UNIVERSITY OF
CAMBRIDGE

Decentralized Training of Acoustic Models for Speech Recognition

PhD Proposal

Yan Gao

Fitzwilliam

First year report submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Contents

Chapter 1

Introduction

Neural networks are now the state-of-the-art in automatic speech recognition (ASR) tasks. This success highly relies on powerful computer hardware (e.g. GPUs in the data centre) and large-scale data to train the model. With the increasing proliferation of mobile devices (e.g. phones, tablets), an unprecedented amount of user data collected by the devices' microphones provide an opportunity for even more robust and accurate ASR models – if this additional data could be harnessed. However, mobile devices usually have constrained computational capabilities, limited communication/network and power consumption. Additionally, audio and speech data is by its nature very sensitive, requiring strong anonymity and privacy guarantees of any training method that may utilize it.

To exploit the treasured user data for model training without violating privacy, federated learning (FL) [81] provides a possible solution. FL is a decentralised computation paradigm that can be used to train neural networks directly on-device. Particularly, it reduces privacy and security risks, while still being able to utilise user data in the training process. Federated training involves the data that is distributed on the mobile devices (refer to as *clients*), and learns a global model by aggregating updates from local computation on a central *server*. The local training data on each client is never uploaded to the server, and only the locally-computed updates are communicated, from which model training is decoupled from the requirement of directly accessing the raw user data. FL has achieved much success within a diverse range of practical applications, such as Gboard mobile keyboard [94, 48, 126], Pixel phones [1], Android Messages [113], medical research [24], hot-word detection [70], etc. However, federated training of acoustic models for speech recognition has received very little scientific study, and even the most basic questions as to how to train ASR models using federated methods remain unresolved.

There are many challenges impeding the development of federated training for ASR models. First, the training data on each mobile device is generated from a particular user, and the recording conditions of the devices are varied due to diverse types of microphones and acoustic environments. This is likely to cause the distribution of user data not

to be identical and independent, and hence not to be representative of the population distribution. Second, the amount of local data on the devices may dramatically vary, perhaps even by several orders of magnitude, because some users may make more frequent use of their devices than others. Third, a more complex model is required for ASR tasks to gain an acceptable level of performance relative to other types of data (e.g. images), due to the complex structure of audio (long sequences, variable-length and high-dimension) [120]. Speech models usually contains a robust encoder to extract features from raw data and a decoder for transcription, leading to a much larger model size [2, 98, 20]. Training ASR models on resource-constricted mobile devices will therefore likely cause computational efficiency problems. Forth, speech models require more data in each client to converge [46, 2, 110], but this may not be available on the clients - especially those devices that are less frequently used. Fifth and finally, labels of speech data might not be easy to collect on mobile devices than other data, e.g., image labels can be defined by natural user interaction with their photo app, but labeling is much less intuitive and difficult to segment correctly [81, 58].

In my thesis, I will dive into the difficulties that prevent effective speech model to be trained under federated settings and propose solutions associated with the challenges above. Concretely, I plan to divide my investigation into three stages as follows. First, study the existing approaches in federated training and cosider how they can be integrated into state-of-the-art methods for ASR modeling. Second, explore novel FL training schemes by introducing new techniques including knowledge distillation (KD), data selection, self-supervision and neural architecture search (NAS). Third, enable the proposed algorithms to be realised in real edge devices. The overarching goal of my thesis is to achieve efficient federated training of ASR models while diminishing communication and compute bottlenecks and in doing so enable clients to produce models customized to users and their environments.

Towards advancing these various research aims I have completed three projects in my 1st year. The first one [34] proposes three multi-teacher distillation strategies for acoustic models, which integrates the error rate metric to the teacher selection. In this way, it directly distills and optimises the student toward the relevant metric for speech recognition. I developed these methods have the potential to address a number of the challenges faced under FL settings. Currently they have been studied under a centralized training context, but the next step will be to study them under FL conditions. The second one [35] presents the first empirical study on attention-based Seq2Seq ASR model for realistic FL scenarios with three aggregation weighting strategies – standard FedAvg, loss-based aggregation and a novel word error rate (WER)-based aggregation. The methods are evaluated with *cross-silo* and *cross-device* FL with up to 2k clients on the naturally-partitioned and heterogeneous French Common Voice dataset. The third project [67]

develops an automated processing pipeline that integrates existing computer vision and signal processing techniques to convert videos of human activity into virtual streams of IMU data, and hence achieve robust and generalised on-body sensor-based human activity recognition (HAR). Similar to the first project, this work - in its current form - also lacks a direct FL element. However, I intend to extend this pipeline so that it can be trained under an FL setting. This first design assumes centralized training. But enabling on-body sensor devices to collectively train and revise their models will be a key next step, and how to do this remains largely an open question.

This document starts by providing a full detailed literature survey related to this report in Chapter 2. Chapter 3 elaborates the projects carried out in my first year. To further explain and justify the direction of FL training of acoustic models for speech recognition, Chapter 4 details a concrete proposal of work for the next two years of my PhD. This document concludes a research timeline proposed in Chapter 5.

Chapter 2

Background and Related Work

Following the challenges and problems raised in the introduction, I investigate the related work and existing methods in this chapter. Since this thesis links to many different areas, this chapter is organised by introducing the recent development and new training schemes of ASR, followed by the separate field of FL, and finally elaborating two techniques that could enhance federated training.

2.1 End-to-end Automatic Speech Recognition (ASR)

Automatic speech recognition, as a very natural human-machine interacting mechanism, has been widely studied in machine learning area since the 1970s [12], with linear prediction [55] and dynamic programming technology [117] been introduced into ASR. In the 1980s, hidden Markov model (HMM) technology began to be applied to speech recognition, and for a long time, the HMM-based models were the mainstream framework of speech recognition [68]. The speech states were modelled by HMM and use Gaussian mixed model (GMM) to model HMM states' observation probability, making milestone level progress in ASR tasks. More recently, deep neural networks (DNNs) were applied to speech recognition and have been integrated with HMM [23]. At the same time, deep learning techniques also aroused the development of an alternative approach — end-to-end ASR. Compared to the HMM-based model, the end-to-end model directly maps audio to transcriptions using a single model without domain expertise requirements. Here, we first make a comparison between conventional models (HMM-based) and end-to-end models.

- **HMM-Based Model**. The HMM-based model typically contains three independent components: acoustic, pronunciation and language model (Figure 2.1). The acoustic model aims to map input audio to feature sequence (e.g. phoneme). The pronunciation model is to build a mapping between phonemes and graphemes, which is constructed by professional human linguists. The language model is responsible

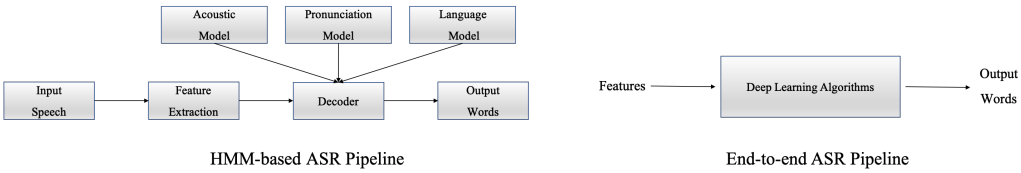

Figure 2.1: HMM-based and end-to-end ASR pipelines.

for mapping the character sequence to the final transcription. These three models are trained separately and require expert knowledge to create pronunciation lexicon and define phoneme sets.

- **End-to-End Model**. Other than disconnecting the training process in HMM-based model, the end-to-end model directly maps input speech to the sequence of words, which implicitly or explicitly contains two parts — 1) encoder, extracting feature from raw speech sequence; 2) decoder, building the alignment between feature sequence and transcription, and decoding the final prediction sentences (Figure 2.1). End-to-end speech recognition dramatically simplifies the process of HMM-based speech recognition without carefully-designed intermediate states and links the optimisation to final evaluation criteria (typically error rate). Based on the types of alignment, the end-to-end model can divide into three different categories: connectionist temporal classification (CTC)-based, transducers, and attention-based.

2.1.1 CTC-based model

In end-to-end training, the loss functions of DNN are calculated based on each time step of the sequence, which requires the explicit alignment between the output sequence of DNN and target sequence (labels). To solve this data alignment problem, CTC loss was proposed in [42] as a milestone of the development of end-to-end speech recognition. In the CTC process, the output sequence of the network, typically longer than the label sequence, can be regarded as a probability distribution over all possible label sequences, conditioned on a given input sequence. Note that there are multiple ways to align label sequences with the input sequence (paths) [42]. Then, the total probability can be calculated by summing the probabilities of all paths, followed by a path aggregation step via a dynamic programming method. Since the objective function is differential, the whole network can be trained with back propagation (Figure 2.2).

The emergence of CTC technology not only solves data alignment problem in end-to-end speech recognition but also can directly output the target transcriptions without any human expertise to build various dictionaries. Since then, there were plenty of works applying CTC in ASR tasks.

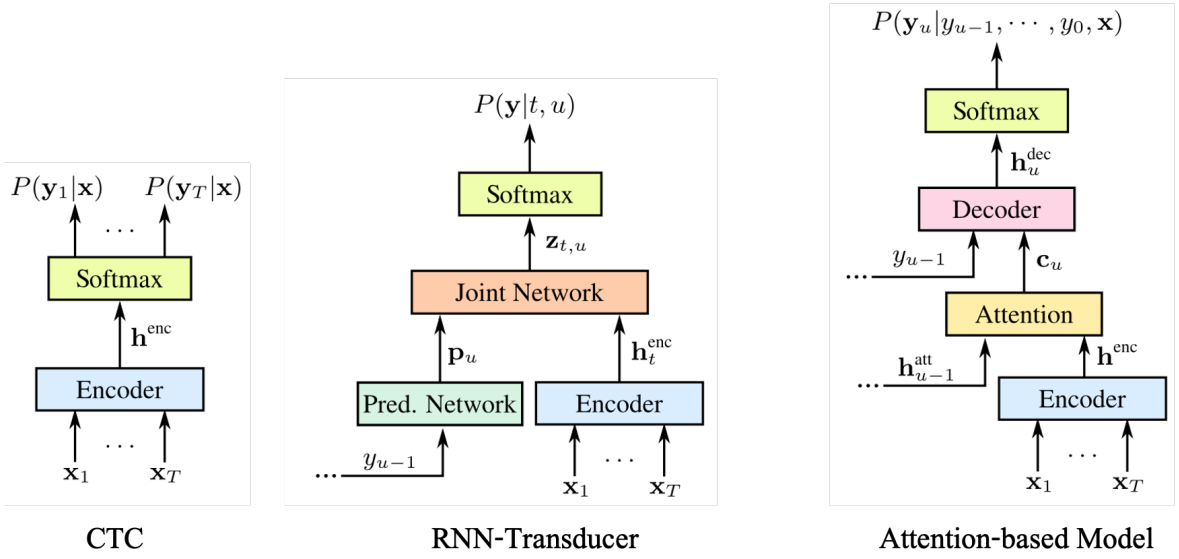

Figure 2.2: Architectures of end-to-end ASR models.

The early work [31] designed a 3-layer network including one feed forward layer and two Long Short-Term Memory (LSTM) layer, trained with CTC loss. The results showed that the recognition performance could be effectively improved by increasing the network's depth and the number of units for each layer, which has also been confirmed by [72]. The work [41] trained a deeper network with five layers of bidirectional LSTM having 500 hidden units for each layer, and achieved state-of-the-art performance at that time. Encouraged by this five-layer network structure, several other works [46, 47, 80, 105] were conducted for further improvements.

Afterwards, people made a great exploration of networks in terms of structure and depth. Song et al. [111] introduced a convolutional neural network (CNN) to extract more robust feature before RNN layers. [131] designed a pure CNN-based model trained by CTC loss, which conducted convolution operations in both time and frequency dimensions. In terms of network depth, deep speech 2 [2] expanded the network to nine layers, with a work [110] training a model with seven layers of bidirectional LSTM having 1000 hidden units each layer. However, the research in networks' structure and depth does not mean that the deeper model works in any situation. More recent works [4, 73, 5] chose to use shallower networks as their experimental datasets are not suitable for training deeper networks.

2.1.2 RNN-transducer

In CTC-based training process, 1) the model can not infer the interdependence within the different tokens of the output sequence as it assumes that elements of output sequence are

independent, 2) and it only works in the scenario where output sequences are shorter than inputs. RNN-transducer was proposed in [40], which solves the aforementioned defects of CTC. RNN-transducer model includes three components: 1) transcription network, playing a role of an acoustic model, which can map input audio to output sequences; 2) prediction network, which is an RNN network and can models the interdependence within output sequence (playing the role of language model); 3) joint network, connecting both components and mapping to the final output sequence. (Figure 2.2) Since one input speech generates a label sequence of arbitrary length, RNN-transducer is capable of mapping input sequence to output sequence with any length. Also, prediction network can learn interdependence within the output sequence, achieving joint training of language model and the acoustic model.

The later work [43] improved original RNN-transducer by changing the joint network from simple addition to a layer connection, and pre-training the transcription and prediction network. The experiment results demonstrated that training RNN-transducer from scratch is difficult. Another work [98] enhanced RNN-transducer by increasing the depth of transcription and prediction network and using a pre-training strategy.

RNN-transducer has its advantages over the CTC-based model. However, it also causes other defects: it may generate many unreasonable paths due to its flexible schemes. For instance, the first frame of audio may incorrectly produce all output sequences, leading to other positions all empty. To solve this problem, a new auxiliary framework was proposed in [106, 29], which developed a recurrent neural aligner that can restrict each input frame only producing one output.

2.1.3 Attention-based model

Attention-based models were first used on machine translation tasks in the paper [8]. In this work, given a input text the encoder generates a sequence of vectors rather than a single vector, and the decoder uses an attention mechanism at each time step of the output by assigning different weights to each vector in this sequence. The prediction of the next time step of output is determined by two elements: the historical output sequence and a weighted summation of the encoding result sequence. Attention-based end-to-end model can also be split into two sub-networks: encoder and decoder (with attention mechanism) (Figure 2.2). This technique has the potential to be applied to speech recognition due to the similar sequence-to-sequence process.

Latency problem. The attention mechanism is applied to the whole encoding sequence, so the training process moves on to decoder only when the encoding process is total completed, which highly increases the delay of model training. Additionally, the encoding

process may lead to lots of useless, redundant information to attention mechanism. The early works using attention for speech recognition [19, 20] ignored this issue. Bahdanau et al. [9] first noticed this problem and introduced time-dimension pooling during the encoding phase, which highly accelerates the model. Another work, LAS [17], adopted different solutions. The encoder consists of 4 bidirectional LSTM layers, where each layer uses the concatenation of two consecutive frames from the previous layer as its input. A pyramid structure is built in encoder and can reduce RNN loop steps.

Attention types. The attention mechanism may have three types: context-based, location-based, and hybrid. As for context-based, it only considers the input sequence and the previous hidden state to compute the weight at each time step [8]. The location-based method uses the previous weight as location information at each time step to compute current weight, without considering input feature sequence. The hybrid approach [19, 124] considers all three elements: input feature sequence, the previous hidden state and the previous weight.

2.2 New training schemes for ASR

With the rapid development of end-to-end ASR, many works began to explore new training situations using various approaches. Here, I introduce two popular techniques (knowledge distillation and self-supervised learning) in the area of deep learning and their applications in ASR.

2.2.1 Knowledge distillation

Knowledge distillation (KD) [51], also known as teacher-student training, is commonly used to narrow the gap of performance between a smaller and larger models. A typical KD training procedure consists of two stages. First, a deep neural network referred as the teacher is trained in line with standard supervised training rules based on numerous samples and their corresponding ground truth labels. Second, a compressed network, the student model, is trained on a selection of original ground truths and soft targets labelled by the teacher. These soft targets are the posterior probabilities obtained from the pre-trained teacher. There are two primary purposes using KD: one is to reduce the student size while matching its performance to that of the teacher, the other one focuses solely on increasing the performances of the student model without considering its complexity.

End-to-End ASR models are particularly well suited for KD as the whole pipeline is composed of neural networks only. One set of E2E ASR systems commonly rely either on the CTC loss [41], Sequence to Sequence models (Seq2Seq) [9], or a combination of the

two [62]. The KD works on ASR system also developed along with these types of training models.

KD for CTC-based model. The naive approach of KD typically minimises Kullback-Leibler (KL) divergence between posterior distributions of student and teacher models at each frame, assuming that both models have the same frame-wise alignments between the input speech and corresponding output sequences. As for the CTC-based models, however, the output symbols of teacher and student models may be different at the same time step. [66] proposed an improved frame-level KD method for CTC, which first selects a similar posterior distribution from teacher model at the preceding or the same time steps, and then train the student by minimising the KL divergence. Another work [27] used a method of dynamic time warping to align the output of student and teacher while proposing the other approach that splits the sequence of teacher output into small segments. Then, they extract N-best hypotheses and their posterior probabilities for each segment, which can be used to train student model. Theses two works aforementioned assume certain alignments exist between student and teacher models at the frame level. Other work [54] conducted sequence-level KD on a CTC-based model, which calculates the posterior distribution given the whole input audio and the teacher model, instead of computing the teacher posterior distribution at each frame. The N-best hypotheses from teacher model are extracted by beam search; then the student is trained in the fashion of cross entropy KD. Several similar works [114, 59, 115] explored the sequence-level KD on different types of models.

KD for attention-based model. Raden et al. [87] first applied KD on attention-based ASR model. Similar to [54], they extract the hypotheses from a pre-trained teacher model using beam search and train the student on the sequence-level cross-entropy criterion. More recent work [61] proposed a KD method on self-attention ASR models, which introduces an exponential weight to the sequence-level knowledge distillation loss function reflecting the word error rate of the teacher model output based on the ground-truth word sequences.

Multi-teacher distillation. Ensembles of teacher models capture complementary information by making different errors that can be further distilled to a student model. A critical aspect of multi-teacher distillation in the context of ASR is to find suitable strategies to maximise the distillation with respect to a specific set of teachers. [18] proposed to pre-assign weights to teachers based on their oracle error rate, to control the impact on the distilled information. Another work [33] uses the same pre-assign weights to all teachers or randomly selects the considered teachers . However, both strategies may give higher weighting, and thus higher importance, to teachers that perform worse than others in the teacher set when applied to specific sentences. The work I completed last

year (in Appendix A) proposed three new multi-teacher distillation strategies, integrating the error rate metric to the teacher selection during each mini-batch training.

2.2.2 Self-supervised learning

The success of deep learning techniques mainly relies on large-scale labelled training data. However, collecting large amounts of annotated samples is very costly and time-consuming. A possible way to alleviate these issues is self-supervised learning, where targets are calculated from the signal itself. Although self-supervision has been used in computer vision fields [28, 38, 84], applying self-supervised learning to speech is challenging since the speech signal is characterised by long, variable-length and high-dimensional sequences. This is, hence, very hard to infer without ground truth labels (e.g. phonemes). There are many works focusing on learning general and meaningful representations via self-supervised tasks to solve these issues. Here, we introduce four popular approaches.

Contrastive Predictive Coding (CPC). Van Den Oord et al. proposed a CTC approach to learn robust representations from unlabelled data. The main component of this framework is a multi-layer CNN that transforms the raw input data into general representation. The objective function is a contrastive loss [45], training by distinguishing a true future encoded representation from negatives given the past context as input. This method has been demonstrated on different domains, including phoneme classification in speech, image and text. Another work, Wav2vec [107], applied the learned representations from the CPC method to improve strong supervised ASR systems.

