2024 IEEE 6th International Conference on Cognitive Machine Intelligence (CogMI) | 979-8-3503-8672-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/CogMI62246.2024.00014

The Case for Micro Foundation Models to Support Robust Edge Intelligence

Tomoyoshi Kimura†, Ashitabh Misra†, Yizhuo Chen†, Denizhan Kara†, Jinyang Li†, Tianshi Wang†, Ruijie Wang†, Joydeep Bhattacharyya*, Jae Kim‡, Prashant Shenoy+, Mani Srivastava§, Maggie Wigness*, Tarek Abdelzaher†
†*University of Illinois at Urbana-Champaign,* **DEVCOM Army Research Laboratory,*
+*University of Massachusetts, Amherst,* ‡*Boeing Inc.*
§*University of California, Los Angeles*

***Abstract*—This paper advocates the concept of micro foundation models (μFMs), recently introduced by the authors to describe a category of self-supervised pre-training solutions that we argue are necessary to support robust intelligent inference tasks in Internet of Things (IoT) applications. The work is motivated by the fact that collecting sufficient amounts of *labeled* data in IoT applications to train AI/ML tasks is challenging due to the difficulties in labeling such data after the fact. In the absence of sufficient labeled data, supervised training solutions become brittle and prone to overfitting. Self-supervised training obviates the collection of labeled data, allowing pre-training with the more readily available *unlabeled* data instead. Specifically, the μFMs discussed in this paper use *self-supervised* pre-training to develop an encoder that maps input data into a semantically-organized latent representation in a manner *agnostic* to the downstream inference task. Our preliminary work shows that this (unsupervised) encoder can be *moderately sized*, yet produce a latent representation that simultaneously supports the fine-tuning of *multiple* downstream inference tasks, each at a minimal labeling cost. We demonstrate the efficacy of this pre-training/fine-tuning pipeline using a vibration-based μFM as a running case study. The study shows that the fine-tuning of inference tasks on top of the aforementioned encoder-produced latent representation needs orders of magnitude fewer labels than supervised training solutions, and that the resulting tasks are significantly more robust to environmental changes and easier to adapt to domain shifts compared to their supervised counterparts. Furthermore, we show that inference algorithms based on our example μFM can be executed in real time on a Raspberry Pi device, making the approach viable for the IoT space. We conclude that μFMs are a preferred (and likely necessary) route to support robust intelligent sensing on IoT devices in subareas where labeled data collection is challenging. The paper is a call for the research community to invest in μFM research for IoT applications.**

***Index Terms*—Foundation Models, Self-Supervised Learning, Internet of Things**

I. Introduction

The notion of *micro foundation models* (μFMs) was first introduced by the authors in a recent paper [1], advocating the advantages of self-supervised learning in IoT contexts. This paper argues that μFMs are not only desirable but in fact necessary for deployment of robust edge intelligence. To support this claim, we explain the pitfalls of supervised techniques that lead to brittleness and possibly to catastrophic failures. We then offer evidence that self-supervised solutions avoid the pitfalls of their supervised counterparts.

Early work on edge AI made significant strides in neural network compression [2], [3] and attention-based data prioritization [4], among other topics [5], allowing the execution of non-trivial inference tasks at the edge. The work promised significant reductions in edge-based inference latency by allowing the execution of requisite inferences at the point of need (i.e., closer to the sensors that produce the underlying data). The initial solutions relied on supervised learning [6] to enable specific inference tasks. Unfortunately, in the absence of large amounts of labeled training data, supervised approaches are prone to overfitting, and thus brittle to domain shifts [7]. Thus, recently, self-supervised training emerged instead [1], [8], [9]. Self-supervised learning leverages *unlabeled data*, thereby circumventing the scarcity of data labels that causes overfitting and brittleness.

Self-supervised techniques differ conceptually (from their supervised counterparts) in the philosophy underlying their training; rather than training to accomplish a specific downstream inference task, self-supervised techniques focus on finding the most appropriate and concise latent data representation to enable a wide range of downstream inferences. This difference has far-reaching run-time implications on inference efficiency and robustness. First, since the latent representation is decoupled from the specifics of any downstream task, it allows for easier task adaptation (or fine-tuning) to changes in the environment, noise, targets, and adversarial disruptions, thereby enhancing run-time robustness [1]. Moreover, being much more concise than the original sensor data, this representation enables more efficient operation, when several nodes must share data to allow downstream inference.

This paper illustrates examples of brittleness of supervised solutions, reviews recently proposed self-supervised techniques, and discusses their robustness and efficiency advantages. Challenges are presented in extending efficient and robust inference to multimodal multi-vantage settings on resource-constrained devices.

The rest of the paper is organized as follows. We describe a key challenge facing the development of intelligent IoT applications in Section II. Section III argues the main advantages of μFMs – our proposed solution to the aforementioned challenge. To offer a proof of concept, we present an evaluation harness developed for testing intelligent IoT applications

979-8-3503-8672-1/24/$31.00 ©2024 IEEE
DOI 10.1109/CogMI62246.2024.00014

in Section IV. Section V describes our experimental design, followed by a preliminary evaluation of the μFM concept Section VI. We discuss remaining issues and challenges in developing μFMs for IoT sensing applications in Section VII, and cover recent related literature in Section VIII. Lastly, we conclude the paper in Section IX.

