t-READi: Transformer-Powered Robust and Efficient Multimodal Inference for Autonomous Driving

Pengfei Hu ⓘ, Yuhang Qian ⓘ, Tianyue Zheng, Ang Li ⓘ, Zhe Chen ⓘ, Yue Gao ⓘ, *Fellow, IEEE*, Xiuzhen Cheng ⓘ, *Fellow, IEEE*, and Jun Luo ⓘ, *Fellow, IEEE*

Abstract—**Given the wide adoption of multimodal sensors (e.g., camera, lidar, radar) by *autonomous vehicle*s (AVs), deep analytics to fuse their outputs for a robust perception become imperative. However, existing fusion methods often make two assumptions rarely holding in practice: i) similar data distributions for all inputs and ii) constant availability for all sensors. Because, for example, lidars have various resolutions and failures of radars may occur, such variability often results in significant performance degradation in fusion. To this end, we present t-READi, an adaptive inference system that accommodates the variability of multimodal sensory data and thus enables robust and efficient perception. t-READi identifies variation-sensitive yet *structure-specific* model parameters; it then adapts only these parameters while keeping the rest intact. t-READi also leverages a cross-modality contrastive learning method to compensate for the loss from missing modalities. Both functions are implemented to maintain compatibility with existing multimodal deep fusion methods. The extensive experiments evidently demonstrate that compared with the status quo approaches, t-READi not only improves the average inference accuracy by more than 6% but also reduces the inference latency by almost 15× with the cost of only 5% extra memory overhead in the worst case under realistic data and modal variations.**

Index Terms—**Autonomous vehicle, robust perception, multimodal learning, object detection, semantic segmentation.**

I. INTRODUCTION

AUTONOMOUS driving, with its worldwide developments [60], [65], [70], promises to achieve greater safety,

less harmful emissions, increased lane capacity, and reduced travel time [52]. At the core of autonomous driving, the perception capability of *autonomous vehicle*s (AVs) leverages the data collected from various sensors (e.g., camera, lidar, and radar) to better understand AVs' surrounding, thus enabling applications such as object detection [19], [27], [76] and semantic segmentation [11], [17], [58]. To power the perception capability, *deep neural network*s (DNN) are adopted to process and fuse the multimodal data [39] into a unified representation. Therefore, maintaining the effectiveness of these AV-DNN models is crucial to autonomous driving.

The lifecycle of an AV-DNN model starts with its design, then it gets pre-trained model by individual manufacturers, before being deployed to respective AVs equipped multimodal sensors for performing perception-related inferences. Although DNN-based multimodal fusion for on-device inference in general has attracted attention from both academia [18], [20] and industry (e.g., Google [38], as well as Intel and Ford [23]), only part of them have targeted multimodal DNN for autonomous driving (e.g., [1], [45]). Since this latter batch of proposals largely focuses on DNN architecture design at the manufacturer side, they often disregard the per-vehicle variations in terms of input data and modality, resulting in a **missing link**, pertaining to in-vehicle inference, between DNN models and their practical adoptions.

One critical issue with existing proposals is the fixed set of multimodal sensors and environment conditions used for model training. In practice, both sensor modalities and environmental conditions may experience drastic variations. For example, different lighting conditions may force the camera to change its exposure, different velocities can result in motion blurs to various degrees, and adverse weather conditions (e.g., rain, fog, and snow) often scatter laser light, thus forcing lidar to perform more intensive scans. Consequently, those two modalities introduce more drastic variations. Moreover, sensor occlusion and/or malfunction may lead to missing modalities that compromise the multimodal fusion. All these realistic scenarios yield variations in the input data and modality, as illustrated in Fig. 1. They often significantly degrade the performance of well-designed DNN models, causing erroneous decision-making and even traffic accidents. Therefore, it is crucial to address adverse external conditions and internal sensor malfunctions carefully. As highlighted in the survey by [61], adverse weather conditions and faulty sensors are common contributors to autonomous vehicle accidents, such as the 2019 Tesla Model 3 incident [62]. A naive
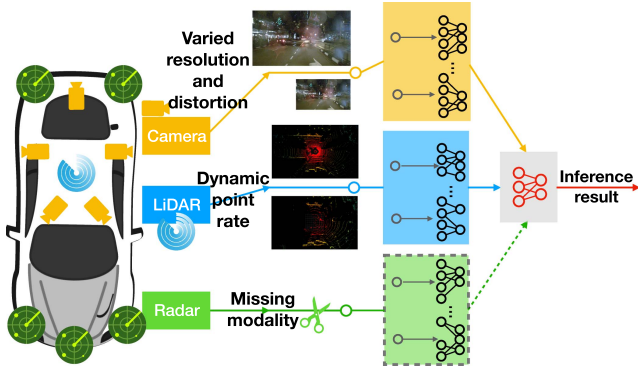
Fig. 1. The variations in data and modality.

solution to mitigate performance degradation is to pre-train a large set of models for different scenarios and switch among them during runtime based on the available data and modality variations. However, this solution is not *robust* against continuously varying sensor and environment conditions since we may only have a countable number of models. Besides, model switching incurs excessive memory footprints and intolerable inference latency.

Whereas, it is non-trivial to realize robust and efficient multimodal inference for autonomous driving. First of all, as robustness and efficiency are often at odds, designing a multimodal system that copes with variations without consuming excessive resources remains an uncharted area. Second, it's hard to strike an adaptive balance between these two objectives since it depends on continuously varying conditions. For instance, the latency requirements for DNN models in high-speed cruising and low-speed maneuvering scenarios are different because of the DNN responsiveness needed to match the vehicle speed. Last but not least, since DNN models embedded in vehicles are typically compiled low-level codes optimized for specific DNN architecture [1], [45], [74], it is largely impossible to modify the architecture in response to, for example, missing modalities and in turn their sensory data as part of the input. To address these challenges, we propose t-READi (i.e., transformer-powered Robust and Efficient multimodal inference for Autonomous Driving) as a novel AV inference system; it adaptively accommodates the variations in multimodal sensory data and missing modalities. Specifically, t-READi employs a variation-aware model adaptation algorithm to handle data and modality variations under memory constraints while producing multiple variants of models deviated from the pre-trained model. For each variant, we exploit the parameter-efficient fine-tuning techniques of Large Language Models (LLMs) [29], but extend to widely-used modules in AV-DNN models in addition to transformer modules in LLMs. For example, t-READi injects trainable rank decomposition matrices into residual blocks leading to a significantly decreased number of trainable parameters. With multiple variants of models and the pre-trained model, we can switch different variants according to current input indications, such as weather conditions.

Moreover, t-READi leverages a contrastive learning framework to overcome the issue of missing modalities, while maintaining a comparable performance as using complete modalities. In particular, by contrasting the samples with and without a potential missing modality, t-READi drives the latent representation under missing modality towards semantically correlated with that under full modalities; this renders the DNN model robust to missing modalities without altering its architecture. Our key contributions can be summarized as follows:

- We design t-READi to address the variations of sensory data and modality in autonomous driving so that t-READi adapts to various run-time environments with both robust and efficient manners.
- We propose a variation-aware model adaptation algorithm to perform model adaptations with sparse parameter updates. The technique minimizes memory footprints, hence allowing for loading a large number of model variations into memory and in turn eliminating extra latency caused by model reloading.
- We design a cross-modal contrastive learning method to compensate for input data loss due to missing modalities. It avoids runtime modifications on model architecture and thus preserves the overall efficiency.
- We implement a prototype of t-READi and evaluate it with extensive experiments. The promising results confirm that t-READi indeed offers robust and efficient multimodal inference for autonomous driving.

Though proposals on multimodal fusion for on-device inference do exist [18], [20], [23], t-READi is still the first to design such a system for autonomous driving with practical considerations, i.e., robustness and efficiency in driving environments. The rest of the paper is organized as follows. Section II introduces the background and motivation. The detailed system design and implementation are described in Sections III and IV, respectively. Section V reports the evaluation results. Related works are presented in Section VI, along with limitations and future directions of t-READi. Finally, Section VIII concludes our paper.

II. BACKGROUND AND MOTIVATION

In this section, we first investigate the impact of sensors' parameter changes (e.g., lidar and camera) on DNN inference performance. We then show how missing modalities significantly degrade the performance of DNN for autonomous driving. Finally, we explain why model reloading is not feasible for mitigating these negative impacts due to high inference latency and memory usage.

A. Sensor Parameter Variation

Most existing DNN-powered perception systems use a pre-trained model assuming that inference and training data follow the same probability distribution [1], [16], [45]. However, this assumption does not always hold true under AV scenarios: the camera of an AV may capture video with varying exposure and motion blur; the lidar of the AVs may retrieve data streams with a dynamic density of point cloud. Consequently, the parameter variation will result in a non-iid (independent and identically
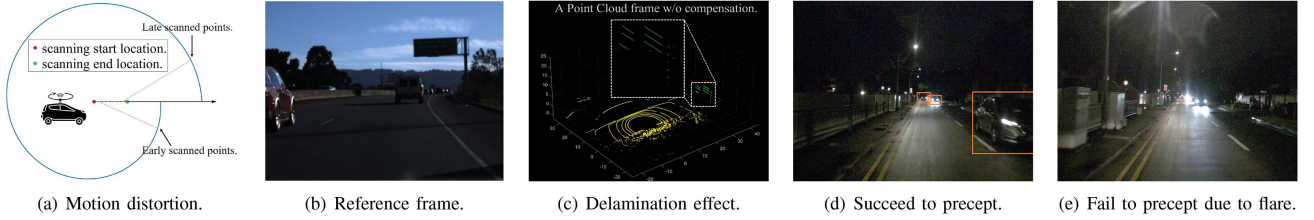
(a) Motion distortion.   (b) Reference frame.   (c) Delamination effect.   (d) Succeed to precept.   (e) Fail to precept due to flare.

Fig. 2.   Sensory data variations dramatically degrade the performance of pre-trained AV-DNN models in reality. (a)∼(c): fail to detect the traffic sign due to lidar point cloud variation. (d)∼(e): fail to detect vehicles due to heavy flare effect on camera.

distributed) distribution that cannot be well handled by the AV-DNN model and hence lead to degraded performance.

