PATCH: A Plug-in Framework of Non-blocking Inference for Distributed Multimodal System

JUEXING WANG, Michigan State University, USA
GUANGJING WANG, Michigan State University, USA
XIAO ZHANG, Michigan State University, USA
LI LIU, Michigan State University, USA
HUACHENG ZENG, Michigan State University, USA
LI XIAO, Michigan State University, USA
ZHICHAO CAO, Michigan State University, USA
LIN GU, RIKEN AIP, JAPAN The University of Tokyo, JAPAN
TIANXING LI, Michigan State University, USA

Recent advancements in deep learning have shown that multimodal inference can be particularly useful in tasks like autonomous driving, human health, and production line monitoring. However, deploying state-of-the-art multimodal models in distributed IoT systems poses unique challenges since the sensor data from low-cost edge devices can get corrupted, lost, or delayed before reaching the cloud. These problems are magnified in the presence of asymmetric data generation rates from different sensor modalities, wireless network dynamics, or unpredictable sensor behavior, leading to either increased latency or degradation in inference accuracy, which could affect the normal operation of the system with severe consequences like human injury or car accident. In this paper, we propose PATCH, a framework of speculative inference to adapt to these complex scenarios. PATCH serves as a plug-in module in the existing multimodal models, and it enables speculative inference of these off-the-shelf deep learning models. PATCH consists of 1) a Masked-AutoEncoder-based cross-modality imputation module to impute missing data using partially-available sensor data, 2) a lightweight feature pair ranking module that effectively limits the searching space for the optimal imputation configuration with low computation overhead, and 3) a data alignment module that aligns multimodal heterogeneous data streams without using accurate timestamp or external synchronization mechanisms. We implement PATCH in nine popular multimodal models using five public datasets and one self-collected dataset. The experimental results show that PATCH achieves up to 13% mean accuracy improvement over the state-of-art method while only using 10% of training data and reducing the training overhead by 73% compared to the original cost of retraining the model.

CCS Concepts: • **Computing methodologies → Multi-task learning**; • **Computer systems organization → Cloud computing**; • **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

Authors' addresses: Juexing Wang, wangjuex@msu.edu, Michigan State University , 428 S Shaw Ln, East Lansing, Michigan, USA , 48824; Guangjing Wang, wanggu22@msu.edu, Michigan State University , 428 S Shaw Ln, East Lansing, Michigan, USA , 48824; Xiao Zhang, zhan1387@msu.edu, Michigan State University , 428 S Shaw Ln, East Lansing, Michigan, USA , 48824; Li Liu, liuli9@msu.edu, Michigan State University , 428 S Shaw Ln, East Lansing, Michigan, USA , 48824; Huacheng Zeng, hzeng@msu.edu, Michigan State University , 428 S Shaw Ln, East Lansing, Michigan, USA , 48824; Li Xiao, lxiao@cse.msu.edu, Michigan State University , 428 S Shaw Ln, East Lansing, Michigan, USA , 48824; Zhichao Cao, Michigan State University , 428 S Shaw Ln, East Lansing, Michigan, USA , 48824, caozcthu@gmail.com; Lin Gu, RIKEN AIP, Tokoyo, Tokoyo, JAPAN and The University of Tokyo, Tokoyo, Tokoyo, JAPAN, lin.gu@riken.jp; Tianxing Li, Michigan State University , 428 S Shaw Ln, East Lansing, Michigan, USA , 48824, litianx2@msu.edu.

2474-9567/2023/9-ART130 $15.00
https://doi.org/10.1145/3610885

Additional Key Words and Phrases: Multimodal Learning, Neural Networks, Multi-task Learning, Non-blocking Inference

ACM Reference Format:
Juexing Wang, Guangjing Wang, Xiao Zhang, Li Liu, Huacheng Zeng, Li Xiao, Zhichao Cao, Lin Gu, and Tianxing Li. 2023. PATCH: A Plug-in Framework of Non-blocking Inference for Distributed Multimodal System . *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 3, Article 130 (September 2023), 24 pages. https://doi.org/10.1145/3610885

1 INTRODUCTION

Recent years have seen substantial interest in deploying deep neural networks on Internet of Things (IoT) systems that combine data from a large number of distributed sensors and analyze them to derive actionable insights [62]. The emerging of IoT-cloud systems using multimodal inference brings great opportunities to significantly improve inference accuracy and robustness under complex scenarios, such as extreme weather conditions and no line of sight [15, 72] and greatly overcome the limitation of single-modality in the complex scenarios. These systems has shown significant potential in healthcare [38], activity recognition [61, 71], event detection [66, 70], and autonomous driving [14].

Real-world multimodal systems always require accurate predictions in a timely manner. Most of the popular multimodal models are trained and make their predictions based on intact well-constructed data. However, the data collected by the multimodal systems may become corrupted due to various factors, such as fluctuations in network bandwidth, power failures, abnormal sensor statuses, and noisy environments. Some of these factors introduce noise to the collected data, which is difficult to be discerned the true signal. Other factors, like fluctuations in network bandwidth, can result in partial or complete loss of data or transmission delay between the sensor and the server. Therefore, we use the term "corrupted data" or "data corruption" to refer to the data affected by (1) noise, (2) partial or complete loss, and (3) transmission delay between the sensor and server, and encompass all of the most common phenomena in this paper. For example, in cellular networks using LTE, the average throughput can fluctuate from 3 Mbps to 300 Kbps within one second, causing a nine-second delay when transferring a 3 Mb audio file [36]. In addition, existing wireless sensor networks may have thousands of sensor nodes with limited battery capabilities. When the battery level is below a certain level (e.g., 70% [19]), the reliability of the wireless sensors drops to 0.57, causing significant sensor failure and data loss. On the other hand, real-time and accurate predictions are crucial for real-world multimodal systems. For an autonomous driving car at 40 km per hour in an urban area, the desired response time of the system must be less than 90ms [52] to have timely feedback and avoid severe consequences.

Prior efforts have been exploring the problem of model partition [50] and model compression [7] to deal with this issue. However, there has been less work on the corrupted heterogeneous data imputation challenge to reduce the multimodal system latency while maintaining the system's accuracy. Recent work also demonstrated the capability of speculative inference to adapt to asymmetries data generation rates across modalities and other abnormal sensor behaviors [51]. However, three fundamental limitations still remain. First, the existing method only works for the specified type of multimodal model (late-fusion[23]) since it leverages fixed feature pairs to perform the data imputation for the missing/corrupted part of the data, which limits the system's compatibility and usability and degrades its performance. Second, the system is not compatible with heterogeneous input features. It requires substantial feature engineering efforts to construct a symmetric feature structure and make a hard-coding feature selection. These two vulnerabilities lead to significant performance drops (up to 13% in our experiments). Meanwhile, the training overhead is high in complex multimodal models (20 times more than our method.) Finally, the data alignment in the system is not an individual section. It depends heavily on the model of data imputation, which not only increase the computing load but also bring more unstable factor to the alignment result.

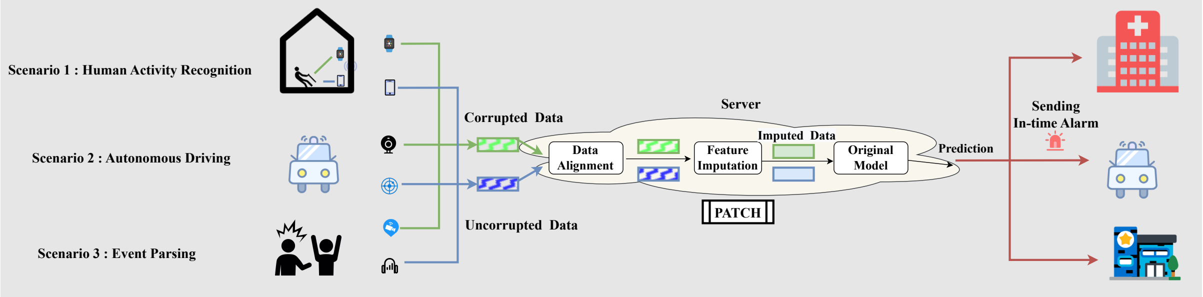


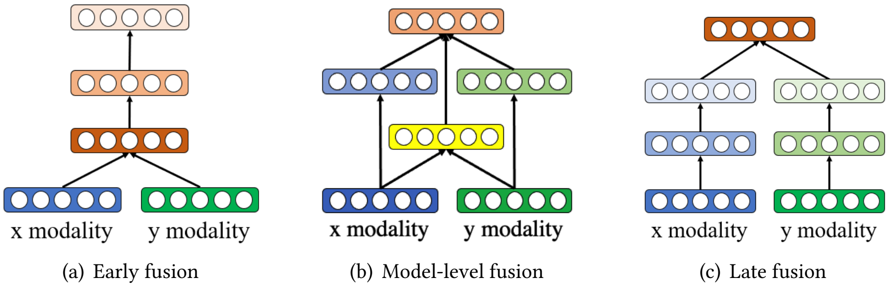
Fig. 1. Application scenarios of PATCH in real-time multimodal systems.

The focus of this paper is to overcome the above limitations and unleash the practical potential of non-blocking inference in existing distributed multimodal systems. To this end, we propose PATCH, the first inference framework that can be plugged into various existing multimodel to enable non-blocking inference under complex and noisy environments.

Figure 1 illustrates three application scenarios of PATCH in Human Activity Recognition (HAR), Autonomous Driving, and Event Parsing. In the case of HAR, a mobile phone sensor and a smartwatch sensor collect data about a person's activities and behaviors for health monitoring. In the case of Autonomous Driving, a radar and a camera gather information about the road and surroundings for collision avoidance. Event Parsing involves the collaborative use of a camera and a microphone to gather information pertaining to security events. However, due to heterogeneous data sizes and fluctuations in sensor states, the server may receive partial or corrupted data from individual IoT devices. When corrupted data reaches the server, PATCH first aligns the heterogeneous data from various sensor modalities and uses the correlated features to impute the missing information. The imputed feature will be fed into the original multimodal model to generate the final prediction. Then, the server can deliver accurate and timely alerts, guidance, and incidents to hospitals, autonomous vehicles, and police departments, respectively.

Besides the above application scenarios, PATCH can benefit a variety of existing multimodal models to adapt to abnormal sensor behaviors and unpredictable network dynamics. First, unlike existing machine learning solutions which require retraining the entire original multimodal model to address the issue, the data imputation model of PATCH can be trained as a plug-in module, significantly reducing the training overhead and preventing potential privacy leakage risk [3]. Second, PATCH works for all fusion strategies, including early fusion [23, 56], model-level fusion [43], and late fusion[23], making it compatible with various existing multimodal models and sensor modalities. Third, the overall framework of PATCH is a lightweight module compared to the prior work [51]. It decreases the computation cost to one-twentieth and increases the system performance by 13%, dramatically expanding PATCH application scenarios.

To realize PATCH in practice, we face three main challenges. First, it is challenging to design a cross-modality imputation algorithm that is broadly applicable across various fusion strategies (e.g., early, middle, and late fusion) and heterogeneous data streams with different sources, dimensionalities, windows sizes, and noise. Second, due to the variety of existing multi-modal models, the number of feature imputation configurations, which denote the combination of potential base input features and target imputation features, can be enormous in existing multi-modal models (e.g., up to 500 configurations [13, 20, 59, 66]), introducing significant training overhead and requiring a large amount of training data. Finally, due to computation and network limitations, sensor data from IoT devices may not be well-synchronized in the cloud. And external synchronization mechanisms are not always available due to noisy environments or sensor malfunction.