Autoregressive Predictive Coding (APC). Motivated by RNN-based language models (LMs) for text, the method of APC [21] built an RNN model to encode temporal information of past acoustic sequence but replace the Softmax layer in LMs with a regression layer. This way, the RNN output at each time step can produce future frames. The model is optimised with reconstruction loss (L1 loss). Compared to CPC-based works, this method focuses on predicting the spectrum of a future frame rather than a wave sample. Also, CPC focuses on the most discriminative information with respect to the target and negative frames, while APC encodes information more sufficient. On the other hand, in contrast to the fully convolutional architecture in CPC, APC is an RNN-based model over time, potentially leading to an efficiency problem.

BERT-based. The recently proposed vq-wav2vec [7] first applied BERT [25] algorithm on audio data to learn high-level representation from unlabelled data. The raw speech is encoded by discretizing to a K-way quantized embedding space, which is effective but computing resources-consuming. An improved work [76] modified the BERT algorithm.

This approach exploited a multi-layer transformer encoder and multi-head self-attention to extract representations, achieving bidirectional encoding. Unlike unidirectional methods, this framework can integrate both past and future contexts into computation metrics at the same time. To train the model in an unsupervised fashion, they proposed a masked acoustic modelling task, where the frames of input speech are masked randomly, and the model is trained to learn reconstructing and predicting the original frames. As for objective function, the L1 loss is used to minimise reconstruction error between prediction and ground-truth frames.

Multi-tasks. Multiple self-supervised tasks may bring different view or constraint on learning representation. Problem-agnostic speech encoder (PASE) [91] is designed to learn general, robust, and transferable features via training with various self-supervised tasks. The encoder maps the raw speech waveform into a representation after several CNN blocks, then feeding into four regressors including waveform, log power spectrum (LPS), mel-frequency cepstral coefficients (MFCC) and prosody, and three discriminators, including local info max (LIM) [99], global info max (GIM) and sequence predicting coding (SPC). The regressors are trained to minimise the mean squared error (MSE) between the target features and the network predictions, while discriminators are trained to minimise binary cross-entropy by feeding positive or negative samples. As an improved version of PASE, PASE+ [100] introduced an online speech distortion module to transform clean audio to contaminated variants via reverberation, additive noise, temporal/frequency masking, clipping, and overlapped speech. Also, the original CNN encoder from PASE has been enhanced by integrating with a quasi-recurrent neural network (QRNN) [99] that can learn long-term dependencies across the time steps. Additionally, some novel regressors are also introduced into this framework.

2.3 Federated learning

Edge devices and the modern internet of things (IoT), such as smartphones, wearable devices, and autonomous vehicles, have gained rapid development in the recent decades. These devices generate a wealth of data each day from their various sensors (e.g. images, text, etc.) in real-time. It is non-trivial to build a robust model to power applications by jointly learning the user data across a large pool of edge devices. However, users may not be willing to share their data due to their privacy requirements, and the connectivity of each device might be limited (e.g. bandwidth/battery power).

Due to the growing storage and computational capabilities of edge devices, it is possible to push model computation to the edge. Federated learning [81] has the potential to train statistical models directly on devices while storing data locally to alleviate the privacy

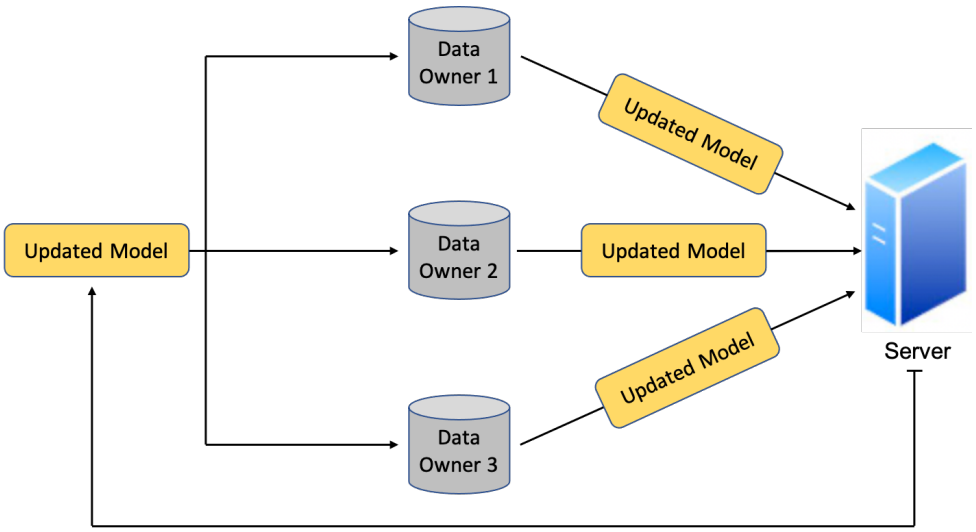

Figure 2.3: General federated learning architecture.

issue. This technique has been used in various real-world applications, such as next-word prediction [48], medical research [53, 24] and hot-word detection [70]. Here, we provide a brief survey of federated learning in terms of its core challenges, schemes for federated optimisation, and its application in the speech field.

2.3.1 Core challenges

Unlike centralised training, there are more constraints when training model on devices. The core challenges of federated learning are divided into four aspects.

- **Limited communication**. In a real federated learning setting, there are a massive number of devices in the federated networks, and communication between devices and server is slow due to the limited resources (e.g. bandwidth, energy, and power). It is crucial to develop communication-efficient approaches to send model updates during the training process, instead of sending the entire dataset over the network. Two key aspects could consider to reduce communication cost, 1) the number of communication rounds, and 2) size of the sending messages at each round.

- **Systems heterogeneity**. The storage, computational, and communication ability of each edge device in the federated system may differ due to their diverse hardware situations. In addition, these constraints may cause only a small partition of the devices being active at the same time, and the active devices may drop out at any time due to connectivity or energy problems. These system-based challenges require that the federated learning approaches must tolerate the heterogeneous hardware and be robust to low training participants and dropped devices.

- **Data heterogeneity**. Since the frequency of using certain applications on edge devices for specific users varies dramatically, each user's amount of training data generated from there is also different. This cause the distribution of user data highly heterogeneous, which violate the assumption of independent and identical distribution (i.i.d.) in centralised optimisation. Many works are focusing on this issue, such as the earlier method leveraging a single global model [81], and FL in multitask learning frameworks [109] or meta-learning [71].

- **Privacy guarantee**. As a major concern in federated learning, privacy could be protected by only sharing model updates, instead of the raw data, in the training process. However, the users' sensitive information may still be revealed during communication between edge devices and server. Some recent works [13, 82] enhance the privacy by secure multiparty computation (SMC) or differential privacy, while the performance or system efficiency drops. It is a challenging problem to balance these trade-offs.

2.3.2 Federated optimization

To achieve robust model training in federated settings, especially data in a non-i.i.d. fashion, a proper optimisation method is the key point. There exist many works using meta-learning and multitask learning to tackle data heterogeneity problem. [109] proposed a federated optimisation framework, MOCHA, which allows each device to learn model separately and aggregate using a shared representation via multitask learning. This method achieves personalisation for updating convex objectives but it is hard to scale to massive networks. Another work [22] designed a Bayesian network within a star topology, which can execute variational inference during learning. This approach can deal with non-convex functions but also has generalisation problem to large networks. Khodak et al. [60] used meta-learning framework to learn a within-task learning rate by treating devices as different tasks and gained improved accuracy over FedAvg method. The work [30] proposed semi-cyclic federated training framework which can dynamically select global or device-specific models.

Fairness is another crucial aspect when conducting federated optimisation across devices. The learned model may drift away from the original task and become biased toward devices containing a larger number of data. Some recent works are aiming to reduce the variance of the local training on the devices. Mohri et al. [85] proposed an agnostic federated learning method that uses a mini-max optimisation strategy to force the global model to a mixed distribution of specific devices. Another more general work, *q-FFL* [75], assigns weights to each device based on their training loss, from which the devices with higher loss are given higher weight to boost less variance in the final accuracy distribution.

Forcing the federated model convergence is much more challenging than centralised training. There exist several works [103, 121, 129] analysing the convergence behaviour of FedAvg and its related variants, but the results depend on the i.i.d. assumption which is not the real-world federated environment. Another recent work FedProx [74] conducted an analysis of FedAvg performance in heterogeneous settings. They modified the vanilla FedAvg method to allow partial training work to be executed across devices and then used a proximal term to integrate the partial work. This method can highly adapt the heterogeneous environments that some devices may be disabled due to system constraints, which boosts more well-behaved local updates and provides convergence guarantees for convex and non-convex functions. In addition to these provable approaches, there are several heuristic methods proposed for the problem of data heterogeneity via sharing local data on devices or using proxy data on server-side [53, 56]. However, these approaches can not provide privacy guarantee due to the data sharing process.

2.3.3 Federated learning in speech area

To my best knowledge, there are few works regarding federated learning in the speech area. Based on the type of tasks, the works involve keyword spotting, speaker verification and ASR.

FL for keyword spotting. Keyword spotting (KWS) has become an important research area for virtual assistants, which is used to start an interaction with a voice assistant, such as Apple's "Hey Siri" or Google's "OK Google". Compared to standard ASR, KWS is a relatively easy speech-based task. [70] first trained keyword spotting model in FL manner. The model they built is CNN-based, and training was performed with the Adam optimiser. As for federated optimization, the federated averaging (FedAvg) algorithm [81] was used in this paper. The other work [49] trained KWS model with a more real FL environment on non-IID data. An encoder-decoder architecture with multiple SVDF (single value decomposition filter) layers [88] was used for this task. Then, they investigated various federated optimization methods and implemented them to model training, including FedAvg [81], FedAdam [63] and FedYogi [127], while also replacing classical momentum with Nesterov accelerated gradients (NAG) [89] for each method.

FL for speaker verification. Speaker verification aims to determine whether the speaker is a specific person or someone else, typically used for securely accessing the devices via a "wake-up phrase". Granqvist et al. [39] first exploited federated learning with privacy to improve on-device speaker verification. They conducted federated training using FedAvg [81] method on a vocal classification model with speaker characteristics as ground truth stored privately on devices. This trained model can provide side information to the

downstream speaker verification system, and hence improving speaker recognition accuracy. To further protect user privacy, differential privacy is applied to add noise to the model updates during communication with the server.

FL for ASR. [26] is the only work on FL for ASR. They trained an ASR model in federated fashion on LibriSpeech dataset using FedAvg [81] method and achieved good performance on the test set. There are several improved aspects: 1) two separate optimiser are established for client and server; 2) running an additional training iteration over held-out data on the server after aggregation, in order to avoid the model drifted away from the original task; 3) using softmax values of losses of each client as weights for aggregation step, to avoid the case that some clients contain data that are represented by the model. However, the training setting is not pure FL environment as they used a pre-trained model for initialisation on the server side, and LibriSpeech dataset is not designed for FL where the data distribution is relatively uniform. There has been no work on classic FL for ASR, and also in the non-i.i.d. setting.

2.4 Data distillation and extraction

Deep learning models have achieved remarkable performance on various tasks, which requires a considerable amount of labelled data to learn their large number of parameters. In addition, the accuracy of deep learning models is often not saturated with increasing dataset size. However, labelling a dataset is an expensive and time-consuming task. On the other hand, their performance is susceptible to the structure and domain of the training data, and training on out-of-domain data can cause worse model accuracy. Hence, a data extraction process is non-trivial before or during model training, from which the small and valuable core set can be generated. This concentrated dataset is capable of accelerating training and gaining competitive performance over the whole dataset, which has the potential to be applied to federated training. Here, I briefly introduce the existing techniques in this direction.

Data selection. Data selection methods were used in domain adaptation problem for a wide range of tasks in some early works [86, 6, 57, 32]. These approaches used heuristic algorithms to measure domain similarity. Another application of data selection is denoising or dealing with undesirable data [118, 93], where they selected training data similar to data on valid set. Yet, these methods rely on features specific to the task. More recent work [123] proposed a more robust data usage optimisation method using reinforcement learning (RL). They use scorer network to minimise the model loss on the valid set while the main model being trained. Also, the gradient similarity between training examples

and the validation set is used as a reward signal to train the scorer network. This bi-level optimisation framework is generalisable to various tasks.

Instance weighting. Instead of pruning dataset, some works are focusing on enhancing training by weighing the examples in the dataset. [108] reweighs data based on a computed weight vector optimised by minimising the error rate on the validation set. However, in this work, only a single number is used to weigh the subgroups of augmented data and requiring a heuristic approach to update the weights. In contrast, another work [104] uses meta-learning to compute a locally optimised weight vector for each processing mini-batch.

Curriculum learning. Difficulty-based curriculum learning [95, 130, 44] can rank the presentation order of data based on human understanding of the hardness of examples, such that the model can learn faster and effectively from the easier examples. These approaches have limited generalisation ability since the difficulty measurement process is task-specific. The other direction of curriculum learning, namely self-paced learning [69, 65], determines the difficulty level of the data based on the loss of training model and with an assumption that the model should learn from easy examples.

Dataset distillation. The work [122] proposed distilling the dataset: keeping the model fixed and synthesising a small number of data points from a huge dataset, which approximates the original data distribution and achieves close to the original performance. Concretely, they extract the model weights as a differentiable function of the target generated training data, followed by optimising the feature values (e.g. pixels in an image) of the distilled data. As the results show, this method can compress 60, 000 training images of MNIST dataset into 10 synthetic distilled images (one each class) and achieve the almost same accuracy as original training with only a few steps of gradient descent.

2.5 Neural Architecture Search (NAS)

Neural architecture search (NAS) aims to automatically design neural network architectures and optimise hyperparameter, by sampling the search space and evaluating candidate architectures based on certain metrics to gain improvements of accuracy or co-optimisation with latency and memory consumption in recent hardware-aware NAS. In federated settings, NAS could be used to search decent data and models for each edge device to achieve personalisation. Here, I briefly introduce the main search algorithms with their applications.

2.5.1 Search algorithms

Reinforcement learning (RL) can be used to search over the search space, where an agent modifies the candidate architectures using a set of actions via Q-learning [10] or proximal policy optimisation [135]. Alternatively, Bayesian optimisation is another popular method for hyperparameter search. Zela et al. [128] applied Bayesian optimisation to NAS achieving jointly optimise for a network's architecture and hyperparameters by searching a categorical distribution. Evolutionary algorithms are also be used to design neural networks, early raised by Miller et al. [83] in 1989, by choosing parent neural architectures and abandon the worse ones in a crossover step with a certain mutation probability [101, 102]. Note that in this process, the crossover and mutation steps require meticulous design.

The aforementioned algorithms, however, have relatively high overhead of sampling. To alleviate this issue, several techniques were developed, such as weight sharing [92] which initialise the weights of current candidate model by the ones from previous candidates, or hyper-networks [14] that generates the weights of the main model conditioned on that model's architecture and search process can be completed in a single training run. Similarly, Liu et al. [77] proposed an auxiliary model to predict the accuracy of a candidate network in light of its architecture, albeit introducing another model to predict performance may lead to potential problems (e.g. lack of training data). The recently proposed work, a differentiable NAS — DARTS [78], significantly reduces the overhead of sampling as all candidate models can be trained at once.

Memory consumption is another concern for NAS algorithms. ProxylessNAS [15] reduces the memory cost by relaxing the constraint of training on the proxy tasks but causes a longer run-time penalty. Another work [119] accelerates ProxylessNAS while retaining the memory consumption.

2.5.2 NAS applications

In addition to some popular tasks in computer vision [102, 116], and natural language processing [64], NAS can also apply to other training situations and problems, such as producing compact models [112, 79]. Other work has developed a NAS approach to train models suited for deployment across a variety of devices within training a single model [16]. NAS can also be used to find graph neural network (GNN) architectures using RL-based methods [36] or differential NAS scheme [132]. In speech recognition, NAS has been applied to search model architectures [11] or hyperparameters [52] for further improvement of performance.

There are several works introducing NAS to FL. Zhu et al. [133] proposed to optimise network architectures in FL by a multi-objective evolutionary algorithm, albeit

all clients participating in training, which significantly increasing both computation and communication costs. Client sampling can be used to alleviate this issue [125]. To achieve simultaneous optimization of weight training and architecture search, the gradient-based [50] and EA-based methods [134] were proposed in the federated environment. However, jointly optimising the global on clients causes much heavier computation and memory consumption, impeding the deployment to edge devices. Another work [134] proposed a light-weighted real-time evolutionary NAS framework (RT-FedEvoNAS) to reduce the memory usage of local devices.

Chapter 3

Completed Work

In my first year at Oxford, I have mainly carried out three projects that lead to conference and journal papers, respectively. All of them are included in the Appendix A section. The first one [34] aims to improve the performance of acoustic models using multi-teacher distillation and proposed three error rate based strategies by linking the error rate metric to the teacher selection. The second project presents the first study on end-to-end ASR model for realistic FL scenarios including *cross-silo* and *cross-device* FL settings. A new aggregation strategy based on WER was proposed to further integrate the specificity of ASR to FL. Both above papers are currently to be submitted to a proper conference. The third paper [67] developed an automatic extraction pipeline of virtual on-body Accelerometry from the video for human activity recognition, which has been published at the "ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT) 2020".

3.1 Multi-teacher distillation of acoustic models

Modern deep learning-based ASR systems have been shown to benefit from multi-teacher distillation strategies strongly [18, 33], as different E2E ASR systems often lead to different transcriptions given a fixed audio sample strongly increasing the diversity of the teachable distributions that could be distilled to the student.

"Distilling Knowledge from Ensembles of Acoustic Models for Joint CTC-Attention End-to-End Speech Recognition" [34] (Appendix A) proposed novel multi-teacher distillation methods for joint CTC-attention end-to-end ASR systems, which considers error rate as an indicator to assess the teacher quality. This way, it directly distillates and optimises the student toward the relevant metric for speech recognition (Figure 3.1). The distillation strategies are depicted as follows:

- *Weighted* strategy enables the student to directly assign weights to all the teachers in the course of training based on the average observed ER on the training processed

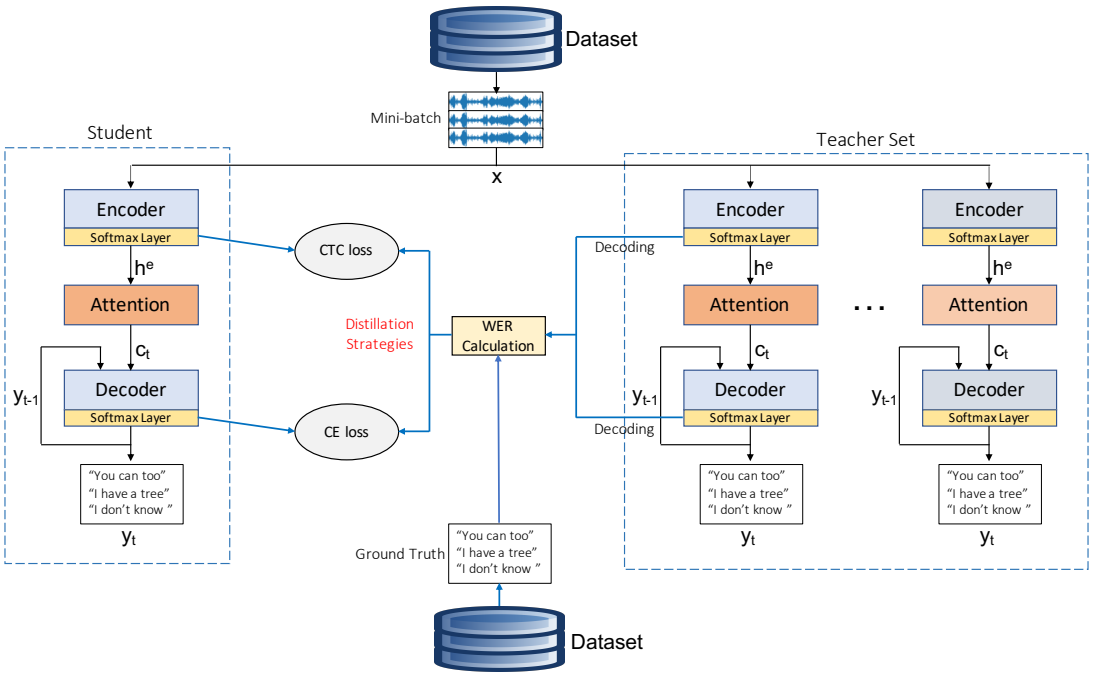

Figure 3.1: Illustration of the error rate multi-teacher distillation strategies connected to a Joint CTC-Attention E2E ASR system.

mini-batch. The impact of the teachers is, therefore, dynamically changed between mini-batches.