Lidar is a vital sensor used in AVs due to its depth estimation capability. Currently, the automotive industry primarily uses rotating lidar sensors, which are powered by mechanical rotation [69]. For lidar sensors, the spinning rate refers to the number of revolutions in a second, and a higher spinning rate results in lower azimuth/elevation angular resolution. For example, when the spinning rate sweeps from 5 Hz to 20 Hz, the azimuth angular resolution deteriorates from $0.09°$ to $0.36°$. However, the pre-setting spinning rate cannot accommodate varying driving speed of the AV on run-time, resulting in a dynamic density of point cloud. The reason is shown in Fig. 2(a) that lidar cannot catch point cloud timely, leading to depth difference in each scanning period (inverse of spinning rate). Fig. 2(b) and (c) demonstrate a road sign lies in front of the view, and the reflected lidar points are delaminated significantly, as the fixed spinning rate cannot catch different driving speeds of the AV.

The camera is another sensor widely used in AVs but much more affordable, which can provide richer semantic information than lidar. However, its performance can be significantly affected by surrounding environments. For example, a high or low intensity of light can cause cameras to malfunction. A common issue is the flare effect. Generally, a model's inference performance will degrade for unseen and bad environments with high possibility. As shown in Fig. 2(d) and (e), even though scenarios of two images are similar, the pre-trained object detection model can work well on its training dataset in Fig. 2(d), but fails to detect the target in Fig. 2(e) due to flare effect. In a nutshell, input variations of multiple modalities severely affect AV-DNN performance.

B. Missing Modality

In the realm of multimodal sensing data fusion, most existing research presumes the availability of all modalities during both the training and inference stages [3], [72]. However, this assumption does not always hold true, as sensors can malfunction or become obstructed during inference, leading to missing modalities, as demonstrated for the RGB-camera modality in [6]. Such failures pose significant challenges to DNN-based perception tasks in autonomous driving, primarily due to mismatches between the partial input data and the model architecture. A common solution to this problem is data imputation, which involves filling in missing modalities with predefined values, such as zeros [66], [73]. However, this approach introduces bias during inference, resulting in non-iid issues.
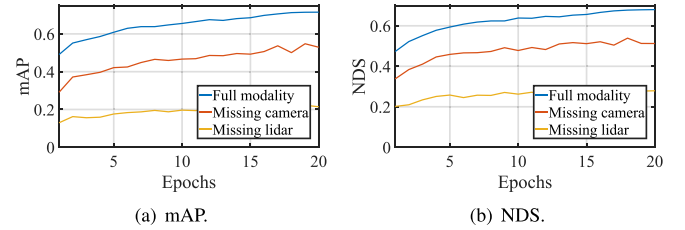

(a) mAP.   (b) NDS.

Fig. 3.   Damaging effects of missing modality.

We demonstrate this in Fig. 3 by showing the mAP (mean average precision) and NDS (details of the metrics will be introduced in Section IV) of BEVFusion [45] under both full and missing modalities of the nuScene Dataset [5] with zero-filling. The results reveal that BEVFusion inference with all modalities outperforms inference with missing camera data (filled with zeros) by over 15%. Moreover, training with missing lidar modality results in significantly lower mAP and NDS, due to the absence of rich geometric information provided by 3D lidar point cloud. These findings confirm that merely filling missing modalities with zeros falls short. They highlight the need for novel methods that can effectively harness the complementary information from multiple modalities, thereby reducing the performance gap between inference with full and missing modalities.

However, designing an effective mechanism to compensate for missing modalities without additional computational overhead, while maintaining comparable performance to using complete modalities, is challenging. To address this, t-READi introduces a novel method that effectively utilizes complementary information from multiple modalities to narrow the performance gap between full and missing modalities. This method will be presented in Section III-C.

C. Model Reloading is Impractical

To address the performance degradation caused by environment variance, sensor parameter changes, and missing modalities, a straightforward solution is to pre-train a set of distinct DNN models for different data input variations and missing modalities. These models can be reloaded as needed in corresponding scenarios. However, changing environments can lead to time-varying parameter changes and missing modalities, requiring frequent model reloading to accommodate these variations. This places a heavy burden on AV embedded systems in terms of memory and latency, making this solution impractical. To illustrate the memory and latency demands imposed by
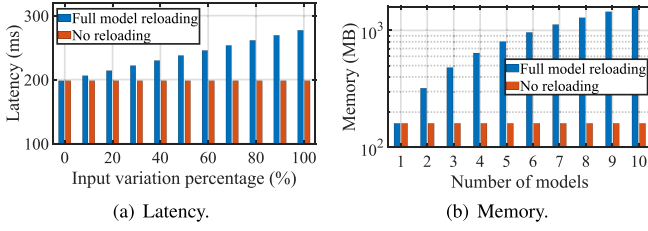
Fig. 4.   Latency and memory requirements caused by model reloading.

frequent DNN model reloading, we present the memory and latency requirements with and without model reloading in Fig. 4 using aforementioned pre-trained models.

As Fig. 4(a) shows, model reloading from hard drives can increase the inference latency by up to 40% compared to not reloading the model. However, as reported in [13], the median driving reaction time in urban street is 370 ms, which makes hundreds of millisecond latency unacceptable indeed. In fact, if we can fit all pre-trained models into memory instead of reloading them from hard drives, the inference latency will be comparable to the case of not reloading. However, as Fig. 4(b) demonstrates, each time the AV reloads the model, it requires an additional 160 MB of hard disk space (i.e., typical size of our trained model) to store the new model parameters compared to not reloading model. More importantly, the memory capacity is still limited (e.g., Tesla Model 3 has 8GB RAM), and the memory needs to run multiple applications simultaneously. Furthermore, contemporary AV inference engines heavily rely on neural network processors, which use SRAM with capacities comparable to L3 cache at best. The prolonged latency and additional memory requirements are unacceptable for real-time, memory-constrained AV embedded systems. Thus, a more efficient solution for DNN model adaptation is needed to handle variations in input data and modalities. Although parameter-efficient fine-tuning methods [2], [28], [29] have succeeded in LLMs for adaptation, these strategies are not directly applicable to AV-DNN models. Adapting such fine-tuning approaches to address the specific data variation issues in AV-DNN models remains an unresolved challenge. AV-DNN models incorporate diverse modules, such as transformers, convolutional layers, and residual convolution blocks. The challenge lies in developing a versatile fine-tuning method that integrates seamlessly with these diverse structures. To this end, we will present t-READi in Section III and tackle the challenge in Section III-B.

D. Transformer for Autonomous Driving

DNN in computer vision has long been dominated by CNN (convolutional neural networks), and these architectures are enhanced with greater scale, more extensive connection, and more sophisticated form of convolution. Recently, the Transformer architecture [67] is adapted from NLP (natural language processing) to vision community [15], [44]. Vision transformer provides the capability to encode distant dependencies or heterogeneous interactions, which is crucial for autonomous driving scenario,

and is qualified to be a powerful backbone as achieves better performance with similar complexity against convolutional-based backbone counterparts. The attention module, as a component of transformer, plays a critical role in modeling the interactive relation. Mathematically speaking, it is computed as

$$A_X = X^T W_Q^T W_K X / \sqrt{d}. \tag{1}$$

Where $X \in \mathbb{R}^{f \times n}$ denotes f-dimensional n features, usually as the intermediate results translated from sensor data with encoders. $W_Q, W_K \in \mathbb{R}^{d \times f}$ are feature projection matrices which project vectors to d-dimensional ones. Many attention modules split relative large projected dim d into pieces as known as "multi-head attention" to obtain effective performance, which implies $d \ll n$.

Transformer-powered DNNs are notoriously difficult to train from scratch, particularly in the presence of noisy data, often due to ill-conditioned attention modules. t-READi effectively tunes these AV-DNNs by addressing and correcting issues within the attention modules.

III. SYSTEM DESIGN

This section introduces the design of t-READi. First we give an overview, then we introduce the variation-aware model adaptation and cross-modal contrastive learning. Finally, we put everything together and summarize the training strategy.

A. Overview

Motivated by the observation in Section II, we design t-READi, a system consisting of two key components: i) a variation-aware model adaptation mechanism for efficient multi-modal inference under memory and latency constraints, and ii) a cross-modal contrastive learning algorithm that addresses missing modalities and improves inference robustness. As shown in Fig. 5, t-READi maintains compatibility with existing multi-modal DNNs for AVs, while incorporating two aforementioned components that will be elaborated in Sections III-B and III-C, respectively. Given a multimodal DNN (e.g., BEVFusion) and sensory data (e.g., lidar and camera), the first component identifies, selects and injects the variation-aware model parameters. The second component compensates for information loss due to missing modalities. The design of t-READi resolves three key technical challenges. i) Adapting techniques from LLM to AV-DNN to manage data variations (Section III-B), as introduced in Section II-C. ii) Narrowing the performance gap between full and missing modalities (Section III-C), as discussed in Section II-B. iii) As most training samples are collected in a normal environment, we may only have limited types of data variations for fine-tuning the pre-trained model. However, the AV-DNN models always encounter unseen variations in reality, which may result in unexpected performance drop. Therefore, our final challenge is to expand the capability of fine-tuned models to deal with unseen variations (Section III-D).
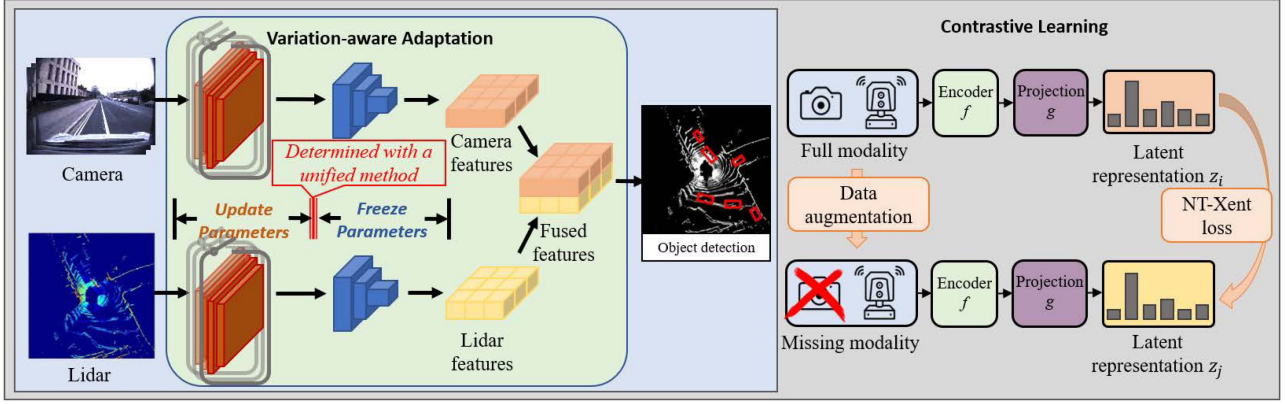
Fig. 5.   t-READi architecture: Optimized multimodal inference with contrastive data augmentation.
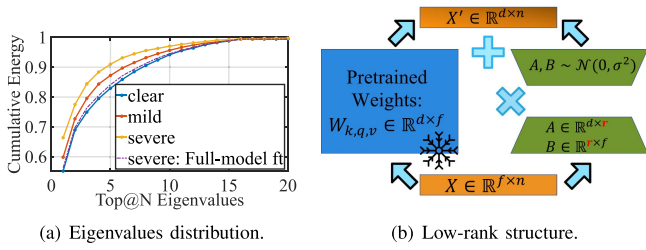

(a) Eigenvalues distribution.   (b) Low-rank structure.