(a) Early fusion (b) Model-level fusion (c) Late fusion

Fig. 2. Three multi-modal fusion strategies based on deep networks: early fusion, model-level fusion, and late fusion.

We designed three key components to address the above challenges. First, we propose a cross-modality imputation technique to predict the partial, corrupted, and heterogeneous sensor data across various modalities by leveraging Feature-level Masked AutoEncoder. We then design a lightweight feature ranking module to select the optimal data imputation configuration, which significantly limits the feature searching space and decreases the computation load of the searching process. Finally, we design a Transformer-based data alignment module with low computation overhead.

Our main contributions are as follows:

- We present PATCH, a generic plug-in framework composed of three modules: cross-modality feature imputation module via feature-level masked autoEncoder (FMAE), feature ranking module, and data alignment module. This framework is designed to facilitate non-blocking multimodal inference under challenging situations, including uneven data generation rates across modalities, noisy/missing/delayed sensor data, fluctuating wireless networks, and complex fusion models. Furthermore, these components significantly enhance the computational efficiency and robustness of PATCH.
- To demonstrate the compatibility and generality of PATCH, we evaluate the system under all fusion strategies using nine off-the-shelf multimodal models and five public datasets. We also evaluate PATCH in four real-world cases, including battery level variation, cellular network fluctuation, one end-to-end human activity recognition task based on a self-collected dataset, and one end-to-end event parsing task.
- Experimental results show that PATCH achieves up to 13% mean accuracy improvement and average 64% inference latency reduction against the state-of-the-art method [51] while only using 27.4% of the training time compared to the original cost of retraining the model. This has the potential to improve the existing system with minimal energy cost and privacy leakage risk.

2 BACKGROUND

In this section, we introduce the background of fusion strategies of multimodal systems and feature imputation.

2.1 Fusion Strategies of Multimodal System

A multimodal system is a system that integrates information from two or more different sources to overcome the limitations of a single signal modality and enhance its recognition precision. Figure 2 shows the three main strategies (early fusion, model-level fusion, and late fusion) in existing multimodal systems [23, 43, 47, 88].

2.1.1 Early Fusion. Early fusion is a conventional method of integrating features extracted from multiple data sources before they are processed by the neural network. The fusion layer is always positioned at the start of the neural network. For this strategy to be effective, it is necessary to have a strong alignment between various modalities so that the correlations can be captured at a high level [23, 56].

2.1.2 *Model-level fusion.*

Model-level fusion, also known as intermediate fusion, allows the system to learn a joint representation of all modalities by merging their individual representations into a single hidden layer [43]. This fusion layer is located at a middle depth within the neural network, providing the multimodal system with a more versatile fusion stage.

2.1.3 *Late Fusion.*

Late fusion can effectively handle heterogeneous data and minimize independent errors from different sensor modalities. In multimodal systems using the late fusion strategy, features extracted from raw data are processed separately within the neural network [23]. The fusion layer is located at the final stage, just before making predictions.

2.2 Feature Imputation

To mitigate the impact of delayed or corrupted data and lower the latency of multimodal systems, we utilize intermediate features from existing models for data imputation. A feature pair in this process consists of one input feature (either clean or damaged) and a corrupted target feature. The aim of feature imputation is to create a new feature from the input feature, which then replaces the corrupted target feature in the original neural network, resulting in improved system performance. There are three commonly used data imputation methods in the machine learning domain.

2.2.1 *Generative Adversarial Network.*

The Generative Adversarial Network (GAN) [6, 17, 34, 35, 39, 57, 81, 89, 94] is a deep convolutional neural network that is trained using adversarial methods for image-to-image translation. Its goal is to map between the input domain and target domain using unaligned image pairs. The work presented in MobiSys21 [51] uses the CycleGAN approach [94] and provides a novel solution for low-latency, distributed multimodal systems. However, the CycleGAN is designed for image-to-image tasks, and is not well-suited for imputing heterogeneous multimodal data. The rigid data structure and fixed position of corrupted intermediate features are the main limitations of [51].

2.2.2 *Variational Autoencoder.*

Variational Autoencoders (VAEs) [4, 18, 26, 33, 40, 85]consist of an encoder, a decoder, and a loss function. The encoder transforms the input data into a latent space through a convolutional or linear network. Instead of forwarding the latent values directly to the decoder, VAEs calculate a mean and standard deviation to regularize the input to the decoder and produce a latent space with desirable properties. Our baseline method involves combining the VAE [45] and GAN [94] approaches to establish a strong baseline result. The results of the experiment showed that the Variational Autoencoder (VAE) method is not suitable for inputs with short durations, which are commonly seen in multimodal systems for applications like autonomous driving and human activity recognition.

2.2.3 *Masked Autoencoder.*

The masked autoencoder (MAE) is a cutting-edge method for self-supervised computer vision tasks [31]. The MAE [31] model masks randomly selected patches of the input image before it is fed into the encoder transformer. By incorporating an extra mask function, the model is able to reconstruct the desired feature with increased precision efficiently. Our Feature-level Masked Autoencoder (FMAE) method leverages this concept by allowing for the choice of either a clean or corrupted input feature as the input. By dividing the input feature into several patches, the FMAE method can perform a more detailed simulation for short-duration target features, such as features from autonomous driving or human activity recognition.

3 SYSTEM DESIGN

We introduce PATCH, a generic framework of non-blocking inference for multimodal learning. The system takes data from corrupted or noisy data streams as the inputs, imputes intermediate features, and proactively computes final predictions within the original multimodal models. Compared with the state-of-the-art approach [51],

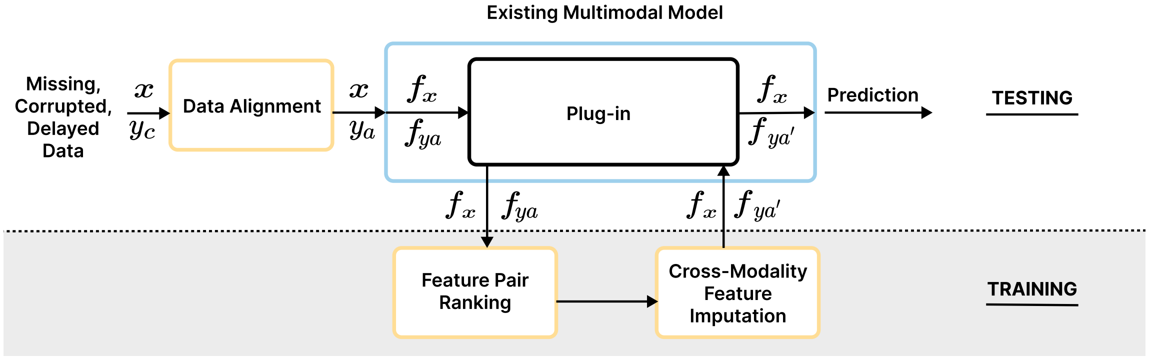


Fig. 3. Overview of PATCH. Three key modules (Data Alignment, Feature Pair Ranking, FMAE) and the workflow.

our system provides more compatibility and flexibility to various multimodal models and reduces computation overhead while maintaining good accuracy.

Figure 3 illustrates the overview of PATCH, containing a plug-in cross-modality feature imputation module, a lightweight feature pair ranking module, and a data alignment module. Given a pair of corrupted unsynchronized data x (*i.e.* sound modality) and y_c (*i.e.* image modality), these corrupted input data initially be aligned by our data alignment module. Then, the aligned data x and y_a are sent to the original multimodal system to extract the intermediate features f_x and f_{ya}. Feature Pair Ranking module exhausts all possible imputation configurations inside the existing multimodal system and searches for the optimal feature pair configuration with minimum noise and maximum mutual information. The optimal configuration is applied to the Feature Masked AutoEncoder (FMAE) module to impute target latent features $f_{ya'}$ from f_x and f_{ya}. Finally, the reconstructed feature $f_{ya'}$ and feature f_x are sent back to the original multimodal model and generate the speculative inference.

3.1 Cross-modality Feature Imputation Using Feature Masked AutoEncoders

The cross-modality feature imputation serves as a plug-in module in the original multimodal model to impute any missing, corrupted, or delayed data. Existing methods mainly focus on generating raw modality data from other modality data using Generative Adversarial Networks (GAN) [6, 17, 34, 35, 39, 57, 81, 94] or Autoencoders [4, 18, 26, 33, 40, 85]. However, the synthesized sensor data often lose significant details and limit the system performance of the original model due to inherent differences between the raw sensor data and their noise sources. Rather than imputing the raw sensor data, we propose to impute intermediate features within the original multimodal models. Specifically, we design a Feature Masked AutoEncoder (FMAE) to impute the corrupted sensor data on the feature level. FMAE deliberately masks the intermediate feature with random patches and then attempts to reconstruct its missing/masked data with a Vision Transformer (ViT) structure from unmasked patches. It can force ViT to directly learn the latent structure from massive amounts of sensor data with this training technique while using a small amount of computing and memory. Utilizing the mask function can boost the accuracy and efficiency of our model when reconstructing the desired feature. It can swiftly substitute the masked clean data with the structure of the target feature and expedite the computing process for the masked corrupted data by promptly refilling it. Unlike the traditional Masked AutoEncoders[31] that can only learn the data structure by reconstructing from raw sensor data, our FMAE can recover cross-modality latent features inside the original multimodal model. Our FMAE also differs from the original MAE in the patch-dividing strategy. Unlike MAE partition image data into several patches, FMAE handles latent features $f^{Nx1xL} \in R^{NxCxL}$ and treats each channel in C as a patch.

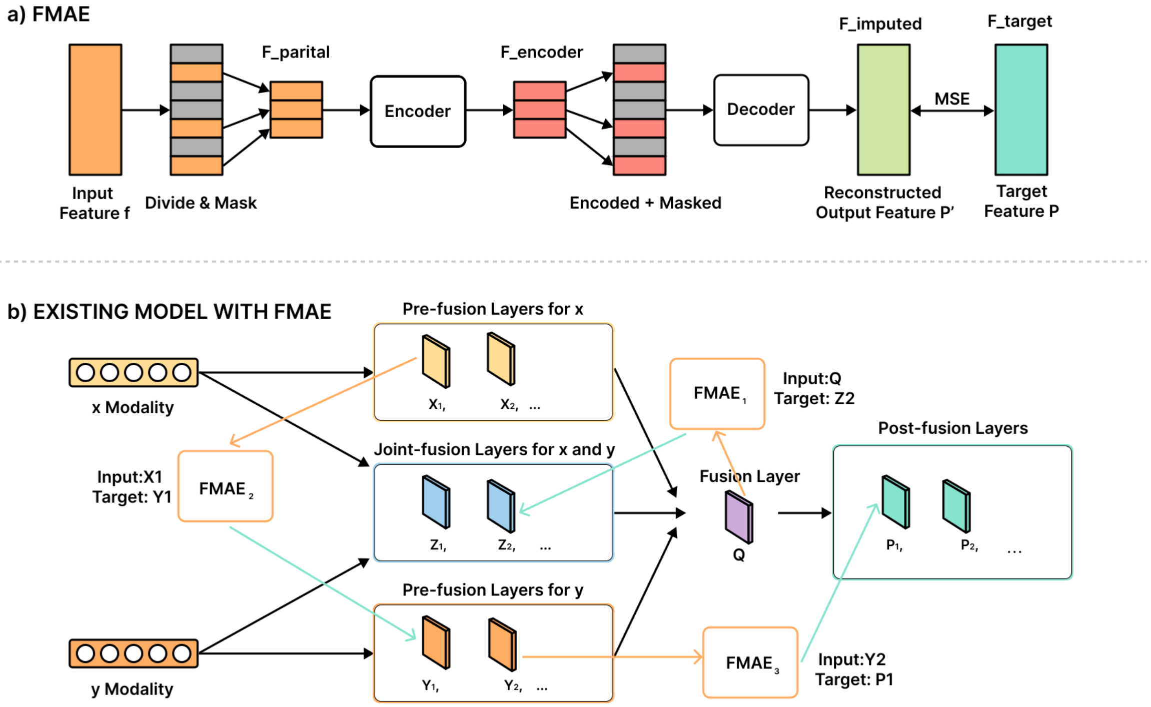


Fig. 4. An overview of cross-modality feature imputation. (a) Feature Masked AutoEncoder (FMAE) includes an asymmetric encode-decode architecture, where the encoder takes in only the visible patches, and the decoder reconstructs the target feature. (b) Existing multimodel models with FMAE as a plug-in module, where FMAE leverages the optimal feature pair configuration to impute missing features within the original models.