- *Top-1* strategy offers the student an option to choose a single teacher with respect to the best ER observed at the sentence level on the processed mini-batch.

- *Top-k* strategy allows the student to learn from a set of best teachers that perform equally in terms of error rate on the processed mini-batch.

- *Word-level distillation*. This strategy enables the student to select teacher models at each word position of the processed sentence. In the case that certain words are never recognised by all teacher models, supervised training (annotation from the dataset as training labels) is conducted in stead of distilling from pre-trained teacher models. This extension is yet integrated into this paper.

The strategies are evaluated on the TIMIT [37] phoneme recognition task and achieve a Phoneme Error Rate (PER) of 13.11% representing a state-of-the-art score for end-to-end ASR systems. The proposed distillation methods could be exploited in federated learning settings, which are elaborated in the following chapter.

3.2 Acoustic model training in federated conditions

Federated learning (FL) offers new opportunities to advance ASR quality given the unprecedented amount of user data directly available on-device. With FL, the training process leverages large and diverse amounts of data collected locally by user devices, while also offering the requisite privacy protection. An existing paper [26] presented a first study on FL for ASR model training, but their experiments were conducted on LibriSpeech [90] with clean and homogeneous audio data which is not realistic FL setting.

"End-to-End Speech Recognition from Federated Acoustic Models" [35] (Appendix A) investigates FL in a more realistic setting with the French Common Voice (CV) dataset [3]. It provides a large set of speakers that used their own devices to record a given set of sentences, naturally fitting to federated learning with various speakers, acoustic conditions, microphones and accents. We conduct an empirical study of three different weighting strategies during model aggregation to approach the difficulty of non-IID FL. In particular, this work introduces a word error rate (WER) based strategy to further adapt ASR training to federated learning. The methods were evaluated with both a *cross-silo* and a *cross-device* (i.e. large number of clients with few non-IID data) FL setups.

Table 3.1: Speech recognition results (WER %) observed on the test set of French Common Voice dataset for different scenarios and weighting strategies.

Training Scenario		WER (%)
Centralised	training on all data (lower bound)	20.18
	training on 1st half data	25.26
	online training on 2nd half data	20.94
10-client FL	count-based	21.26
	loss-based	21.10
	WER-based	20.99
2k-client FL	count-based	22.83
	loss-based	22.67
	WER-based	22.42

Table 3.1 shows the main results presented in this paper. Compared to different weighting strategies, WER-based and loss-based methods obtain a better performance, which indicates that weakening the effects of low-quality clients can assist the aggregation process in federated training with heterogeneous data distribution.

3.3 Virtual IMU extraction pipeline

Labelled data in human activity recognition is scarce and hard to come by, as sensor data collection is expensive, and the annotation is time-consuming and sometimes even impossible for privacy or other practical reasons.

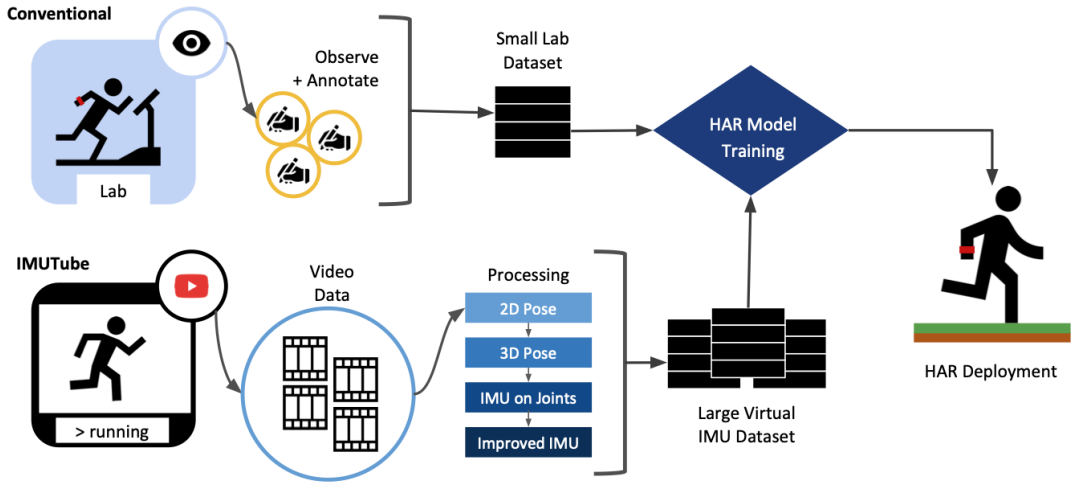

Figure 3.2: The proposed IMUTube system replaces the conventional data recording and annotation protocol (upper left) for developing sensor-based human activity recognition (HAR) systems (upper right). We utilise existing, large-scale video repositories from which we generate virtual IMU data that is then used for training the HAR system (bottom part).

"IMUTube: Automatic Extraction of Virtual on-body Accelerometry from Video for Human Activity Recognition" [67] (Appendix A) develops an automated framework that exploits existing video data from large-scale repositories, such as YouTube, and automatically generate data for virtual, body-worn movement sensors (IMUs) that will then be used for deriving sensor-based human activity recognition systems.

The extraction steps are as follows. First, we apply standard pose tracking and 3D scene understanding techniques to estimate full 3D human motion from a video segment that captures a target activity. Second, visual tracking information is translated into virtual motion sensors (IMU) placed on dedicated body positions. Then, we adapt the virtual IMU data towards the target domain through distribution matching. Finally, the activity recognisers are derived from the generated virtual sensor data, potentially enriched with small amounts of real sensor data.

The pipeline (Figure 3.2) integrates several off-the-shelf computer vision and graphics techniques so that IMUTube is fully automated and thus directly applicable to a wide variety of existing videos. We demonstrate the virtually-generated IMU data can improve the performance of a variety of models on known HAR datasets, and this should lead to on-body, sensor-based HAR becoming yet another success story in large-dataset breakthroughs in recognition.

Chapter 4

Proposed Research

As the background chapter describes, training effective ASR models in federated environments is still an unsolved problem. The core objective of this PhD is to address this situation, and radically improve our ability to federate speech models. This this end, I aim to develop novel FL training approaches, and extending them to diverse federated acoustic applications, with the end-goal of enabling these algorithms to be realised in real edge devices. The following three research directions are tentative, with the exact scope and purpose of each project expected be refined during their execution.

4.1 Federated multi-teacher distillation

Due to non-i.i.d. and unbalanced data distributions on edge devices, some clients may contain data not represented by the model. This causes the training model to drift away from the original task, and hence leading to a performance reduction or even preventing the training process altogether. Filtering or alleviating the negative effect of these clients is very crucial for high-quality federated learning. I believe, a multi-teacher distillation based approach has untapped potential to solve this problem, by considering clients as teacher models. Examples of these are described in my completed work [34] (Appendix A). The distillation strategies in that work can link error rate (ER) metric to the teacher selection. In this way, the effects of each client are measured by their ERs, and only highly-performed local models are selected to train the global model.

Under conventional distillation, the student model is trained on a selection of ground truth and soft targets (posterior probabilities) labelled by the teacher on the same training dataset. This is, however, impractical in federated settings as data sharing is not allowed due to user privacy guarantee. Additionally, transmitting posterior distributions from clients to the server is also restricted due to the resource-constraint environment on edge devices (e.g. bandwidth).

To achieve distillation in FL settings while reducing communication, the knowledge

distillation (KD) training could be established on the server side via a proxy dataset, after receiving parameters of models from local training on devices. Concretely, soft targets are collected from the inference outputs on the proxy dataset using pre-trained local models. A student model with the same architecture is then trained in line with distillation rules proposed in [34]. The distilled model weights are transmitted back to the clients while triggering the next round of local training. In this way, the out-of-scope clients are filtered from the distillation procedure, ensuring a robust federated training within non-i.i.d. environments.

While the out-of-shelf distillation strategies proposed in [34] can be directly integrated into FL settings, the following aspects are non-trivial to be considered for further improvements.

- **Model architectures exploration**. Different architectures may have their advantages over the challenges in FL settings. Attention-based sequence-to-sequence models within distillation training framework have achieved success in our completed work, yet other model architectures (e.g. transformer, RNN-Transducer) have not been verified. Exploring diverse model architectures within FL environments would establish a good foundation for the downstream federated optimisation.

- **Unified error rate-based distillation scheme**. Typical ASR models are trained by CTC or cross-entropy loss which corresponds to improving the log-likelihood of the data. However, system performance is usually measured in terms of error rate, not log-likelihood. The process of teacher selection is directed toward the metric relevant for speech recognition (i.e. error rate) in my completed work. Intuitively, ASR system can further benefit from integrating minimum word error rate training [96], an error rate-based optimisation criteria, into distillation framework to build a more unified distillation training scheme.

- **Recursive training mechanism**. During the distillation process on the server side described above, all local models are simultaneously integrated into calculation metrics to generate soft targets for student model. This may be non-applicable in real-word federated environments as the number of clients is typically in hundreds or thousands level, which will cause an unaccepted memory consumption. To alleviate this issue, a clients sampling operation could be conducted before model inference, from which the student model can be trained with only partial clients at once. The distilled model is then recursively re-trained with a new set of clients as teachers from another sampling step. In case the student model is highly converged on the first round of distillation, the last layer is re-initialised randomly at the beginning of each training round.

4.2 Data enhancement for efficient federated learning

A crucial concern in federated learning is the resource-constrained environments (e.g. CPU, memory and network connectivity) on edge devices. This leads to limited capabilities of computational, storage and communication. Training deep networks on these devices usually costs much longer time to obtain high performance, especially for ASR models — typically requires larger model size and more training data [98, 2, 110].

To achieve efficient training in FL settings, a potential effort direction is to reduce the scale of the training dataset. Some existing works, such as core-set construction [123] and dataset distillation [122], aim to summarise the entire dataset and hence decrease the amount of training data while remaining the same performance. These techniques could be integrated on each client before local training. This way, the reduced data size could match the computational capability of edge devices, leading to an adequate training time.

Indeed, this method could integrate with the federated distillation framework in Section 4.1 in order to accelerate the training process — extracting a compressed set from the proxy dataset on the server side, then conducting distillation training. Additionally, if the size of the compressed dataset is smaller than that of parameters of the model, we could transmit this dataset to the clients instead of model weights. This way, the KD process is migrated to edge device, reducing the overhead of communication.

4.3 Unified FL system via multi-objective search

In a real-word federated environment, the training situations are much more challenging. In addition to the aforementioned difficulties (e.g. privacy guarantee, non-i.i.d. and unbalanced distribution, limited communication), there is a myriad of practical issues: 1) the local dataset changes with data adding and deleting by users; 2) client availability affects the local data distribution; 3) clients may not respond or send updates. If considering user's preferences, the case will be more complex (e.g. users have different requirements with respect to accuracy, latency, etc.). A more robust and unified FL training system, by integrating all situations above into optimisation metrics, is required.

Neural architecture search (NAS) [135], aiming to automate the process of designing and tweaking neural network architectures, is a potential approach to tackle these issues. However, the existing NAS methods in FL [133, 125, 50] only consider the architecture optimisation based on model accuracy, ignoring communication costs as the evaluation metric. Also, data and users' preference are not integrated into search space. In this regard, a multi-objective search system can be built, which not only customises both model and data for each client, but also provides trade-off solutions between accuracy and latency based on the user's preferences. This is a large-scale project but worthy of

investigation, if time allowing.

4.4 Self-supervised federated learning

The success of deep learning techniques mainly relies on large-scale labelled training data. Federated learning provides an opportunity to exploit the unprecedented amount of user data for more robust mode training. However, collecting large amounts of annotated samples is very expensive, time-consuming, and even error-prone. This is more deteriorating in federated settings as the data stores on inaccessible users' device, and the annotation can only be collected by inferring from user interaction. Thus, it is intuitive to promote federated learning towards self-supervision on unlabelled data. With the development of self-supervision at the avenue of end-to-end speech recognition [91, 76, 100, 100], these techniques could be integrated into the federated training system.

Instead of training local models using ground truth labels in standard FL, self-supervision is conducted with unlabelled data on devices. Then, the weights of all self-supervised models are aggregated on the server side as standard federated training. This global model obtained after certain training rounds could be used for any downstream tasks (e.g. speech recognition, speaker verification, etc.).

In addition to taking advantage of unlabelled user data on clients, accessible unlabelled data from the server side also has the potential to enhance federated training. Here, I introduce two explorations in line with this assumption.

- **Self-supervised feature extraction**. First, a robust self-supervised model is trained using numerous unlabelled data on the server side, followed by transmitting the model to clients. Then, this pre-trained model can be used to extract the enhanced features (representations) via inference process on user data. These deep features will then be used as input for standard federated training, instead of using the surface features (e.g. Mel filter banks, MFCCs) that can poorly reveal the abundant information within speech.

- **Improve FL via semi-supervision**. To further enhance the trained model under standard federated training rules, data distillation [97] could be conducted on the server side. Concretely, we first generate pseudo labels on unlabelled data using the pre-trained model and then re-train the model using these extra generated annotations to gain further improvement.

Chapter 5

Timeline

Term	Planned work
MT 2020	Re-visit existing methods in FL for speech and implement KD framework into FL. Replicate popular self-supervision methods in speech area.
LT 2021	Explore new approaches based on proposal for further improvement of KD-based FL. Achieve self-supervised FL framework. Summarise the completed works and write up for conference submissions if promising results. Clean up related code and open source.
ET 2021	Take steps towards data side, implementing data extraction approaches and integrating into FL. Sequentially, integrating this with KD-based FL.
Summer 2021	Internship
MT 2021	Begin literature review on architecture design and data selection. Summarise completed work and build a unified FL training system via NAS.
LT 2022	Run our methods on real edge devices. Write up summary of all completed projects. Begin thesis write-up
ET 2022	Thesis outline, complete outstanding experiments and thesis write-up.
Summer 2022	Thesis write-up

Table 5.1: Timeline

Bibliography

[1] ai.google. Under the hood of the pixel 2: How ai is supercharging hardware. `https://support.google.com/messages/answer/9327902`, 2019.

[2] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182, 2016.

[3] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*, 2019.

[4] Kartik Audhkhasi, Bhuvana Ramabhadran, George Saon, Michael Picheny, and David Nahamoo. Direct acoustics-to-word models for english conversational speech recognition. *arXiv preprint arXiv:1703.07754*, 2017.

[5] Kartik Audhkhasi, Brian Kingsbury, Bhuvana Ramabhadran, George Saon, and Michael Picheny. Building competitive direct acoustics-to-word models for english conversational speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4759–4763. IEEE, 2018.

[6] Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, 2011.

[7] Alexei Baevski, Steffen Schneider, and Michael Auli. vq-wav2vec: Self-supervised learning of discrete speech representations. *arXiv preprint arXiv:1910.05453*, 2019.

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[9] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. End-to-end attention-based large vocabulary speech recognition. In *2016*

IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 4945–4949. IEEE, 2016.

[10] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing neural network architectures using reinforcement learning. *arXiv preprint arXiv:1611.02167*, 2016.

[11] Ahmed Baruwa, Mojeed Abisiga, Ibrahim Gbadegesin, and Afeez Fakunle. Leveraging end-to-end speech recognition with neural architecture search. *arXiv preprint arXiv:1912.05946*, 2019.

[12] Yoshua Bengio. Markovian models for sequential data. *Neural computing surveys*, 2 (199):129–162, 1999.

[13] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[14] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Smash: one-shot model architecture search through hypernetworks. *arXiv preprint arXiv:1708.05344*, 2017.

[15] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. *arXiv preprint arXiv:1812.00332*, 2018.

[16] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.

[17] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964. IEEE, 2016.

[18] Yevgen Chebotar and Austin Waters. Distilling knowledge from ensembles of neural networks for speech recognition. In *Interspeech*, pages 3439–3443, 2016.

[19] Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. End-to-end continuous speech recognition using attention-based recurrent nn: First results. *arXiv preprint arXiv:1412.1602*, 2014.

[20] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. *Advances in neural information processing systems*, 28:577–585, 2015.

[21] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass. An unsupervised autoregressive model for speech representation learning. *arXiv preprint arXiv:1904.03240*, 2019.

[22] Luca Corinzia and Joachim M Buhmann. Variational federated multi-task learning. *arXiv preprint arXiv:1906.06268*, 2019.

[23] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2011.

[24] Walter de Brouwer. The federated future is ready for shipping, 2019.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[26] Dimitrios Dimitriadis, Kenichi Kumatani, Robert Gmyr, Yashesh Gaur, and Sefik Emre Eskimez. A federated approach in training acoustic models. In *Proc. Interspeech*, 2020.

[27] Haisong Ding, Kai Chen, and Qiang Huo. Compression of ctc-trained acoustic models by dynamic frame-wise distillation or segment-wise n-best hypotheses imitation. In *INTERSPEECH*, pages 3218–3222, 2019.

[28] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2051–2060, 2017.

[29] Linhao Dong, Shiyu Zhou, Wei Chen, and Bo Xu. Extending recurrent neural aligner for streaming end-to-end speech recognition in mandarin. *arXiv preprint arXiv:1806.06342*, 2018.

[30] Hubert Eichner, Tomer Koren, H Brendan McMahan, Nathan Srebro, and Kunal Talwar. Semi-cyclic stochastic gradient descent. *arXiv preprint arXiv:1904.10120*, 2019.

[31] Florian Eyben, Martin Wöllmer, Björn Schuller, and Alex Graves. From speech to letters-using a novel neural network architecture for grapheme based asr. In *2009*

IEEE Workshop on Automatic Speech Recognition & Understanding, pages 376–380. IEEE, 2009.

[32] George Foster, Cyril Goutte, and Roland Kuhn. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 451–459, 2010.