Fig. 6.   (a) Even severer low-rank problem. (b) Proposed LoRa module to alleviate the adverse effect.

B. Variation-Aware Model Adaptation

Existing parameter-efficient fine-tuning techniques [2], [28], [29], [34] are mostly designed for downstream tasks of LLMs, and how to effectively apply those approaches to AV-DNN models is barely studied. To investigate the effectiveness of the parameter-efficient fine-tuning method for AV-DNN models, we start with the models with the same transformer modules as LLMs, and then extend to other widely-used modules in AV-DNN models. Drawing inspirations from BitFit [2], which focuses on fine-tuning lightweight inductive-bias terms only, we begin by tuning all the normalization layers and task-specific heads in the AV-DNN model. This foundational operation of t-READi is elaborated in Section V-H, and results demonstrate that it significantly outperforms the conventional practice of only fine-tuning task-specific heads.

a) Low-Rank Adaptation: Our exploration continues into the transformer module, where the attention mechanism is a critical component. We conduct an experiment by setting $n = 200, d = 16$ in (1), calculating the cumulative sum of eigenvalues of Ax for each sample, and averaging them to gauge its rank. As Fig. 6(a) shows, it is noticeable that even on clean data that the attention module is most familiar with, it already exhibits a low-rank instance as the top 3% eigenvalues account for more than 85% energy. This bias amplifies as sensor data distortion intensifies. In other words, the matrix Ax, inherently low-rank due to multi-head operations, degrades when processing distorted data. However, this is not a consequence of low-quality input

data since comprehensive fine-tuning can substantially mitigate this effect. The observed rank collapse phenomenon was first recognized in NLP, specifically when fine-tuning the pre-trained model for distinct tasks. To overcome this, a Low-Rank Adaptation method [29] is proposed by of injecting rank-decomposition matrices to transformer blocks. Drawing on this concept, we design low-rank modules to fine-tune transformer blocks in AV-DNNs based on sensory data variations. Fig. 6(b) presents its detailed architecture. In general, the low-rank module comprises a pair of matrices A, B, which coexist parasitically for each inherent projection weight matrix. Each matrix has a low rank, bounded by a hyperparameter r, and is transparent due to their near-identity initialization. During fine-tuning, we keep the bulky pre-trained weights frozen, allowing only updates to these low-rank modules. As our objective is to restore the pre-trained matrix, and its rank is relatively low in optimal scenarios (as indicated in Fig. 6(a)), we can efficiently bound r by a small integer. Additionally, the two distinct paths that are demonstrated in Fig. 6(b) can operate in parallel, causing no distinctive latency overhead.

b) Generalization to Non-Tranformer Modules: It is worth noting that not all AV-DNNs rely on transformers. To make our adaptation approach general and not specific to particular models, we take a closer look at the similarities and distinctions between transformer-heavy DNNs and others. Our key insight is that transformers or residual convolution blocks widely adopted cutting-edge AV-DNNs are deeply interconnected. For example, they both utilize the skip connection, significantly mitigates the rate of rank degeneration [14]. Furthermore, it has been demonstrated that a convolution-only network with a relatively large kernel size can achieve performance on par with its transformer-based counterparts [46], suggesting that the attention mechanism of transformers can be emulated with larger convolution kernels. Moreover, we find parallels in viewing the attention module as an ensemble of shallow networks, as studies of ResNet point out [68]. Based on these insights, with the intention of reviving convolution kernels, we inject a module with a similar structure as the low-rank module into residual blocks, resulting in a different style of adaptation in the attention module [28]. This module, demonstrated in Fig. 7, is placed before batch
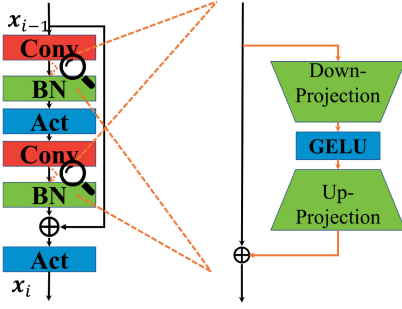
Fig. 7.   A variant of widely-adopted residual convolution block, we inject such transparent module.

normalization layers with skip connection, a decision informed by a similar layer arrangement in transformers. We complement this setup with an activation layer, given the absence of an apparent low-rank attention matrix. By harmonizing the above three techniques, we ensure our adaptation method remains efficient and universally applicable.

C. Cross-Modal Contrastive Learning

Recall that the preliminary experiments in Section II-B showed that the conventional data imputation methods, such as filling the missing modalities with zeros, incur information loss and even introduce biases into the DNN model. To compensate for the information loss caused by missing modality, we resort to representation learned from full modalities (e.g., camera and lidar) to guide uni-modal data (e.g., camera) towards a unified multimodal representation space. To this end, we design a cross-modal contrastive learning framework. The key idea behind the this framework is that even when some modalities are missing due to sensor occlusion or malfunction, the latent representation learned from missing modalities should be as similar as possible to the representation extracted from complete modalities. For example, if only one camera on an AV, which are equipped with multiple cameras and lidars, is missing, the inference performance can be preserved if the missing camera's latent representation can be compensated by the remaining sensors, which typically have overlapping fields of view (FoV). To make the latent representations with and without missing modalities as similar as possible, follow work [33], [77], t-READi employs four components (i.e., data augmentation, feature extractor, projection head, and contrastive loss) in its contrastive learning framework, as shown in the right side of Fig. 5. First, t-READi uses a stochastic data augmentation module to remove each sample of some modalities with a probability of 10%, resulting in one missing-modality dataset paired with the original full-modality one. The full-modality dataset and the one with missing modality are denoted as x and $\alpha(x)$, respectively. We consider $(x_i, a(x_i)) \in \{(x_i, a(x_i)) : i \in |x|\}$ as a positive pair, and $(x_i, a(x_j)) \in \{(x_i, a(x_j)) : i, j \in |x|, \text{and } i \neq j\}$ as a negative pair since x_i and $a(x_j)$ describe different scenes. Then, t-READi leverages a neural network based encoder $f(\cdot)$ to extract representation vectors from a pair of samples from x and $\alpha(x)$. Our framework allows various choices of network

architecture without any constraints, and we opt for simplicity, and reuse the feature extractor used in the object detection network. Thereafter, we use a multilayer perceptron with one hidden layer as $g(\cdot)$, which is used to map representations to the latent space where the contrastive loss is applied. Last but not least, we use a contrastive loss function, termed NT-Xent [9], i.e., the normalized temperature-scaled cross-entropy loss, to train the encoder $f(\cdot)$. The loss function enforces that the features between a positive pair as similar as possible, while enlarging the distance between the features of a negative pair.

We randomly sample a mini-batch of N samples from x and one sample from $a(x)$, i.e., $a(x_i)$ and $x_i \in \{x_1, \ldots, x_N\}$ resulting in $N + 1$ samples. Given a positive pair $(x_i, a(x_i))$, we consider the other $N - 1$ samples from x as negative samples with regard to $a(x_i)$. Then the contrastive loss for a positive pair $(x_i, a(x_i))$ can be formally expressed as

$$\mathcal{L}_i = -\log \frac{\exp\left(\mathbf{sim}\left(z_i, \hat{z}_i\right)/\tau\right)}{\sum_{k=1}^{N} \mathbb{1}_{[k \neq i]} \exp\left(\mathbf{sim}\left(z_k, \hat{z}_i\right)/\tau\right)}, \quad (2)$$

where $z_i = g(f(x_i))$ is the projected feature for scene x_i and $\hat{z}_j = g(f(\alpha(x_j)))$ is the projected feature for scene x_j where some missing modality happens, $\mathbf{sim}(\cdot)$ is a function that calculates cosine similarity of two latent representations, $\mathbb{1}_{[k \neq i]} \in \{0, 1\}$ is an indicator function evaluating to 1 if $k \neq i$, and τ denotes a temperature hyper-parameter, whose appropriate tuning can help the model learn from hard negatives as it controls the penalties on hard negative sample. The final loss is computed across all positive pairs in a N size mini-batch. Minimizing the contrastive fusion loss will force the projected features from the same scene but modality-missing conditions are different (i.e., $(z_i, \hat{z}_i)$ pair) together, while pushing projected features from different scenes (i.e., $(z_k, \hat{z}_i), k \neq i$ pairs) apart. A previously trained but not robust model is incorporated into the contrastive learning framework, during the subsequent adaptation phase, these additional components are removed. The contrastive learning framework approximately doubles the forward propagation time, however, we find that a relatively small number of epochs is sufficient, and it remains completely transparent during inference. Therefore, we consider it introduces an affordable computational cost.

D. Putting Everything Together

We summarize the training strategy of the t-READi, and present the overall workflow as follows. With the strategy presented in Section III-C, t-READi re-trains the model under missing modality settings, allowing each sensor to fail independently. During the following tuning phase, which addresses various scenarios with variation, t-READi is constrained by two parameters, namely rank bound k and projection squeeze ratio r, as we have discussed their insights in Section III-B above.