Figure 4 illustrate the overview of FMAE, which leverages input features Y to impute target features P. In the training phase, Y is split into n patches and masked by a function using a high mask ratio (0.75 in our implementation). Then the remaining features $F_{partial}$ are sent to the encoder part. For encoding, PATCH leverages Vision Transformer (ViT) [21] as the encoder $g_{encoder}$ to embed the unmasked patches $F_{partial}$ into latent representation $F_{encoder}$, which can be derived in Eq.1.

$$F_{encoder} = g_{encoder}(F_{partial}) \tag{1}$$

For decoding, both the latent representation $F_{encoder}$ and masked feature patches $F_{mask}$are applied to the decoder $g_{decoder}$ with a positional embedding h to maintain the organized structure of imputed feature $F_{imputed}$, which can be derived in Eq.2.

$$F_{imputed} = h(g_{decoder}(F_{encoder} + F_{mask})) \tag{2}$$

Finally, we compute the loss $\mathcal{L}_{\in}$ in Eq. 3 using all of the target feature $f_{target}(i)$ and the reconstructed imputed feature $f_{imputed}(i)$. Both masked and unmasked patches of features are included for the imputed feature.

$$\mathcal{L} = \frac{1}{n} \sum_{i \in n} |\mathbf{F}_{imputed}(i) - \mathbf{F}_{target}(i)|^2 \tag{3}$$

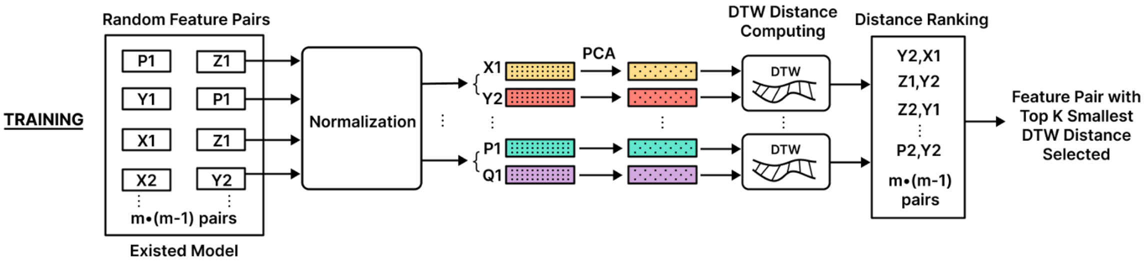


Fig. 5. Overview of Feature Pair Ranking Algorithm. PATCH leverages the top-K most correlated intermediate feature pair to filter out irrelevant imputation configurations.

Figure 4(b) illustrates how FMAE serves as plug-in modules in off-the-shelf multimodal models. Specifically, various intermediate feature pairs within the existing model are sent to the feature imputation modules $FMAE_i$. After imputing features using FMAE, the imputed feature $F_{imputed}$ is transmitted back to the original multimodel model and generates the non-blocking inference.

To further improve the robustness of the feature imputation module, we utilize the Multi-Task Learning(MTL) method and leverage the hard parameter-sharing approach in the training phase. MTL involves training a model to perform multiple tasks simultaneously, while hard parameter sharing allows the intermediate feature of the existing multimodal model to acquire information from both the FMAE module and the final prediction. We consider the FMAE functions that recover the corrupted features as an auxiliary task to the main task of the original model in MTL, and merge the FMAE loss function with the original model's loss function. MTL not only enhances the robustness of the feature imputation module but also lowers the risk of overfitting. Our experiments in Sec. 4.7 show that MTL works well in all existing multimodal models, improving an average of 8.5% accuracy compared to the single FMAE module.

3.2 Feature Ranking Using the Top-K Most Correlated Feature Pairs

This module aims to search for the optimal feature pair that produces the best cross-modality feature imputation performance in the imputation configuration. Existing multi-modal models often involve complex learning models like Alex-Net and VGG-Net [13, 20, 46, 59, 64, 66]. For example, when the multimodal system comprises 20 intermediate features spanning two modalities, the number of possible data imputation configurations for these deep learning models can exceed 380. Therefore, the computation overhead can be significant if we use the naive brutal-force method to rank all imputation configurations. In this paper, we design a lightweight feature ranking algorithm to effectively search for the optimal imputation configuration (pair of the input and the target features) with minimum computation overhead.

During the feature imputation experiments, we observe that *when a pair of intermediate features are highly correlated, the imputed features always provide significant cross-modality information and thus improve the inference accuracy of the original multimodal model.* Inspired by this observation, PATCH ranks intermediate feature pairs by their correlation and then filters out less-correlated feature pairs.

Figure 5 shows the overview of the feature pair ranking algorithm. Specifically, the intermediate features P_1, Z_1... from the existing model are randomly selected and normalized to minimize the impact of the absolute value divergence between different modalities. However, the vast size of the intermediate feature introduces significant overhead on computing feature correlations. We leverage Principal Component Analysis (PCA) to reduce the dimensionality of all normalized intermediate features linearly. Eq. 4 derives the intermediate features

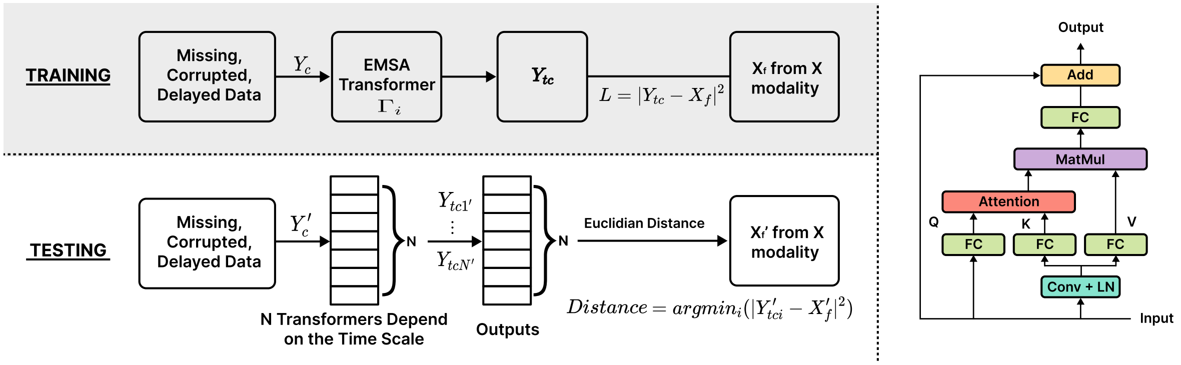


Fig. 6. Overview of Data Alignment Module. The module leverages the inherent similarity between sensor modalities to align multi-modal data streams.

$f_i'^{(CxL_2)}$ after applying PCA.

$$f_i'^{(CxL_2)} = \phi_{pca}(f_i^{(CxL)}) \quad (L_2 \le C < L) \tag{4}$$

where $\phi_{pca}()$ is the PCA function, L is the original dimensionality, and L_2 is the reduced dimensionality.

After dimension reduction, PATCH ranks the feature pair configuration by computing the Dynamic Time Warping (DTW) distance among all feature pairs $Y_2X_1, Z_1Y_2, \cdots$. Specifically, w_1, w_2,..w_{L_2} make up the warp route W between feature f_1' and feature f_2' , where w_m is the m^{th} element of the warp path. The warping path starts at $w_1 = (1, 1)$ and finishes at $w_{L_2} = (L_2, L_2)$. The path must traverse every component of the two input features. Then, PATCH can filter out less-correlated feature configurations based on DTW distance $Dist(f_1', f_2')$, derived in Eq. 5.

$$Dist(f_1', f_2') = \sum_{k=1}^{k=L_2} Dist(f_1'[k, i], f_2'[k, j]) \tag{5}$$

Using the most correlated feature pair as the optimal imputation configuration risks incorrectly eliminating the true optimal imputation configuration. For example, intermediate features from the same modality also have strong correlations with each other, but the imputed features will not generate any meaningful cross-modality information to improve the accuracy of the original multimodal model. Based on this observation, we hypothesize that while the most correlated feature pair and the optimal imputation configuration often do not match, the optimal imputation configuration is highly likely to fall within the *top-K* (K = 20% in our experiments) most correlated feature pairs. Therefore, PATCH searches for the optimal imputation configuration using FMAE within the top-K most correlated feature pairs instead of just the top result.

3.3 Data Alignment Using Transformer

The aim of this module is to synchronize the data streams of multiple modalities before feature imputation, all while avoiding dependence on external synchronization mechanisms, as these mechanisms may prove unreliable or inaccessible in complex environments. The key idea is to leverage the inherent similarity between sensor modalities to align multi-modal data streams.

Specifically, PATCH leverages full sensor data from one modality as the reference and then uses the corrupted partial sensor data from other modalities to impute the full sensor data. We have observed that heavy modalities with high data stream requirements or energy consumption are more susceptible to severe data missing or corruption. Therefore, we can select the lightweight modality as the reference modality for the data alignment

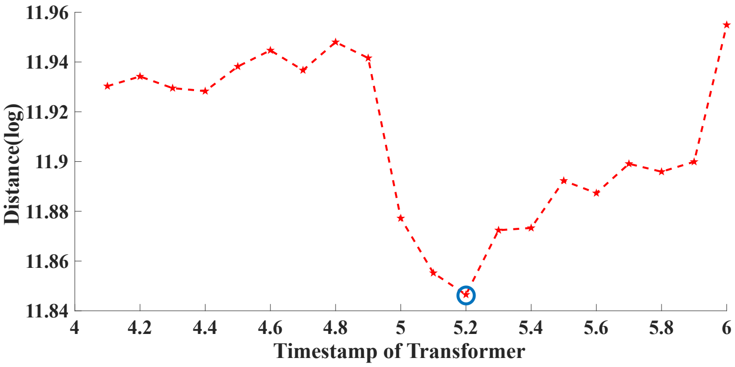


Fig. 7. An example of data alignment. The blue circle indicates the minimum L2 distance between the reference and the imputed sensor data.

module. To make our data alignment module broadly applicable to various sensor modalities, we design a Transformer-based algorithm to take advantage of one modality's clean sensor data.