[33] Takashi Fukuda, Masayuki Suzuki, Gakuto Kurata, Samuel Thomas, Jia Cui, and Bhuvana Ramabhadran. Efficient knowledge distillation from an ensemble of teachers. In *Interspeech*, pages 3697–3701, 2017.

[34] Yan Gao, Titouan Parcollet, and Nicholas Lane. Distilling knowledge from ensembles of acoustic models for joint ctc-attention end-to-end speech recognition. *arXiv preprint arXiv:2005.09310*, 2020.

[35] Yan Gao, Titouan Parcollet, Javier Fernandez-Marques, Pedro PB de Gusmao, Daniel J Beutel, and Nicholas D Lane. End-to-end speech recognition from federated acoustic models. *arXiv preprint arXiv:2104.14297*, 2021.

[36] Yang Gao, Hong Yang, Peng Zhang, Chuan Zhou, and Yue Hu. Graphnas: Graph neural architecture search with reinforcement learning. *arXiv preprint arXiv:1904.09981*, 2019.

[37] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *STIN*, 93:27403, 1993.

[38] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[39] Filip Granqvist, Matt Seigel, Rogier van Dalen, Áine Cahill, Stephen Shum, and Matthias Paulik. Improving on-device speaker verification using federated learning with privacy. *arXiv preprint arXiv:2008.02651*, 2020.

[40] Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.

[41] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pages 1764–1772, 2014.

[42] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent

neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[43] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[44] Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. *arXiv preprint arXiv:1704.03003*, 2017.

[45] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010.

[46] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[47] Awni Y Hannun, Andrew L Maas, Daniel Jurafsky, and Andrew Y Ng. First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns. *arXiv preprint arXiv:1408.2873*, 2014.

[48] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[49] Andrew Hard, Kurt Partridge, Cameron Nguyen, Niranjan Subrahmanya, Aishanee Shah, Pai Zhu, Ignacio Lopez Moreno, and Rajiv Mathews. Training keyword spotting models on non-iid data with federated learning. *arXiv preprint arXiv:2005.10406*, 2020.

[50] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Fednas: Federated deep learning via neural architecture search. *arXiv preprint arXiv:2004.08546*, 2020.

[51] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[52] Shoukang Hu, Xurong Xie, Shansong Liu, Mengzhe Geng, Xunying Liu, and Helen Meng. Neural architecture search for speech recognition. *arXiv preprint arXiv:2007.08818*, 2020.

[53] Li Huang, Yifeng Yin, Zeng Fu, Shifa Zhang, Hao Deng, and Dianbo Liu. Loadaboost: Loss-based adaboost federated machine learning on medical data. *arXiv preprint arXiv:1811.12629*, 2018.

[54] Mingkun Huang, Yongbin You, Zhehuai Chen, Yanmin Qian, and Kai Yu. Knowledge distillation for sequence model. In *Interspeech*, pages 3703–3707, 2018.

[55] Fumitada Itakura. A statistical method for estimation of speech spectral density and formant frequencies. *Electronics and Communications in Japan, A*, 53(1):36–43, 1970.

[56] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.

[57] Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 264–271, 2007.

[58] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[59] Naoyuki Kanda, Yusuke Fujita, and Kenji Nagamatsu. Sequence distillation for purely sequence trained acoustic models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2018.

[60] Mikhail Khodak, Maria-Florina F Balcan, and Ameet S Talwalkar. Adaptive gradient-based meta-learning methods. In *Advances in Neural Information Processing Systems*, pages 5917–5928, 2019.

[61] Ho-Gyeong Kim, Hwidong Na, Hoshik Lee, Jihyun Lee, Tae Gyoon Kang, Min-Joong Lee, and Young Sang Choi. Knowledge distillation using output errors for self-attention end-to-end models. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6181–6185. IEEE, 2019.

[62] Suyoun Kim, Takaaki Hori, and Shinji Watanabe. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4835–4839. IEEE, 2017.

[63] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[64] Nikita Klyuchnikov, Ilya Trofimov, Ekaterina Artemova, Mikhail Salnikov, Maxim Fedorov, and Evgeny Burnaev. Nas-bench-nlp: Neural architecture search benchmark for natural language processing. *arXiv preprint arXiv:2006.07116*, 2020.

[65] M Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23:1189–1197, 2010.

[66] Gakuto Kurata and Kartik Audhkhasi. Improved knowledge distillation from bi-directional to uni-directional lstm ctc for end-to-end speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 411–417. IEEE, 2018.

[67] Hyeokhyen Kwon, Catherine Tong, Harish Haresamudram, Yan Gao, Gregory D Abowd, Nicholas D Lane, and Thomas Ploetz. Imutube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition. *arXiv preprint arXiv:2006.05675*, 2020.

[68] Kai-Fu Lee. On large-vocabulary speaker-independent continuous speech recognition. *Speech communication*, 7(4):375–379, 1988.

[69] Yong Jae Lee and Kristen Grauman. Learning the easy things first: Self-paced visual category discovery. In *CVPR 2011*, pages 1721–1728. IEEE, 2011.

[70] David Leroy, Alice Coucke, Thibaut Lavril, Thibault Gisselbrecht, and Joseph Dureau. Federated learning for keyword spotting. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6341–6345. IEEE, 2019.

[71] Jeffrey Li, Mikhail Khodak, Sebastian Caldas, and Ameet Talwalkar. Differentially private meta-learning. *arXiv preprint arXiv:1909.05830*, 2019.

[72] Jie Li, Heng Zhang, Xinyuan Cai, and Bo Xu. Towards end-to-end speech recognition for chinese mandarin using long short-term memory recurrent neural networks. In *Sixteenth annual conference of the international speech communication association*, 2015.

[73] Jinyu Li, Guoli Ye, Rui Zhao, Jasha Droppo, and Yifan Gong. Acoustic-to-word model without oov. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 111–117. IEEE, 2017.

[74] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

[75] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

[76] Andy T Liu, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6419–6423. IEEE, 2020.

[77] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018.

[78] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[79] Yu Liu, Xuhui Jia, Mingxing Tan, Raviteja Vemulapalli, Yukun Zhu, Bradley Green, and Xiaogang Wang. Search to distill: Pearls are everywhere but not the eyes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7539–7548, 2020.

[80] Andrew Maas, Ziang Xie, Dan Jurafsky, and Andrew Y Ng. Lexicon-free conversational speech recognition with neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 345–354, 2015.

[81] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

[82] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

[83] Geoffrey F Miller, Peter M Todd, and Shailesh U Hegde. Designing neural networks using genetic algorithms. In *ICGA*, volume 89, pages 379–384, 1989.

[84] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016.

[85] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. *arXiv preprint arXiv:1902.00146*, 2019.

[86] Robert C Moore and Will Lewis. Intelligent selection of language model training data. 2010.

[87] Raden Mu'az Mun'im, Nakamasa Inoue, and Koichi Shinoda. Sequence-level knowledge distillation for model compression of attention-based sequence-to-sequence speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6151–6155. IEEE, 2019.

[88] Preetum Nakkiran, Raziel Alvarez, Rohit Prabhavalkar, and Carolina Parada. Compressing deep neural networks using a rank-constrained topology. 2015.

[89] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

[90] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[91] Santiago Pascual, Mirco Ravanelli, Joan Serrà, Antonio Bonafonte, and Yoshua Bengio. Learning problem-agnostic speech representations from multiple self-supervised tasks. *arXiv preprint arXiv:1904.03416*, 2019.

[92] Hieu Pham, Melody Y Guan, Barret Zoph, Quoc V Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. *arXiv preprint arXiv:1802.03268*, 2018.

[93] Minh Quang Pham, Josep M Crego, Jean Senellart, and François Yvon. Fixing translation divergences in parallel corpora for neural mt. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2967–2973, 2018.

[94] Sundar Pichai. Privacy should not be a luxury good. *The New York Times, May*, 7, 2019.

[95] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv:1903.09848*, 2019.

[96] Rohit Prabhavalkar, Tara N Sainath, Yonghui Wu, Patrick Nguyen, Zhifeng Chen, Chung-Cheng Chiu, and Anjuli Kannan. Minimum word error rate training for attention-based sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4839–4843. IEEE, 2018.

[97] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4119–4128, 2018.

[98] Kanishka Rao, Haşim Sak, and Rohit Prabhavalkar. Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 193–199. IEEE, 2017.

[99] Mirco Ravanelli and Yoshua Bengio. Learning speaker representations with mutual information. *arXiv preprint arXiv:1812.00271*, 2018.

[100] Mirco Ravanelli, Jianyuan Zhong, Santiago Pascual, Pawel Swietojanski, Joao Monteiro, Jan Trmal, and Yoshua Bengio. Multi-task self-supervised learning for robust speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6989–6993. IEEE, 2020.

[101] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc Le, and Alex Kurakin. Large-scale evolution of image classifiers. *arXiv preprint arXiv:1703.01041*, 2017.

[102] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Aging evolution for image classifier architecture search. In *AAAI Conference on Artificial Intelligence*, 2019.

[103] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International Conference on Artificial Intelligence and Statistics*, pages 2021–2031. PMLR, 2020.

[104] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*, 2018.

[105] Haşim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays. Fast and accurate recurrent neural network acoustic models for speech recognition. *arXiv preprint arXiv:1507.06947*, 2015.

[106] Hasim Sak, Matt Shannon, Kanishka Rao, and Françoise Beaufays. Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping. In *Interspeech*, volume 8, pages 1298–1302, 2017.

[107] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019.

[108] Sunit Sivasankaran, Emmanuel Vincent, and Irina Illina. Discriminative importance weighting of augmented training data for acoustic model training. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4885–4889. IEEE, 2017.

[109] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. *Advances in neural information processing systems*, 30:4424–4434, 2017.

[110] Hagen Soltau, Hank Liao, and Hasim Sak. Neural speech recognizer: Acoustic-to-word lstm model for large vocabulary speech recognition. *arXiv preprint arXiv:1610.09975*, 2016.

[111] William Song and Jim Cai. End-to-end deep neural network for automatic speech recognition. *Standford CS224D Reports*, 2015.

[112] Dimitrios Stamoulis, Ruizhou Ding, Di Wang, Dimitrios Lymberopoulos, Bodhi Priyantha, Jie Liu, and Diana Marculescu. Single-path nas: Designing hardware-efficient convnets in less than 4 hours. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 481–497. Springer, 2019.

[113] support.google. Your chats stay private while messages improves suggestions. `https://www.intel.ai/federated-learning-for-medical-imaging/`, 2019.

[114] Ryoichi Takashima, Sheng Li, and Hisashi Kawai. An investigation of a knowledge distillation method for ctc acoustic models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5809–5813. IEEE, 2018.

[115] Ryoichi Takashima, Li Sheng, and Hisashi Kawai. Investigation of sequence-level knowledge distillation methods for ctc acoustic models. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6156–6160. IEEE, 2019.

[116] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

[117] Taras K Vintsyuk. Speech discrimination by dynamic programming. *Cybernetics*, 4 (1):52–57, 1968.

[118] Yogarshi Vyas, Xing Niu, and Marine Carpuat. Identifying semantic divergences in parallel text without annotations. *arXiv preprint arXiv:1803.11112*, 2018.

[119] Alvin Wan, Xiaoliang Dai, Peizhao Zhang, Zijian He, Yuandong Tian, Saining Xie, Bichen Wu, Matthew Yu, Tao Xu, Kan Chen, et al. Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12965–12974, 2020.

[120] Dong Wang, Xiaodong Wang, and Shaohe Lv. An overview of end-to-end automatic speech recognition. *Symmetry*, 11(8):1018, 2019.

[121] Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. *arXiv preprint arXiv:1808.07576*, 2018.

[122] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

[123] Xinyi Wang, Hieu Pham, Paul Michel, Antonios Anastasopoulos, Jaime Carbonell, and Graham Neubig. Optimizing data usage via differentiable rewards. In *International Conference on Machine Learning*, pages 9983–9995. PMLR, 2020.

[124] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi. Hybrid ctc/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253, 2017.

[125] Mengwei Xu, Yuxin Zhao, Kaigui Bian, Gang Huang, Qiaozhu Mei, and Xuanzhe Liu. Neural architecture search over decentralized data. *arXiv preprint arXiv:2002.06352*, 2020.

[126] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*, 2018.

[127] Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. *Advances in neural information processing systems*, 31:9793–9803, 2018.

[128] Arber Zela, Aaron Klein, Stefan Falkner, and Frank Hutter. Towards automated deep learning: Efficient joint neural architecture and hyperparameter search. *arXiv preprint arXiv:1807.06906*, 2018.

[129] Sixin Zhang, Anna E Choromanska, and Yann LeCun. Deep learning with elastic averaging sgd. In *Advances in neural information processing systems*, pages 685–693, 2015.

[130] Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. An empirical exploration of curriculum learning for neural machine translation. *arXiv preprint arXiv:1811.00739*, 2018.

[131] Ying Zhang, Mohammad Pezeshki, Philémon Brakel, Saizheng Zhang, Cesar Laurent Yoshua Bengio, and Aaron Courville. Towards end-to-end speech recognition with deep convolutional neural networks. *arXiv preprint arXiv:1701.02720*, 2017.

[132] Yiren Zhao, Duo Wang, Xitong Gao, Robert Mullins, Pietro Lio, and Mateja Jamnik. Probabilistic dual network architecture search on graphs. *arXiv preprint arXiv:2003.09676*, 2020.

[133] Hangyu Zhu and Yaochu Jin. Multi-objective evolutionary federated learning. *IEEE transactions on neural networks and learning systems*, 31(4):1310–1322, 2019.

[134] Hangyu Zhu and Yaochu Jin. Real-time federated evolutionary neural architecture search. *arXiv preprint arXiv:2003.02793*, 2020.

[135] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

Appendix A

Attached papers

The following three papers are produced during my PhD so far.

Distilling Knowledge from Ensembles of Acoustic Models for Joint CTC-Attention End-to-End Speech Recognition

Yan Gao[1]*, Titouan Parcollet*[2]*, Nicholas D. Lane*[1,3]

[1]University of Cambridge, United Kingdom, [2]Avignon University, France
[3]Samsung AI, Cambridge, United-Kingdom

yg381@cam.ac.uk, titouan.parcollet@univ-avignon.fr
ndl32@cam.ac.uk

Abstract

Knowledge distillation has been widely used to compress existing deep learning models while preserving the performance on a wide range of applications. In the specific context of Automatic Speech Recognition (ASR), distillation from ensembles of acoustic models has recently shown promising results in increasing recognition performance. In this paper, we propose an extension of multi-teacher distillation methods to joint CTC-attention end-to-end ASR systems. We also introduce three novel distillation strategies. The core intuition behind them is to integrate the error rate metric to the teacher selection rather than solely focusing on the observed losses. This way, we directly distillate and optimize the student toward the relevant metric for speech recognition. We evaluated these strategies under a selection of training procedures on the TIMIT phoneme recognition task and observed promising error rates for these strategies compared to common baselines. Indeed, the best obtained phoneme error rate of 13.11% represents a state-of-the-art score.

Index Terms: End-to-end speech recognition, attention models, CTC, multi-teacher knowledge distillation.

1. Introduction

Knowledge distillation (KD) [1], also known as teacher-student training, is commonly used to narrow the gap of performance between a smaller model and a larger one [2, 3, 4, 5, 6]. A typical KD training procedure consists of two stages. First, a deep neural network referred as the *teacher* is trained in line with standard supervised training rules based on numerous samples and their corresponding ground truth labels. Second, a compressed network, the *student* model, is trained on a selection of original ground truths and soft targets labelled by the teacher. These soft targets are the posterior probabilities obtained from the pre-trained teacher. Knowledge distillation has been shown to be particularly efficient to reduce the student size while matching its performance to that of the teacher. Common applications include Computer Vision (CV) [4, 7, 8], Natural Language Processing [9, 10, 11] (NLP) and Automatic Speech Recognition (ASR) [12, 13, 14, 15].

An alternative approach to KD focuses solely on increasing the performances of the student model without considering its complexity. Distillation from ensembles of teachers has been commonly conducted under this approach. This method is referred as the multi-teacher distillation [16, 17].

Modern deep learning based ASR systems have been shown to strongly benefit from multi-teacher distillation strategies [12, 17]. Empirically, ensembles of teacher models capture complementary information by making different errors that can be further distillate to a student model. A critical aspect of multi-teacher distillation in the context of ASR is to find suitable strategies to maximize the distillation with respect to a specific set of teachers. For instance, [12] proposed to pre-assign weights to teachers to control their impact on the distilled information. Another strategy is to sample the considered teachers randomly [17]. However, both strategies may give higher weighting, and thus higher importance, to teachers that are performing worse than others in the teacher set when applied to specific sentences.

End-to-End ASR models are particularly well suited for KD as the whole pipeline is composed of neural networks only [13, 15, 18]. One set of E2E ASR systems commonly rely either on the Connectionist Temporal Classification (CTC) loss [19], Sequence to Sequence models (Seq2Seq) [20], or a combination of the two [21]. While single teacher distillation to achieve acoustic model compression have been widely investigated on the CTC and Seq2Seq families of models [13, 3], works on ensembles of teachers to enhance the performances remain scarce.

Multi-teacher setup holds untapped potential as different E2E ASR systems often lead to different transcriptions given a fixed audio sample, which strongly increases the diversity of the teachable distributions that could be distilled to the student. Therefore, it is of crucial interest to explore the use of diverse set of E2E teachers to increase both the robustness and the performance of the student acoustic model. Potential use-cases include: Federated Learning (FL) [22, 23] with hundreds of potential acoustic models being trained concurrently, thus needing a proper aggregation or distillation strategy to further reduce the error rate and training time. Production-oriented training pipelines of ASR systems relying on strong hyper-parameters tuning phases with multiple models that could be further used rather than discarded to improve the quality of the final model.

In this paper, we first propose and investigate an extension of multi-teacher KD strategies to joint CTC-attention based ASR models. Motivated by error-weighted ensemble methods [24], we introduce three novel Error Rate-based (ER) multi-teacher distillation strategies. Indeed, common distillation strategies only consider the loss as an indicator to assess the teacher quality, while a more relevant scheme for ASR is to optimise our student toward the transcription quality.

First, the *weighted* strategy enables the student to directly assign weights to all the teachers in the course of training based on the average observed ER on the training processed mini-batch. The impact of the teachers is therefore dynamically changed between mini-batches. Then, The strategy *top-1* offers the student an option to choose a single teacher with respect to the best ER observed at the sentence level on the processed mini-batch. Finally, The strategy *top-k* allows the student to learn from the a set of best teachers that perform equally in terms of error rate on the processed mini-batch.