In reality, a cocktail of domain-specific variations muddles the situation, rendering the tuning of the entire domain space anything from unmanageable to impossible. Yet, by employing tuned parameters from individual variations, we squeeze the combined domain's tuning scope from $M \times N$ to a more manageable $M + N$ (respecting lidar and camera modalities).

Furthermore, we find unseen domains like nighttime often share traits with seen domains, such as underexposure. We propose a conditional melding of two tuned variants $\mathcal{L}c'$ and $\mathcal{L}l'$: for exclusive layers $(\cup_{i\in\mathbf{N}}\mathcal{L}ci') \oplus (\cup_{j\in\mathbf{N}}\mathcal{L}lj')$, where $\oplus$ indicates a symmetric difference operation, we opt for a straightforward update. More complicated are the overlapping layers $(\cup_{i\in\mathbf{N}}\mathcal{L}ci') \bigcap (\cup_{j\in\mathbf{N}}\mathcal{L}lj')$, for which we choose interpolation using $\lambda_c * \mathcal{P}c(t) + \lambda_l * \mathcal{P}l(t)$, where t denotes overlapping layers and $\mathcal{P}c, l(t)$ represents the model-specific parameter set for t, with the additional constraint $\lambda_c + \lambda_l = 1$.

While it's possible to overwrite the tuned layers, t-READi takes a different approach for the sake of efficiency. It simultaneously loads all the layers that are tuned under different variations, and further refines the less significant bits within the "shared" layers through pruning. These layers are organized as the *values* in a *map*. During inference, t-READi utilizes the information provided by various sensors (e.g., brightness) to encode the *key* that switches to the desired layer set. While a more self-contained method that only uses input data to switch parameters might be ideal, it can be more specific to certain modalities and less general. For example, it might be easier to implement for cameras but more challenging for lidar. Simple solutions like a lightweight filter are capable of rating images, mapping the ratings to keys, and selecting the tuned parameters to switch. However, it is not as straightforward when dealing with lidar data. The complexity of lidar data, with its 3D point cloud representation, makes it challenging to use the same methodology to rate, map, and switch parameters as easily as we do with images. Therefore, we consider the development of a more self-contained approach to be part of our future work. This would enhance t-READi's adaptability to different variations in sensory data and modalities.

IV. IMPLEMENTATION AND EXPERIMENT SETUP

In this section, we first present the details of t-READi's implementation, then we apply t-READi to develop two widely-used applications. Finally, we describe the metrics that we use to comprehensively evaluate the performance of t-READi.

A. Implementation

We implement the vehicle detection application on a server equipped with an Intel Xeon Gold 6226 CPU [31], 128 GB RAM, and NVIDIA GeForce RTX 3080 Ti GPU [48]. As for the software, Python 3.7 and PyTorch 1.9.1 [49] are used for implementing the application. Our object detection and segmentation model is built upon mmDetection [8], which is an open-source toolbox that provides state-of-the-art OD models. In particular, the model components and settings for t-READi are as follows:

- The encoder $f(\cdot)$ consists of two modality-specific encoders. For the camera and lidar modalities, Swin Transformer-T [44] and VoxelNet [78] are used as the encoders, which is transformer intensive and residual block intensive respectively.
- We choose to project both camera's and lidar's data to a unified bird's-eye view. For lidar, we flatten the sparse lidar features along the height dimension, hence not creating

TABLE I
DATASETS USED FOR T-READI

Dataset	NuScenes[5]	DENSE[4]
Sensor Setup		
Cameras	6	2
Camera Resolution	1600×900	1920×1024
lidars	1	2
lidar Resolution	32	32, 64
Dataset Statistics		
Images	1.4M	104K
Bounding boxes	1.4M	13.5K
Frames	390K	104K
Point density per frame	35K	30K, 55K
Annotations	1.17M	212K

geometric distortion. For camera, we cast each camera feature pixel back into a ray in the 3D space, which can result in a feature map that retains full semantic information from the cameras.
- Even though all sensory inputs are converted to a unified representation, the lidar features and camera features can still be spatially misaligned to some extent due to the inaccurate depth in the view transformer. To this end, we apply a fully convolutional encoder (with a few residual blocks) to compensate for such local misalignments.
- In the tuning process, the AdamW optimizer, which employs the decoupled weight decay regularization [47] is used by setting a fixed learning rate of 5×10^{-5} and weight decay as 0.01. We also enabled gradient clip, which confines the L2 norm to be less than 35.

B. Tasks and Dataset

We perform two tasks to evaluate of t-READi, i.e., *object detection* and *semantic segmentation*, which are primary tasks for autonomous driving perception. We evaluate our scheme on two large-scale public datasets for autonomous driving: nuScenes [5] and DENSE [4]. Details of the two datasets are given in Table I.
- nuScenes includes 1000 driving scenes under different weather and illumination conditions in Boston and Singapore, which are known for dense traffic and challenging driving situations. There are approximately 1.4M camera images and 390k lidar frames which are annotated with 1.4M accurate bounding boxes for 23 classes.
- DENSE is captured during two test drives in February and December 2019 for two weeks, each under different weather (i.e., rain, snow, light/dense fog). There are approximately 104K camera images and 104K lidar frame which are annotated with 13.5K accurate bounding boxes.

C. Evaluation Metrics

To evaluate the performance of object detection, follow the previous works, we consider the widely used metric mean Average Precision (mAP), along with the specialized nuScenes detection score (NDS) tailored for the nuScenes. Since semantic segmentation can be considered as a *pixel-wise* classification task, we employ Intersection over Union (IoU) to measure the overlap between pixel set pairs of ground-truth and prediction.

- *Average Precision (AP):* Our predictions consists of 4 categories: True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). Based on whether our prediction agrees with the corresponding ground truth (T/F) and the condition of our prediction (P/N). Precision (prec) is then calculated as $\frac{|TP|}{|TP|+|FP|}$ to evaluate the likelihood of making false positive reports. Consequently, AP is defined as the integration over recall: $AP = \int_0^1 \text{prec}(r)\mathrm{d}r$.

- *mAP:* mAP is calculated by averaging the AP values across different thresholds and categories. These thresholds are based on IoU typically (like DENSE). However in NuScenes, the thresholds are center-distance based represented by a set of thresholds $\mathbb{D}$. The overall object category space, denoted as $\mathbb{C}$, is heavily biased and consists of 10 categories. The mAP can be expressed as

$$mAP = \frac{1}{|\mathbb{C}||\mathbb{D}|} \sum_{c \in \mathbb{C}} \sum_{d \in \mathbb{D}} AP_{c,d}. \tag{3}$$

- NDS: NDS is designed to address the limitations of mAP in capturing all aspects of general detection tasks, such as vehicle velocity. To overcome these limitations, it decomposes the detection error into individual normalized metric components, such as translation, orientation, etc.

All metrics are bounded between 0 and 1, with higher values indicating better performance. We refer readers to the original papers [4], [5] for more dataset-specific metric details.

V. EVALUATION

In this section, we evaluate t-READi under various sensory variation and modality-missing scenarios.

A. Benchmark

One primary motivation of t-READi is to accommodate various distortion inputs. We now summarize several common distortion types which prevail in daily driving conditions:

- *Fog-induced distortion:* Fog affects lidar systems by distorting point clouds at short distances and reducing intensity information. We adopt the approach from [22] to model lidar as a Linear Time Invariant (LTI) system, and calibrate sensors according to the manuals as necessary. We use α to characterize the meteorological optical range (MOR) explicitly and capture the fog density implicitly.

- *Snow-induced distortion:* Lidar in snow poses unique challenges compared to fog scenarios. In snow conditions, the air can be considered as a low-humidity medium with high-reflectivity snowflakes.
  We adopt the approach [21], and utilize β to represent the snowfall rate.

- *Motion blur:* This effect is common in images when the vehicle moves rapidly. For simplicity, we use a convolution with Gaussian kernel to simulate this blurring effect. A smaller kernel size corresponds to a lower speed and less pronounced blurring, and vice versa.

- *Exposure condition:* Auto exposure of cameras can result in poor image quality under poor illumination conditions. We
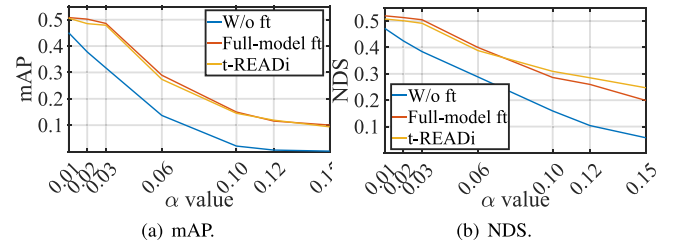

Fig. 8. t-READi maintains high performance when fog density deviates from pre-training.

use Gamma calibration to characterize different exposure levels. An image is classified as under-exposed for $\gamma \in [0, 1)$ and over-exposed for $\gamma \in (1, +\infty)$.

We adopt two models as baselines to demonstrate the capability of t-READi. The first one is a pre-trained BEVFusion model *without further fine-tuning* (**w/o ft**). It is trained with a relatively high point density ($\approx$ 200k points) along with $6\times$ well-exposed RGB images from different perspective resized to 256×704 pixels.

No distortion is adopted during training, and the framework proposed in Section III-C is not applied. We also employ a *fine-tuned model* (**full-model ft**) as a baseline, all parameters in the model can be fine-tuned.