Figure 6 shows the overview of the data alignment module as well as the Transformer model. The Transformer is a multi-head self-attention module that projects the input into three different matrices: query matrix **Q**, key matrix **K**, and value matrix **V**. The first step of the model is to perform the convolution and liner projection operation to the matrix **K** and **V** to compress memory space and obtain the query matrix **Q** by the line projection in the left side. After the reshaping process, the module computes the attention function on matrix **Q**, **K**, and **V**. The output values of each head are concatenated together, and the model performs the linear projection to the concatenated feature to generate the final output.

In the training phase, we train the Transformers Γ_i (i =1,2...N) under various delay configurations and get the reference features $X_f = \Gamma_{base}(f)$ from well-aligned raw sensor data. In the testing phase, PATCH leverages these N Transformers to generate N transformed sensor data $Y_{tc}1', Y_{tc}2' \cdots Y_{tc}N'$. Then, we align the multimodal data streams by computing the minimal $L2$ distance between the reference sensor data and the imputed sensor data. Eq. 6 shows the objective function of the data alignment module

$$D_{min} = \arg\min_{i=1}^{N} |\Gamma_i(\mathbf{Y}'_c) - \Gamma_{base}(f')|^2 = \arg\min_{i=1}^{N} |\mathbf{Y}_{tc}i' - X'_f|^2 \tag{6}$$

where D_{min} is minimum distance between our referral feature X'_f and the transformed corrupted data $\mathbf{Y}_{tc}i'$.

Figure 7 shows an example of the data alignment. The aligned data streams match the actual delay only if the $L2$ distance is the minimum. The misaligned data streams will introduce noise to the corresponding Transformer, increasing the $L2$ distance between the imputed sensor data and the reference. Unlike the prior efforts that rely on the structure of the multimodal model to align sensor data [51], PATCH does not assume the dimensionality of the sensor input and the structure of the multimodal model are *identical*. Thus, PATCH is broadly applicable to complex multimodal models and can align both structural and non-structural sensor data. Moreover, our data alignment module utilizes the raw data as the input for alignment to the Transformer, which prevents the buildup of errors from the decreased performance of the feature imputation module. This results in improved accuracy and efficiency.

Table 1. Experimental settings: we evaluate PATCH under four fusion strategies using nine off-the-shelf multimodal models and five public datasets.

Model Name	Dataset	Fusion Strategies	Applications	# of Feature Pairs	# of Modalities	Evaluation Metric (Ori.)	Modality Types
AVE [66]	AVE [66]	Late Fusion	Event Localization	30+	2	Accuracy	Audio + Video
CMRAN [76]	AVE [66]	Model-Level Fusion	Event Localization	50+	2	Accuracy	Audio + Video
PSP [93]	AVE [66]	Model-Level Fusion	Event Localization	50+	2	Accuracy	Audio + Video
AVVP [65]	LLP [65]	Late Fusion	Event parsing	50+	3	F1 score	Audio + Video(2D+3D)
MAAVVP [75]	LLP [65]	Late Fusion	Event parsing	50+	3	F1 score	Audio + Video(2D+3D)
DAN [24]	UCI OPPORTUNITY [10]	Early Fusion	Human Activity Recognition	30+	3	Accuracy	IMU Sensors
DDNN [61]	UCI HAR [2]	Early Fusion	Human Activity Recognition	20+	3	Accuracy	IMU Sensors
S-HAR	Self-collected	Early Fusion	Human Activity Recognition	30+	3	Accuracy	IMU Sensors
Transfuser1 [60]	CARLA [22]	Late Fusion	Autonomous Driving	30+	2	Loss Point	Image + Lidar
Transfuser2 [60]	CARLA [22]	Model-level Fusion	Autonomous Driving	30+	2	Loss Point	Image + Lidar

4 SYSTEM EVALUATION

In this section, we describe the experimental settings, datasets, and multimodal models we use, followed by a comprehensive evaluation.

4.1 Experimental Settings

4.1.1 Implementation. We implement PATCH on a server utilizing two Nvidia Tesla V100S with 32 GB RAM, an Nvidia Tesla K80 GPU with 16GB RAM, and an Intel(R) Xeon(R) Platinum 8260 CPU 2.40GHz as our main experimental platform. We also evaluate the performance baseline methods on the same device.

4.1.2 Baselines. We compare PATCH with four baselines. The first baseline is naive blocking, in which the cloud will ask the client to re-transmit the partial sensor data. The inference pipeline will be completely blocked until full sensor data is available in the cloud. The second, on the hand, is naive non-blocking, in which the cloud will leverage the partial-available sensor data to force the inference model to generate an output. The third baseline is retraining the multi-modal models themselves with varying amounts of data corruption ratio. The number of retrained models is equal to the number of data corruption configurations. During the test phase, it will measure the data corruption ratio and select a retrained model that leverages the partial data to get the outputs without blocking the inference pipeline. Finally, the most advanced technique currently available (MobiSys21 [51]), referred to as the fourth baseline, uses CycleGan and Variational Autoencoder (VAE) to imputed the delayed sensor data, allowing for inference without blocking.

4.1.3 Unified Evaluation Metric. In the evaluation, we aim to standardize the evaluation metric for the prediction results to facilitate consistent comparison across different multimodal models. Therefore, we define the accuracy loss as the evaluation metric, which is the accuracy difference between the original models (i.e., gold standard) without any data corruption and models with different data corruption ratios. A smaller accuracy loss indicates better system performance. We then define performance improvement as the normalized difference between the performance of PATCH and that of the baselines.

We conducted experiments with various types of data corruption, varying in percentage and distribution forms, to determine the variable that caused the most significant accuracy loss. For instance, in datasets containing input from both time and spatial domains (e.g., AVE) we found that a 50% corruption rate in the time domain had a more significant impact on system accuracy than 30% corruption rates in both the time and spatial domains separately (50% in total). Therefore, we chose data corruption in the time domain as the primary variable for AVE, LLP, and all HAR models. We selected spatial domain corruption as the primary variable for the CARLA dataset, which contains image and Lidar data. Moreover, we opted for the continuous distribution of partial data loss as the primary form of data corruption, as it produces more severe damage to the data and prediction results compared to a uniform distribution.

4.2 Multimodal Models and Datasets

Table 1 summarizes the off-the-shelf multimodal models and datasets we used in the evaluation. Specially, we implement both PATCH and the baseline methods in **nine** off-the-shelf multimodal models and evaluate the performance using **five** public datasets and **one** self-collected Human Activity Recognition(s-HAR) dataset. These multimodal models cover a variety of application scenarios (Event location, Event parsing, Activity Recognition, and Autonomous driving) and fusion strategies (Early fusion, Model-level fusion, and Late fusion) with numerous intermediate features (20+ to 50+) extracted from different sensor modalities (audio, video, IMU, LIDAR).

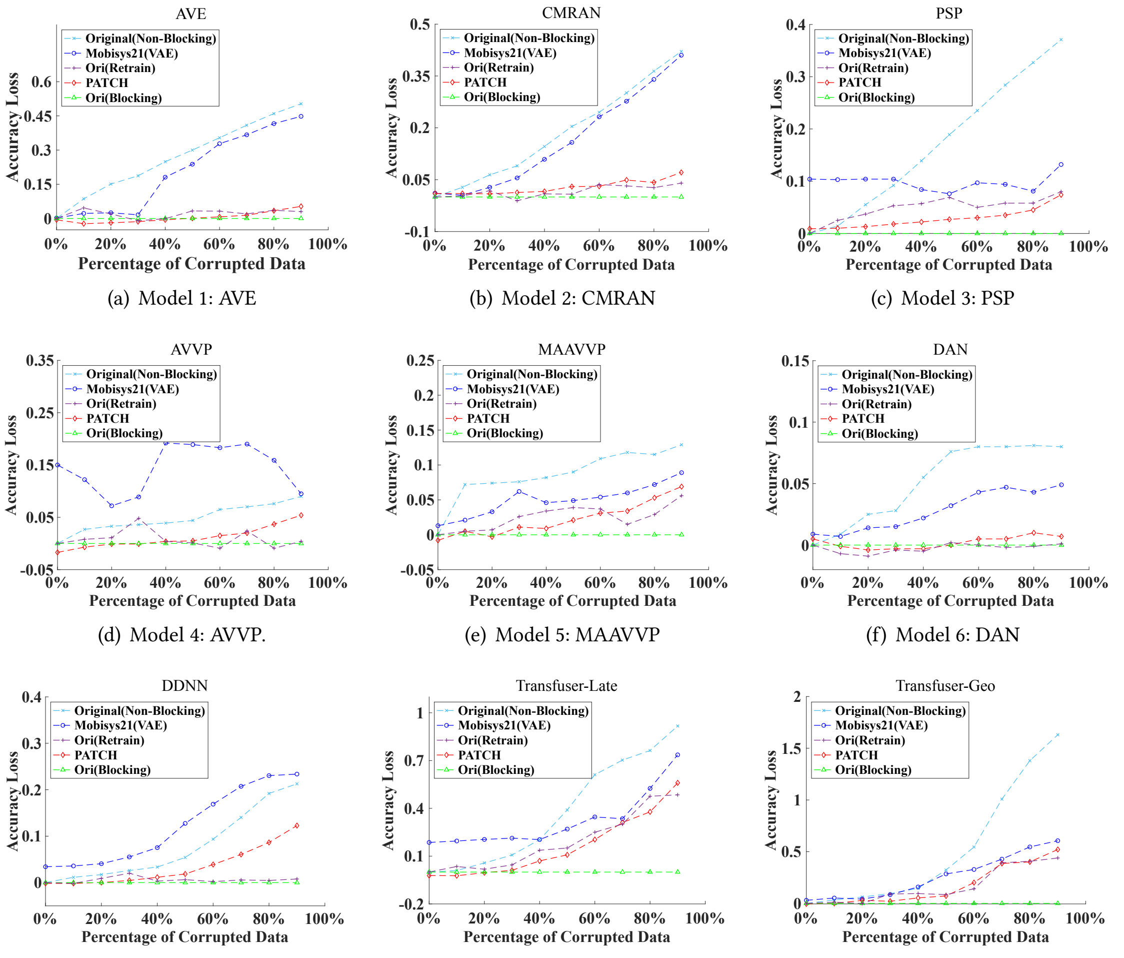
4.2.1 Dataset. The Event Detection Dataset (AVE)[66] contains over 4,000 10-second videos covering 28 audio-visual events. The Event Parsing Dataset(LLP)[65] contains 11849 YouTube video clips covering 25 event categories. The UCI HAR[2] dataset is recorded by 30 volunteers with a smartphone on their waist to record the information of the accelerometer and gyroscope in six types of activities. UCI Opportunity dataset [10] are collected from 4 volunteers from multiple body-worn sensors. The Autonomous Driving Dataset uses the same setting with [60] CARLA 0.9.10 data simulator for training and testing. It consists of 8 publicly available towns with around 2500 routes and 14 kinds of weather data.

4.2.2 Models. As shown in Table 1, the AVE [66], CMRAN [76], and PSP [93] models concentrate on the task of Audio-Visual Event Localization with two modalities and are trained on the AVE [66] dataset. The AVE model is a late fusion model, while the other two models are model-level fusion models. Each of these three models contains more than 30 available intermediate feature pairs that might be beneficial from the feature ranking and imputation step of PATCH. AVVP [65] and MAAVVP [75] models are trained on the LLP[65] dataset to explore more detailed event information from input sources with three modalities. Their multimodal models have three different kinds of features: audio features, 2D frame-level features, and 3D snippet-level features. More than 50 intermediate features and a complex model structure can be extracted from these two models. Both the DAN [24] and DDNN [61] models are early fusion models with three modalities. Before processing in the neural network, incoming data from various sources is fused at an early stage. The majority of these modality data are gathered by IMU sensors or mobile IoT devices. The Transfuser [60] model contains two individual multimodal model structures with different processing procedures and fusion strategies that contribute to the original system performance. Implementing PATCH with these off-the-shelf multimodal models can explore the affection of fusion type under the same model settings.