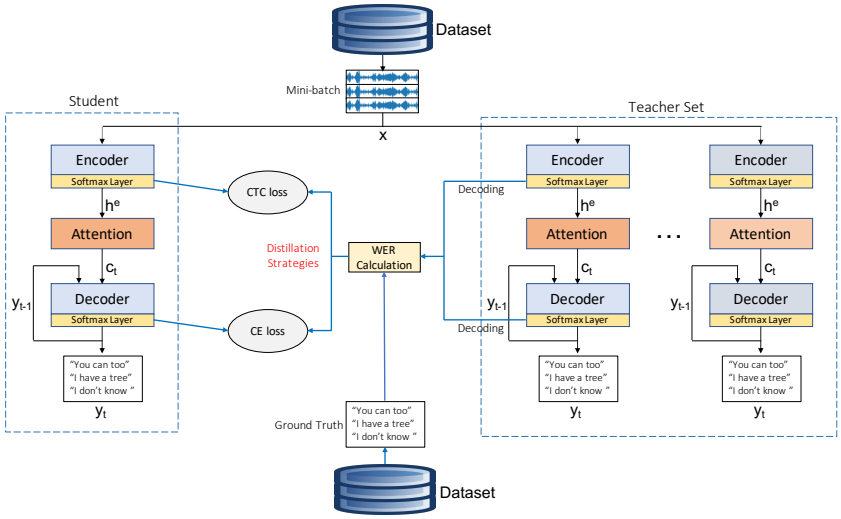

Figure 1: *Illustration of the error rate multi-teacher distillation strategies connected to a Joint CTC-Attention E2E ASR system.*

In short, our contributions are: a. Introduce multi-teacher distillation for ER reduction on joint CTC-attention based E2E systems (Sec. 2 & Sec. 3); b. Propose three novel distillation strategies focusing on the reduction of the ER (Sec. 3); c. Compare all the models on the TIMIT dataset [25] and release the code and the models within the SpeechBrain [26] [1] toolkit (Sec. 4). Following these experiments, a Phoneme Error Rate (PER) of 13.11% is reported on TIMIT, thus improving the performance over all previously investigated supervised-only E2E ASR systems.

2. Distillation for Joint CTC-Attention Speech Recognition

Joint CTC-Attention E2E systems [21] combine a seq-to-seq attention-based model [20] with the CTC loss [19]. The CTC is applied to facilitate the training of the attention decoder by directing the attention toward the correct alignment.

A typical Seq2Seq model includes three modules: an encoder, a decoder and an attention module. The *encoder* processes an input sequence $\mathbf{x} = [x_1, ..., x_{T_x}]$ with a length T_x, and creates an hidden latent representation $\mathbf{h}^e = [h^e_1, ..., h^e_{T_x}]$. Then the *decoder* attends $\mathbf{h}^e$ combined with an attention context vector c_t obtained with the attention module to produce the different decoder hidden states $\mathbf{h}^d = [h^d_1, ..., h^d_{T_y}]$, where T_y corresponds to the length of the target $\mathbf{y}$. Note that in a speech recognition scenario, the length of the original signal T_x is much longer than the utterance length T_y.

The standard supervised training procedure of the Joint CTC-Attention ASR pipeline is based on two different losses. First, the CTC loss is derived with respect to the prediction obtained from the encoder module of the Seq2Seq model:

$$\mathcal{L}_{CTC} = -\sum_{S} \log p(\mathbf{y}'|\mathbf{h}^e), \tag{1}$$

with S denoting the training dataset and $\mathbf{y}' = \mathbf{y} \cup \{blank\}$. Note that the $blank$ token is added to enable the alignment between T_x and T_y.

Second, the attention-based decoder is optimized following the Cross Entropy (CE) loss.

[1] https://speechbrain.github.io

$$\mathcal{L}_{CE} = -\sum_{S} \log p(\mathbf{y}|\mathbf{h}^d). \tag{2}$$

Both losses are combined and controlled with a fixed hyperparameter α ($0 \leq \alpha \leq 1$) as:

$$\mathcal{L} = \alpha \mathcal{L}_{CE} + (1-\alpha)\mathcal{L}_{CTC}. \tag{3}$$

In the context of knowledge distillation, we can enhance both losses by considering the different posterior probabilities obtained with a teacher for all their targets. For instance, the CE loss applied in our distillation process for the attention decoded can be rewritten as:

$$\mathcal{L}_{CE-KD} = -\sum_{S}\sum_{y \in Y} p_{tea}(y|\mathbf{h}^d) \log p_{st}(y|\mathbf{h}^d), \tag{4}$$

with y being one of the target of the label set Y. Here, $p_{tea}(y|\mathbf{h}^d)$ represents the posterior probability given by the teacher model with respect to the label y, and $p_{st}(y|\mathbf{h}^d)$ the one estimated by the student model.

These hypotheses are then used as new soft targets for the student model as following:

$$\mathcal{L}_{CTC-KD} = -\sum_{S}\sum_{n=1}^{N} p'_{tea}(\mathcal{H}_n|\mathbf{h}^e) \log p_{st}(\mathcal{H}_n|\mathbf{h}^e), \tag{5}$$

with $\mathbf{h}^e$ the hidden vector representation of the encoder and $\mathcal{H}_n$ the n-th hypothesis from the set of N-best hypothesis for the teacher. $p'_{tea}(\mathcal{H}_n|\mathbf{h}^e)$ is the normalised posterior probability of the teacher:

$$p'_{tea}(\mathcal{H}_n|\mathbf{h}^e) = \frac{p_{tea}(\mathcal{H}_n|\mathbf{h}^e)}{\sum_{n=1}^{N} p_{tea}(\mathcal{H}_n|\mathbf{h}^e)}. \tag{6}$$

Then, Eq. 3 is extended to knowledge distillation:

$$\mathcal{L}_{KD} = \alpha \mathcal{L}_{CE-KD} + (1-\alpha)\mathcal{L}_{CTC-KD}. \tag{7}$$

Finally, the global loss is computed by combining knowledge distillation and the supervised training as:

$$\mathcal{L}_{total} = \beta\mathcal{L}_{KD} + (1-\beta)\mathcal{L}, \quad (8)$$

with $\beta \in (0, 1]$ an hyperparameter controlling the impact of KD during the training.

3. Multi-teacher Error Rate Distillation

Different E2E ASR models make different mistakes while transcripting the same audio recording. Therefore, distillation from multiple pre-trained teachers has potential to help the student model to improve considerably. Finding a good teacher weight assignment strategy, however, is not trivial.

An existing approach [12, 17] is to simply compute an average over the set of teachers:

$$\mathcal{L}_{multi} = \sum_m w_m \mathcal{L}_m, \quad (9)$$

with $w_m \in [0, 1]$ the pre-assigned weight corresponding to the m-th teacher model and equal to $1/M$. M is the total number of teachers composing the ensemble. However, this method gives to a poor teacher the same importance as a good one while a natural solution would be to associate well-performing teachers with higher weights.

We propose to consider the error rate metric as a proxy to determine which teacher loss to consider during distillation. Indeed, cross-entropy and CTC losses are not directly linked to error rates, and there is no evidence that a teacher with the lowest global loss also provides the lowest error rate. Nonetheless, in speech recognition applications, the standard metric to measure performances remains error rate. The multi-teacher distillation can, therefore, benefit from the introduction of this metric to the training procedure.

More precisely, the sequence level distillation detailed in [15] and Eq. 5 can easily be extended to multi-teacher distillation and ER by replacing the N best hypothesis with the number of teachers M:

$$\mathcal{L}_{CTC-KD} = -\sum_S \sum_{m=1}^{M} p'_{tea}(\mathcal{H}_m|\mathbf{h}^e) \log p_{st}(\mathcal{H}_m|\mathbf{h}^e), \quad (10)$$

with $p'_{tea}(\mathcal{H}_m|\mathbf{h}^e)$ computed with respect to the ER:

$$p'_{tea}(\mathcal{H}_m|\mathbf{h}^e) = \frac{\exp(1-er_m)}{\sum_{m=1}^{M}\exp(1-er_m)}, \quad (11)$$

and er_m the average error rate (*e.g.* word, phonemes or concept error rates) observed on the current training mini-batch for the m-th teacher model. To complete the integration of the ER to $\mathcal{L}_{KD}$, we propose to derive three different strategies to modify $\mathcal{L}_{CE-KD}$.

***Weighted* Strategy**: Similar to $\mathcal{L}_{CTC-KD}$, we benefit from all the teachers by assigning them a weight w.r.t. their average error rates of a mini-batch, such that the weights would dynamically change adapting to different mini-batches. More precisely, w_m from Eq.9 is computed as the *softmax* distribution obtained from the ER of the current training mini-batch:

$$w_m = \frac{\exp(1-er_m)}{\sum_{m=1}^{M}\exp(1-er_m)}. \quad (12)$$

However, this approach may exhibits two potential weaknesses: 1) the worst teacher would still impact negatively the training, even though it has lowest weight; 2) the variation of ERs in one mini-batch could be large and the average ER may not reflect properly the quality of a teacher. To overcome these issues, a sentence level distillation strategy is proposed.

***Top-1* Strategy**: Here, instead of computing an average over the error rate in a mini-batch, we consider only the best performing teacher at the sentence level, *i.e.* the posterior probabilities of the best teacher for each sentence composing the mini-batch are distilled. Note that a single teacher is used for each sentence. Then, $\mathcal{L}_{CE-KD}$ can dynamically be computed following Eq. 4 during training. This approach slightly reduces the computational complexity, but also suffers from a lack of diversity. Indeed, the same teacher will always be picked for a specific sentence from one epoch to an other one. To mitigate this issue, a third strategy is introduced.

***Top-K* Strategy**: this method proposes to consider all the teachers that obtain identical error rates at the sentence level as candidates for distillation. In particular, identical ER do not necessarily mean that posterior probabilities are also equivalent as different word-level mistakes could be observed. Consequently, Eq. 4 is extended to the K-best teachers as:

$$\mathcal{L}_{CE-KD} = -\sum_S \sum_{k=1}^{K} \sum_{y\in Y} \frac{1}{K} p_{tea}(y|\mathbf{h}^d) \log p_{st}(y|\mathbf{h}^d). \quad (13)$$

Finally, the global losses $\mathcal{L}_{KD}$ and $\mathcal{L}_{total}$ are computed with the new $\mathcal{L}_{CE-KD}$ and $\mathcal{L}_{CTC-KD}$ based on Eq. 7 and Eq. 8 respectively.

4. Experiments

The multi-teacher knowledge distillation approach for joint CTC-attention E2E ASR systems (Sec. 4.2) and the proposed distillation strategies are investigated and discussed (Sec. 4.4) under different training strategies (Sec. 4.3) on the TIMIT [25] phoneme recognition task (Sec. 4.1).

4.1. The TIMIT phoneme recognition task

The TIMIT [25] dataset consists of the standard 462-speaker training set, a 50-speakers development set and a core test set of 192 sentences for a total of 5 hours of clean speech. During the experiments, the SA records of the training set are removed and the development set is used for tuning.

4.2. Model architectures

Table 1 shows the different teacher architectures and hyperparameters with their performance on validation and test sets. 80-dimensional Mel filter banks energies are extracted from the raw waveform and used as input features to the model. One CNN encoder block is composed of two 2D CNN of 64 filters and a kernel size equal to 3 with a stride of 1.

To increase the diversity in the set of teachers we also changed the recurrent neural network employed from LSTM to GRU with different numbers of layers (*i.e.* from 4 to 5) and neurons (*i.e.* from 320 to 640). Attention dimensions have also been changed across the teachers. Additionally, the models were trained with different set of hyperparameters.

The output layer of the encoder consists of 40 classes corresponding to the 39 phonemes and 1 blank label, while the decoder output size is 40 with an EOS token. All models were

Table 1: *List of the different teacher models used to compose the ensemble. "RC" is the number of repeated convolutional blocks and "data_aug" represents whether data augmentation (Y) is applied or not (N).*

RC	rnn_type	n_neurons	n_layers	dropout	data_aug	batch size	PER valid set	PER test set
2	GRU	512	4	0.15	Y	8	12.38	13.94
2	GRU	512	4	0.3	N	16	13.51	14.61
2	GRU	512	4	0.3	Y	16	13.36	14.17
2	LSTM	512	5	0.2	N	8	12.64	14.31
2	GRU	512	4	0.3	N	8	12.87	14.32
2	LSTM	320	4	0.3	N	8	14.56	15.61
1	LSTM	320	4	0.3	N	8	15.31	16.81
2	GRU	640	4	0.15	N	8	13.44	15.15
2	LSTM	512	5	0.3	N	8	12.65	14.36
2	GRU	512	4	0.15	N	8	13.27	15.20

Table 2: *Results expressed in term of Phoneme Error Rate (PER) % (i.e. lower is better) observed on the test set of the TIMIT dataset for different distillation strategies. Models are evaluated on the test with respect to the best validation performance. Original results give the PER obtained by the teacher model selected to be the student architecture prior to distillation. Single represents single teacher distillation. Average is the baseline multi-teacher KD strategy detailed in Eq. 9. Weighted (global) is a variation of Weighted considering the validation set PER rather than mini-batch-level PER to attribute weights.*

Strategies	PER (%)
Original	13.94
Single	14.15
Average	14.58
Top-1	**13.15**
Top-k	**13.13**
Weighted (global)	14.06
Weighted	**13.11**

trained for 100 epochs including pre-training and knowledge distillation. Training was performed with the Adam learning rate optimizer with vanilla hyperparameters [27]. Data augmentation is performed with a variation of SpecAugment [28] implemented within SpeechBrain.

4.3. Student selection

The student model architecture is based on the best performing teacher from the ensemble. Therefore, we propose to pick the best teacher with respect to the best PER on the validation set of TIMIT. Then, the selected model is trained with KD following the four training strategies detailed in Sec. 3 and compared to some other baselines (Sec. 4.4). For the initialization scheme, we propose to start from the pre-trained teacher neural parameters except for the last layer that is re-initialized randomly. Then, we fine-tune the whole architecture.

4.4. Speech recognition results

Table 2 shows the Phoneme Error Rate (PER) of the tested student-teacher strategies on the TIMIT test set. It is important to note that results are obtained w.r.t. the best validation performance (*i.e.* not tuned on the TIMIT test set). We compare our proposed strategies with several baselines, including singe teacher distillation, averaging weights and a strategy with fixed global weights based on ERs from the whole validation set. It is worth emphasising that TIMIT is a challenging task for E2E ASR systems due to the small amount of training samples (*i.e.* less than 5 hours). Nevertheless, the very clean recording conditions alongside with the lack of language modalities allow for a good benchmarking of pure acoustic models. Interestingly, the original teacher offers a PER of 13.94% and three baselines fail at matching this level of performance. In fact, the best reported PER of 13.11% is a state-of-the-art result on TIMIT for both E2E ASR and HMM-DNN ASR systems trained on a supervised manner only. Indeed, the previously reported SOTA with a deep CNN model and CTC is 17.7% [29] and 13.8% for a HMM-DNN pipeline [30] excluding data augmentation.

First, compared to single teacher distillation, our *weighted* multi-teacher KD strategy reaches the best performance (surpass *single* by 1.04 and *original* by 0.83). Note that in *single* strategy, teacher and student models have the same architectures (*i.e.* self distillation), where the student could not gain extra information during distillation, and thus would be tough to achieve further improvement.

Second, the very good overall performance observed with *weighted* strategy could be easily explained by the nature of the strategy and the PER statistics obtained from the ensemble of teachers. All teachers are exploited during training and the importance is determined by their PERs. Indeed, all the errors and uncertainties are helpful to build more robust students. This finding also supports the recent empirical research on the importance of the diversity in teacher ensembles [31].

Finally, *top-k* strategy offers slightly better performance than *top-1* strategy, mainly due to considering more well-quality teachers into computing metric. Additionally, it is interesting to note that the strategy relying on globally computed statistics (*i.e.* over the whole validation dataset) obtain worse performance compared to our dynamic approaches, thus highlighting the importance of sentence and mini-batch level distillation.

5. Conclusion

This paper introduces multi-teacher distillation for joint ctc-attention end-to-end ASR systems, and three novel distillation strategies relying on a combination of the error rate and the losses of the teachers. The conducted experiments on the TIMIT dataset have highlighted promising performance improvements achieved under these strategies with a state-of-the-art phoneme error rate of 13.11%. For future work, some of our results illustrated that teacher diversity is crucial for increasing the system performance. Nevertheless, it is not clear how one would measure this relationship between diversity and error rate. Developing such a measure is of utmost importance since it will allow for formulating ensemble forming strategies that produce better-constructed teacher sets.

6. References

[1] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[2] A. A. Rusu, S. G. Colmenarejo, C. Gulcehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu, and R. Hadsell, "Policy distillation," *arXiv preprint arXiv:1511.06295*, 2015.

[3] Y. Kim and A. M. Rush, "Sequence-level knowledge distillation," *arXiv preprint arXiv:1606.07947*, 2016.

[4] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Advances in Neural Information Processing Systems*, 2017, pp. 742–751.

[5] A. Mishra and D. Marr, "Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy," *arXiv preprint arXiv:1711.05852*, 2017.

[6] J. Wang, W. Bao, L. Sun, X. Zhu, B. Cao, and S. Y. Philip, "Private model compression via knowledge distillation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1190–1197.

[7] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," *arXiv preprint arXiv:1802.05668*, 2018.

[8] Y. Liu, L. Sheng, J. Shao, J. Yan, S. Xiang, and C. Pan, "Multi-label image classification via knowledge distillation from weakly-supervised detection," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 700–708.

[9] J. Cui, B. Kingsbury, B. Ramabhadran, G. Saon, T. Sercu, K. Audhkhasi, A. Sethy, M. Nussbaum-Thom, and A. Rosenberg, "Knowledge distillation across ensembles of multilingual models for low-resource languages," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4825–4829.

[10] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "Tinybert: Distilling bert for natural language understanding," *arXiv preprint arXiv:1909.10351*, 2019.

[11] S. Sun, Y. Cheng, Z. Gan, and J. Liu, "Patient knowledge distillation for bert model compression," *arXiv preprint arXiv:1908.09355*, 2019.

[12] Y. Chebotar and A. Waters, "Distilling knowledge from ensembles of neural networks for speech recognition." in *Interspeech*, 2016, pp. 3439–3443.

[13] G. Kurata and K. Audhkhasi, "Guiding ctc posterior spike timings for improved posterior fusion and knowledge distillation," *arXiv preprint arXiv:1904.08311*, 2019.

[14] H.-G. Kim, H. Na, H. Lee, J. Lee, T. G. Kang, M.-J. Lee, and Y. S. Choi, "Knowledge distillation using output errors for self-attention end-to-end models," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6181–6185.

[15] R. Takashima, S. Li, and H. Kawai, "An investigation of a knowledge distillation method for ctc acoustic models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5809–5813.

[16] M. Freitag, Y. Al-Onaizan, and B. Sankaran, "Ensemble distillation for neural machine translation," *arXiv preprint arXiv:1702.01802*, 2017.

[17] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran, "Efficient knowledge distillation from an ensemble of teachers." in *Interspeech*, 2017, pp. 3697–3701.

[18] M. Huang, Y. You, Z. Chen, Y. Qian, and K. Yu, "Knowledge distillation for sequence model." in *Interspeech*, 2018, pp. 3703–3707.

[19] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*, 2014, pp. 1764–1772.

[20] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.

[21] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[22] D. Dimitriadis, K. Kumatani, R. Gmyr, Y. Gaur, and S. E. Eskimez, "A federated approach in training acoustic models," in *Proc. Interspeech*, 2020.

[23] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[24] L. I. Kuncheva, *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2014.

[25] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, 1993.

[26] M. Ravanelli, T. Parcollet, A. Rouhe, P. Plantinga, E. Rastorgueva, L. Lugosch, N. Dawalatabad, C. Ju-Chieh, A. Heba, F. Grondin, W. Aris, C.-F. Liao, S. Cornell, S.-L. Yeh, H. Na, Y. Gao, S.-W. Fu, C. Subakan, R. De Mori, and Y. Bengio, "Speechbrain," https://github.com/speechbrain/speechbrain, 2021.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[28] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[29] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.