B. Object Detection

We have demonstrated in Section II-A that sensory input variation considerably impacts inference performance. To evaluate the task- and model-agnostic nature of t-READi, we first focus on the object detection task and lidar modality. We assess its effectiveness by systematically varying each type of lidar input to specific levels. Subsequently, we present the performance of t-READi for each variation setting.

a) Effects of Fog-induced Distortion: We introduce noise points by simulating scattering using fog density indicators of $\alpha \in \{0.01, 0.02, 0.03, 0.06, 0.1, 0.12, 0.15\}$. As a rule of thumb, when $\alpha = 0.06$, the MOR is approximately 50 m. Fig. 8 demonstrates the high inference accuracy of t-READi across all quantified fog density levels. In contrast, w/o ft experiences significant accuracy degradation when the fog density deviates from the training data. t-READi achieves 15% average accuracy gain over the w/o ft model while only 2% average accuracy loss over the full-model ft, but with much higher memory efficiency, at a cost of only 2.6% memory and 2.1% inference time overhead respectively. Our key observations are as follows. First, training a model from scratch with severely deviated data (e.g., $\alpha > 0.1$) is risky and results in corruption at an early stage, leading to a result far from full-model ft. Second, only a small fraction of distorted points ($\approx 5\%$ on average when $\alpha = 0.03$) significantly impairs the inference accuracy. In these scenarios, radar data can provide more valuable insights.

b) Effects of Snow-induced Distortion: We generate snowflake-scattered and high-reflectivity points with the snowfall rate indicator $\beta \in \{0.5, 1.0, 1.5, 2.0, 2.5\}$. Fig. 9 demonstrates the high inference accuracy of t-READi across all quantified snowfall rate levels. t-READi achieves an average accuracy

TABLE II
3D AVERAGE PRECISION ON DENSE DENSE FOG SPLIT

| Model	Method	Car AP@.5IoU easy mod hard	Cyclist AP@.25IoU easy mod hard	Pedestrian AP@.25IoU easy mod hard
PointRCNN [55]	train with clear:	42.43 42.24 40.29	22.52 23.52 25.62	43.23 40.16 37.05
	t-READi:	47.94 46.07 42.07	27.60 27.65 29.21	45.52 43.38 41.10
	full ft:	48.52 46.77 42.19	27.66 27.94 29.37	45.65 43.23 41.15
	train with dense fog:	49.31 47.33 42.94	27.89 27.89 29.29	45.79 43.47 41.33
PV-RCNN [54]	train with clear:	40.19 40.93 39.66	24.33 24.63 24.63	42.67 41.04 39.59
	t-READi:	46.69 47.38 46.51	29.63 28.50 28.22	46.03 45.23 43.76
	full ft:	47.36 47.55 46.89	29.92 29.01 27.54	46.51 45.44 44.91
	train with dense fog:	47.81 47.86 47.12	30.11 29.42 27.91	46.73 45.61 45.12
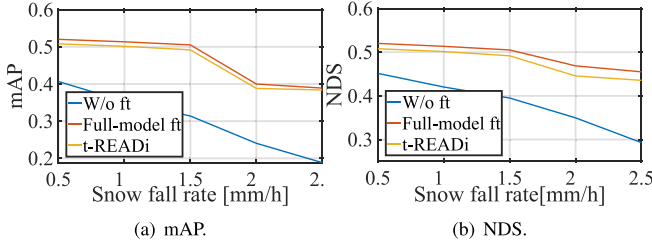
Fig. 9. t-READi maintains high performance when snow fall rate deviates from pre-training.
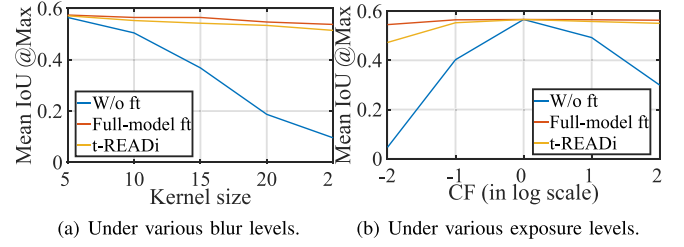

Fig. 10. t-READi maintains high performance when camera modality performance deviates from pre-training.

Fig. 11. t-READi suffers from marginal overhead in additional memory or load/inference time.

loss of only $2 \sim 3\%$ compared to the full-model ft, while offering higher memory efficiency. We note that snowflake scatter has less impact than reflection from melted snow, this also explains why fog affects lidar more than snow.

C. Primary Results on DENSE Dataset

To justify the methodology of t-READi as neither dataset- nor model-specific, we focus on lidar modality under practical dense fog conditions. We evaluate our approach using PointR-CNN [55] and PV-RCNN [54], which differ from the model used in Section V-B in terms of how raw points are represented and how proposals are generated. We train each model on the clear-split dataset (model M_c) and the dense-fog-split dataset from scratch. We then fine-tune or apply t-READi to M_c and compare their performance on the dense-fog-split dataset.

The results of the class-wise AP metric are shown in Table II, and we chose $r = 4$ and $k = 0$ as the parameters for t-READi, as neither is based on transformer. An interesting observation is dataset-specific: for DENSE, when testing a model trained on clear data with severe adverse weather data, the performance degradation ($\approx 7AP$) is much smaller than that on NuScenes ($\approx 15mAP$). We hypothesize that the multi-lidar setup in DENSE enhances its robustness and that the weather-based split achieves better consistency within the dataset. Despite this, t-READi demonstrates its generalizability: for every class, it incurs at most a $1 \sim 1.5\%$ AP loss compared to the full-model fine-tuning approach. Moreover, for subtle instances occupied by fewer points, the performance gap between t-READi and full-model fine-tuning is even more marginal.

D. Semantic Segmentation

We focus on the camera modality in the segmentation task, where the pre-trained model relies heavily on transformers. We pick the maximum IoU under each variation setting.
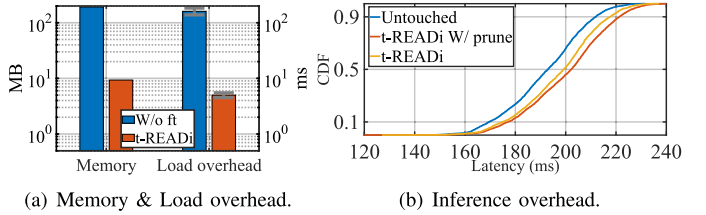
a) Effects of Motion Blur: We apply different kernel sizes chose from $\{5, 10, 15, 20, 30\}$ to the original datasets to obtain blurred datasets. In Figs. 10(a) and 11(a), t-READi demonstrates robustness even under heavy distortion, with only slight $1 \sim 2\%$ IoU loss compared to the full-model ft approach. Moreover, t-READi achieves this by updating only 3% of the parameters. This approach also offers higher memory efficiency, with a memory overhead of 4.9% and an load time overhead of 5.8%. On average, it takes 160 ms with a standard deviation of 25 ms to load a 200 MB model, whereas t-READi completes the switch within 10 ms in most cases.

b) Effects of Poor Exposure: Exposure variations in images can blur the boundaries between different textured zones. We apply pixel-wise gamma calibration with values of $\gamma \in \{0.25, 0.5, 1, 2, 4\}$ (Where "CF" denotes Calibration Factor for short in Fig. 10(b)). To our surprise, when the image is heavily under-exposed (e.g., $\gamma = 0.25$), t-READi shows a significant margin of 7 IoU points compared to the full-model ft. Additionally, the properties of under-exposure effects may differ from over-exposure effects, as indicated by the marginal gap of approximately 1.5 IoU points at $\gamma = 4$. Furthermore, we observed that both t-READi and the full-model ft can improve segmentation performance when the camera is pushed to its limits, whereas this trend is not observed for detection

and lidar modality. This discrepancy can be attributed to the relatively simple and less lossy compression applied to the camera-generated variations, while the fog/snow simulation on lidar introduces more noise. Exploring a more comprehensive framework for modeling the behavior of cameras under adverse conditions would be an interesting avenue for future research.

E. Discussions on Inference Latency

t-READi significantly mitigates memory and load time overhead in scenarios necessitating multiple model variants, while the exact inference time per frame can be compensated. Leveraging the `torch.profiler` interface in PyTorch, we compare the overall inference time distributions between the original model and its t-READi-enhanced counterpart, as illustrated in Fig. 11(b). t-READi introduces an approximate 3.2% overhead in both averaged and tail latency, while this increment exceeds the parameter injection overhead (calculated as $4.9\% - 3\% = 2.9\%$), it remains modest. Surprisingly, the tail-latency (90th percentile) to averaged-latency ratio is notably large (≈ 1.12). Upon profiling the latency contributed by the encoder from the camera modality, we observe a corresponding ratio as low as 1.04. Furthermore, we identify that the prolonged tail latency primarily stems from the quantization pipeline in the lidar modality encoder, which is beyond the scope of t-READi.

In addition, considering the significant variability in latency as advocated by [42], tail latency emerges as a more pertinent metric for meeting real-time constraints. To alleviate tail latency within the context of t-READi, we take a simple step, employ pruning in the vast frozen layers. Specifically, we reset all parameters whose absolute value is less than 1×10^{-3}, resulting in approximately 10% parameter pruning. We find that this approach nearly halves the overhead with no discernible accuracy degradation. Moreover, there exists potential for further reduction in tail latency by applying more advanced techniques such as parameter quantization [79] to the vast frozen layers, along with the implementation of more efficient lidar point cloud quantization operators.

F. Missing Sensing Modalities

We have shown in Section II-B that missing modality incurs information loss, and filling the missing modality with 0's does not help because doing so only introduces bias into the network. Correspondingly, we propose to improve t-READi's robustness to missing modality by contrastive learning. We evaluate the robustness of t-READi to missing modalities by removing a certain number of sensors during inference and observe how the DNN's performance drops. In the following, we show the mAP and NDS of t-READi under different missing modality scenarios. On the x-axis of the figures, "Full" denotes there are no missing sensors, and "number+C/L" denotes the number of remaining cameras and lidars (there is a total of 6 cameras and 1 lidar).

a) Missing camera: We first present how our contrastive learning framework deals with missing cameras. It can be seen in Fig. 12 that mAP and NDS under contrastive learning decrease
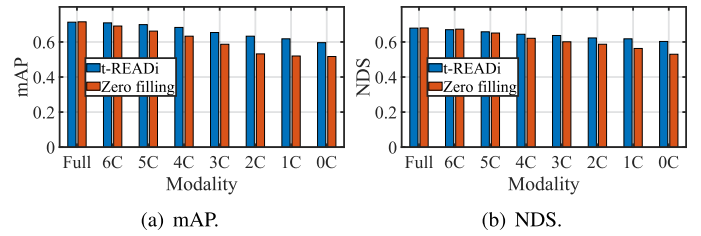

Fig. 12.   t-READi compensates for the information loss of missing cameras.
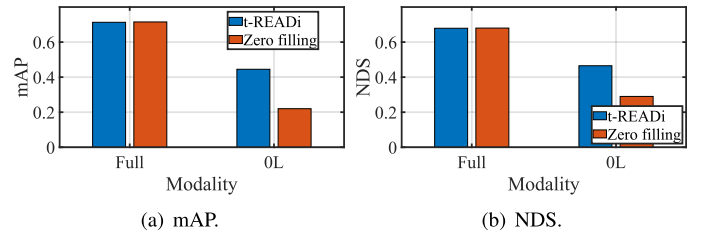

