\mathcal{S}} \times \text{ACC}_{\mathcal{U}}}{\text{ACC}_{\mathcal{S}} + \text{ACC}_{\mathcal{U}}}, \tag{5}$$

A lower ACC_H means that the unseen class accuracy $\text{ACC}_{\mathcal{U}}$ is lower than seen class accuracy $\text{ACC}_{\mathcal{S}}$, indicating that a GZSL method is biased towards the seen classes.

5.4.2 Results

As shown in Table 2, our approach achieves the best $\text{ACC}_{\mathcal{U}}$ and ACC_H on all datasets compared with all baselines. Although some

baselines have higher $\text{ACC}_{\mathcal{S}}$, it is not practical to only consider seen classes because recognizing both seen and unseen classes is critical for most IoT sensing tasks. Specifically, our approach outperforms embedding-based approaches ALE, DCN, and BERT on ACC_H because we construct better text embeddings by using cross-attention to integrate soft prompt and hard prompt while using contrastive loss to make text-IoT embedding alignment more accurate and robust. Moreover, these methods are trained only with uni-modal textual data, whereas the CLIP text encoder is trained from multi-modal data of both images and text, which generates more effective text embeddings for data-text alignment [6]. Compared with generative methods f-CLSWAGAN and FREE, our approach still achieves superior performance. The generative methods' results on small-scale IoT datasets are less satisfactory because their performance relies on a large amount of training data. For our approach, in addition to using the generative model for synthesizing unseen class data to alleviate the biasing problem, the open-set detection also helps our method further classify the seen and unseen data correctly, achieving better performance.

Dataset	Method	Performance		
		$\text{ACC}_{\mathcal{S}}$	$\text{ACC}_{\mathcal{U}}$	ACC_H
MM-Fi (mmWave)	ALE	86.5±0.1%	0.01±0.0%	2.0±0.0%
	DCN	67.0±1.3%	30.2±1.3%	40.3±0.9%
	BERT	71.8±0.0%	36.9±0.6%	48.3±0.5%
	f-CLSWGAN	77.2±0.3%	29.7±0.5%	42.3±0.4%
	FREE	87.7±0.1%	25.3±0.8%	38.3±1.1%
	Ours	73.3±0.0%	40.4±0.5%	**51.7**±0.3%
USC-HAD	ALE	92.5±0.0%	0.6±0.0%	1.1±0.0%
	DCN	56.6±3.2%	37.1±1.5%	43.3±1.1%
	BERT	74.9±0.1%	41.6±1.3%	52.2±0.7%
	f-CLSWGAN	81.3±0.5%	29.2±3.5%	39.5±4.9%
	FREE	90.9±0.1%	14.0±0.6%	23.2±1.6%
	Ours	73.1±0.5%	54.8±1.8%	**61.1**±0.7%
PAMAP2	ALE	70.1±3.9%	12.1±1.9%	15.5±3.6%
	DCN	42.2±0.9%	33.1±0.2%	36.7±0.3%
	BERT	74.7±0.0%	49.9±0.7%	59.3±0.0%
	f-CLSWGAN	92.4±0.2%	27.8±1.1%	41.7±1.5%
	FREE	87.7±0.3%	37.2±0.2%	52.1±0.2%
	Ours	74.6±0.1%	53.7±0.4%	**62.1**±0.2%
MM-Fi (Wi-Fi)	ALE	52.2±6.0%	9.5±0.5%	11.8±0.5%
	DCN	60.1±1.8%	18.7±0.2%	28.2±0.4%
	BERT	62.5±0.0%	29.5±0.5%	39.5±0.5%
	f-CLSWGAN	84.7±0.0%	6.2±0.1%	11.4±0.1%
	FREE	80.0±0.1%	30.4±0.3%	43.6±0.3%
	Ours	75.1±0.0%	35.3±0.5%	**47.6**±0.4%

Table 2. Generalized zero-shot learning performance.

5.5 Ablation Study

To analyze the effectiveness of the prompt engineering, open-set detection, and data augmentation modules, we conduct ablation studies by removing one of these components. The results are shown in Table 3.