4.3 Overall Performance

We start with the end-to-end improvement of PATCH against the three baselines. To simulate the impact of missing, corrupted, and delayed sensor data in a complex environment, we randomly drop 0-90% of the sensor data in both time and spatial domains. For example, to drop 50% of sensor data, we first drop 30% of available frames and then drop 30% of available blocks within the not dropped frames. Blocks represent the smallest unit of dimensionality in the spatial domain for each input feature. This strategy can simulate data loss during data generation and transmission when the sensor data (i.e., images) are compressed using compression techniques like H.264 or H.265 Codec. For the baselines using original models (blocking or non-blocking), we assume the data streams are perfectly aligned since they need to rely on external synchronization mechanisms to align data streams. For PATCH and MobiSys21, we introduce misaligned data streams and leverage their data alignment modules to align sensor data prior to the data imputation module. Figure 8 summarizes the overall performance of PATCH as well as the three baselines.

4.3.1 PATCH vs. Original Models (Non-blocking). As shown in Figure 8, PATCH outperforms the original models with non-blocking strategy by 18.7% on the average improvement of accuracy loss, showing that PATCH is capable of dealing with missing, corrupted, or delayed sensor data in complex multimodal models. When more



(a) Model 1: AVE

(b) Model 2: CMRAN

(c) Model 3: PSP

(d) Model 4: AVVP.

(e) Model 5: MAAVVP

(f) Model 6: DAN

(g) Model 7: DDNN

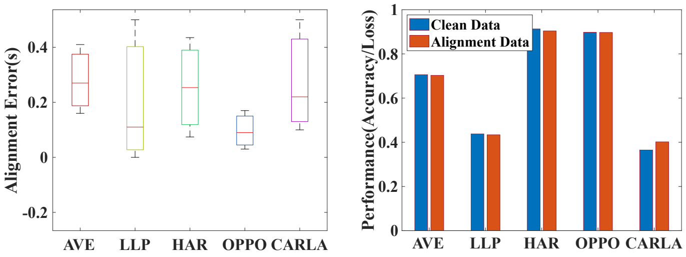
(h) Model 8: Transfuser-Late Fusion

(i) Model 9: Transfuser-Geometric Fusion

Fig. 8. Overall performance: End-to-end accuracy improvement between PATCH and the three baselines. In this experiment, we randomly mask the sensor data in both time and spatial domains.

sensor data are masked, the gain of PATCH over the original models (non-blocking) becomes larger. We observe that the complexity of the multimodal model also affects the gain of PATCH. For example, CMRAN contains more than six inference layers, including two fusion layers. In this case, PATCH outperforms the original models (non-blocking) by 40% since the masked sensor data significantly degrades the accuracy of the original multimodal model.

4.3.2 PATCH vs. Original Models (Blocking). To maintain high inference accuracy, the original models may impede the inference pipeline and request IoT devices to resend sensor data. However, this approach significantly increases the end-to-end inference latency, which may be unacceptable in many real-time systems. For instance, autonomous driving applications typically demand an end-to-end latency of less than 25 ms [11]. Therefore, the extra latency caused by retransmit the data (e.g., 50 ms) is not tolerable and may lead to severe accidents in the



(a) Data Alignment Errors.

(b) Inference Performance

Fig. 9. Impact of Data Alignment.

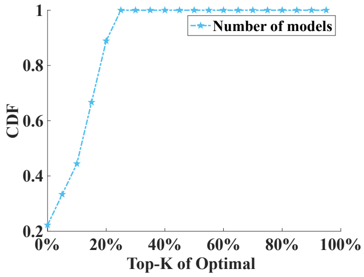
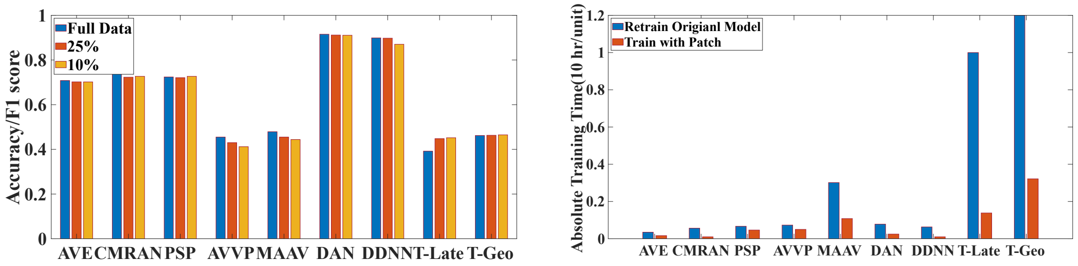


Fig. 10. Impact of Top-K Feature Pair Ranking

real world. Our observations indicate that when the data corruption ratio is below 50%, PATCH can achieve similar level inference accuracy to that of the original models that block the inference pipeline. In addition, the inference latency of PATCH is only 11% of the original models that block the pipeline, as shown in Fig 12. These findings demonstrate that PATCH can enable non-blocking inference without compromising the inference performance if the data corruption ratio is small.

4.3.3 PATCH vs. Original Models (Retrain). We observe that PATCH outperforms the retrained models in five models (AVE, PSP, MAVVP, T-Late, and T-Geo) with an average accuracy improvement of 2%, while it matches the performance of four other models (CMRAN, AVVP, DAN, DDNN) with a negligible difference. This performance was assessed across a range of data corruption ratios from 0% to 90%. In scenarios with a low data corruption ratio (less than 35%), our method significantly improved the average accuracy loss across nine models by 2.9%, compared to the results obtained from retraining models. However, in situations of high data corruption (greater than 65%), the retrained models exhibited superior performance. Overall, the average accuracy loss improvement is 0.3% against the retrained models. Additionally, as demonstrated in Fig. 11(b), PATCH significantly reduces the training overhead. On average, PATCH reduces the training overhead by 73% compared to retraining methods. This marked reduction in training time, coupled with the improvement in accuracy loss, underscores the superiority of our PATCH method over the conventional approach of retraining models.

4.3.4 PATCH vs. MobiSys21. The result in Fig. 8, shows that PATCH outperforms the state-of-the-art (MobiSys21[51]) by up to 13% mean accuracy loss improvement , meanwhile the inference latency of PATCH is only 36% of the MobiSys21 system. MobiSys21 leverages a hard-coded feature pair to impute intermediate features. Thus, the gain of PATCH will be significant if the hard-coded feature pair is not optimal for cross-modality data imputation. For the early-fusion multimodal models (e.g., Model 6: DAN and Model: DDNN), the hard-coded feature pair is not even within the top-20% best candidates, which significantly degrades the performance. Unlike MobiSys21, PATCH searches for the optimal feature pair that provides the most cross-modality information to the original multimodal model, which guarantees to generate optimal inference results in all models. Besides, the data imputation in MobiSys21 is based on Pix2Pix GAN [39] and Variational Autoencoder(VAE), which is designed to train a deep convolutional neural network for image-to-image reconstruction. Compared with the GAN and VAE, FMAE in PATCH is more generic and robust in reconstructing non-image data like audio or IMU data. Besides, the data alignment error of PATCH is only one-fourth of MobiSys21, improving system robustness in the case of long delays (more than 50% data drop) or complex multimodal models (30+ intermediate features). The alignment module in PATCH is an independent section that takes the raw sensor data as the referral base. However, the alignment method of MobSys21 strongly relies on the cross-modality imputation module, which degrades the system performance when the imputation module accumulates the inference error.



(a) Impact of Training Data Size.

(b) Absolute Training Time

Fig. 11. Impact of Training Overhead. In this experiment, we train FMAE using a subset (10% or 25%) of the training data

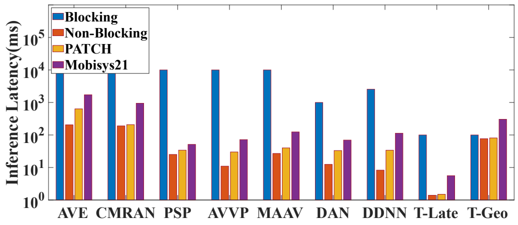


Fig. 12. Inference Latency.

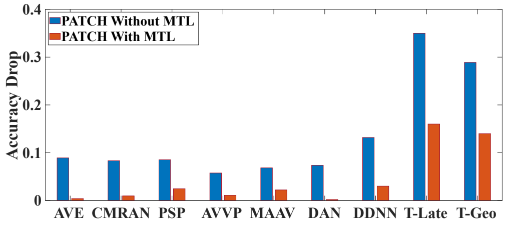


Fig. 13. Impact of Multi-Task learning.

4.4 Impact of Data Alignment

We now look at the impact of our data alignment module. In this experiment, we randomly set the ratio of data corruption uniformly distributed in the 0–90% range to the collected sensor data in the five datasets. Figure 9(a) shows the mean data alignment error in the second covering 90% confidence interval. Overall, the mean data alignment error is 0.09–0.27 seconds for these five datasets. Since we search for the minimal L2 distance between the imputed data and the reference, the minimum data alignment resolution is one frame (e.g., 0.1 s for 10 FPS video). Figure 9(b) shows the comparison of inference performance between perfect data alignment and our method. Since there is no statistical difference between the two methods, the data alignment error is negligible in PATCH. The data alignment module assumes that the data from the lightweight modality is clean and serves as a reference for aligning the corrupted modality. However, if the reference modality is also corrupted, the alignment module will be affected. Although the lightweight modality is typically more robust in multimodal systems and the Transformer-based alignment module can provide tolerance to the noise, the alignment error will accumulate when the data corruption ratio dramatically increases in the reference modality

4.5 Impact of Top-K Feature Pair Ranking

The Top-K feature pair ranking module is used to reduce the computation overhead of ranking feature pair configurations. Figure 10 summarizes the impact of the K across all these nine models with more than 400 feature pairs, which controls how many feature pairs will be reserved to search for the optimal imputation configuration. A large K will reserve more feature pairs, which guarantees the optimal imputation configuration will be selected by PATCH. However, a large K will also introduce significant training overhead since each imputation configuration needs to train a feature mapping using FMAE. We observe that the optimal feature pairs always fall within the top 20% of the feature pair for all the multimodal models we tested. Therefore, we select $K = 20\%$ as the best trade-off between accuracy and computation overhead in PATCH.

4.6 Impact of Training Overhead and Running Time

In this experiment, we assess the training overhead in the training phase and the inference latency of each model in the testing phase. Fig.11(a) illustrates the relationship between the training data size and the inference accuracy. Since PATCH serves as a plug-in module to the original model, it can achieve comparable inference accuracy with only 10% of the original training data. The T-Late and T-Geo models use the loss point as their evaluation metric, which leads to an inverse relationship between the metric value and system performance. A lower loss point value indicates better performance for these models, which is in contrast to the other models.

Fig.11(b) illustrates the training overhead of PATCH compared to the overhead of retraining the original model. Across the nine models considered in this study, the average training overhead of PATCH is only 27.4% of the time required for retraining the original model. The computation overhead mainly arises from training the FMAE to select the optimal feature pair. We observe that even for the most complex multimodal models, such as T-Late, the training overhead is only 13.8% of the retraining time for the original model.