Fig. 13.   t-READi compensates for the information loss of missing lidar.

from 0.71 to 0.59 and from 0.68 to 0.60, respectively, and the mAP and NDS under zero filling decrease from 0.72 to 0.52 and from 0.68 to 0.53, respectively. Although we can still observe mAP and NDS decrease when contrastive learning is adopted, the decreased mAP and NDS is smaller than those of zero filling, proving t-READi's robustness to missing modalities. Moreover, we find that in some cases (e.g., "3C" for zero filling and "1C" for t-READi), t-READi can achieve comparable performance with zero filling with fewer camera sensors. The result shows that fewer sensors can be used by t-READi to achieve the same performance than previous solutions, thus proving t-READi's better efficiency.

b) Missing lidar: We then demonstrate t-READi's robustness to missing lidar. As Fig. 13(a) and (b) show, removing the lidar and filling the missing modality with 0's renders the system unusable since it only gives an mAP and NDS of 0.22 and 0.29, respectively. The poor performance can be attributed to the fact that lidar is the dominant sensor and removing it results in major information loss. As for the contrastive learning adopted by t-READi, we find that while maintaining the same performance under full modalities, it greatly improves t-READi's performance with missing lidar. The result demonstrates that the contrastive learning of t-READi can effectively compensate for missing lidar information by learning from the complementary radar modality.

c) Mixed missing sensors: We also evaluate t-READi's robustness in cases where radar and lidar data are. It can be seen in Fig. 14 that even when there is no radar or lidar data, t-READi still performs significantly better than the zero filling baseline in terms of mAP and NDS. This demonstrates that t-READi is able to effectively make use of remaining camera data, even when there are only a small number of cameras. Overall, our results show that t-READi offers both efficiency and robustness in multi-modal perception tasks.
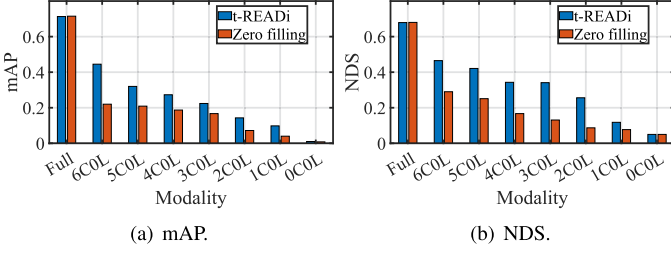
(a) mAP. (b) NDS.

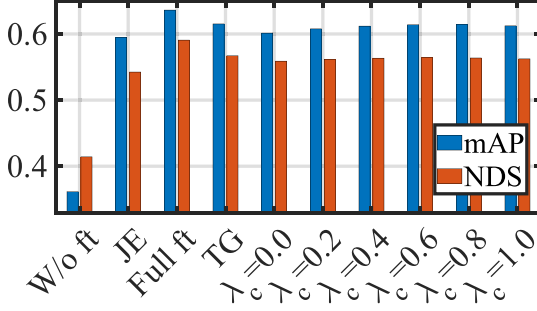Fig. 14. t-READi compensates for both missing cameras and lidar information loss.

Fig. 15. t-READi is capable of handling multi-variation domain with mono-variation domain.

Fig. 16. Loss profile of various tuning schemes, as parameters of t-READi vary.

TABLE III
IMPACT OF MODULE'S PARAMETERS

	mAP	NDS	Exist budget	Add budget
no op	31.71	38.43	N/A	N/A
just pred	37.11	41.91	0.63%	N/A
k, r=null	42.23	45.30	0.87%	N/A
k=4, r=2	47.90	49.17	0.87%	1.83%
k=4, r=4	47.23	48.62	0.87%	1.00%
k=8, r=2	47.67	48.87	0.87%	2.00%
alt org	46.43	47.62	0.87%	3.51%

G. Combating Cross-Domain Variations

To evaluate t-READi's ability of handling of cross-domain variation using previously seen mono-domain variation, we extend the experiment in Section V-B. The lidar/camera variation levels are set to $\alpha = 0.03$ and $\gamma = 0.5$ respectively. Our results are shown in Fig. 15. Compared to the full model ft operation in column 3 and tuning both variation domains together (referred to as **TG** in legend), t-READi achieves significant performance with mono-variation domain variants. Specifically, updating the exclusive layers, i.e., modality-specific encoders (referred to as **JE** in legend), already achieves an impressive performance loss of only $4 \sim 5$ mAP and NDS compared to full model ft. With interpolation, the local maximum is achieved at $\lambda_c = 0.8$, reducing the gap to $2 \sim 2.5$ mAP and NDS loss. The approximation to the TG variant is remarkable, with a marginal gap of only 0.1 and 0.4 for mAP and NDS, respectively. In challenging cross-domain variation scenarios, we propose interpolating from mono-variation domains as a simple yet effective method, yielding sub-optimal performance but with greater scalability. This insight has the potential to be extended to more diverse sensor systems and over-the-air variations. When variations from different sensors are superimposed, t-READi achieves exceptional parameter efficiency compared to full model ft. It combines interpolation with variation-aware adaptation, ensuring scalability: the overall parameter space increases linearly with the number of variation quantization levels, regardless of the number of modalities.

H. Ablation Study and Micro-Benchmark

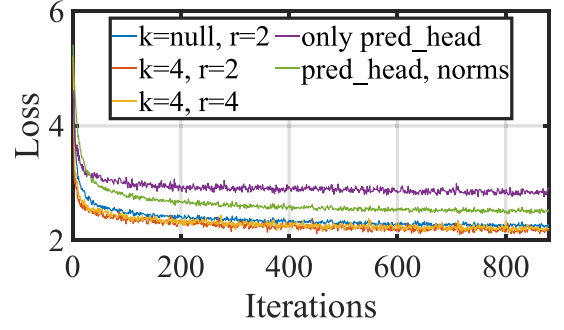In this section, we present how the components of t-READi are fused properly and evaluate the critical hyperparameters to justify our choices. The settings are consistent with Section V-B, with parameter α set to 0.03.

a) Ablation Study: We first study the effectiveness of individual module of t-READi. We set the default values of intrinsic rank upper bound k and squeeze ratio r to 4 and 2, respectively. Fig. 16 provides a clear illustration: compared to the common practice of adjusting only a few downstream layers (typically lightweight prediction heads), t-READi demonstrates its effectiveness and efficiency by prioritizing the adjustment of Batch Normalization (BN) layers as the first step. This approach outperforms the conventional practice, which often reaches a plateau early on. Moreover, injecting transparent tiny modules further boosts t-READi's performance. By injecting the tiny modules, t-READi achieves significant loss reduction in the first epoch compared to conventional schemes that require up to 10 epochs or more.

b) Performance and Overhead Trade-off: In addition, we conducted an evaluation of the effects of k and r on mAP, NDS, and memory budgets. Table III presents a summary of the results. Here we have 4 insights: 1) BN contribute a small percentage ($\approx 0.12\%$) to memory usage, ensuring t-READi's memory efficiency. 2) Increasing r for adapter squeezing memory usage but leads to fluctuating and slower convergence, we choose 2 as the default ratio. 3) t-READi's low overhead is due to bulky 3D convolution modules, compared to 2D counterparts. This makes adapter less efficient for 2D convolution-intensive scenarios. While with the default configuration, the budget still remains within 5%. A no-projection fully connected layer was omitted in earlier versions of the adapter to save space, sacrificing a maginal gain of 0.5 points. 4) A relatively small k effectively mitigates rank collapse. Larger k (e.g., 8) worsen performance even more than not using this option. One may complain this improvement is more marginal, while can be attributed to the

reduced reliance on the transformer and the minimal impact on the additional budget.

c) The Effectiveness of t-READi beyond Adding More Parameters: One might argue that the improvement achieved by t-READi is simply a result of adding more parameters. However, we show that this is not the case by comparing t-READi with an alternative configuration that allocates the budget to convolution modules instead of t-READi modules. The configuration has more parameters than t-READi, but it performs worse, as shown in the last row of Table III. This demonstrates that t-READi is effective not because it has more parameters, but because it has a better organization of them.

VI. RELATED WORK

a) Multi-modal Sensor Fusion: Multi-modal fusion is essential for autonomous driving systems' perception, combining data from multiple sensors to enhance accuracy and robustness. Major paradigms for multi-modal fusion can be categorized into early, deep, and late fusion based on the *stage* (raw data, feature, proposal) where the fusion occurs [10], [30], [56], [71]. Each paradigm can employ different fusion policies, ranging from simple concatenation or element-wise addition to more sophisticated methods using differentiable and learnable functions. These traditional paradigms have limitations as they fuse all modalities simultaneously. Recent works have attempted to fuse modalities selectively, achieving more robust performance [7].

b) Multi-modal Detection & Segmentation: Detection and segmentation are vital tasks in autonomous driving perception. Multi-modal 3D detection and segmentation techniques leverage complementary modalities due to the deficiencies of the camera's depth information and the lidar's semantic information. For 3D multi-modal object detection, [41] uses cross-modal attention to fuse camera and lidar features. [19] combines edge-assisted 2D detection with on-device 3D boxes for lightweight, hybrid 3D object detection. For 3D semantic segmentation, [11] projects 2D CNN-extracted image features to 3D space and fuses them with lidar data for voxel-wise segmentation. In [58], permutohedral lattice representation fuses multi-modal data for 3D semantic segmentation.

c) Combating Missing Modalities: Multimodal systems face challenges with missing modalities, negatively impacting performance. Researchers proposed methods for missing modality imputation and performance improvement. [64] uses cascaded residual autoencoder (CRA) for missing modality imputation, stacking residual autoencoders (RAs) that iteratively output the difference between incomplete and complete data. [50] learns robust joint representations by translating between modalities, using a translation network to establish a consistent, robust common space. This approach allows the system to handle missing modalities at test time by inferring them from available modalities.