[30] M. Ravanelli, T. Parcollet, and Y. Bengio, "The pytorch-kaldi speech recognition toolkit," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6465–6469.

[31] X. Zhu, S. Gong *et al.*, "Knowledge distillation by on-the-fly native ensemble," in *Advances in neural information processing systems*, 2018, pp. 7517–7527.

End-to-End Speech Recognition from Federated Acoustic Models

Yan Gao[1]*, Titouan Parcollet*[2]*, Javier Fernandez-Marques*[3]
Pedro P. B. de Gusmao[1]*, Daniel J. Beutel*[1,4]*, Nicholas D. Lane*[1]

[1]University of Cambridge, [2]Avignon University, [3]University of Oxford, [4]Adap GmbH

yg381@cam.ac.uk, titouan.parcollet@univ-avignon.fr, javier.fernandezmarques@cs.ox.ac.uk
pp524@cam.ac.uk, daniel@adap.com, ndl32@cam.ac.uk

Abstract

Training Automatic Speech Recognition (ASR) models under federated learning (FL) settings has recently attracted considerable attention. However, the FL scenarios often presented in the literature are artificial and fail to capture the complexity of real FL systems. In this paper, we construct a challenging and realistic ASR federated experimental setup consisting of clients with heterogeneous data distributions using the French Common Voice dataset, a large heterogeneous dataset containing over 10k speakers. We present the first empirical study on attention-based sequence-to-sequence E2E ASR model with three aggregation weighting strategies – standard FedAvg, loss-based aggregation and a novel word error rate (WER)-based aggregation, are conducted in two realistic FL scenarios: *cross-silo* with 10-clients and *cross-device* with 2k-clients. In particular, the WER-based weighting method is proposed to better adapt FL to the context of ASR by integrating the error rate metric with the aggregation process. Our analysis on E2E ASR from heterogeneous and realistic federated acoustic models provides the foundations for future research and development of realistic FL-based ASR applications.

Index Terms: End-to-end ASR, federated learning

1. Introduction

Neural networks are now widely adopted in state-of-the-art automatic speech recognition (ASR) systems [1]. This success mostly relies on centralised training of deep neural architectures with large amounts of data and computational power [2, 3, 4]. But decentralized alternatives are becoming more practical due to the proliferation of powerful mobile devices (e.g. phones, tablets) and rapid developments of communication technologies (e.g. 5G). Such ingredients make federated and on-device training of ASR a feasible and an attractive alternative to traditional centralised training [5]. Federated learning (FL) offers new opportunities to advance ASR quality given the unprecedented amount of user data directly available on-device. For example, such data could be leveraged to better adapt ASR to the users' usage, or to simply improve the robustness of models to realistic scenarios [6]. However, decentralized training with users' data require strong anonymity and privacy guarantees, this in turn limits how such training maybe performed and presenting a series of significant challenges.

With FL, the training process leverages large and diverse amounts of data collected locally by user devices, while also offering the requisite privacy protection [7]. In practice, FL allows for the training of machine learning models, such as deep neural networks, collaboratively between a number of devices – assisted by a central server [7, 5, 6]. In a standard setup, a global model is learned from aggregating updates obtained from computation performed locally on the considered pool of mobile devices (often referred to as clients). While the aggregation step is performed on a central server, users' data is never shared with it and remains local to the clients.

However, training E2E ASR models in a realistic FL setting comes with numerous challenges. First, it is notoriously complicated to train a deep learning model with FL on non independent and identically distributed data (non-IID) [6, 8, 9]. Unfortunately, on-device speech data is, by its very nature, extremely non-IID (e.g. different acoustic environments, words being spoken, languages, microphones, amount of available speech, etc.). Second, state-of-the-art (SOTA) E2E ASR models are computationally intensive and potentially not suited to on-device training phases of FL. Indeed, the latest ASR systems rely on large Transformers [10, 11], Transducers [12, 13] or attention sequence-to-sequence (Seq2Seq) models [14, 15] that process high-dimensional acoustic features. In addition, E2E ASR training is difficult and very sensitive at early optimisation stages due to the complexity of learning a proper alignment between the latent speech representation and the transcription. Because of these three issues, training ASR models from scratch on low-resources languages [16, 17, 18] is particularly challenging.

Despite the growing number of studies applying FL on speech-related tasks [19, 20, 21, 22, 23], very few of these have investigated its use for end-to-end (E2E) ASR. To our best knowledge, existing works on FL for ASR typically rely on strong simplifying assumptions for many of these challenges – and this results in their experimental settings being still far away from the conditions in which a FL ASR would need to function. In addition, some works are evaluated on the LibriSpeech [24] dataset, further limiting the realism as recordings are from users reading books in a controlled setting without background noise. For instance, the work [22] introduces a novel FL client-based adaptive training in a specific setup known as *cross-silo* (i.e. reduced number of clients with high amount of homogeneous data) to train a HMM-DNN based ASR system, thus relinquishing two of the constraints (i.e. non-IID and complexity of the model with simplified HMM alignments). Then, [21] proposes a federated transfer learning platform with improved performance using enhanced federated averaging and hierarchical optimization for E2E ASR. While the alignment issue is alleviated with a careful centralised pre-training phase, the non-IID constraint remains mostly unconsidered as the FL training is performed on LibriSpeech.

In this paper, we investigate FL in a more realistic setting with the French Common Voice (CV) dataset. It provides a large set of speakers that used their own devices to record a given set of sentences, naturally fitting to federated learning with various speakers, acoustic conditions, microphones and accents. We evaluate both a *cross-silo* and a *cross-device* (i.e. large number of clients with few non-IID data) FL setups while

training a SOTA E2E ASR system. We conduct an empirical study of three different weighting strategies during model aggregation to approach the difficulty of non-IID FL. In particular, this work introduces a word error rate (WER) based strategy to further adapt ASR training to federated learning. In short, our contributions are:

1. Present the first study on attention-based Seq2Seq E2E ASR model for realistic FL scenarios. Our setup approaches previously overlooked challenges such as extremely heterogeneous recording conditions.
2. Evaluate both *cross-silo* and *cross-device* FL with up to 2k clients on the naturally-partitioned and heterogeneous French Common Voice dataset.
3. A new aggregation strategy based on WER to further integrate the specificity of ASR to FL.
4. Release the source code using Flower [25] and SpeechBrain [26] to facilitate replication and future research[1].

2. End-to-end Speech Recognizer

To ensure realistic conditions, the considered E2E ASR system relies on the wide spread joint connectionist temporal classification (CTC) with attention paradigm [14]. This method combines a Seq2Seq attention-based model [27] with the CTC loss [28].

A typical ASR Seq2Seq model includes three modules: an encoder, a decoder and an attention module. Given a speech input sequence (i.e. speech signal or acoustic features) $\mathbf{x} = [x_1, ..., x_{T_x}]$ with a length T_x, the *encoder* first converts it into an hidden latent representation $\mathbf{h}^e = [h_1^e, ..., h_{T_x}^e]$. Then the *decoder* attends to the encoded representation $\mathbf{h}^e$ combined with an attention context vector c_t obtained with the attention module, to produce the different decoder hidden states $\mathbf{h}^d = [h_1^d, ..., h_{T_y}^d]$, with T_y corresponding to the length of the target $\mathbf{y}$. In a speech recognition scenario, the length of the original signal T_x is usually longer than the utterance length T_y.

The standard training procedure of the joint CTC-Attention ASR pipeline is based on two different losses. First, the CTC loss is derived with respect to the prediction obtained from the encoder module of the Seq2Seq model:

$$\mathcal{L}_{CTC} = -\sum_{S} \log p(\mathbf{y}'|\mathbf{h}^e), \tag{1}$$

with S denoting the training dataset and $\mathbf{y}' = \mathbf{y} \cup \{blank\}$. The $blank$ token enables the alignment between T_x and T_y. Second, the attention-based decoder is optimised following the cross entropy (CE) loss:

$$\mathcal{L}_{CE} = -\sum_{S} \log p(\mathbf{y}|\mathbf{h}^d). \tag{2}$$

The losses are combined with a hyperparameter $\mu \in [0, 1]$ as:

$$\mathcal{L} = \mu\mathcal{L}_{CE} + (1 - \mu)\mathcal{L}_{CTC}. \tag{3}$$

In practice the CTC loss facilitates the early convergence of the system due its monotonic behavior while the attentional decoder needs to first figure out where to attend in the hidden representation of the entire input sequence.

[1] https://github.com/yan-gao-GY/Flower-SpeechBrain

3. Federated Training of Acoustic Models

The process of training an end-to-end acoustic model using federated learning follows four steps: 1) Following [21], model weights are initialised with a pre-training phase on a centralised dataset; 2) The centralised server samples K clients from a pool of M clients and uploads to these clients the weights of the model. 3) The clients train the model for t_{local} local epochs in parallel based on their local user data and send back the new weights or gradients to the server. 4) The server aggregates the weights and restart at step 2. This procedure is executed for T rounds until the model converges on a dedicated validation set.

3.1. Federated Optimisation

Federated Averaging (FedAvg) [7], as a typical aggregation strategy based on averaging local stochastic gradient descent (SGD) updates, has been widely applied in various FL tasks [29]. At the beginning of a new round, the server sends to all participating clients the global model, which contains the resulting model after the aggregation stage. During each training round, each client $k \in K$, consisting of n_k samples of audio data, runs $t \in [0, t_{local}]$ iterations with learning rate η_l to locally update the model based on the loss function Eq. 3,

$$w_{t+1}^{(k)} = w_t^{(k)} - \eta_l \tilde{g_k}. \tag{4}$$

where w_k is the local model weights in client k, and $\tilde{g_k}$ denotes an average gradient over local samples. After training for t_{local} local epochs in the global round T, the updated weights $w_T^{(k)}$ of the client k are sent back to the server. Then, the local gradient $g_T^{(k)}$ can be approximated by computing the difference between the latest updated model and the previous global model w_{T-1}:

$$g_T^{(k)} = w_T^{(k)} - w_{T-1}. \tag{5}$$

Then, the gradients from all clients are aggregated as follows:

$$\Delta_T = \sum_{k=1}^{K} \alpha_T^{(k)} g_T^{(k)}, \tag{6}$$

where $\alpha_T^{(k)}$ denotes different weighting strategies described in Section 3.2. The updated global model weights w_T are computed with a server learning rate η_s according to:

$$w_T = w_{T-1} - \eta_s \Delta_T, \tag{7}$$

During FL training, especially with heterogeneous data, the global model may deviates away from the original task or simply not converges [6, 8, 9], and therefore lead to performance degradation. To alleviate this issue, and motivated by [21], we propose an additional training iteration over a small batch of held-out data on the server, after the standard model update procedure with Eq. 7. This way, the global model would be pulled back to the direction of interest and the convergence would accelerate. Once the aggregated global model has been computed, the server sends it back to the clients and re-iterates.

3.2. Weighting Strategies

In the original FedAvg algorithm, the weighting $\alpha_T^{(k)}$ for the aggregation step is based on the number of client samples each:

$$\alpha_T^{(k)} = \frac{n_k}{\sum_{k=1}^{K} n_k}, \tag{8}$$

In realistic FL settings with heterogeneous client data distribution, however, the situation becomes challenging. First, some clients may contain data that is skewed and not representative of the global data distribution (e.g. audio samples with different languages or multiple speakers). As a result, the aggregated model might simply not converge if such clients have proportionally more training samples than others. Second, clients containing low quality data would introduce unexpected noise into the training process (e.g. extreme noise in the background). Either scenarios could lead to model deviation in the aggregation step, which can not be solved via the standard FedAvg weighting method (Eq. 8). A potential solution, instead, is to use the averaged training loss as a weighting coefficient, thus reflecting the quality of the locally trained model. Intuitively, higher loss would indicate that the global model struggles to learn from the client' local data. More precisely, we compute the weighting with the *Softmax* distribution obtained from the the negative training loss from Eq. 3. Eq. 8 can be modified as follows:

$$\alpha_T^{(k)} = \frac{\exp\left(-\mathcal{L}_k\right)}{\sum_{k=1}^{K} \exp\left(-\mathcal{L}_k\right)}. \tag{9}$$

In the context of ASR, WER is commonly used as the final evaluation metric for the model instead of the training loss. Intuitively, we propose a WER-based weighting strategy for aggregation. Similarly, this approach utilizes the values $(1-wer)$ obtained on the validation set as weighting coefficients $\alpha_T^{(k)}$, after passing them through a *Softmax* function:

$$\alpha_T^{(k)} = \frac{\exp\left(1-wer_k\right)}{\sum_{k=1}^{K} \exp\left(1-wer_k\right)}. \tag{10}$$

In this way, we directly optimise the model towards the relevant metric for speech recognition.

4. Experimental Settings

In this section we present the model, the dataset used in our experiments and describe our realistic FL experimental setup.

4.1. E2E Speech Recognizer

The experiments are based on an attention Seq2Seq model trained with the joint CTC-attention objective [14]. The encoder is made of a 2D CNN block with 128 filters and a 5-layer bidirectional LSTM with 1024 units. The decoder is a single layered attentional GRU. The E2E acoustic model is trained to predict subwords units. No language model fusion is performed to properly assess the impact of the training procedure on the acoustic models. Data is augmented in the time-domain during training. The complete details of the architecture and hyperparameters can be found in our GitHub[1]. The model has been implemented within SpeechBrain [26] and is therefore extremely easy to manipulate, customise and retrain.

4.2. Common Voice French

In our experiments, we used the French set of the Common Voice dataset (version 6.1) [30]. Common Voice (CV) allows us to simulate a realistic FL setup as it contains a total of $328k$ utterances (475 hours in total) with diverse accents which were recorded by more than $10K$ French-speaking participants. More precisely, the train set consists of 4190 speakers (425.5 hours of speech), while the validation and test sets contain around 24 hours of speech from 2415 and 4247 speakers respectively. Such recording, accent, and acoustic environment

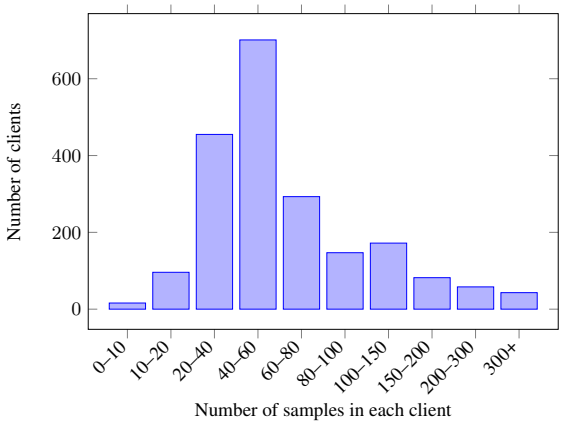

Figure 1: *Illustration of the sample distribution across the* 2036 *FL clients from the French Common Voice dataset.*

diversity highly correlates with the requirements needed for real-world FL. This level of realism, especially on the variety of acoustic environments, is not possible with other datasets such as LibriSpeech used in the closest works to our own [22, 21].

4.3. Realistic Federated Learning

Based on the natural partitioning on the CV dataset we propose to conduct two sets of experiments reflecting real usages of FL:

***Cross-silo* FL**. In this scenario, clients are generally few, with high availability during all rounds, and are likely to have similar data distribution for training [6], e.g. a consortium of hospitals, each of which having large amounts of data from a large set of users. In this context, shared data is often independent and identically distributed. To achieve *cross-silo* FL, and following [21], the dataset is split in 10 random partitions with no overlapping speakers each containing roughly the same amount of speech data. Each partition is assigned to one FL client.

***Cross-device* FL**. On the other hand, a *cross-device* setup will likely encompass thousands of clients having very different data distributions (non-IID) participating in just a few rounds [6]. To reproduce this scenario, we randomly divided the CV datasets into 2036 partitions. This results in each client containing data from two different speakers. In this way, we simulate the realistic scenario where two users use the same device (e.g personal assistants or smart car). Figure 1 precisely depicts the sample distribution over all considered clients.

4.4. Federated Learning for ASR: a hybrid approach

Training E2E ASR models in a federated learning manner is challenging in many aspects. First, jointly learning the alignment and the latent speech representation is a difficult task that commonly requires large datasets. Therefore, and as we experienced during our analysis, it is nearly impossible to train an E2E ASR model from scratch in a realistic FL setup. This is because most of the clients can only provide a few minutes of speech, resulting in a slow model convergence or no convergence at all. To overcome this issue we first pre-train the global model on half of the data samples. We do this by distinctly partitioning the original dataset into a small subset of speakers (with many samples) for centralized training (referred to sub-

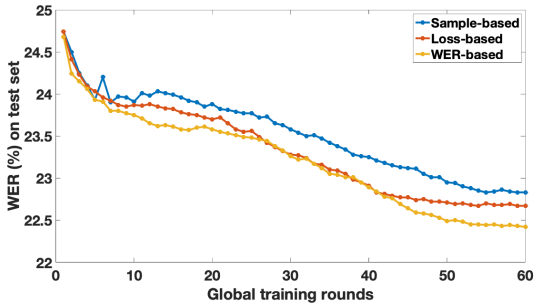

Figure 2: *Word error rate (WER) for 3 weighting strategies with respect to global training rounds in the 2K-client setting.*

Table 1: *Speech recognition results on the test set of French Common Voice for different scenarios and weighting strategies.*

Training Scenario		**WER** (%)
Centralised	Train on all data (lower bound)	20.18
	Train on 1st half (*warm-up* only)	25.26
	Train on 2nd half (after *warm-up*)	20.94
10-clients FL *Cross-silo*	Standard FedAvg	21.26
	Loss-based aggregation	21.10
	WER-based aggregation	20.99
2K-clients FL *Cross-device*	Standard FedAvg	22.83
	Loss-based aggregation	22.67
	WER-based aggregation	22.42

sequently as the *warm-up dataset*) and a much larger subset of speakers (having fewer samples each) for the FL experiment. The small subset contains 117 speakers, leaving the remaining 4073 speakers to continue training the E2E ASR model in a federated fashion. We argue that this scenario remains realistic as, in practice, centralised data is often available and can therefore be used to pre-train models.

The number of clients that participate in each round influences the outcome of the experiment. More precisely, at the beginning of each round, K clients are randomly selected from the available set. Higher K lead to slight improvement of the performance but also increase the communication overhead and potential memory usage on the server side (i.e. more clients to aggregate), while lower K induce an increased number of rounds to converge. In addition to setting the number of global rounds for the FL experiment, we must as well set the number of local epoch (i.e. on each client). This, however, is a non-trivial task [7]. In practice, we found that increasing the number of local epochs leads to instabilities as longer training would cause over-fitting the local client data. Hence, clients are locally trained for only 5 epochs.

For the *cross-silo* setup, all clients are selected at each round ($K = 10$) while *cross-device* training relies on $K = 100$. Indeed, we decided to follow the strategy investigated by previous large scale FL works [5].

For evaluation, we infer the trained models on the test set of French Common Voice dataset with beam search. The results are shows on Table 1. Note that the test set is smaller in number of speech hours but contains more speakers (4247 speakers) than the training set, making this a challenging but realistic task.