II. The Challenge in Edge Intelligence

Applying supervised deep-learning-based solutions to real-field problems is challenging because it requires significant amounts of labeled training data. Labeled IoT training data are hard to collect because IoT signals (e.g., acoustic and seismic data) are difficult to label after the fact. Unlike images, for example, that are generally self-describing and can thus be labeled in retrospect, in IoT applications, unless the environment in which the time-series data were collected was documented at the time of collection, the lack of interpretability of the recorded sensor waveforms makes it hard to label the recorded phenomena simply by inspecting the data in hindsight. In addition, time-series data often conflate foreground objects with background influences. For example, the sound and vibrations emitted by a given target, such as a passing vehicle, may differ substantially based on the terrain (e.g., sand, dirt, snow, or tar). Thus, adequately training supervised AI to detect such targets requires labeled samples taken in all possible environmental conditions. The need for diverse training data combined with the difficulties in labeling is a key problem facing the development of intelligent IoT applications. Labeled datasets are often insufficient in size or diversity. Using inadequate amounts of training data results in overfitting, which occurs when the number of input data samples available for training is not significantly larger than the number of parameters being trained. As a result, the trained model may simply memorize the individual samples without the ability to generalize well to new ones, causing it to fail in the field.

In practice, another challenge compounds the inadequacy of AI/ML training in IoT scenarios. Since running field experiments with desired IoT sensors is often expensive, intelligent IoT applications often rely on *previously collected* traces (or datasets) for AI/ML training and evaluation. It is in this context that another pitfall manifests [7]. Namely, it is often possible to achieve deceptively good evaluation outcomes on test datasets, whereas the underlying algorithms might remain prone to catastrophic failure in practice. This challenge was illustrated in recent experiments [7], where we designed two target detection and classification algorithms. One was based on a neural network architecture for embedded AI [2] with over one million parameters. The other was based on a traditional decision-tree-based machine learning approach with domain-inspired input feature engineering and under 50,000 parameters. The neural network approach outperformed the traditional one on the test dataset. Yet, it failed catastrophically in later deployment.

To explain why this occurred, it is important to understand how dataset-based evaluation leads to unintentional overfitting. On the surface, to guard against overfitting, the prevailing evaluation methodology calls for a separation between training and testing data; the neural network is trained on one dataset but tested on another. The problem with this methodology lies in the way it is implemented in a typical research environment. Specifically, due to difficulties accessing real physical systems, researchers often acquire training, validation, and testing data ahead of time. The tested algorithms are then developed iteratively. When the first iteration of the algorithm fails to do well on the test dataset, the researchers take notes and re-design the algorithm. This continues until a version is reached that does well on test data. The practice creates an unintended feedback loop from testing one version of the algorithm to designing the next. This loop ultimately results in overfitting the developed algorithm to the testing data, despite using proper cross-validation when testing any one iteration of the algorithm [7] (i.e., despite the fact that each version of the algorithm is trained on one dataset and tested on another). A solution is needed to fundamentally reduce reliance of AI on labeled training data. In sensing applications, unlabeled data are much easier to collect. If one can exploit the available volumes of unlabeled data for training, overfitting (due to data scarcity) is significantly ameliorated. Below, we describe how unlabeled data and self-supervised training can successfully reduce reliance of intelligent IoT application development on labeled samples.

III. The Argument for Micro Foundation Models

In this section, we detail the notion of μFMs then describe the insights why they improve robustness of edge AI. In a nutshell, μFMs are trained using a two step process. The first, *pre-training*, produces an encoder using *only unlabeled data*. The second, *fine-tuning*, produces lightweight decoders (as small as a single linear layer each) to support downstream tasks. Fine-tuning needs a minimal amount of labeled data. By eliminating the need for labeled data in pre-training, the reliance on labeled data is vastly reduced, mitigating overfitting-related problems.

A. Preliminaries of μFMs

Micro Foundation Models (borrowing the concept of *foundation models* from the vision and NLP domains [10]) are meant to extract domain knowledge in an unspervised manner that homogenizes inferencing across multiple downstream tasks. In prior work [1], we defined μFMs to have the following essential properties:

- *Domain-specific:* Unlike models such as ChatGPT that attempt to offer general intellligence, the μFMs we advocate are not designed to be "everything for everyone". Since different application domains call for encapsulating different types of expertise within the foundation model, we encourage specialization. Limiting the construction of the model from scratch to an application domain can significantly reduce its needed size. An example would be a foundation model for medical image analysis, urban traffic monitoring, target tracking, or network security.

- *Modality-specific:* The number of different types of sensors of interest to IoT scenarios can be very large. To limit the size of the model further, we constrain it to the handling of a limited number of sensing modalities.
- *Self-supervised:* This is a core property of all foundation models and is inherited by μFMs.
- *Task-agnostic:* As alluded to in the introduction, self-supervised pre-training is fundamentally different from its supervised counterpart. Rather than training a neural network to perform a particular inference task, as done in supervised training, the purpose of the self-supervised approaches we advocate for training μFMs is simply to train an encoder to offer a better and more concise representation of input data. This is done in a manner agnostic to the particular downstream inference tasks that we ultimately need to implement. As a result, the computed representation may facilitate multiple very different downstream tasks. Such training is called *task-agnostic*.
- *Moderately-sized:* As the name "μFM" suggests, the intent is to find models that use only a moderate number of parameters (e.g., in the millions, not billions) and correspondingly moderate amounts of data for pre-training.

It remains to argue why μFMs stand to significantly improve robustness of edge AI applications. The relevant background and insights are described next.

B. Reducing Reliance on Labeled Data

This section offers an important piece of preliminary background needed to appreciate μFMs. Namely, how does one train a model in an unsupervised manner to be task agnostic, and why would such training improve downstream inference robustness? Two common training techniques for foundation models are (i) *contrastive learning*, and (ii) *masking*:

- *Contrastive learning* teaches the foundation model a notion of semantic similarity by applying label-invariant transformations (such as image rotation) to individual data samples and contrasting the transformed samples with random other ones. Specifically, an encoder is trained to project input data into a latent space. The loss function rewards the encoder for placing similar samples closer together (in the latent space) and different samples further apart. Note that, no labels are needed for encoder training. For example, an image and its rotation can be presented as examples of similar samples without understanding (or labeling) the content of the image.
- *Masking*, in contrast to contrastive learning, encourages the encoder to extract latent structures that allow it to guess the masked elements in the input. Specifically, some parts of the input are masked. The remaining parts are given to the encoder that maps them to a latent space from which a decoder is designed to restore the missing pieces. The loss function rewards the encoder/decoder pair for reconstruction accuracy of those missing pieces. Note that, no data labels are needed either. The hypothesis is that if the decoder has learned to reconstruct the missing pieces of input correctly, it must be that the latent representation produced by the encoder has extracted higher-level semantics from the input that allow the auto-filling.