Prompt Engineering. To demonstrate that prompt engineering brings improvement to GZSL, we remove it by replacing the prompt engineering part with a fixed prompt template, "The human action of [CLASS]". As shown in Table 3, we can see that there is an accuracy drop in $\text{ACC}_{\mathcal{U}}$ and ACC_H by disabling the prompt engineering. The prompt engineering provides tailored text embeddings by integrating the soft prompt and hard prompt, helping the model to align text embeddings and IoT embeddings, resulting in better GZSL results.

Open-Set Dectection. To validate the effectiveness of open-set detection, we remove it and directly match the IoT embeddings with all seen and unseen text embeddings. The class label that has the largest matching score will be the classification result. As shown in Table 3, although there is an increase for $\text{ACC}_{\mathcal{S}}$, the $\text{ACC}_{\mathcal{U}}$ and ACC_H experience a huge decline by removing the open-set detection module. This is because the open-set detection helps the model eliminate the bias problem in GZSL, leading to classifying more unseen data correctly.

Data Augmentation. To investigate the effectiveness of data augmentation, we remove the step of fine-tuning the model using synthetic data. From Table 3, we can observe that by using data augmentation, the $\text{ACC}_{\mathcal{U}}$ is improved since the synthetic unseen data helps the model to reduce the bias problem of unseen IoT embeddings.

Dataset	P.E.	O.S.	D.A.	Performance		
				$\text{ACC}_{\mathcal{S}}$	$\text{ACC}_{\mathcal{U}}$	ACC_H
MM-Fi (mmWave)		✓	✓	60.0±0.1%	38.1±0.4%	46.4±0.2%
	✓		✓	88.6±0.0%	8.4±0.1%	15.0±0.5%
	✓	✓		72.0±0.0%	39.8±0.2%	51.1±0.1%
	✓	✓	✓	73.3±0.0%	40.4±0.5%	**51.7**±0.3%
USC-HAD		✓	✓	74.8±0.2%	40.3±3.0%	49.9±1.5%
	✓		✓	83.8±1.1%	14.1±0.9%	22.8±1.7%
	✓	✓		73.1±0.3%	51.3±1.1%	59.5±0.5%
	✓	✓	✓	73.1±0.5%	54.8±1.8%	**61.1**±0.7%
PAMAP2		✓	✓	73.5±0.1%	53.1±0.9%	61.2±0.4%
	✓		✓	92.9±0.2%	7.7±0.9%	12.9±2.1%
	✓	✓		74.7±0.1%	52.7±0.2%	61.6±0.0%
	✓	✓	✓	74.6±0.1%	53.7±0.4%	**62.1**±0.2%
MM-Fi (Wi-Fi)		✓	✓	65.6±0.1%	31.6±0.3%	42.1±0.2%
	✓		✓	80.0±0.2%	10.2±0.1%	17.7±0.5%
	✓	✓		74.8±0.0%	34.4±0.4%	46.7±0.4%
	✓	✓	✓	75.1±0.0%	35.3±0.5%	**47.6**±0.4%

Table 3. Ablation study. P.E. indicates prompt engineering, O.S. represents open-set detection, and D.A. is the data augmentation.

6 Conclusion

In this work, we have explored the potential of foundation models (FMs) for zero-shot IoT sensing. We leverage the generalized knowledge encoded in FMs and employ novel techniques to bridge the semantic gap between IoT data and text embeddings. Our proposed approach utilizes cross-attention for effective prompt engineering and data augmentation to mitigate bias. The evaluation has demonstrated the superior performance of our approach compared with existing baselines in both open-set detection and generalized zero-shot learning tasks across USC-HAD, PAMAP2, MM-Fi datasets of IMU, mmWave, Wi-Fi modalities. Future research directions include exploring the integration of additional modalities and investigating the adaptability of our approach to different types of IoT sensors and applications. Besides, exploring the explainability and interpretability of FM-based zero-shot IoT sensing would be valuable for understanding the decision-making process.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7):1425–1438, 2015.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901, 2020.

[4] C.-F. R. Chen, Q. Fan, and R. Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 357–366, 2021.

[5] S. Chen, W. Wang, B. Xia, Q. Peng, X. You, F. Zheng, and L. Shao. Free: Feature refinement for generalized zero-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 122–131, 2021.

[6] Z. Chen, G. Chen, S. Diao, X. Wan, and B. Wang. On the difference of bert-style and clip-style text encoders. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13710–13721, 2023.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023.

[9] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2016.