Fig.12 presents the inference latency of each model in the testing phase.PATCH only need 11% and 36% inference time compared to the Blocking method and MobiSys 21, respectively. Although the inference latency of PATCH is higher than the Non-Blocking method due to the extra computing overhead of the plug-in framework, PATCH brings 14.8 % accuracy loss reduction on average, which makes the trade-off between accuracy and latency still worth. In systems where latency is critical, such as autonomous driving, PATCH introduces only a 6% average inference latency increase for the T-late and T-geo models. These results demonstrate that the PATCH is a practical approach in both training and testing phases for distributed multimodal systems.

Overall, given the varied neural network structures and features, PATCH offers an individual-specific solution for nine multimodal models designed for non-blocking inference. Compared to the solution of retraining the original model, PATCH significantly reduces the training overhead by an average of 72.6% using just 10% of the original training data. This makes the training cost for PATCH feasible. Additionally, it delivers an average prediction accuracy boost of 14.8% with a mere 6% increase in inference time for time-sensitive scenarios.

4.7 Impact of Multi-Task Learning

Figure 13 shows the impact of multi-task learning on PATCH. Overall, multi-Task learning brings a 9% accuracy improvement on average for the nine multimodal models, compared to the single FMAE task training strategy. For seven out of nine models, the accuracy difference between PATCH with MTL and the original models without data missing (gold standard) is less than 5%, significantly improving the robustness of PATCH against missing/corrupted sensor data. By setting the FMAE as an auxiliary task to the original model, PATCH avoids overfitting while providing more mutual information to the existing multimodal models.

4.8 Case Study

Finally, we evaluate PATCH in three real-world cases: battery reliability variation, wireless network fluctuation, and real-world human activity recognition.

4.8.1 End-to-End Human Activity Recognition Task.

In this study[1], we collect a Human Activity Recognition (HAR) dataset from 10 participants (8 males and 2 females with ages ranging from 20 to 30+). For each participant, we collect eight actions of their daily activities (e.g., touching surfaces, drinking, picking up the phone, walking, sitting, running) using a Moto 360 smartwatch and a Samsung Galaxy A02S smartphone. One participant wears

[1] Our study is conducted under the IRB approval at the local institution.

the smartwatch and puts the smartphone in their pocket during the data collection process. And we collect the accelerometer and the gyroscope data on the smartphone and smartwatch.

In this HAR task, data corruption occurs for two reasons. First, the position of the smartwatch is crucial for our system to perform stable activity recognition. However, the orientation/position of the smartwatch may change irrelevantly during the experiment process, such as moving up and down or rotating from side to side. The noise introduced by these changes in wearing status can obscure the true signal from the user activity. Secondly, due to the adapted Bluetooth communication between the smartwatch and smartphone, the throughput of the smartwatch is not always stable, owing to the sniff mode [95]. This fluctuation in throughput causes the Bluetooth throughput to drop to near-zero levels and interrupts data transmission. Data corruption resulting from improper smartwatch orientation or positioning makes up 10.1% of the entire dataset, while issues stemming from Bluetooth represent 7.9% of corrupted data. In public applications, users might not consistently ensure proper device positioning or maintain stable Bluetooth connections between smartphones and smartwatches, which could lead to even higher rates of data corruption than we observed in our experiment. These two factors contribute significantly to the misalignment and corruption of the data during the experiment.

In the PATCH System, the Data Alignment section serves as the preliminary processing step for input data under situations with unknown delays. The Data Alignment section will assess the data corruption ratio (ranging from 0% to 90%) of the input modality based on the reference modality and selects the corresponding setting for feature selection and pre-trained data imputation sections. Fig.14 illustrates the average accuracy of an RNN-based model with and without PATCH. Our study finds that PATCH significantly improves the accuracy of the original RNN-based model by about 4% across different users. Nonetheless, PATCH does not benefit every user in our study. For instance, as demonstrated in Fig.14, the prediction accuracy for User 5 decreased by 0.3% when using PATCH compared to the original RNN-based model. This drop in performance is attributed to imprecise data alignment, which leads to the inappropriate selection of the data corruption ratio and the corresponding pre-trained data imputation module. This misalignment impacts the performance of data imputation. Consequently, any anomalous behavior that results in inaccurate alignment can also constrain the effectiveness of PATCH. The overall experimental result shows that PATCH not only improves the stability of the self-build HAR model under network fluctuation and data corruption but also makes the original model more robust to user diversity and noise of the experiment.

4.8.2 End-to-End Event Parsing Task. In this study, we conducted an Event Parsing task in a real-world scenario. It is known that network bandwidth fluctuation is a common phenomenon in daily life. For instance, during an online ZOOM meeting, the network status affects sound and video quality. In most cases, while the sound continues, the video may stall at a previous scene when network bandwidth fluctuates, greatly affecting the quality of the meeting. Similarly, network fluctuation can affect multimodal systems, such as real-time event parsing based on video and audio inputs. In this study, we performed the Event Parsing task using the public AVE dataset with 400 test cases to simulate various daily events (like a dog barking or a train passing with a whistle). In the experiment, two mobile devices were used. The first device streamed films featuring specific events with synchronized video and audio signals. This device was carried by an individual moving through a predetermined route in a building. Due to uneven network coverage within the building, certain points along the route experienced robust network connections, while others suffered from poor connectivity, potentially resulting from signal blockage. As the device moved, it encountered varying network conditions, leading to potential data corruption. Meanwhile, the second device was stationed in a room with consistent and reliable network coverage. This stationary device received the video and audio signals transmitted by the mobile device and conducted event parsing. For the purposes of this study, we classified the corruption levels of the received signal into three categories: Low (with less than 35% corrupted data), Medium (with 35% to 65% corrupted data), and High (with over 65% corrupted data).

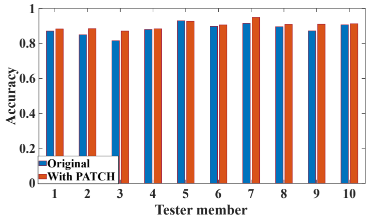
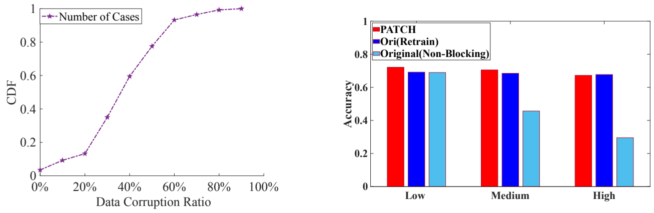


Fig. 14. System Performance on the Self-Collected Dataset of HAR Task.



(a) Distribution of data corruption ratio for all 400 cases.

(b) System Performance Comparison Between Differnt Level of Data Corruption

Fig. 15. Experiemtn Result on the End-to-End Event Parsing Task

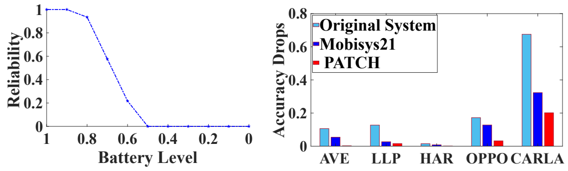
Fig.15(a) depicts the cumulative distribution of the data corruption ratio for 400 cases. While the results indicate there only 35% of the case are suffering data corruption at the Low level, more than 65% of cases are experiencing a medium or heavy level of data corruption, significantly reducing the accuracy of event parsing. Fig.15(b) shows a comparison between the prediction results of the original AVE model and PATCH across three levels of data corruption. PATCH outperforms the original model (non-blocking) and retrained model, achieving a prediction accuracy improvement of 4.9% and 1.8%. This study illustrates the compatibility of PATCH with various data corruptions that result from the partial/total data missing in real-world scenarios.

4.8.3 Power Reliability. The power reliability is strongly correlated to the work status of the wireless sensor. The abnormal working status of the sensor node will also affect the performance of the distributed multimodal system. Existing studies have revealed the correlation between the battery level of IoT nodes and the reliability of the sensor [19]. Based on this correlation, we simulate the battery drain curves when the sensor nodes were deployed in the wild. Fig. 16(a) shows an example of the battery drain curves. We then evaluate PATCH on various power reliability levels. Fig. 16(b) shows the mean accuracy drops under the three battery levels (e.g., 33%, 66%, 100%). Overall, the mean accuracy improvement is 10.4% across the five application scenarios when the battery level is low (e.g., 33%). Also, we observed the state-of-the-art suffers from a significant accuracy drop in autonomous driving (CARLA) and human activity recognition (OPPO), which can cause severe car accidents or delayed medical helps in practice.

4.8.4 Network Dynamics. Network dynamics play an essential role in the performance of multimodal systems. In this experiment, we evaluate PATCH using real-world wireless network traces [80]. The real-world network data was collected in a radius of 250 meters with 1184 clusters. We evaluated PATCH within 20 randomly selected separate clusters across this range. Fig. 17(a) shows an example of the wireless network traces when the user downloads a 1MB file in different clusters. We observe that the utilization rate ranges from 41% to 92%. Fig. 17(b) shows the mean accuracy drop under various network bandwidth fluctuations. We observe that PATCH improve the prediction accuracy by up to 20% compared to the original model and 13% against the state-of-the-art. For all network traces, PATCH achieves an acceptable accuracy drop level of 1.9%, increasing the stability of these systems and generating reliable predictions under various network bandwidth fluctuations.

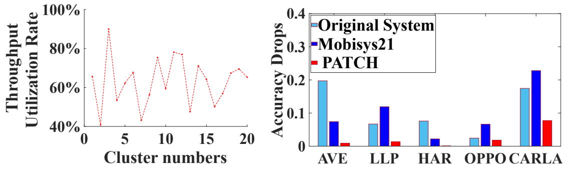
5 RELATED WORK

Cross-modality Data Imputation The purpose of a cross-modality imputation is to predict one modality from another. The methods in this field can be classified into three categories: supervised [39, 48, 58, 67, 78, 79], semi-supervised [49, 74], and unsupervised models [44, 94], depending on if the training data contains the ground truth labels. Recently deep learning models have been used to automatically learn features with better descriptive



(a) Battery Reliability. (b) System Performance Comparison

Fig. 16. Impact of Power Reliability to the nine multimodal systems in the five datasets.



(a) Network Fluctuation. (b) System Performance Comparison

Fig. 17. Impact of Network Fluctuation to the Nine Multimodal System in the Five Datasets.

power [32, 53, 67, 78, 79]. These methods, however, exhibit certain limitations when it comes to data imputation or synchronization.

For instance, in the study conducted by [58], deep convolutional and LSTM Recurrent Neural Networks were utilized for Multimodal Wearable Activity Recognition. Although this framework enhances system performance in activity recognition when applied to homogenous sensor data, it relies on linear interpolation and normalization to process missing data. While linear interpolation may be useful for handling isolated missing values, it becomes less reliable for feature extraction as the amount of missing data increases over time, leading to a decline in accuracy.

In another study by [67], a multiple-layer perception (MTL) was proposed as a multi-task model for context recognition. To manage absent sensors, they utilized sensor dropout and weighted each available feature. However, the importance of each sensor type is not uniform. This approach might be applicable for less critical sensors such as accelerometers or gyroscopes, but it falls short if key sensor data in a multimodal system is missing. This limitation becomes particularly evident in event parsing where visual data is significantly more important than audio data, especially when only two modalities are available.