The above process describes *pre-training*. The outcome of pre-training (using either of the above approaches) is an encoder that maps inputs into a *sematically well-organized* latent space. In the case of contrastive learning, this property arises because similar inputs are mapped closer together in the latent space. In the case of masking, this property arises because the training forces the latent representation to efficiently encode the higher-level semantics needed for the reconstruction of masked regions. The decoder used in pre-training can be discarded.

Once an encoder has been designed (using either of the above approaches) to map input data to the latent space, it becomes possible to support multiple inference tasks. Given the semantically well-organized latent space, it becomes easy to map from that space to a variety of inferences (e.g., inferences on observed target classes, activities, environmental conditions, etc), which is called *fine-tuning*. Intuitively, fine-tuning becomes easy because the mapping from a semantically well-organized space to a semantic output is generally simpler than the mapping from the original data space to the same output. Given a set of labeled concepts that we need to recognize in input data, we train a decoder to recognize the regions of the latent space to which such concepts map. A simple one-layer linear decoder network is often sufficient to delineate such regions given a small number of labeled samples.

To summarize, the key observation that simplifies decoder design is that the latent space is semantically well-organized. Thus, the desired concepts of interest to a given downstream task tend to map to points in the latent space that are well-clustered together and thus easy to delineate with a few linear constraints. The simplicity of the decoder structure implies a significant reduction in the number of labeled data samples needed for fine-tuning. In other words, it explains the significant reduction attained in the reliance on labeled data when using the aforementioned pre-training/fine-tuning approach. Moreover, simplifyimg the decoder (and thus needing less labeled data to train it) is the key to avoiding overfitting. A small number of labeled samples is usually sufficient to train the task-specific decoder given the low number of decoder parameters. Note that changes in the environment and other domain shifts can now be easily accommodated by retraining or updating the simple decoder, as opposed to retraining the whole neural network model. Consequently, the architecture described above is also much more adaptive to environmental changes and domain shifts, a conjecture we later demonstrate empirically in our evaluation.

We refer by μFM to the actual pre-trained neural network encoder model and any used task-specific decoders. This section summarized our argument for why μFMs offer more robust edge AI. To complete the discussion of μFM advantages, below, we point out another useful property attained as a side-effect of reduced decoder complexity.

C. Improving Structural Edge Resilience

In addition to reducing the need for labeled data, μFMs can improve the *structural resilience* of edge systems. Specifically, we conjecture that these machine learning techniques can both (i) help reduce unnecessary dependencies in intelligent data fusion workflows, thus decreasing the propagation of local failure effects, and (ii) help create pools of more easily interchangeable, retargetable components that can take over each other's functions, thus increasing resilience in the face of component failures. We explain these conjectures below.

- *Increasing resilience by breaking unnecessary dependencies:* The separation of our μFM architecture into an encoder that maps inputs into a latent space followed by task-specific decoders that map the latent representation to task-specific inferences breaks down end-to-end stovepipes of supervised models that directly connect sensing to inference. Data from multiple sensor modalities (e.g. both acoustic and seismic data as we show later in this paper) can be mapped to the same joint latent space. Different downstream analytics can then be implemented based on the shared joint latent representation. The model architecture thus decouples the sensors that encode their respective modality measurements (into the common latent space) from the analytics that utilize the shared representation for various tasks. The decoupling, in turn, increases workflow resilience; failures of individual sensors can be mitigated by other sensors that generate the same unified semantic representation without impacting downstream analytics, thus reducing disruption to the rest of the data fusion pipeline. Similarly, downstream analytics can be changed without impacting the front of the sensor data processing pipeline.
- *Increasing resilience by promoting retargetable components:* The ability of self-supervised pre-training to produce latent representations that support a multiplicity of downstream tasks (thanks to lightweight task-specific fine-tuned decoders) creates further opportunities for improved resilience. As we show in the evaluation, different decoders can be fine-tuned to support different task-specific inferences such as target classification, distance estimation, and others. Since the decoders are very lightweight, nodes pre-loaded with a pre-trained, self-supervised encoder model can now be easily re-targeted to offer new functionality as needed, simply by adding the relevant decoders, possibly covering for the loss of other nodes with specific functions elsewhere in the system.

The separation of end-to-end inference stovepipes into easily combinable mix-and-match components together with the ability of components to take on each other's roles is akin to the concept of *degeneracy* in biological networks [11]. In biology, *degeneracy* refers to a condition where (i) agents can perform one of multiple functions depending on context, and (ii) the same function can be performed by one of several agents. For example, individuals in an organization might allocate their time to any of a set of possible roles. Similarly, the same role can possibly be performed by any of multiple individuals. It is shown that degeneracy improves system resilience by facilitating reconfiguration to adapt to perturbations [11]–[13]. The architecture of μFMs improves degeneracy of edge networks in that certain sensors can take on roles of other sensors (by mapping to the same latent space) and nodes can take on the analytics of other nodes simply by adding a lightweight decoder (which is far easier in terms of incremental overhead than adding an entire new supervised model). While we do not explicitly explore this angle in the evaluation for space limitations, the prospect of improving resilience to structural perturbations is one of the most intriguing advantages of μFMs.