d) Networking and system support for AI inference: The scope of support for AI spans the entire lifecycle of AI services, from *data collection* and *model training* to *model inference* [53]. In the *inference* phase, multi-dimensional QoS can be achieved. To ensure power efficiency and low latency for end users with limited computing capabilities and battery power, model partition techniques divide a DNN into multiple sub-models, embedding them into different network nodes to conduct inference collaboratively [59], [75]. Additionally, model compression leverages techniques such as weight pruning [25], [37], parameter quantization [79], and encoding [51] to fit DNNs into *Application Specific Integrated Circuit* (ASIC), thereby overcoming memory and power constraints [24]. However, while a typical driving system is not as energy-constrained as energy-harvesting systems, real-time reaction to traffic conditions is critical, necessitating that processing always meets strict deadlines. Communication between the vehicle and edge servers (e.g., smart lampposts) is too costly, as an autonomous driving vehicle can generate hundreds of megabytes of raw sensor data per second. Most model compression techniques and ASIC designs focus on "single-branch" networks like AlexNet [36], VGG [57], while thoroughly compressing or designing ASICs for "multi-branch" networks like ResNet [26] is challenging. This is even more complex for attention-based advanced networks, which are the de facto solutions for autonomous driving [63].

e) Efficient Fine-Tuning for LLM: Several methods for tuning LLM are proposed to achieve parameter efficiency. Low-Rank Adaptation (LoRA) [29] updates every parameter in low-rank matrices at each iteration. Prefix-tuning [40] adds a series of trainable vectors, known as prefix tokens, to each layer in LLM; these tokens can be tailored to specific tasks. Unlike prefix-tuning, prompt-tuning allows tokens to be inserted either as a prefix or anywhere within the input tokens. Advanced techniques, such as those introduced in [43], incorporate continuous prompts at each layer rather than solely at the input layer, boosting performance in natural language understanding tasks. Additionally, memory efficiency is also a critical topic. Models can first be quantized or decomposed into low-precision data types, and only the LoRAs corresponding to specific downstream tasks are tuned [12]. In [32], scalar vectors resulting from decomposition are also updated. Our method is greatly inspired by LoRA, thanks to its scalability.

VII. DISCUSSION

In this section, we discuss the existing limitations of t-READi's pipeline and potential techniques that could enhance t-READi when integrated.

a) High-Fidelity variations: t-READi employs an *online* version of camera modality variation to conserve disk space. However, an *offline* approach, such as estimating the blur kernel and then generating artificial images accordingly, would be a more comprehensive method. Concurrently with t-READi, a study by [35] also explores lidar-perception robustness, simulating blur effects for lidar by introducing jitter noise subject to certain distributions.

b) Additional Modalities: t-READi currently focuses on camera and lidar binary modalities. When additional modalities are introduced, simple fusion policies, such as concatenation followed by element-wise multiplication [4], have been shown to be less robust [7]. Consequently, t-READi's robustness can be

enhanced further through the adoption of more advanced fusion methods.

c) AV-DNN Accelerators: t-READi functions akin to a gear mechanism, facilitating the swift switching of AV-DNNs rather than serving as an inference accelerator. Although it only marginally compromises inference latency, as discussed in Section V-E, we have identified that the primary bottleneck in current inference latency resides within the lidar quantization pipeline, which falls outside the scope of t-READi. Addressing this bottleneck can be approached through software- or hardware-based optimizations, such as refining quantization parameters or offloading the quantization preprocessing to ASICS. Both methods are compatible with t-READi and hold significant potential for accelerating this stage.

VIII. CONCLUSION

Taking an important step towards full driving automation, we have proposed t-READi in this paper for robust and efficient multimodal inference for autonomous driving. Employing a novel partial weight adaptation mechanism and a data imputation method by autoencoder, t-READi gracefully handles the heterogeneous data caused by sensor parameter variation and missing modality, thus releasing its full potential in the vehicle detection task. With extensive experiments under highly heterogeneous scenarios and comparisons with other baselines, we demonstrate the promising performance of t-READi in vehicle detection for autonomous driving. Overall, we believe that t-READi represents an important step forward in the development of robust and efficient multimodal inference for autonomous driving.

REFERENCES

[1] X. Bai et al., "TransFusion: Robust LiDAR-Camera fusion for 3D object detection with transformers," in *Proc. 35th IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 1080–1089.

[2] E. Ben-Zaken, S. Ravfogel, and Y. Goldberg, "BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," 2021, *arXiv:2106.10199*.

[3] M. Bijelic et al., "Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather," in *Proc. 33rd IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11682–11692.

[4] M. Bijelic et al., "Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather," in *Proc. 33rd IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11679–11689.

[5] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. 33rd IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11618–11628.

[6] A. Ceccarelli and F. Secci, "RGB cameras failures and their effects in autonomous driving applications," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 4, pp. 2731–2745, Jul./Aug. 2023.

[7] C. Chen, S. Rosa, C. X. Lu, B. Wang, N. Trigoni, and A. Markham, "Learning selective sensor fusion for state estimation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 03, 2022, doi: 10.1109/TNNLS.2022.3176677.

[8] K. Chen et al., "MMDetection: Open MMLab detection toolbox and benchmark," 2019, *arXiv: 1906.07155*.

[9] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[10] Y. Chen, J. Yu, L. Kong, H. Kong, Y. Zhu, and Y.-C. Chen, "RF-Mic: Live voice eavesdropping via capturing subtle facial speech dynamics leveraging RFID," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 7, no. 2, pp. 1–25, 2023.

[11] A. Dai and M. Nießner, "3DMV: Joint 3D-multi-view prediction for 3D semantic scene segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 452–468.

[12] T. Dettmers, A. Pagnoni, and A. Holtzman, "QLoRA: Efficient finetuning of quantized LLMs," 2023, *arXiv:2305.14314*.

[13] V. V. Dixit, S. Chand, and D. J. Nair, "Autonomous vehicles: Disengagements, accidents and reaction times," *PLoS One*, vol. 11, 2016, Art. no. e0168054.

[14] Y. Dong, J.-B. Cordonnier, and A. Loukas, "Attention is not all you need: Pure attention loses rank doubly exponentially with depth," 2021, *arXiv:2103.03404*.

[15] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv: 2010.11929*.

[16] M. Dundar, B. Krishnapuram, J. Bi, and R. B. Rao, "Learning classifiers when the training data is not IID," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, 2007, pp. 756–61.

[17] D. Feng et al., "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1341–1360, Mar. 2021.

[18] S. Grigorescu, T. Cocias, B. Trasnea, A. Margheri, F. Lombardi, and L. Aniello, "Cloud2Edge elastic AI framework for prototyping and deployment of AI inference engines in autonomous vehicles," *Sensors*, vol. 20, no. 19, pp. 5450, 2020.

[19] Y. Guan, X. Hou, N. Wu, B. Han, and T. Han, "DeepMix: Mobility-aware, lightweight, and hybrid 3D object detection for headsets," in *Proc. 20th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2022, pp. 28–41.

[20] T. Guo, "Cloud-based or on-device: An empirical study of mobile deep inference," in *Proc. IEEE Int. Conf. Cloud Eng.*, 2018, pp. 184–190.

[21] M. Hahner et al., "LiDAR snowfall simulation for robust 3D object detection," in *Proc. 35th IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16343–16353.

[22] M. Hahner, C. Sakaridis, D. Dai, and L. V. Gool, "Fog simulation on real LiDAR point clouds for 3D object detection in adverse weather," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 15263–15272.

[23] S. Han, "Putting AI on diet: TinyML and efficient deep learning," in *Proc. IEEE Int. Symp. VLSI Des. Autom. Test*, 2021, pp. 1–1.

[24] S. Han et al., "EIE: Efficient inference engine on compressed deep neural network," 2016, *arXiv:1602.01528*.

[25] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," 2016, *arXiv:1510.00149*.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. 29th IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[27] M. Hnewa and H. Radha, "Object detection under rainy conditions for autonomous vehicles: A review of state-of-the-art and emerging techniques," *IEEE Signal Process. Mag.*, vol. 38, no. 1, pp. 53–67, Jan. 2021.

[28] N. Houlsby et al., "Parameter-efficient transfer learning for NLP," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 2790–2799.

[29] E. Hu et al., "LoRA: Low-rank adaptation of large language models," 2021, *arXiv:2106.09685*.

[30] K. Huang, B. Shi, X. Li, X. Li, S. Huang, and Y. Li, "Multi-modal sensor fusion for auto driving perception: A survey," 2022, *arXiv:2202.02703*.

[31] Intel Corporation, "Intel xeon gold 6226 processor," 2022. Accessed: Jul. 28, 2022. [Online]. Available: https://www.intel.com/content/www/xa/en/products/sku/193957/intel-xeon-gold-6226-processor-19-25m-cache-2-70-ghz/specifications.html

[32] J. Kim et al., "Memory-efficient fine-tuning of compressed large language models via sub-4-bit integer quantization," 2023, *arXiv:2305.14152*.

[33] H. Kong, L. Lu, J. Yu, Y. Chen, and F. Tang, "Continuous authentication through finger gesture interaction for smart homes using WiFi," *IEEE Trans. Mobile Comput.*, vol. 20, no. 11, pp. 3148–3162, Nov. 2021.

[34] H. Kong et al., "m3Track: MmWave-based multi-user 3D posture tracking," in *Proc. 20th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2022, pp. 491–503.

[35] L. Kong et al., "Robo3D: Towards robust and reliable 3D perception against corruptions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 19937–19949.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, pp. 84–90, 2012.

[37] G. Lee and K. Lee, "DNN compression by ADMM-based joint pruning," *Knowl.-Based Syst.*, vol. 239, 2022, Art. no. 107988.

[38] J. Lee et al., "On-device neural net inference with mobile GPUs," 2019, *arXiv: 1907.01989*.

[39] D. Li et al., "Motion inspires notion: Self-supervised visual-LiDAR fusion for environment depth estimation," in *Proc. 20th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2022, pp. 114–127.

[40] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," 2021, *arXiv:2101.00190*.

[41] Y. Li et al., "DeepFusion: LiDAR-camera deep fusion for multi-modal 3D object detection," in *Proc. 35th IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17182–17191.

[42] S.-C. Lin et al., "The architectural implications of autonomous driving: Constraints and acceleration," in *Proc. 23rd Int. Conf. Architectural Support Program. Lang. Operating Syst.*, 2018, pp. 751–766.