The research by [78] and [79] offers valuable strategies for managing IoT sensor data noise using a unified Deep Learning Framework, which also evaluates the impact on system latency and energy consumption and presents solutions for data alignment. Nevertheless, in some multimodal applications, the resilience to noise is already integrated into the model. Consequently, data corruption that leads to partial or complete data loss or delay can have a more damaging effect on system performance. Our cross-modality imputation strategy could provide a more robust solution to severe instances of data corruption.

Low-latency Deep Learning in IoT Systems Prior work mainly focuses on optimizing resource-accuracy trade-offs to reduce latency in IoT systems [9, 28, 29, 41, 86]. These systems can find an optimal configuration by analyzing the impact of configurations/knobs (e.g., resolution) on resource consumption (e.g., bandwidth) and accuracy. Although significant progress has been made, these systems will all block inference when employed in a distributed multi-modal context — i.e., if one stream is much larger than the other or if wireless network bandwidth results in one data stream being more impacted than another, the slower stream will block the entire inference pipeline. Therefore, the problem of asymmetric data transmission and missing data due to network dynamics and sensor malfunction still remains a grand challenge in distributed multi-modal learning. Recent work attempted to investigate non-blocking inference by finding the optimal linear prediction using block-missing multi-modal data streams without imputing missing data [82, 84]. It decomposes the multi-modal model into a set of regression tasks and then builds regression models for these tasks. However, since these optimizations are model-specific, it is difficult to integrate the regression model into various multi-modal models that leverage different fusion techniques. PATCH serves as a plug-in module to enable non-blocking inference in the existing multimodal model. It is a generic and configurable software solution that can handle asymmetric data generation and is applicable to complex multi-modal models.

Data Alignment on Distributed Wireless Sensor Networks In distributed multi-modal learning, a shared time can be costly, and IoT devices are often equipped with low-cost clocks that can drift quickly and unpredictably [83]. Dynamic time warping [1, 27, 42, 90] and canonical correlation analysis [30, 63, 91, 92] were explored to align data. However, their methods assume the raw data is from a single modality. Thus, the existing clock synchronization methods are not ideal for aligning multimodal data streams in IoT systems [55, 69]. PATCH contains a lightweight data alignment module that is robust to data-missing or poorly aligned data streams without requiring external synchronization mechanisms.

6 DISCUSSION

Potential Security Risks PATCH imputes missing data using partially-available sensor data. However, the sensor dataset could be compromised by various attacks such as data poisoning [37, 77, 87], backdoor attacks [25, 54, 68]. For example, researchers have proposed data poisoning attacks against autoencoder-based anomaly detection models [8]. In addition, a number of defense methods [5, 12, 16, 73] have been proposed to mitigate the security and privacy issues of machine learning systems. In the future, we will adopt security and privacy strategies against potential malicious attacks. Moreover, apart from machine learning security issues, we acknowledge the existence of cyber-physical security issues in the distributed multimodal system. In the future, we will also enhance the software and network security (e.g., using firewalls, Anti-Virus software) to protect the internal deep learning-based models.

Generalisation and Training Labels Since each multimodal model has a distinct network and feature design, even with the same type of input, the best imputation feature pair will also not be the same. As a consequence, each PATCH framework needs to be trained separately to fit the unique feature structure and network to perform the auxiliary task of data imputation. However, the learning approach of PATCH is also limited by the original model. All ten multimodal models evaluated in this study are supervised learning, which requires labeled training data. We plan to explore the combination of PATCH with other semi-supervised or self-supervised learning models to minimize reliance on labeled data. Additionally, we aim to develop a framework tailored to the sensor type, capable of accommodating multiple models within a single trained framework.

7 CONCLUSION

We present PATCH, a framework to enable non-blocking inference of distributed multimodal models. PATCH serves as a plug-in module in the existing multimodal model and thus does not need to retrain the original deep learning model. PATCH consists of a cross-modality feature imputation module, a lightweight feature pair ranking module, and a data alignment module. We implement PATCH in nine off-the-shelf multimodal models using five public datasets and one self-collected dataset. And we also evaluate PATCH in four real-world scenarios. Experimental results show that PATCH can support various existing multimodal models and fusion strategies and it outperforms the state-of-the-art by up to 13% mean accuracy using only 10% of the training data.

ACKNOWLEDGMENTS

Dr. Lin Gu was supported by JST Moonshot R&D Grant Number JPMJMS2011, Japan.

REFERENCES

[1] John Aach and George M. Church. 2001. Aligning gene expression time series with time warping algorithms . *Bioinformatics* 17, 6 (06 2001), 495–508.

[2] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra Perez, and Jorge Luis Reyes Ortiz. 2013. A public domain dataset for human activity recognition using smartphones. In *Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning*. 437–442.

[3] Ho Bae, Jaehee Jang, Dahuin Jung, Hyemi Jang, Heonseok Ha, Hyungyu Lee, and Sungroh Yoon. 2018. Security and privacy issues in deep learning. *arXiv preprint arXiv:1807.11655* (2018).

[4] Pierre Baldi. 2012. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning*. JMLR Workshop and Conference Proceedings, 37–49.

[5] Hritik Bansal, Nishad Singhi, Yu Yang, Fan Yin, Aditya Grover, and Kai-Wei Chang. 2023. CleanCLIP: Mitigating Data Poisoning Attacks in Multimodal Contrastive Learning. *arXiv preprint arXiv:2303.03323* (2023).

[6] Avi Ben-Cohen, Eyal Klang, Stephen P Raskin, Shelly Soffer, Simona Ben-Haim, Eli Konen, Michal Marianne Amitai, and Hayit Greenspan. 2019. Cross-modality synthesis from CT to PET using FCN and GAN networks for improved automated lesion detection. *Engineering Applications of Artificial Intelligence* 78 (2019), 186–194.

[7] Anthony Berthelier, Thierry Chateau, Stefan Duffner, Christophe Garcia, and Christophe Blanc. 2021. Deep model compression and architecture optimization for embedded systems: A survey. *Journal of Signal Processing Systems* 93, 8 (2021), 863–878.

[8] Giampaolo Bovenzi, Alessio Foggia, Salvatore Santella, Alessandro Testa, Valerio Persico, and Antonio Pescapé. 2022. Data poisoning attacks against autoencoder-based anomaly detection models: A robustness analysis. In *ICC 2022-IEEE International Conference on Communications*. IEEE, 5427–5432.

[9] Christopher Canel, Thomas Kim, Giulio Zhou, Conglong Li, Hyeontaek Lim, David G Andersen, Michael Kaminsky, and Subramanya R Dulloor. 2019. Scaling video analytics on constrained edge nodes. *arXiv preprint arXiv:1905.13536* (2019).

[10] Ricardo Chavarriaga, Hesam Sagha, Alberto Calatroni, Sundara Tejaswi Digumarti, Gerhard Tröster, José del R Millán, and Daniel Roggen. 2013. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letters* 34, 15 (2013), 2033–2042.

[11] Djabir Abdeldjalil Chekired, Mohammed Amine Togou, Lyes Khoukhi, and Adlen Ksentini. 2019. 5G-slicing-enabled scalable SDN core network: Toward an ultra-low latency of autonomous driving service. *IEEE Journal on Selected Areas in Communications* 37, 8 (2019), 1769–1782.

[12] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. 2018. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728* (2018).

[13] Hao Chen, Youfu Li, and Dan Su. 2019. Multi-modal fusion network with multi-scale multi-path and cross-modal interactions for RGB-D salient object detection. *Pattern Recognition* 86 (2019), 376–385.

[14] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. 2022. TransFuser: Imitation with Transformer-Based Sensor Fusion for Autonomous Driving. *arXiv preprint arXiv:2205.15997* (2022).

[15] Max Chu, Annette Patton, Josh Roering, Cora Siebert, John Selker, Cara Walter, and Chet Udell. 2021. SitkaNet: A low-cost, distributed sensor network for landslide monitoring and study. *HardwareX* 9 (2021), e00191.

[16] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. 2019. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*. PMLR, 1310–1320.

[17] Pedro Costa, Adrian Galdran, Maria Ines Meyer, Meindert Niemeijer, Michael Abràmoff, Ana Maria Mendonça, and Aurélio Campilho. 2017. End-to-end adversarial retinal image synthesis. *IEEE transactions on medical imaging* 37, 3 (2017), 781–791.

[18] Karren Dai Yang, Anastasiya Belyaeva, Saradha Venkatachalapathy, Karthik Damodaran, Abigail Katcoff, Adityanarayanan Radhakrishnan, GV Shivashankar, and Caroline Uhler. 2021. Multi-domain translation between single-cell imaging and sequencing data using autoencoders. *Nature Communications* 12, 1 (2021), 1–10.

[19] Antônio Dâmaso, Nelson Rosa, and Paulo Maciel. 2014. Reliability of wireless sensor networks. *Sensors* 14, 9 (2014), 15760–15785.

[20] Jose Dolz, Karthik Gopinath, Jing Yuan, Herve Lombaert, Christian Desrosiers, and Ismail Ben Ayed. 2018. HyperDense-Net: a hyper-densely connected CNN for multi-modal image segmentation. *IEEE transactions on medical imaging* 38, 5 (2018), 1116–1126.

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR* (2021).

[22] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*. PMLR, 1–16.

[23] Konrad Gadzicki, Razieh Khamsehashari, and Christoph Zetzsche. 2020. Early vs late fusion in multimodal convolutional neural networks. In *2020 IEEE 23rd international conference on information fusion (FUSION)*. IEEE, 1–6.

[24] Wenbin Gao, Lei Zhang, Qi Teng, Jun He, and Hao Wu. 2021. DanHAR: Dual attention network for multimodal human activity recognition using wearable sensors. *Applied Soft Computing* 111 (2021), 107728.

[25] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Mądry, Bo Li, and Tom Goldstein. 2022. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 2 (2022), 1563–1580.

[26] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[27] Feng Han, Lan Zhang, Xuanke You, Guangjing Wang, and Xiang-Yang Li. 2019. SHAD: Privacy-Friendly Shared Activity Detection and Data Sharing. In *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 109–117.

[28] Seungyeop Han, Haichen Shen, Matthai Philipose, Sharad Agarwal, Alec Wolman, and Arvind Krishnamurthy. 2016. Mcdnn: An approximation-based execution framework for deep stream processing under resource constraints. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*. 123–136.

[29] Mark Hardiman, Ying Ou, Ryan Frazier, Zeyi Lee, and Longxiang Cui. 2015. *Project NoScope*. Master's thesis. EECS Department, University of California, Berkeley. http://www2.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-51.html

[30] M. A. Hasan. 2009. On multi-set canonical correlation analysis. In *2009 International Joint Conference on Neural Networks*. 1128–1133.

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

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[33] Nathan Henderson, Andrew Emerson, Jonathan Rowe, and James Lester. 2019. Improving sensor-based affect detection with multimodal data imputation. In *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 669–675.

[34] Yuta Hiasa, Yoshito Otake, Masaki Takao, Takumi Matsuoka, Kazuma Takashima, Aaron Carass, Jerry L Prince, Nobuhiko Sugano, and Yoshinobu Sato. 2018. Cross-modality image synthesis from unpaired data using CycleGAN. In *International workshop on simulation and synthesis in medical imaging*. Springer, 31–41.

[35] He Huang, Philip S Yu, and Changhu Wang. 2018. An introduction to image synthesis with generative adversarial nets. *arXiv preprint arXiv:1803.04469* (2018).