[10] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

[11] S. Liu, M. Long, J. Wang, and M. I. Jordan. Generalized zero-shot learning with deep calibration network. *Advances in Neural Information Processing Systems*, 31, 2018.

[12] M. Matsuki, P. Lago, and S. Inoue. Characterizing word embeddings for zero-shot sensor-based human activity recognition. *Sensors*, 19(22): 5043, 2019.

[13] Y. Ming, Z. Cai, J. Gu, Y. Sun, W. Li, and Y. Li. Delving into out-of-distribution detection with vision-language representations. *Advances in Neural Information Processing Systems*, 35:35087–35102, 2022.

[14] E. Nijkamp, B. Pang, H. Hayashi, L. Tu, H. Wang, Y. Zhou, S. Savarese, and C. Xiong. Codegen: An open large language model for code with multi-turn program synthesis. In *The Eleventh International Conference on Learning Representations*, 2022.

[15] H. Ohashi, M. Al-Naser, S. Ahmed, K. Nakamura, T. Sato, and A. Dengel. Attributes' importance for zero-shot pose-classification based on wearable sensors. *Sensors*, 18(8):2485, 2018.

[16] F. Pourpanah, M. Abdar, Y. Luo, X. Zhou, R. Wang, C. P. Lim, X.-Z. Wang, and Q. J. Wu. A review of generalized zero-shot learning methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[17] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[18] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[19] A. Reiss and D. Stricker. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th International Symposium on Wearable Computers*, pages 108–109. IEEE, 2012.

[20] Y. Sun, Y. Ming, X. Zhu, and Y. Li. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, pages 20827–20840. PMLR, 2022.

[21] B. Tang, J. Zhang, L. Yan, Q. Yu, L. Sheng, and D. Xu. Data-free generalized zero-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5108–5117, 2024.

[22] C. Tong, J. Ge, and N. D. Lane. Zero-shot learning for imu-based activity recognition using video embeddings. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(4):1–23, 2021.

[23] W. Wang, C. Miao, and S. Hao. Zero-shot human activity recognition via nonlinear compatibility based method. In *Proceedings of the International Conference on Web Intelligence*, pages 322–330, 2017.

[24] T. Wu, Y. Chen, Y. Gu, J. Wang, S. Zhang, and Z. Zhechen. Multi-layer cross loss model for zero-shot human activity recognition. In *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11–14, 2020, Proceedings, Part I 24*, pages 210–221. Springer, 2020.

[25] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5542–5551, 2018.

[26] H. Xie and T. Virtanen. Zero-shot audio classification via semantic embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1233–1242, 2021.

[27] B. Yang, L. He, N. Ling, Z. Yan, G. Xing, X. Shuai, X. Ren, and X. Jiang. Edgefm: Leveraging foundation model for open-set learning on the edge. *arXiv preprint arXiv:2311.10986*, 2023.

[28] J. Yang, X. Chen, H. Zou, C. X. Lu, D. Wang, S. Sun, and L. Xie. Sensefi: A library and benchmark on deep-learning-empowered wifi human sensing. *Patterns*, 4(3):100703, 2023. ISSN 2666-3899. doi: https://doi.org/10.1016/j.patter.2023.100703. URL https://www.sciencedirect.com/science/article/pii/S2666389923000405.

[29] J. Yang, H. Huang, Y. Zhou, X. Chen, Y. Xu, S. Yuan, H. Zou, C. X. Lu, and L. Xie. Mm-fi: Multi-modal non-intrusive 4d human dataset for versatile wireless sensing. *Advances in Neural Information Processing Systems*, 36, 2024.

[30] M. Zhang and A. A. Sawchuk. Usc-had: A daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 1036–1043, 2012.

[31] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He, et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*, 2023.

[32] K. Zhou, J. Yang, C. C. Loy, and Z. Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9): 2337–2348, 2022.

[33] Y. Zhou, J. Yang, H. Zou, and L. Xie. Tent: Connect language models with iot sensors for zero-shot activity recognition. *arXiv preprint arXiv:2311.08245*, 2023.

[34] B. Zhu, B. Lin, M. Ning, Y. Yan, J. Cui, W. HongFa, Y. Pang, W. Jiang, J. Zhang, Z. Li, et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. In *The Twelfth International Conference on Learning Representations*, 2023.