5. Speech Recognition Results

When comparing results across the different training setups, we may notice from Table 1 that training on the entire dataset in a *centralised* way gives us the best WER with 20.18%. This lower-bound is expected as the system has full visibility of the data and can sample the inputs in an almost IID fashion. On the other hand, when using only the *warm-up* dataset, we notice the effect of having fewer data points for training as the WER increases to 25.26%. This is expected as the system has now less data to learn from. This sheds some light on the inherent lower-bound limitations of FL, limited to partial data observations in each round. The third centralised scenario trains the warmed-up model on the 2nd half of data in an on-line training fashion. This model provides a slightly lower WER compared to all FL models. However, we should note that this is an unrealistic setting as training models in a centralised way would void all the privacy guarantees that FL offers.

The effect of data visibility can indeed be seen in both *cross-silo* and *cross-device* scenarios, which do not have uniform access to data. However, since this problem is less severe in the former setup, with the correct choice of aggregation strategy we are still able to obtain a WER of 20.99%, which is very close to the centralised lower bound of 20.18%. As for the more challenging *cross-device* scenario, the effect of non-IID data distribution among devices leads to its best WER being 22.43%. This value is larger than the worst *cross-silo* result, showing the strong effects of the non-IID nature of the data partitioning and also suggesting that *cross-silo* results could offer a more realistic lower-bound results for FL in general.

Compared to different weighting strategies, WER-based and loss-based methods obtain a better performance and converge faster (Figure 2), which indicates that weakening the effects of low-quality clients can assist the aggregation process in federated training with heterogeneous data distribution. Herein, we have two types of indicators reflecting the quality of clients. WER-based method exceeds loss-based strategy after 40 training rounds in 2k-client setting (Figure 2). The results in Table 1 show that WER-based strategy obtain the lowest WER in both settings, surpassing the *warm-up* model in the *cross-silo* setting by 4.3%. This could be easily explained by the nature of the strategy which directly optimise the model toward the relevant metric for speech recognition.

6. Conclusion

In this paper, we presented the first study on attention-based Seq2Seq E2E ASR model with three aggregation weighting strategies – standard FedAvg, loss-based aggregation and a novel WER-based aggregation, for realistic FL scenarios. We evaluated both *cross-silo* and *cross-device* FL on the French Common Voice dataset which, unlike other datasets such as LibriSpeech, includes recordings from a large number of users in a diverse set of scenarios. Our WER-based aggregation strategy, aware of the data quality of clients with respect to the task, enables complex FL trained E2E ASR models to perform as well as centralised trained ones. Our work sets the foundations for future research and development of realistic FL-based ASR applications. For future work, we plan to investigate other ASR model architectures and optimisers to better adapt to challenging FL environments.

7. References

[1] A. Kumar, S. Verma, and H. Mangla, "A survey of deep learning techniques in speech recognition," in *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. IEEE, 2018, pp. 179–185.

[2] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[3] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International conference on machine learning*, 2016, pp. 173–182.

[4] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word lstm model for large vocabulary speech recognition," *arXiv preprint arXiv:1610.09975*, 2016.

[5] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[6] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[7] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[8] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[9] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3400–3413, 2019.

[10] A. Mohamed, D. Okhonko, and L. Zettlemoyer, "Transformers with convolutional context for asr," *arXiv preprint arXiv:1904.11660*, 2019.

[11] A. Zeyer, P. Bahar, K. Irie, R. Schlüter, and H. Ney, "A comparison of transformer and lstm encoder decoder models for asr," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 8–15.

[12] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.

[13] E. Battenberg, J. Chen, R. Child, A. Coates, Y. G. Y. Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, "Exploring neural transducers for end-to-end speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 206–213.

[14] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[15] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[16] A. Rosenberg, K. Audhkhasi, A. Sethy, B. Ramabhadran, and M. Picheny, "End-to-end speech recognition and keyword search on low-resource languages," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5280–5284.

[17] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Pre-training on high-resource speech recognition improves low-resource speech-to-text translation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 58–68.

[18] J.-Y. Hsu, Y.-J. Chen, and H.-y. Lee, "Meta learning for end-to-end low-resource speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7844–7848.

[19] A. Hard, K. Partridge, C. Nguyen, N. Subrahmanya, A. Shah, P. Zhu, I. L. Moreno, and R. Mathews, "Training keyword spotting models on non-iid data with federated learning," *arXiv preprint arXiv:2005.10406*, 2020.

[20] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated learning for keyword spotting," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6341–6345.

[21] D. Dimitriadis, K. Kumatani, R. Gmyr, Y. Gaur, and S. E. Eskimez, "A federated approach in training acoustic models," in *Proc. Interspeech*, 2020.

[22] X. Cui, S. Lu, and B. Kingsbury, "Federated acoustic modeling for automatic speech recognition," *arXiv preprint arXiv:2102.04429*, 2021.

[23] F. Granqvist, M. Seigel, R. van Dalen, Á. Cahill, S. Shum, and M. Paulik, "Improving on-device speaker verification using federated learning with privacy," *arXiv preprint arXiv:2008.02651*, 2020.

[24] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[25] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.

[26] M. Ravanelli, T. Parcollet, A. Rouhe, P. Plantinga, E. Rastorgueva, L. Lugosch, N. Dawalatabad, C. Ju-Chieh, A. Heba, F. Grondin, W. Aris, C.-F. Liao, S. Cornell, S.-L. Yeh, H. Na, Y. Gao, S.-W. Fu, C. Subakan, R. De Mori, and Y. Bengio, "Speechbrain," https://github.com/speechbrain/speechbrain, 2021.

[27] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.

[28] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*, 2014, pp. 1764–1772.

[29] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[30] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

IMUTube: Automatic Extraction of Virtual on-body Accelerometry from Video for Human Activity Recognition

HYEOKHYEN KWON*, School of Interactive Computing, Georgia Institute of Technology, USA
CATHERINE TONG*, Department of Computer Science, University of Oxford, UK
HARISH HARESAMUDRAM, School of Electrical and Computer Engineering, Georgia Institute of Technology, USA
YAN GAO, Department of Computer Science, University of Oxford, UK
GREGORY D. ABOWD, School of Interactive Computing, Georgia Institute of Technology, USA
NICHOLAS D. LANE, Department of Computer Science, University of Oxford, UK
THOMAS PLÖTZ, School of Interactive Computing, Georgia Institute of Technology, USA

The lack of large-scale, labeled data sets impedes progress in developing robust and generalized predictive models for on-body sensor-based human activity recognition (HAR). Labeled data in human activity recognition is scarce and hard to come by, as sensor data collection is expensive, and the annotation is time-consuming and error-prone. To address this problem, we introduce IMUTube, an automated processing pipeline that integrates existing computer vision and signal processing techniques to convert videos of human activity into virtual streams of IMU data. These virtual IMU streams represent accelerometry at a wide variety of locations on the human body. We show how the virtually-generated IMU data improves the performance of a variety of models on known HAR datasets. Our initial results are very promising, but the greater promise of this work lies in a collective approach by the computer vision, signal processing, and activity recognition communities to extend this work in ways that we outline. This should lead to on-body, sensor-based HAR becoming yet another success story in large-dataset breakthroughs in recognition.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing**; • **Computing methodologies → Artificial intelligence**; *Supervised learning by classification.*

Additional Key Words and Phrases: Activity Recognition, Data Collection, Machine Learning

ACM Reference Format:
Hyeokhyen Kwon, Catherine Tong, Harish Haresamudram, Yan Gao, Gregory D. Abowd, Nicholas D. Lane, and Thomas Plötz. 2020. IMUTube: Automatic Extraction of Virtual on-body Accelerometry from Video for Human Activity Recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 3, Article 87 (September 2020), 29 pages. https://doi.org/10.1145/3411841

*Both authors contributed equally to this research.

Authors' addresses: Hyeokhyen Kwon, hyeokhyen@gatech.edu, School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, USA; Catherine Tong, eu.tong@cs.ox.ac.uk, Department of Computer Science, University of Oxford, UK; Harish Haresamudram, harishkashyap@gatech.edu, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA; Yan Gao, yan.gao@keble.ox.ac.uk, Department of Computer Science, University of Oxford, UK; Gregory D. Abowd, abowd@gatech.edu, School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, USA; Nicholas D. Lane, nicholas.lane@cs.ox.ac.uk, Department of Computer Science, University of Oxford, UK; Thomas Plötz, thomas.ploetz@gatech.edu, School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA, USA.

2474-9567/2020/9-ART87 $15.00
https://doi.org/10.1145/3411841

1 INTRODUCTION

On-body sensor-based human activity recognition (HAR) is widely utilized for behavioral analysis, such as user authentication, healthcare, and tracking everyday activities [5, 14, 50, 78, 97]. Regardless of its utility, the HAR field has yet to experience significant improvements in recognition accuracy, in contrast to the breakthroughs in other fields, such as speech recognition [34], natural language processing [19], and computer vision [32]. In those domains it is possible to collect huge amounts of labeled data, the key for deriving robust recognition models that strongly generalize across application boundaries. In contrast, collecting large-scale, labeled data sets has so far been limited in sensor-based human activity recognition. Labeled data in human activity recognition is scarce and hard to come by, as sensor data collection is expensive, and the annotation is time-consuming and sometimes even impossible for privacy or other practical reasons. A model derived from such a sparse dataset is not likely to generalize well.

Despite the numerous efforts in improving human activity dataset collection, the scale of typical datasets remains small, thereby only covering limited sets of activities [14, 35, 88, 97]. Even the largest sensor-based activity dataset only spans a few dozen users and relatively short durations [5, 72], which is in stark contrast to the massive datasets in other domains that are often several orders of magnitude larger. For example, Daphnet freezing of gait dataset [5] has 5 hours of sensor data from 10 subjects, and PAMAP2 dataset [72] has 7.5 hours of sensor data from 9 subjects. However, for reference, the ImageNet dataset [18] has approx. 14 million images, and the "One billion words" benchmark [15] contains literally one billion words.

In this work, we develop a framework that can potentially alleviate the sparse data problem in sensor-based human activity recognition. We aim at harvesting existing video data from large-scale repositories, such as YouTube, and automatically generate data for virtual, body-worn movement sensors (IMUs) that will then be used for deriving sensor-based human activity recognition systems that can be used in real-world settings. The overarching idea is appealing due to the sheer size of common video repositories and the availability of labels in the form of video titles and descriptions. Having access to such data repositories opens up possibilities for more robust and potentially more complex activity recognition models that can be employed in entirely new application scenarios, which so far could not have been targeted due to limited robustness of the learned models. The challenges for making these vast amounts of existing data usable for sensor-based activity recognition are manyfold, though: *i)* the datasets need to be curated and filtered towards the actual activities of interest; *ii)* even though video data capture the same information about activities in principle, sophisticated preprocessing is required to match the source and target sensing domains; *iii)* the opportunistic use of activity videos requires adaptations to account for contextual factors such as multiple scene changes, rapid camera orientation changes (landscape/portrait), the scale of the performer in the far sight, or multiple background people not involved in the activity; and *iv)* new forms of features and activity recognition models will need to be designed to overcome the short-comings of learning from video-sourced motion information for eventual IMU-based inference.

Our work is part of a growing number of exciting recent research results that explore the generation of cross-modality sensor data from "data-rich" sources such as video and motion capture in various domains [36, 74, 87, 92]. For example, in [36] IMU data was synthesized from high-fidelity motion capture data with high temporal and spatial resolutions for computing human pose in real-time. On a similar vein, [87, 92] generate sensory data from motion capture datasets and demonstrate their effectiveness for activity recognition. Most similar to our work, [74] showed in principle that motion information can be extracted from video and utilized for sensor-based HAR.

In this paper, we present a method that allows us to effectively use video data for training sensor-based activity recognizers, and as such demonstrates the first step towards larger-scale, and more complex deployment scenarios than what is considered the state-of-the-art in the field. Our approach extracts motion information from arbitrary human activity videos, and is thereby not limited to specific scenes or viewpoints. We have developed **IMUTube**, an automated processing pipeline that: *i)* applies standard pose tracking and 3D scene understanding techniques

to estimate full 3D human motion from a video segment that captures a target activity; *ii)* translates the visual tracking information into virtual motion sensors (IMU) that are placed on dedicated body positions; *iii)* adapts the virtual IMU data towards the target domain through distribution matching; and *iv)* derives activity recognizers from the generated virtual sensor data, potentially enriched with small amounts of real sensor data. Our pipeline integrates a number of off-the-shelf computer vision and graphics techniques, so that IMUTube is fully automated and thus directly applicable to a rich variety of existing videos. One notable limitation from our current prototype is that it still requires human curation of videos to select appropriate activity content. However, with advances in computer vision the potential of our approach can be further increased towards complete automation.

The work presented in this paper is our first step towards the greater vision of automatically deriving robust activity recognition systems for body-worn sensing systems. The key idea is to opportunistically utilize as much existing data and information as possible thereby not being limited to the particular target sensing modalities. We present the overall approach and relevant technical details and explore the potential of the approach on practical recognition scenarios. Through a series of experiments on three benchmark datasets—RealWorld [86], PAMAP2 [72], and Opportunity [14]—we demonstrate the effectiveness of our approach. We discuss the overall potential of models trained purely on virtual sensor data, which in certain cases can even reach recognition accuracies that are comparable to models that are trained only on actual sensor data. Moreover, we show that when adding only small portions of real sensor data during model training we are even able to outperform those models that were trained on real sensor data alone. As such, our experiments show the potential of the proposed approach, a paradigm shift for deriving sensor-based human activity recognition systems.

This work opens up the opportunity for the human activity recognition community to expand the general focus towards more complex and more challenging recognition scenarios. We expect the proposed approach to dramatically accelerate the progress of human activity recognition research. With the proposed method it will also be possible to freely experiment with and optimize on-body sensor configurations, which will have a substantial impact on real-world deployments. We discuss possible extensions to the presented approach, and thus define a research agenda towards next-generation sensor-based human activity recognition.

2 EXTRACTING VIRTUAL IMU DATA FROM VIDEOS

The key idea of our work is to replace the conventional data collection procedure that is typically employed for the development of sensor-based human activity recognition (HAR) systems. Our approach aims at making existing, large-scale video repositories accessible for the HAR domain, leading to training datasets of sensor data, such as IMUs, that are potentially multiple orders of magnitude larger than what is standard today. With such a massively increased volume of *real* movement data—in contrast to simulated or generated samples, that often do not exhibit the required quality nor variability—it will become possible to develop substantially more complex and more robust activity recognition systems with a potentially much broader scope than the state-of-the-art in the field. In what follows, we first give an overview of the general approach before we provide the technical details of our procedure that converts videos into virtual IMU data.

2.1 IMUTube Overview

Figure 1 gives on overview of our framework for deriving sensor-based human activity recognition systems. The top left part ("conventional") summarizes the currently predominant protocol. Study participants are recruited and invited for data collection in a laboratory environment. There they wear sensing platforms, such as a wrist-worn IMU, and engage in the activities of interest, typically in front of a camera. Human annotators provide ground truth labeling either directly, i.e., while the activities are performed, or based on the video footage from the recording session. This procedure is very labor-intensive and often error-prone, and, as such, labeled datasets of only limited size can typically be recorded with reasonable efforts.

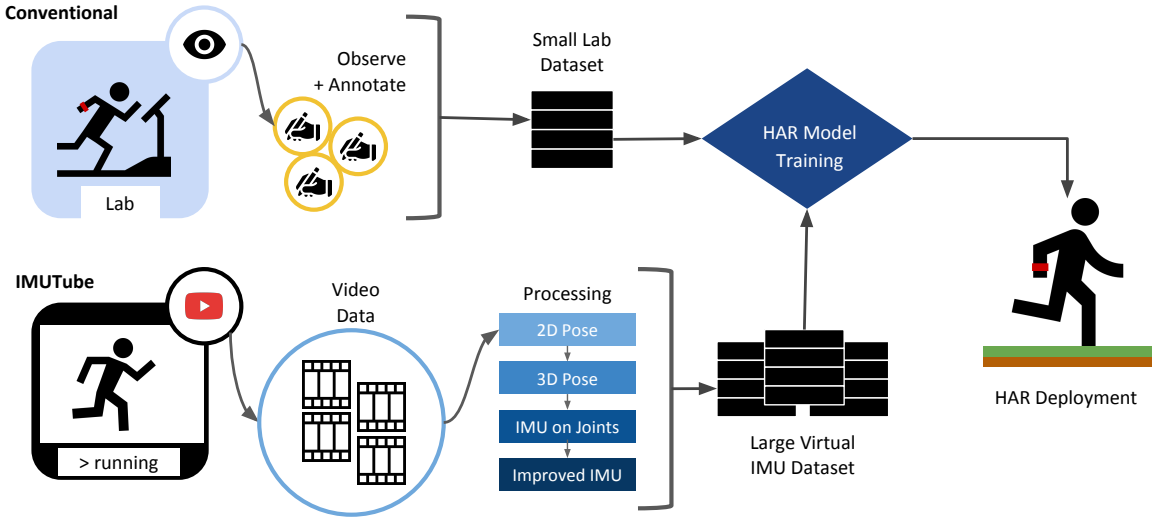

Fig. 1. The proposed IMUTube system replaces the conventional data recording and annotation protocol (upper left) for developing sensor-based human activity recognition (HAR) systems (upper right). We utilize existing, large-scale video repositories from which we generate virtual IMU data that are then used for training the HAR system (bottom part).

In contrast, our approach aims at utilizing existing, large-scale repositories of videos that capture activities of interest (bottom left part of Figure 1 labeled "IMUTube"). With the explosive growth of social media platforms, a virtually unlimited supply of labeled video is available online that we aim to utilize for training sensor-based HAR systems. In our envisioned application, a query for a specific activity delivers a (large) set of videos that seemingly capture the target activity. These results (currently) need to be curated in order to eliminate obvious outliers etc. such that the videos are actually relevant to the task (see discussion in Section 6). Our processing pipeline then converts the video data into usable virtual sensor (IMU) data. The procedure is based on a computer vision pipeline that first extracts 2D pose information, which is then lifted to 3D. Through tracking individual joints of the extracted 3D poses, we are then able to generate sensor data, such as tri-axial acceleration values, at many locations on the body. These values are then post-processed to match the target application domain.

Our work aims at replacing the data collection phase of HAR development. It is universal as it does not impose constraints on model training (top center in Figure 1) nor deployment (right part of Figure 1). In what follows, we describe the technical details of our processing pipeline that make videos usable for training IMU-based activity recognition systems. This description assumes direct access to a video that captures a target activity, i.e., here we do not focus on the logistics and practicalities of querying video repositories and curating the search results.