IV. ACIES-OS: AN EVALUATION HARNESS

To demonstrate that the above insights hold true in the field, we designed an evaluation harness of edge AI solutions, called *Acies-OS* [14]. Acies-OS is developed on the concept of digital twins [15], which provides virtual representations of the deployed physical systems and optimizes its run-time behavior (orchestration). However, dynamic IoT deployment conditions, such as heterogeneous sensors (different types of sensors and the number of deployed sensors), poor inter-communication medium, and diverse running application workloads, often increase the difficulty of monitoring entire edge systems with a static twinned system. Acies-OS addresses these challenges by providing a content-centric platform for edge AI with a dynamic interface for prototyping diverse edge AI systems with digital twins. At the core of Acies-OS is a control plane that manages the services deployed across the edge nodes in real time. The deployed services can include AI models (*e.g.*, μFMs), sensors (*e.g.*, seismic and acoustic), and system status (*e.g.*, heartbeat). The control plane and these services are developed on top of a structured namespace to support flexible and extensible components for deployment. It enables efficient orchestration between the physical systems and the digital twins, which provides several advantages for deploying intelligent IoT applications:

- *Development and Deployment:* Acies-OS simplifies the prototyping of intelligent IoT applications. Its structured namespace allows the developers to easily integrate different components (*e.g.*, different AI models, additional sensors, etc.) into the system. Besides, Acies-OS supports adding or replacing modules independently to enhance deployment flexibility in dynamic IoT environments.
- *System monitoring:*: Acies-OS continuously tracks the status of edge devices and services. The control plane receives periodic updates from each node on the system status and service outputs (e.g., sensor readings or model predictions). These outputs can then be used to detect anomalies in the system or evaluate the running intelligent IoT applications.
- *Dynamic reconfiguration:*: Acies-OS supports rapid and flexible dynamic reconfiguration of services. The control plane allows services to be redeployed or reconfigured in real time without disrupting the rest of the system. This

TABLE I
RASPBERRY SHAKE DEVICE CONFIGURATIONS.

Device Type	Memory	Storage	Sensor
Raspberry Shake 4B Rev 1.4 4D (Node 1&4); 1D (Node 2&3)	8GB	64GB	200 Hz Seismic 16000Hz Acoustic

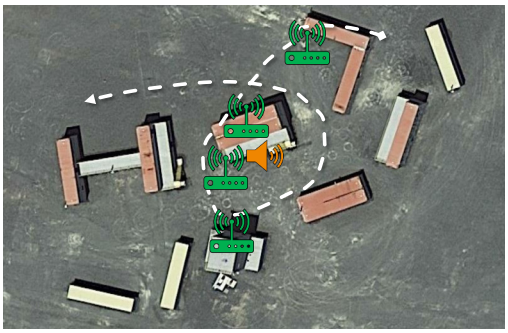


Fig. 1. Experimental site view. The nodes are colored in green. An example route in white is drawn in white for illustration purposes.

adaptability ensures that application performance remains optimal and the system remains resilient. For instance, in an edge application running multimodal models on seismic and acoustic signals, if a failure occurs in the seismic sensor due to malfunction, Acies-OS can detect the issue and quickly reconfigure the system. It can switch from a multi-modal model, which relies on both seismic and acoustic data, to a uni-modal model that uses only the acoustic service.

V. THE EXPERIMENTAL SETUP

To evaluate the robustness and generalizability of μFMs to new tasks and domains, we experimented at an outdoor research facility similar to the experimental site described in prior research [1]. We present an overview of the experimental site in Figure 1. The site is a repurposed neighborhood on gravel roads with four deploy nodes. Each node is a Raspberry Pi class device[1] containing a geophone and a microphone array to collect 200Hz seismic and 16,000Hz acoustic signals. Table I lists the detailed configurations for the deployed nodes. Three targets with distinct vibrational signals are selected: (i) a Polaris ATV, (ii) a Warthog all-terrain unmanned ground robot, and (iii) a standard pickup truck. During each run, a subset of these targets navigates around the neighborhood following a pre-specified route and passes by each node. A GPS receiver is attached to each vehicle to track their locations from the nodes. In Figure 2, we provide sample illustrations of these vehicles to show their differences.

A. Model Pre-training

We evaluate VibroFM, a type of μFMs for vibration application [1], with the SWIN-Transformer [16] as the primary backbone encoder. SWIN-Transformer is a variant of the Vision Transformer [17] that uses a hierarchical structure

[1]https://raspberryshake.org/

(a) Polaris (b) Warthog (c) Truck

Fig. 2. Vehicles used in the experiment. For illustration purposes only.

TABLE II
DATASETS FOR MODEL TRAINING/FINE-TUNING (WITH CROSS-VALIDATION) AND EVALUATION

Task	Train/Fine-tune set	Test Set (Reported)
Vehicle Classification	*Development Set* - Train	*Evaluation Set* - All
Distance Classification	*Evaluation Set* - Train	*Evaluation Set* - Test
Distance Rergession	*Evaluation Set* - Train	*Evaluation Set* - Test

with shifted windows for efficient extraction of the spatio-temporal features. VibroFM is pre-trained with FOCAL [8], a contrastive learning framework, on the large-scale unlabeled vibration dataset [1].

B. Downstream fine-tuning

We focus on two labeled datasets for downstream fine-tuning and evaluations.

(1) Development Set: We utilize the labeled dataset collected in prior work [1] for supervised training. This dataset contains single-target scenarios exclusively to fine-tune the pre-trained model and train their supervised counterparts. During fine-tuning, the pre-trained encoder parameters remain frozen, and only the downstream task-specific layer is optimized. We explore two types of downstream decoders: a single-layer linear classifier and a four-layered multi-layer perception (MLP). Instead of enforcing a single target classification with the softmax activation function, we apply the sigmoid activation function and binary cross entropy (BCE) objective for multi-target classification. We use the same confidence threshold of 0.8 for all evaluated models.