[36] Junxian Huang, Feng Qian, Yihua Guo, Yuanyuan Zhou, Qiang Xu, Z Morley Mao, Subhabrata Sen, and Oliver Spatscheck. 2013. An in-depth study of LTE: Effect of network protocol and application behavior on performance. *ACM SIGCOMM Computer Communication Review* 43, 4 (2013), 363–374.

[37] W Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. 2020. Metapoison: Practical general-purpose clean-label data poisoning. *Advances in Neural Information Processing Systems* 33 (2020), 12080–12091.

[38] Maqbool Hussain, Taqdir Ali, Wajahat Ali Khan, Muhammad Afzal, Sungyoung Lee, and Khalid Latif. 2015. Recommendations service for chronic disease patient in multimodel sensors home environment. *Telemedicine and e-Health* 21, 3 (2015), 185–199.

[39] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125–1134.

[40] Natasha Jaques, Sara Taylor, Akane Sano, and Rosalind Picard. 2017. Multimodal autoencoder: A deep learning approach to filling in missing sensor data and enabling better mood prediction. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 202–208.

[41] Junchen Jiang, Ganesh Ananthanarayanan, Peter Bodik, Siddhartha Sen, and Ion Stoica. 2018. Chameleon: scalable adaptation of video analytics. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. 253–266.

[42] B-H Juang. 1984. On the hidden Markov model and dynamic time warping for speech recognition–A unified view. *AT&T Bell Laboratories Technical Journal* 63, 7 (1984), 1213–1243.

[43] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale Video Classification with Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[44] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. 2017. Learning to discover cross-domain relations with generative adversarial networks. In *International conference on machine learning*. PMLR, 1857–1865.

[45] Diederik P Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. https://doi.org/10.48550/ARXIV.1312.6114

[46] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012), 1097–1105.

[47] Dana Lahat, Tülay Adali, and Christian Jutten. 2015. Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects. *Proc. IEEE* 103, 9 (2015), 1449–1477. https://doi.org/10.1109/JPROC.2015.2460697

[48] Baiying Lei, Zaimin Xia, Feng Jiang, Xudong Jiang, Zongyuan Ge, Yanwu Xu, Jing Qin, Siping Chen, Tianfu Wang, and Shuqiang Wang. 2020. Skin lesion segmentation via generative adversarial networks with dual discriminators. *Medical Image Analysis* 64 (2020), 101716.

[49] Chunyuan Li, Hao Liu, Changyou Chen, Yuchen Pu, Liqun Chen, Ricardo Henao, and Lawrence Carin. 2017. Alice: Towards understanding adversarial learning for joint distribution matching. *Advances in neural information processing systems* 30 (2017).

[50] He Li, Kaoru Ota, and Mianxiong Dong. 2018. Learning IoT in edge: Deep learning for the Internet of Things with edge computing. *IEEE network* 32, 1 (2018), 96–101.

[51] Tianxing Li, Jin Huang, Erik Risinger, and Deepak Ganesan. 2021. Low-latency speculative inference on distributed multi-modal data streams. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*. 67–80.

[52] Liangkai Liu, Sidi Lu, Ren Zhong, Baofu Wu, Yongtao Yao, Qingyang Zhang, and Weisong Shi. 2020. Computing systems for autonomous driving: State of the art and challenges. *IEEE Internet of Things Journal* 8, 8 (2020), 6469–6486.

[53] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3431–3440.

[54] Peizhuo Lv, Chang Yue, Ruigang Liang, Yunfei Yang, Shengzhi Zhang, Hualong Ma, and Kai Chen. 2023. A Data-free Backdoor Injection Approach in Neural Networks. (2023).

[55] Sathiya Kumaran Mani, Ramakrishnan Durairajan, Paul Barford, and Joel Sommers. 2018. A system for clock synchronization in an internet of things. *arXiv preprint arXiv:1806.02474* (2018).

[56] Héctor P Martínez and Georgios N Yannakakis. 2014. Deep multimodal fusion: Combining discrete events and continuous signals. In *Proceedings of the 16th International conference on multimodal interaction*. 34–41.

[57] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[58] Francisco Javier Ordóñez and Daniel Roggen. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* 16, 1 (2016), 115.

[59] Stavros Petridis, Themos Stafylakis, Pingehuan Ma, Feipeng Cai, Georgios Tzimiropoulos, and Maja Pantic. 2018. End-to-end audiovisual speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6548–6552.

[60] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. 2021. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7077–7087.

[61] Hangwei Qian, Sinno Jialin Pan, Bingshui Da, and Chunyan Miao. 2019. A Novel Distribution-Embedded Neural Network for Sensor-Based Activity Recognition.. In *IJCAI*, Vol. 2019. 5614–5620.

[62] Habib F Rashvand and Jose M Alcaraz Calero. 2012. *Distributed sensor systems: practice and applications*. John Wiley & Sons.

[63] S. Shariat and V. Pavlovic. 2011. Isotonic CCA for sequence alignment and activity recognition. In *2011 International Conference on Computer Vision*. 2572–2578.

[64] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[65] Yapeng Tian, Dingzeyu Li, and Chenliang Xu. 2020. Unified Multisensory Perception: Weakly-Supervised Audio-Visual Video Parsing. In *ECCV*.

[66] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. 2018. Audio-visual event localization in unconstrained videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 247–263.

[67] Yonatan Vaizman, Nadir Weibel, and Gert Lanckriet. 2018. Context recognition in-the-wild: Unified model for multi-modal sensors and multi-label classification. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 1–22.

[68] Matthew Walmer, Karan Sikka, Indranil Sur, Abhinav Shrivastava, and Susmit Jha. 2022. Dual-key multimodal backdoors for visual question answering. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*. 15375–15385.

[69] Guangjing Wang, Hanqing Guo, Anran Li, Xiaorui Liu, and Qiben Yan. 2023. Federated IoT Interaction Vulnerability Analysis. In *2023 IEEE 39th International Conference on Data Engineering (ICDE). IEEE*.

[70] Guangjing Wang, Nikolay Ivanov, Bocheng Chen, Qi Wang, ThanhVu Nguyen, and Qiben Yan. 2023. Graph Learning for Interactive Threat Detection in Heterogeneous Smart Home Rule Data. *Proceedings of the ACM on Management of Data* 1, 1 (2023), 1–27.

[71] Guangjing Wang, Lan Zhang, Zhi Yang, and Xiang-Yang Li. 2018. Socialite: Social activity mining and friend auto-labeling. In *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 1–8.

[72] Wei Wang, Dan Wang, and Yu Jiang. 2017. Energy efficient distributed compressed data gathering for sensor networks. *Ad Hoc Networks* 58 (2017), 112–117.

[73] Yuanda Wang, Hanqing Guo, Guangjing Wang, Bocheng Chen, and Qiben Yan. 2023. VSMask: Defending Against Voice Synthesis Attack via Real-Time Predictive Perturbation. *arXiv preprint arXiv:2305.05736* (2023).

[74] Mike Wu and Noah Goodman. 2018. Multimodal generative models for scalable weakly-supervised learning. *Advances in Neural Information Processing Systems* 31 (2018).

[75] Yu Wu and Yi Yang. 2021. Exploring Heterogeneous Clues for Weakly-Supervised Audio-Visual Video Parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[76] Haoming Xu, Runhao Zeng, Qingyao Wu, Mingkui Tan, and Chuang Gan. 2020. Cross-Modal Relation-Aware Networks for Audio-Visual Event Localization. In *ACM International Conference on Multimedia*.

[77] Ziqing Yang, Xinlei He, Zheng Li, Michael Backes, Mathias Humbert, Pascal Berrang, and Yang Zhang. 2022. Data Poisoning Attacks Against Multimodal Encoders. *arXiv preprint arXiv:2209.15266* (2022).

[78] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek Abdelzaher. 2017. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th international conference on world wide web*. 351–360.

[79] Shuochao Yao, Yiran Zhao, Shaohan Hu, and Tarek Abdelzaher. 2018. Qualitydeepsense: Quality-aware deep learning framework for internet of things applications with sensor-temporal attention. In *Proceedings of the 2nd International Workshop on Embedded and Mobile Deep Learning*. 42–47.

[80] Jongwon Yoon, Sayandeep Sen, and Joshua Hare. 2012. CRAWDAD dataset wisc/wiscape (v. 2012-08-03). Downloaded from https://crawdad.org/wisc/wiscape/20120803. https://doi.org/10.15783/C71C7D

[81] Biting Yu, Luping Zhou, Lei Wang, Yinghuan Shi, Jurgen Fripp, and Pierrick Bourgeat. 2019. Ea-GANs: edge-aware generative adversarial networks for cross-modality MR image synthesis. *IEEE transactions on medical imaging* 38, 7 (2019), 1750–1762.

[82] Guan Yu, Quefeng Li, Dinggang Shen, and Yufeng Liu. 2020. Optimal sparse linear prediction for block-missing multi-modality data without imputation. *J. Amer. Statist. Assoc.* 115, 531 (2020), 1406–1419.

[83] Wenpeng Yu, Wenxuan Yao, Xianda Deng, Yinfeng Zhao, and Yilu Liu. 2019. Timestamp Shift Detection for Synchrophasor Data Based on Similarity Analysis between Relative Phase Angle and Frequency. *IEEE Transactions on Power Delivery* (2019).

[84] Lei Yuan, Yalin Wang, Paul M Thompson, Vaibhav A Narayan, Jieping Ye, Alzheimer's Disease Neuroimaging Initiative, et al. 2012. Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data. *NeuroImage* 61, 3 (2012), 622–632.

[85] Martina Zambelli, Antoine Cully, and Yiannis Demiris. 2020. Multimodal representation models for prediction and control from partial information. *Robotics and Autonomous Systems* 123 (2020), 103312.

[86] Haoyu Zhang, Ganesh Ananthanarayanan, Peter Bodik, Matthai Philipose, Paramvir Bahl, and Michael J Freedman. 2017. Live video analytics at scale with approximation and delay-tolerance. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*. 377–392.

[87] Xuezhou Zhang, Xiaojin Zhu, and Laurent Lessard. 2020. Online data poisoning attacks. In *Learning for Dynamics and Control*. PMLR, 201–210.

[88] Yifei Zhang, Désiré Sidibé, Olivier Morel, and Fabrice Mériaudeau. 2021. Deep multimodal fusion for semantic image segmentation: A survey. *Image and Vision Computing* 105 (2021), 104042.

[89] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. 2023. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419* (2023).

[90] F. Zhou and F. De la Torre. 2012. Generalized time warping for multi-modal alignment of human motion. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 1282–1289.

[91] F. Zhou and F. De la Torre. 2016. Generalized Canonical Time Warping. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 2 (2016), 279–294.

[92] Feng Zhou and Fernando Torre. 2009. Canonical Time Warping for Alignment of Human Behavior. In *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta (Eds.). Curran Associates, Inc., 2286–2294. http://papers.nips.cc/paper/3728-canonical-time-warping-for-alignment-of-human-behavior.pdf

[93] Jinxing Zhou, Liang Zheng, Yiran Zhong, Shijie Hao, and Meng Wang. 2021. Positive Sample Propagation along the Audio-Visual Event Line. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[94] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2223–2232.

[95] Xiao Zhu, Yihua Ethan Guo, Ashkan Nikravesh, Feng Qian, and Z Morley Mao. 2019. Understanding the networking performance of wear OS. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 3, 1 (2019), 1–25.