[43] X. Liu et al., "P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks," 2022, *arXiv:2110.07602*.

[44] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. 34th IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9992–10002.

[45] Z. Liu et al., "BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2023, pp. 2774–2781.

[46] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. 35th IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11966–11976.

[47] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. 7th Int. Conf. Learn. Representations*, 2017, pp. 1–18.

[48] NVIDIA, "GeForce RTX 3080 family," 2022. Accessed: Jul. 28, 2022. [Online]. Available: https://www.nvidia.com/en-in/geforce/graphics-cards/30-series/rtx-3080-3080ti/

[49] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," 2019, *arXiv: 1912.01703*.

[50] H. Pham, P. P. Liang, T. Manzini, L.-P. Morency, and B. Póczos, "Found in translation: Learning robust joint representations by cyclic translations between modalities," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 6892–6899.

[51] B. Reagen et al., "Weightless: Lossy weight encoding for deep neural network compression," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 4321–4330.

[52] S. Shaheen and M. A. Bouzaghrane, "Mobility and energy impacts of shared automated vehicles: A review of recent literature," *Curr. Sustainable/Renewable Energy Rep.*, vol. 6, no. 4, pp. 193–200, 2019.

[53] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 1–30, First Quarter 2022.

[54] S. Shi et al., "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. 33rd IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10526–10535.

[55] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. 32nd IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 770–779.

[56] S. Shi et al., "VIPS: Real-time perception fusion for infrastructure-assisted autonomous driving," in *Proc. 28th Annu. Int. Conf. Mobile Comput. Netw.*, 2022, pp. 133–146.

[57] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, *arXiv:1409.1556*.

[58] H. Su et al., "SPLATNet: Sparse lattice networks for point cloud processing," in *Proc. 31st IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2530–2539.

[59] S. Teerapittayanon, B. McDanel, and H. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," in *Proc. 37th IEEE Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 328–339.

[60] Tesla, "Autopilot: Future of driving," 2022. Accessed: Jul. 25, 2022. [Online]. Available: https://www.tesla.com/en_SG/autopilot

[61] Texas personal injury attorneys, "Examining autonomous car accidents," 2023. Accessed: May 02, 2024. [Online]. Available: https://www.lgrlawfirm.com/blog/examining-autonomous-car-accidents-and-statistics-2/

[62] The Washington Post, "Tesla autopilot crash analysis," 2023. Accessed: May 02, 2024. [Online]. Available: https://www.washingtonpost.com/technology/interactive/2023/tesla-autopilot-crash-analysis/

[63] Think Autonomous, "A look at Tesla's occupancy networks," 2022. Accessed: Apr. 10, 2024. [Online]. Available: https://www.thinkautonomous.ai/blog/occupancy-networks/

[64] L. Tran, X. Liu, J. Zhou, and R. Jin, "Missing modalities imputation via cascaded residual autoencoder," in *Proc. 30th IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1405–1414.

[65] Uber Technologies Inc., "Self-driving perception & prediction," 2022. Accessed: Jul. 25, 2022. [Online]. Available: https://www.uber.com/us/en/atg/research-and-development/perception-and-prediction/

[66] S. Van Buuren, *Flexible Imputation of Missing Data*. Boca Raton, FL, USA: CRC Press, 2018.

[67] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[68] A. Veit, M. J. Wilber, and S. J. Belongie, "Residual networks are exponential ensembles of relatively shallow networks," 2016, *arXiv:1605.06431*.

[69] Velodyne, "HDL-64E manual," 2022. Accessed: Jun. 22, 2023. [Online]. Available: https://www.manualslib.com/manual/185826/Velodyne-Hd-Hdl-64e-S2-1.html

[70] Waymo, "We're building the world's most experienced driver," 2022. Accessed: Jul. 25, 2022. [Online]. Available: https://waymo.com/?ncr

[71] L. Xie et al., "PI-RCNN: An efficient multi-sensor 3D object detector with point-based attentive cont-conv fusion module," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 12460–12467.

[72] D. Xu, D. Anguelov, and A. Jain, "PointFusion: Deep sensor fusion for 3D bounding box estimation," in *Proc. 31st IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 244–253.

[73] X. Xu et al., "TouchPass: Towards behavior-irrelevant on-touch user authentication on smartphones leveraging vibrations," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw.*, 2020, pp. 1–13.

[74] J. Yu, L. Lu, Y. Chen, Y. Zhu, and L. Kong, "An indirect eavesdropping attack of keystrokes on touch screen through acoustic sensing," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 337–351, Feb. 2021.

[75] L. Zhang, L. Chen, and J. Xu, "Autodidactic neurosurgeon: Collaborative deep inference for mobile edge intelligence via online learning," in *Proc. ACM Web Conf.*, 2021, pp. 3111–3123.

[76] X. Zhao, P. Sun, Z. Xu, H. Min, and H. Yu, "Fusion of 3D LIDAR and camera data for object detection in autonomous vehicle applications," *IEEE Sensors J.*, vol. 20, no. 9, pp. 4901–4913, May 2020.

[77] T. Zheng, A. Li, Z. Chen, H. Wang, and J. Luo, "AutoFed: Heterogeneity-aware federated multimodal learning for robust autonomous driving," in *Proc. 29th Annu. Int. Conf. Mobile Comput. Netw.*, 2023, pp. 209–223.

[78] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. 31st IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4490–4499.

[79] C. Zhu, S. Han, H. Mao, and W. J. Dally, "Trained ternary quantization," 2017, *arXiv:1612.01064*.
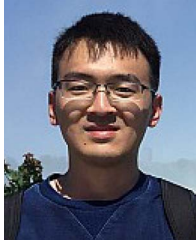
Pengfei Hu received the PhD degree in computer science from UC Davis. He is a professor with the School of Computer Science and Technology, Shandong University. His research interests are in the areas of cyber security, data privacy, and mobile computing. He has published more than 40 papers in premier conferences and journals on these topics. He served as TPC for numerous prestigious conferences, and associate editors for *IEEE Transactions on Wireless Communications* and *IEEE Internet of Things Journal*. He is the recipient of 2022 ACM SIGBED China Rising Star Award.
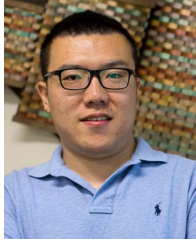
Yuhang Qian received the BE degree in computer science and technology from Shandong University. He is currently working toward the MD degree from the School of Computer Science, Shandong University, China. His research interests lie in vehicular network, RF-based communication and sensing.

Tianyue Zheng received the BEng degree from the Harbin Institute of Technology, China, the MEng degree from the University of Toronto, Canada, and the PhD degree from Nanyang Technological University, Singapore. He is currently an assistant professor with the Department of Computer Science and Engineering, Southern University of Science and Technology, China. His research interests include mobile and pervasive computing, the Internet of Things, and machine learning.

Xiuzhen Cheng (Fellow, IEEE) received the MS and PhD degrees in computer science from the University of Minnesota – Twin Cities, in 2000 and 2002, respectively. She is a professor with the School of Computer Science and Technology, Shandong University. Her current research interests include wireless and mobile security, cyber physical systems, wireless and mobile computing, sensor networking, and algorithm design and analysis. She has served on the editorial boards of several technical journals and the technical program committees of various professional conferences/workshops. She also has chaired several international conferences. She worked as a program director for the US National Science Foundation (NSF) from April to October in 2006 (full time), and from 2008 to 2010 (part time). She received the NSF CAREER Award, in 2004. She is a member of ACM.

Ang Li received the PhD degree in electrical and computer engineering from Duke University, and the PhD degree in computer science from the University of Arkansas. He is an assistant professor with the Department of Electrical and Computer Engineering, University of Maryland College Park. His research interests are in the areas of federated learning, edge computing, and data-driven privacy-enhancing techniques. He has published more than 50 papers in premier conferences and journals on these topics. He served as TPC for numerous prestigious conferences. His research has been recognized with a variety of awards, including the IEEE TCCPS Outstanding Ph.D. Dissertation Award, ACM KDD Best Student Paper Award, in 2020, and the 2022-2023 Duke ECE Department Outstanding Dissertation Award.

Jun Luo (Fellow, IEEE) received the BS and MS degrees in electrical engineering from Tsinghua University, China, and the PhD degree in computer science from EPFL (Swiss Federal Institute of Technology in Lausanne), Lausanne, Switzerland. From 2006 to 2008, he has worked as a postdoctoral research fellow with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada. In 2008, he joined the faculty of the College of Computing and Data Science, Nanyang Technological University in Singapore, where he is currently an associate professor. His research interests include mobile and pervasive computing, wireless networking, machine learning and computer vision, applied operations research, as well as security.

Zhe Chen received the PhD degree in computer science from Fudan University, China, with a 2019 ACM SIGCOMM China Doctoral Dissertation Award. He is an associate researcher within the School of Computer Science, Fudan University, and the co-founder of AIWiSe Ltd. Inc. Before joining Fudan University, he worked as a research fellow with NTU for three years, and his research achievements, along with his efforts in launching products based on them, have thus earned him 2021 ACM SIGMOBILE China Rising Star Award recently.

Yue Gao (Fellow, IEEE) received the PhD degree from the Queen Mary University of London (QMUL), U.K., in 2007. He is a professor with the School of Computer Science, Fudan University, China, and a visiting professor with the University of Surrey, U.K. He worked as a lecturer, a senior lecturer, a reader and the chair professor with QMUL and the University of Surrey, respectively. He has published more than 200 peer-reviewed journal and conference papers. His research interests include sparse signal processing, smart antennas and cognitive networks for mobile and satellite systems. He was a co-recipient of the EU Horizon Prize Award on Collaborative Spectrum Sharing, in 2016 and an Engineering and Physical Sciences Research Council fellow, in 2017. He is a member of the Board of Governors and a distinguished speaker of the IEEE Vehicular Technology Society (VTS), the chair of the IEEE ComSoc Wireless Communication Technical Committee, and the past chair of the IEEE ComSoc Technical Committee on Cognitive Networks. He has been an editor of several IEEE Transactions and Journals, the Symposia chair, the Track chair, and other roles in the organizing committee of several IEEE ComSoC, VTS, and other conferences.