(2) Evaluation Set: We evaluate the fine-tuned VibroFM with the newly collected dataset. This Evaluation Set presents several key distinctions from the Development Set, making it of particular interest:

- *Domain Shift:* Although this dataset is collected from similar vehicles to the Fine-tune set, it was collected at a different time and introduces varied environmental conditions.
- *Multi-Target Scenarios:* In contrast to the single-target fine-tune data, the Evaluation Set contains both single-target and multi-target samples. The multi-target scenarios present a more challenging task compared to the single-target classification. Signals generated by each target interfere with each other. This interference can lead to complex, overlapping vibration signals distinct from those a single target generates.
- *Target Distance:* Each target has a GPS receiver to track its real-time distance from the nodes. This additional

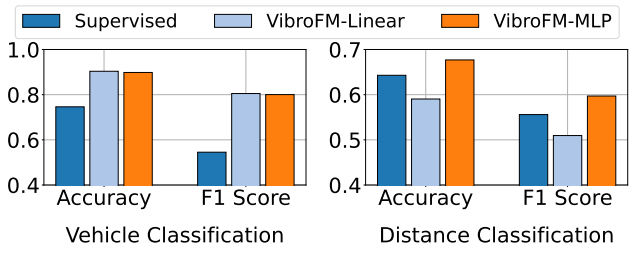


Fig. 3. Vehicle Classification (Left) and Distance Classification (Right) Performance.

information allows us to evaluate VibroFM's generalizability to additional downstream tasks, such as distance classification and distance regression.

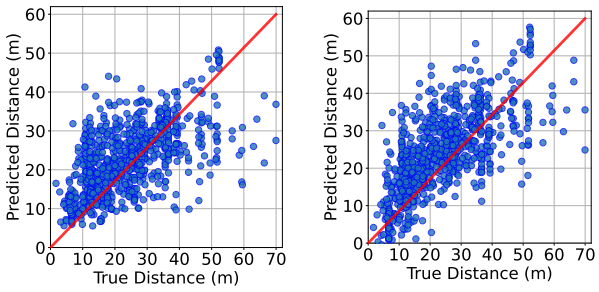
For vehicle classification, we train the supervised model and fine-tune the VibroFM variants on the *Development Set* with 80%, 10%, 10% ratio for cross-validation. We then test the performance on the entire *Evaluation Set* and report the accuracy and F1-score. For distance classification and distance regression, we train and fine-tune the models on the *Evaluation Set* by using the unique distance information. The *Evaluation Set* is split randomly by *runs* with the same 80%, 10%, 10% ratio for training, validation, and testing. The testing performance is reported. We clarify the specific set used to train and evaluate each task in Table II.

VI. Evaluation Results

In this section, we present a comprehensive evaluation of VibroFM. We begin by assessing its performance on the *Evaluation Set* to validate its robustness to new domains. We then evaluate its generalizability by applying VibroFM to additional distance-related downstream tasks. Lastly, we profile the model to demonstrate its efficiency and practicality in multi-task scenarios, focusing on latency and memory consumption.

A. Performance on Vehicle Classification

Figure 3 (Left) compares the VibroFM variants with the supervised model on vehicle classification. VibroFM variants significantly outperform the supervised model. Although the testing data from the *Evaluation Set* is collected from the same domain as the *Development Set*, the supervised model fails to generalize with substantial degradation in performance. In contrast, despite having *fewer parameters* to train on, both VibroFM-Linear and VibroFM-MLP demonstrate exceptional performance with 90% accuracy. The poor performance of the supervised model underscores the pitfall of traditional supervised learning when evaluated against real physical systems, where significant domain shifts are present due to the dynamic deployment conditions. VibroFM variants, leveraging self-supervised learning with more available unlabeled data, exhibit strong resilience to deployment. The semantically organized representations learned by the μFMs during pre-training can be well adapted to the physical systems without excessive reliance on domain-specific labels. This highlights μFMs significant robustness well suited for the diverse edge AI applications.



(a) Supervised (b) MLP

Fig. 4. Distance Regression Scatter Plot - Predicted distance vs. True distance.

B. Performance on Distance Classification

Next, we evaluate the models' performance on the distance classification task. Due to the lack of distance information in the *Development Set*, we train and evaluate the Supervised model and VibroFM variants with the newly collected *Evaluation Set*. We consider three classes: close range ($\leq$15m), medium range (15m-30m), and long range ($\geq$40m). As illustrated in Figure 3 (Right), the Supervised model shows marginal improvement over VibroFM-Linear but has lower performance when compared to VibroFM-MLP. The performance gap between the Supervised model and the VibroFM variants is notably reduced compared to the vehicle classification task. Since the training and testing data for distance classification are drawn from domains more similar to those in the vehicle classification task, the supervised model performs relatively well. However, this validates the challenge of supervised learning we described, which causes supervised models to often generalize poorly when deployed in real physical systems. On the other hand, VibroFM-MLP still achieves superior performance. The task-agnostic pre-trained representations with a lightweight decoder can outperform the task-specific Supervised model. This demonstrates the strength of μFMs in learning *task-agnostic* representations that can be adapted to various downstream IoT applications.

C. Performance on Distance Regression

We evaluate VibroFM on the distance regression task to demonstrate its potential to support a range of intelligent inference tasks beyond mere classification. In addition to the supervised baseline and VibroFM, we also implement a non-linear curve-fitting model using inverse polynomial regression to capture the inverse relation between the energy of audio (a_i) and seismic (s_i) signal with the target distance (d) on the *Development Set*. We formulate this relationship as

$$d_i = \left(\frac{A_a}{(a_i + B_a)^{p_a}} + C_a\right) + \left(\frac{A_s}{(s_i + B_s)^{p_s}} + C_s\right),$$

where $p_{a/s}, A_{a/s}, B_{a/s}$, and $C_{a/s}$ are learnable parameters. Figure 5 provides a quantitative comparison of the distance regression performance using Mean Absolute Error (MAE) and R-squared (R2) score metrics. Both VibroFM variants (Linear and MLP) outperform the supervised baseline and the

TABLE III
INFERENCE COMPUTATION OVERHEAD COMPARISON BETWEEN DIFFERENT TASKS.

Metric	Task	Vehicle Classification		Distance Classification		Distance Regression	
	Encoder	Linear	MLP	Linear	MLP	Linear	MLP
Average Latency (ms)	188.2342	0.1915	1.4341	0.2041	1.4304	0.2130	1.3950
P99 Latency (ms)	255.0525	0.3550	4.9966	0.3965	2.5303	0.3779	2.1056
Size (MiB)	44.9016	0.0069	0.5157	0.0029	0.5078	0.0010	0.5039

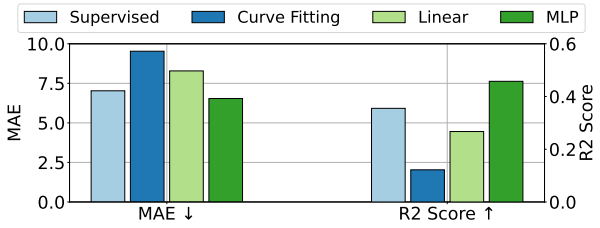


Fig. 5. Mean Absolute Error and R2 Score of Distance Regression.

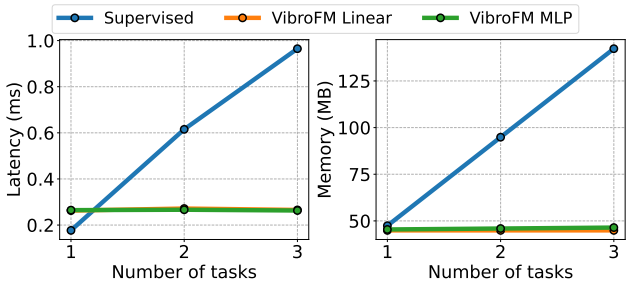


Fig. 6. Latency and model sizes against the number of tasks.

simple curve fitting approach with lower MAE and higher R2 scores, demonstrating μFM's ability to accurately estimate distances based on the latent representations learned during self-supervised pre-training. We present scatter plots in Figure 4 to show the correlation between true distances and predicted distances from the supervised baseline and VibroFM with the MLP decoder. Overall, there is a strong correlation between predicted and actual distances for both models, with more dense and accurate predictions observed in the 0-40 meter range. However, the VibroFM MLP decoder shows tighter clustering around the ideal prediction line (red) in the 20-40 meter range, which is the main range of interest. The relatively lower accuracy for distances beyond 40 meters can be due to the increased signal interference and attenuation over longer distances. As the target moves farther from the sensor, the received signals experience more environmental interference and longer travel times, making the distance estimation task more challenging. Both the supervised model and VibroFM experience deviation, but VibroFM maintains slightly better performance at longer distances. These results highlight the robustness of the representation learned by μFMs during task-agnostic pre-training in handling tasks beyond classification, even when trained without any distance-specific labels.

D. System Performance

Lastly, we conduct system evaluations to show that μFMs (1) are moderately sized and executable in real-time on resource-constrained edge devices despite having exceptional performance, and (2) can significantly benefit the run-time inference and memory consumption as the number of tasks scale for an intelligent IoT application.

1) Inference overhead: We deploy VibroFM on the Raspberry Shake device described in Section V to evaluate its latency and memory consumption, demonstrating the feasibility of using μFMs in real-time IoT applications. For each task, we set the batch size to one and measure the inference time of both the encoder and decoder across 300 samples. Table III presents the mean and P99 latency for each task. VibroFM is highly efficient and can complete an inference in under 260 milliseconds for each two-second sample, meeting the latency requirements for IoT applications. The MLP decoders take slightly longer than their linear counterparts due to the dense computations. However, we observe that most of the latency overhead originates from the encoder, while the decoders have minimal impact on overall performance. In addition to its low end-to-end latency, VibroFM's entire parameter size is under 50 MB, making it highly compact and efficient for deployment on resource-constrained edge devices. This ensures VibroFM is well-suited for various IoT applications where both speed and memory efficiency are required.

2) Multi-task inference: Despite outperforming traditional supervised models in different tasks, a significant advantage of VibroFM is its superior multi-task efficiency. We compare the inference efficiency of VibroFM with task-specific supervised encoders as the number of tasks increases. Figure 6 illustrates the latency and memory overhead of both approaches as tasks scale. For the supervised models, both latency and memory overhead rise significantly with the addition of more tasks. In contrast, VibroFM's Linear and MLP decoders only show marginal increases. This is because VibroFM encodes the input data only once, and the resulting representations are generalizable to all downstream tasks. Therefore, adding new tasks merely requires loading lightweight decoders, leaving the primary computational overhead from the encoder unaffected. On the other hand, task-specific supervised approaches require an entirely new model for each additional task, leading to substantial memory usage and higher latency. Each task requires a full end-to-end inference, which directly contributes to the increased computational overhead and resource consumption as the number of tasks grows. μFMs' task-agnostic nature allows for superior scalability in multi-task scenarios. This efficiency, combined with its competitive performance across various

tasks, makes μFMs a more practical solution for resource-constrained IoT applications, especially for an intelligent IoT application running multiple tasks simultaneously.

VII. Limitations and Challenges

This paper argues μFM's exceptional robustness and adaptability compared to traditional supervised methods when deployed in physical systems. The task-agnostic nature of μFM makes it a practical solution for edge AI applications. In this section, we discuss some of the remaining challenges in developing μFMs and potential directions for future works. Our evaluation highlights that μFMs exhibit superior robustness by leveraging widely available unlabeled data for pre-training. Unlike traditional supervised methods that are *label hungry*, μFMs reduces the reliance on large-scale labeled data to perform well in deployment scenarios with simple decoder designs. μFM's pre-trained embedding shows robustness to domain shifts during deployment, extracting consistent representations of the same target even across different deployment environments. However, the performance of the downstream tasks can still heavily depend on the *quality* of the labeled data used to fine-tune the decoders. This challenge is particularly pronounced in IoT applications, where label quality is extremely difficult to determine due to the diverse conditions (*e.g.*, environment, location of the nodes, the target of interest, etc.) present in the physical system. Due to the simplicity of the decoder designs and fine-tuning objectives, incorrect or noisy labels can mislead the decoders, producing inaccurate mapping even if the pre-trained embedding is well-structured. A promising direction for future work lies in designing new fine-tuning techniques and decoder architectures that are robust to low-quality labels. Additionally, developing effective task-specific fusion strategies for pre-trained embeddings from multiple sensor modalities, such as adaptive weighting schemes that can dynamically adjust each modality based on task-specific requirements and physics-guided signal quality, could further enhance the fine-tuned adaptability of μFMs in dynamic IoT applications.

VIII. Related Work

Self-Supervised Learning: Self-supervised learning has emerged as a promising paradigm to tackle the limitations of supervised learning, particularly in scenarios where labeled data is scarce [1]. It leverages large amounts of unlabeled data to learn meaningful task-agnostic representations that can be fine-tuned for downstream tasks. Two main approaches have gained increasing popularity: contrastive learning and masked reconstruction. Contrastive learning [18]–[23] focuses on learning robust representations by maximizing the agreement between positive pairs of data while pushing negative pairs apart. Prior works [24]–[26] have also studied leveraging the temporal properties for learning time-series data. Contrastive learning for multimodal inputs [27], [28] has gained increasing interest and has been well explored for different IoT sensing applications [8], [29], [30]. On the other hand, masked reconstruction [31]–[35] techniques mask significant portions of the input data and learn to encode meaningful representations that can be used to reconstruct the missing parts. For time-frequency signals, previous works leverage time-frequency spectrogram [36], [37] and adopt physical priors to make meaningful masking [9].

Robust IoT Sensing: Recent advancements in deep learning have significantly contributed to robust signal classification across various domains, especially in dynamic IoT deployment environments where signal integrity is often compromised. In the field of automatic modulation classification, prior work has introduced a threshold autoencoder denoiser CNN (TADCNN) [38] that markedly improved classification accuracy, especially in low signal-to-noise ratio (SNR) conditions. Others have proposed to convert time-domain signals into symmetrized dot patterns and achieve remarkable efficiency in differentiating various vehicle noise types with CNN-based encoders [39]. Siamese CNNs have shown robust performance in classifying wireless signals under low SNR conditions, excelling in scenarios with limited samples and noisy inputs [40]. Additionally, ensemble and wavelet-based approaches have been proposed to maintain high classification accuracy for noisy signals, addressing the limitations of traditional classifiers in noise-interfered environments [41]. However, unlike self-supervised learning, these works heavily rely on supervised learning and require substantial amounts of labeled data to perform well in new deployment scenarios.

IX. Conclusion

In this paper, we have argued the importance of micro foundation models (μFMs) for the IoT community. μFMs serve as a practical solution to address the challenges in edge intelligence. Our evaluation has demonstrated that μFMs are significantly more robust than the supervised model during deployment and support efficient adaptation to different downstream tasks. These properties of μFMs are essential in improving the resiliency and efficiency of intelligent IoT applications.

X. Acknowledgements

Research reported in this paper was sponsored in part by DEVCOM ARL under Cooperative Agreement W911NF-17-2-0196 (ARL IoBT CRA), and in part by NSF CNS 20-38817, and the Boeing Company. It was also supported in part by ACE, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. The views and conclusions contained in this document are those of the authors, not the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] T. Kimura, J. Li, T. Wang, Y. Chen, R. Wang, D. Kara, M. Wigness, J. Bhattacharyya, M. Srivatsa, S. Liu, M. Srivastava, S. Diggavi, and T. Abdelzaher, "Vibrofm: Towards micro foundation models for robust multimodal iot sensing," in *2024 IEEE 21th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. IEEE, 2024.

[2] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *International Conference on World Wide Web*, 2017.

[3] S. Yao, Y. Zhao, H. Shao, S. Liu, D. Liu, L. Su, and T. Abdelzaher, "Fastdeepiot: Towards understanding and optimizing neural network execution time on mobile and embedded devices," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, 2018, pp. 278–291.

[4] S. Liu, S. Yao, X. Fu, R. Tabish, S. Yu, A. Bansal, H. Yun, L. Sha, and T. Abdelzaher, "On removing algorithmic priority inversion from mission-critical machine inference pipelines," in *2020 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 2020, pp. 319–332.

[5] Z. Chang, S. Liu, X. Xiong, Z. Cai, and G. Tu, "A survey of recent advances in edge-computing-powered artificial intelligence of things," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 849–13 875, 2021.

[6] S. Yao, Y. Zhao, A. Zhang, S. Hu, H. Shao, C. Zhang, L. Su, and T. Abdelzaher, "Deep learning for the internet of things," *Computer*, vol. 51, no. 5, pp. 32–41, 2018.

[7] T. Wang, D. Kara, J. Li, S. Liu, T. Abdelzaher, and B. Jalaian, "The methodological pitfall of dataset-driven research on deep learning: An iot example," in *MILCOM 2022-2022 IEEE Military Communications Conference (MILCOM)*. IEEE, 2022, pp. 1082–1087.

[8] S. Liu, T. Kimura, D. Liu, R. Wang, J. Li, S. Diggavi, M. Srivastava, and T. Abdelzaher, "Focal: Contrastive learning for multimodal time-series sensing signals in factorized orthogonal latent space," in *Advances in Neural Information Processing Systems*, 2023.

[9] D. Kara, T. Kimura, S. Liu, J. Li, D. Liu, T. Wang, R. Wang, Y. Chen, Y. Hu, and T. Abdelzaher, "Freqmae: Frequency-aware masked autoencoder for multi-modal iot sensing," in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 2795–2806.

[10] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[11] J. M. Whitacre and A. Bender, "Networked buffering: a basic mechanism for distributed robustness in complex adaptive systems," *Theoretical Biology and Medical Modelling*, vol. 7, pp. 1–20, 2010.

[12] R. Frei and J. Whitacre, "Degeneracy and networked buffering: principles for supporting emergent evolvability in agile manufacturing systems," *Natural computing*, vol. 11, pp. 417–430, 2012.

[13] A. Kott and T. F. Abdelzaher, "Resiliency and robustness of complex systems and networks." *Adaptive, Dynamic, and Resilient Systems*, vol. 67, pp. 67–86, 2014.

[14] J. Li, Y. Chen, T. Kimura, T. Wang, R. Wang, D. Kara, Y. Hu, L. Wu, W. A. Hanafy, A. Souza *et al.*, "Acies-os: A content-centric platform for edge ai twinning and orchestration," in *2024 33rd International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2024, pp. 1–9.

[15] M. Singh, E. Fuenmayor, E. P. Hinchy, Y. Qiao, N. Murray, and D. Devine, "Digital twin: Origin to future," *Applied System Innovation*, vol. 4, no. 2, p. 36, 2021.

[16] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *IEEE/CVF International Conference on Computer Vision (CVPR)*, 2021.

[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[18] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning (ICML)*, 2020.

[19] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *IEEE/CVF International Conference on Computer Vision (CVPR)*, 2021.

[20] 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 (ICML)*, 2021.

[21] Y. Liu, Q. Fan, S. Zhang, H. Dong, T. Funkhouser, and L. Yi, "Contrastive multimodal fusion with tupleinfonce," in *IEEE/CVF International Conference on Computer Vision (CVPR)*, 2021.

[22] R. Nakada, H. I. Gulluk, Z. Deng, W. Ji, J. Zou, and L. Zhang, "Understanding multimodal contrastive learning and incorporating unpaired data," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

[23] R. Brinzea, B. Khaertdinov, and S. Asteriadis, "Contrastive learning with cross-modal knowledge mining for multimodal human activity recognition," in *International Joint Conference on Neural Networks (IJCNN)*, 2022.

[24] Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu, "Ts2vec: Towards universal representation of time series," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

[25] E. Eldele, M. Ragab, Z. Chen, M. Wu, C. K. Kwoh, X. Li, and C. Guan, "Time-series representation learning via temporal and contextual contrasting," in *Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[26] S. Tonekaboni, D. Eytan, and A. Goldenberg, "Unsupervised representation learning for time series with temporal neighborhood coding," in *International Conference on Learning Representations (ICLR)*, 2021.

[27] P. Poklukar, M. Vasco, H. Yin, F. S. Melo, A. Paiva, and D. Kragic, "Geometric multimodal contrastive representation learning," in *International Conference on Machine Learning (ICML)*, 2022.

[28] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *European Conference on Computer Vision (ECCV)*, 2020.

[29] X. Ouyang, X. Shuai, J. Zhou, I. W. Shi, Z. Xie, G. Xing, and J. Huang, "Cosmo: Contrastive fusion learning with small data for multimodal human activity recognition," in *International Conference on Mobile Computing And Networking (MobiCom)*, 2022.

[30] S. Deldari, H. Xue, A. Saeed, D. V. Smith, and F. D. Salim, "Cocoa: Cross modality contrastive learning for sensor data," *ACM Interact. Mob. Wearable Ubiquitous Technol. (IMWUT)*, vol. 6, no. 3, 2022. [Online]. Available: https://doi.org/10.1145/3550316

[31] 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*, vol. 33, pp. 1877–1901, 2020.

[32] 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.

[33] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[34] L. Kong, M. Q. Ma, G. Chen, E. P. Xing, Y. Chi, L.-P. Morency, and K. Zhang, "Understanding masked autoencoders via hierarchical latent variable models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7918–7928.

[35] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu, "Simmim: A simple framework for masked image modeling," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 9653–9663.

[36] Y. Gong, A. Rouditchenko, A. H. Liu, D. Harwath, L. Karlinsky, H. Kuehne, and J. R. Glass, "Contrastive audio-visual masked autoencoder," in *The Eleventh International Conference on Learning Representations*, 2022.

[37] P.-Y. Huang, H. Xu, J. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer, "Masked autoencoders that listen," *Advances in Neural Information Processing Systems*, vol. 35, pp. 28 708–28 720, 2022.

[38] T. T. An and B. M. Lee, "Robust automatic modulation classification in low signal to noise ratio," *IEEE Access*, vol. 11, p. 7860–7872, 2023. [Online]. Available: http://dx.doi.org/10.1109/access.2023.3238995

[39] J.-D. Wu, W.-J. Luo, and K.-C. Yao, "Acoustic signal classification using symmetrized dot pattern and convolutional neural network," *Machines*, vol. 10, no. 2, p. 90, Jan. 2022. [Online]. Available: http://dx.doi.org/10.3390/machines10020090

[40] Z. Langford, L. Eisenbeiser, and M. Vondal, "Robust signal classification using siamese networks," in *Proceedings of the ACM workshop on wireless security and machine learning*, 2019, pp. 1–5.

[41] P. Grant and M. Z. Islam, *A Novel Approach for Noisy Signal Classification Through the Use of Multiple Wavelets and Ensembles of Classifiers*. Springer International Publishing, 2019, p. 195–203. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-35231-8_14