2.2 Motion Estimation for 3D Joints

On-body movement sensors capture local 3D joint motion, and, as such, our processing pipeline aims at reproducing this information but from 2D video. As shown in Figure 2 we employ a two-step approach. First, we estimate 2D pose skeletons for potentially multiple people in a scene using a state-of-the-art pose extractor, namely the *OpenPose* model [10]. Then, each 2D pose is lifted to 3D by estimating the depth information that is missing in 2D videos. Without limiting the general applicability we assume here that all people in a scene are performing the same activity. Although fast and accurate, the *OpenPose* model estimates 2D poses of people on a frame by frame basis only, i.e., no tracking is included which requires post-processing to establish and maintain person

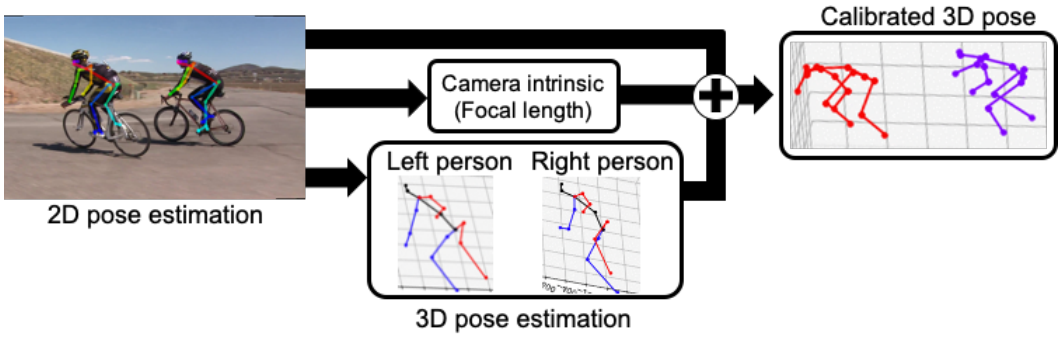

Fig. 2. 3D joint orientation estimation and pose calibration. The multi-person 2D poses are estimated with *OpenPose* followed by lifting to 3D through *VideoPose3D*. The camera intrinsic parameters are estimated using the *DeepCalib* model. Jointly with the pose and camera related parameters, we calibrate the orientation and translation in the 3D scene for each frame.

correspondences across frames. In response, we apply the *SORT* tracking algorithm [7] to track each person across the video sequence. *SORT* utilizes bipartite graph matching with the edge weights as the intersection-over-union (IOU) distance between boundary boxes of people from consecutive frames. The boundary boxes are derived as tight boxes including the 2D keypoints for each person.

To increase the reliability of the 2D pose detection and tracking, we remove 2D poses where over half of the joints are missing, and also drop sequences that are shorter than one second. For each sequence of a tracked person, we also interpolate and smooth missing or noisy keypoints in each frame using a Kalman filter, as poses cannot be dramatically different between subsequent frames. Finally, each 2D pose sequence is lifted to 3D pose by employing the *VideoPose3D* model [63]. Capturing the inherent smooth transition of 2D poses across the frames encourages more natural 3D motion in the final estimated (lifted) 3D pose.

2.3 Global Body Tracking in 3D

Inertial measurement units capture the acceleration from global body movement in 3D, and additionally local joint motion in 3D. Thus, we also have to extract global 3D scene information from the 2D video to track a person's movement in the whole scene. Typical 3D pose estimation models do not localize the global 3D position and orientation of the pose in the scene. Tracking the 3D position of a person in 2D video requires two pieces of information: *i)* 3D localization in each 2D frame; and *ii)* the camera viewpoint changes (ego-motion) between subsequent 3D scenes. We map the 3D pose of a frame to the corresponding position within the whole 3D scene in the video, compensating for the camera viewpoint of the frame. The sequence of the location and orientation of 3D pose is the global body movement in the whole 3D space. For the virtual sensor, the global acceleration from the tracked sequence will be extracted along with local joint acceleration.

2.3.1 3D Pose Calibration. First, we estimate the 3D rotation and translation of the 3D pose within a frame, as shown in Figure 2. For each frame, we calibrate each 3D pose from a previously estimated 3D joint according to the perspective projection between corresponding 3D and 2D keypoints. The perspective projection can be estimated with the Perspective-n-Point (PnP) algorithm [38]. Additionally to 3D and 2D correspondences, the PnP algorithm requires the camera intrinsic parameters for the projection, which include focal length, image center, and the lens distortion parameters [11, 79]. Since arbitrary online videos typically do not come with such metadata, the camera intrinsic parameters are estimated from the video with the *DeepCalib* model [8]. The *DeepCalib* model is a frame-based model that considers a single image at a time so that the estimated intrinsic parameter for each frame slightly differs across the frame according to its scene structure. Hence, we assume that

a given video clip sequence is recorded with a single camera, and aggregate the intrinsic parameter predictions by calculating the average from all frames:

$$c^{int} = \frac{1}{T}\sum_{t=1}^{T} c_t^{int} \tag{1}$$

where $c^{int} = [f, p, d]$ is the averaged camera intrinsic parameters from each frame, x_t at time t, predictions, $c_t^{int} = DeepCalib(x_t)$. $f = [f_x, f_y]$ is the focal length and $p = [p_x, p_y]$ is the optical center for the x and y axis, and d denotes the lense distortion. Once the camera intrinsic parameter is calculated, the PnP algorithm regresses global pose rotation and translation by minimizing the following objective function:

$$\begin{gathered}\{R^{calib}, T^{calib}\} = \arg\min_{R,T} \sum_{i=1}^{N} \|p_2^i - \frac{1}{s} c^{int}(Rp_3^i + T)\| \\ \text{subject to } R^T R = I_3, det(R) = 1\end{gathered} \tag{2}$$

where $p_2 \in \mathbb{R}^2$ and $p_3 \in \mathbb{R}^3$ are corresponding 2D and 3D keypoints. $R^{calib} \in \mathbb{R}^{3\times3}$ is the extrinsic rotation matrix, $T^{calib} \in \mathbb{R}^3$ is the extrinsic translation vector, and $s \in \mathbb{R}$ denotes the scaling factor [98, 101]. For the temporally smooth rotation and translation of a 3D pose across frames, we initialize the extrinsic parameter, R, and T, with the result from the previous frame. The 3D pose for each person, $p_3 \in \mathbb{R}^{3\times N}$, at each frame is calibrated (or localized) with the estimated corresponding extrinsic parameter.

$$p_3^{calib} = R^{calib} p_3 + T^{calib} \tag{3}$$

From the calibrated 3D poses, $p_3^{calib} \in \mathbb{R}^{3\times N}$, we remove people considered as the background. For example, in a rope jumping competition scene, a set of people may rope jump while others are sitting and watching. Depending on the scene, not all people captured may partake in an activity (e.g., bystanders). To effectively collect 3D pose and motion that belongs to a target activity, we thus remove those people in the (estimated) background. We first calculate the pose variation across the frames as the summation of the variance of each joint location across time. Subsequently, we only keep those people with the pose variation larger than the median of all people.

2.3.2 Estimation of Camera Ego-motion. In an arbitrary video, the camera can move around the scene freely. However, the pipeline should not confuse the camera motion with human motion. For example, a person who does not move (much) may appear at a different location in subsequent frames due to the camera movement, which is misleading for our purpose. Also, a moving person can always appear in the center of the frame, and thus erroneously appear static, if the camera follows that person and therefore the movements are effectively compensated for in the video. In these two cases, the virtual sensor should capture no motion (static), or the global body acceleration only, respectively, independently from camera motion. Hence, before generating the virtual sensor data, the 3D poses, which were previously calibrated per frame, need to be corrected for camera ego-motion, i.e., potential viewpoint changes, across the frames.

Camera ego-motion estimation from one viewpoint to another requires 3D point clouds of both scenes [6, 66, 76]. Deriving a 3D point cloud of a scene requires two pieces of information: *i)* the depth map; and *ii)* camera intrinsic parameters. For camera intrinsic parameters, we reuse the parameters previously estimated. The depth map is the distances of pixels in the 2D scene from a given camera center, which we estimate with the *DepthWild* model [25] for each frame. Once we have obtained the depth map and the camera intrinsic parameters, we can geometrically inverse the mapping of each pixel in the image to the 3D point cloud of the original 3D scene. With basic trigonometry, the point cloud can be derived from the depth map using the previously estimated camera intrinsic parameter, $c^{int} = [f_x, f_y, p_x, p_y, d]$. For a depth value Z at image position (x, y), the point cloud value,

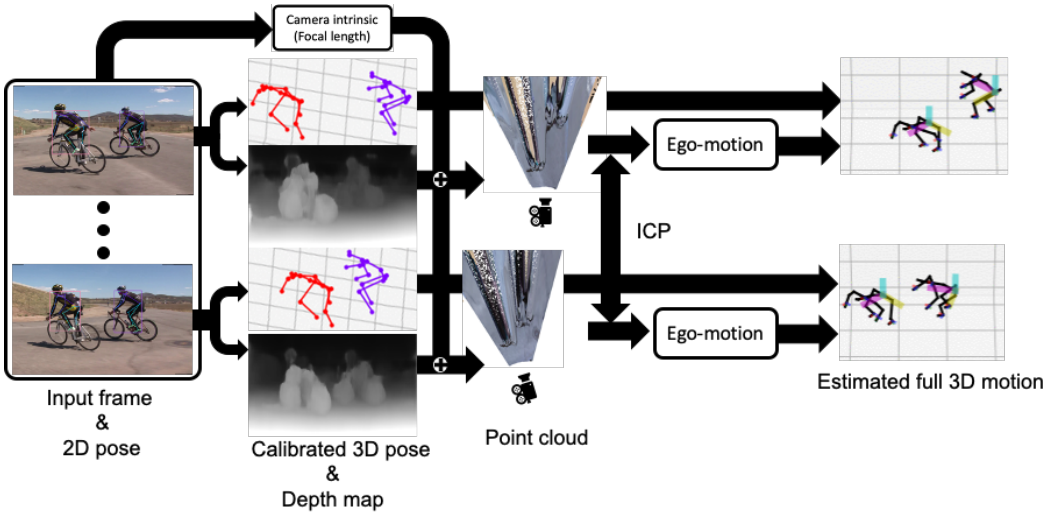

Fig. 3. 3D pose and motion tracking with compensation of the camera motion. The camera motion is estimated through the iterative closest point (ICP) algorithm between subsequent point clouds. Then, calibrated 3D poses per frame are mapped to the location in the entire 3D scene, compensating for camera motion. The calibrated 3D poses from both frames are initially centered in the 3D world origin as the camera follows the cyclists. After incorporating ego-motion information, we can see that two cyclists are moving from right to left, moving closer to each other as in the video (most right figure).

$[X, Y, Z]$, is:

$$[X, Y, Z] = \left[\frac{(x - p_x) \cdot Z}{f_x}, \frac{(y - p_y) \cdot Z}{f_y}, Z\right] \tag{4}$$

Once point clouds are calculated across frames, we can derive the camera ego-motion (rotation and translation) parameters between two consecutive frame point clouds. A popular method for registering groups of point clouds is the Iterative Closest Points (ICP) algorithm [6, 66, 76]. Fixing a point cloud as a reference, ICP iteratively finds closest point pairs between two point clouds and estimates rotation and translation for the other point cloud that minimizes the positional error between matched points [6]. Since we extract color point clouds from video frames, we adopted Park *et al.*'s variant of the ICP algorithm [62], which considers color matching between matched points in addition to the surface normals to enhance color consistency after registration. More specifically, we utilize background point clouds instead of the entire point cloud from a scene because the observational changes for the stationary background objects in the scene are more relevant to the camera movement. We consider humans in the scene as foreground objects, and remove points that belong to human bounding boxes determined from 2D pose detection. The reason for this step is that we noticed that including foreground objects, such as humans, leads to the ICP algorithm confusing movements of moving objects, i.e., the humans, and of the camera. With the background point clouds, we apply the color ICP algorithm between point clouds at time $t-1$ and t, q_{t-1} and q_t respectively. As such, we iteratively solve:

$$\{R_t^{ego}, T_t^{ego}\} = \arg\min_{R,T} \sum_{(q_{t-1}, q_t) \in \mathcal{K}} (1-\delta)\|C_{q_{t-1}}(f(Rq_t + T)) - C(q_{t-1})\| + \delta\|(Rq_t + T - q_{t-1}) \cdot n_{q_{t-1}}\| \tag{5}$$

where $C(q)$ is the color of point q, n_q is the normal of point q. $\mathcal{K}$ is the correspondence set between q_{t-1} and q_t, and $R_t^{ego} \in \mathbb{R}^{3\times3}$ and $T_T^{ego} \in \mathbb{R}^3$ are fitted rotation and translation vectors in the current iteration. $\delta \in [0, 1]$ is the weight parameter that balances the emphasis given to positional or color matches.

The estimated sequence of translation and rotation of a point cloud represents the resulting ego-motion of the camera. As the last step, we integrate the calibrated 3D pose and ego-motion across the video to fully track 3D human motion. Previously calibrated 3D pose sequences, p_3^{calib}, are rotated and translated according to their ego-motion at frame t:

$$p_{3_t}^{track} = R_t^{ego} p_{3_t}^{calib} + T_t^{ego} \tag{6}$$

where $p_3^{track} \in \mathbb{R}^{T \times N \times 3}$ is the resulting 3D human pose and motion tracked in the scene for the video, T is the number of frames, and N is the number of joint keypoints. The overall process of compensating camera ego-motion is illustrated in Figure 3.

2.4 Generating Virtual Sensor Data

Once full 3D motion information has been extracted for each person in a video, we can extract virtual IMU sensor streams from specific body locations. The estimated 3D motion only tracks the locations of joint keypoints, i.e., those dedicated joints that are part of the 3D skeleton as it has been determined by the pose estimation process. However, in order to track how a virtual IMU sensor that is attached to such joints rotates while the person is moving, we also need to track the orientation change of that local joint. This tracking needs to be done from the perspective of the body coordinates. The local joint orientation changes can be calculated through forward kinematics based from the hip, i.e., the body center, to each joint. We utilize state-of-the-art 3D animation software – *Blender* [16], to estimate and track these orientation changes. Using the orientation derived from forward kinematics, the acceleration of joint movements in the world coordinate system is then transformed into the local sensor coordinate system. The angular velocity of the virtual sensor (i.e., a gyroscope) is calculated by tracking orientation changes.

We employ our video processing pipeline on raw 2D videos that can readily be retrieved by, for example, querying public repositories such as YouTube, and combined with subsequent curation (not within the focus of this paper). The pipeline produces virtual IMU, for example, tri-axial accelerometer data. This data effectively captures the recorded activities, yet the characteristics of the generated sensor data will still differ from real IMU data, for instance it will lack any MEMS noise. In order to compensate for this mismatch, we employ the *IMUSim* [95] model to extract realistic sensor behavior for each on-body location. *IMUSim* estimates sensor output considering mechanical and electronic components in the device, as well as the changes of a simulated magnetic field in the environment. As such, this post-processing step leads to more realistic IMU data [4, 42, 64].

2.5 Distribution Mapping for Virtual Sensor Data

As the last step before using the virtual IMU dataset for HAR model training, we define a calibration operation to account for any potential mismatch between the source (virtual) and target (real) domains [13]. We employ a distribution mapping technique to fix such mismatch, where we transfer the distribution of the virtual sensor to that of the target sensor. For computational efficiency, the rank transformation approach [17] is utilized:

$$x_r = G^{-1}(F(X \leq x_v)) \tag{7}$$

where, $G(X \leq x_r) = \int_{-\inf}^{x_r} g(x)dx$ and $F(X \leq x_v) = \int_{-\infty}^{x_v} f(x)dx$ are cumulative density functions for real, x_r, and virtual, x_v, sensor samples, respectively. In our experiments (Section 4.2), we show that only a few seconds to minutes of real sensor data is sufficient to calibrate the virtual sensor effectively for successful activity recognition.

3 TRAINING ACTIVITY RECOGNITION CLASSIFIERS WITH VIRTUAL IMU DATA

We now describe a series of experiments to examine the viability of using IMUTube to produce virtual IMU data useful for HAR. Our first set of experiments consider the performance of virtual IMU data on a HAR dataset providing both video and real IMU data, which enables a fair comparison between virtual and real IMU data. Here,

we see promising results suggesting that training activity classifiers from virtual IMU data alone can perform well on real IMU data. We then move on to show that activity classifiers trained using this virtual IMU data can also perform well on real IMU data coming from common HAR datasets, namely Opportunity [14] and PAMAP2 [72]. Finally, we describe how we curate a video dataset comprising of online videos (e.g., YouTube) in order to extract virtual IMU data for complex activities.

In each experiment, we compare the performances of models on real IMU data (i.e., the test data is from real IMUs), when trained from real IMUs (R2R), trained from virtual IMUs (V2R), or trained from a mixture of virtual and real (Mix2R) IMU data. Throughout our experiments, we consider the Random Forest classifier as our main machine learning back-end for activity recognition, evaluated via leave-one-subject-out scheme. We supplement this primary result by also demonstrating the feasibility to apply deep learning with a hold-out evaluation scheme; in doing so we show our approach is agnostic to the choice of the learning algorithm.

3.1 Feasibility Experiment under Controlled Conditions

There are many potential sources of noise which may impact the activity recognition performance; therefore, in our first experiment we hold constant as many factors as possible. We accomplish this by using the RealWorld dataset [86], an activity recognition dataset that contains not only IMU data but also provides videos of the subjects performing the activities.

Data. The Realworld dataset covers 15 subjects performing eight locomotion-style activities, namely *climbing up, climbing down, jumping, lying, running, sitting, standing*, and *walking*. Each subject wears the sensors for approximately ten minutes for each activity except for jumping (<2 minutes). The video and accelerometer data are not time-synchronized, as each video starts some time (under one minute) before each activity begins. The video is recorded using a hand-held device by the experiment's administrator, who follows the subject as they perform the activity (e.g., running through the city alongside the subject). The videos do not always present a full-body view of the subject, and the video-taker sometimes makes arbitrary changes to the video scene (e.g., he/she might walk past the subject, or rotate the camera from landscape to portrait mode halfway). These factors present extra difficulty in extracting virtual IMU for the full duration of the activities; nonetheless we are able to extract 12 hours of virtual IMU data, this is compared to 20 hours of available real IMU data. As a preprocessing step, we remove the first ten seconds of each video and divide the remainder into two-minute chunks for efficient running of IMUTube. Virtual IMU data are extracted from 7 body locations, i.e. forearm, head, shin, thigh, upper arm and waist/chest, corresponding to where real sensors are placed in Realworld. We assume all IMU data to have a frequency of 30Hz and use sliding windows to generate training samples of duration 1 second and 50% overlap. The resulting real and virtual IMU dataset contains $221k$ and $86k$ windows, respectively.

Method. Our primary evaluations are performed with the Random Forest classifier using ECDF features [28] (15 components), trained using a leave-one-subject-out scheme. On Realworld, we use a train set of 13 subjects, validation set of 1 subject and test set of 1 subject in each fold. This scheme is followed in R2R (where training data is real IMU data), V2R (where training data is virtual, and distribution-mapped only using data from train users), and Mix2R (which contains a mixture of both real and virtual IMU data). We calibrate hyperparameters on the validation subjects by varying the number of trees from 3 to 50 and the minimum number of samples in leaf node from 1 to 50. We report the mean F1-score of the test subjects computed after the completion of all folds.

Separate from this, we train DeepConvLSTM [61] on a hold-out evaluation scheme, where subject 15 is randomly selected as validation, 14 as test, and the rest as the training set. DeepConvLSTM is trained on raw data for a maximum of 100 epochs with an Adam optimizer [43] and early stopping on the validation set with a patience of ten epochs; learning rate is searched from 10^{-6} to 10^{-3}, and weight decay from 10^{-4} to 10^{-3} via grid search. We further regularize model training by employing augmentation techniques from [91] with a probability

Table 1. Recognition results on the Realworld dataset (8 classes) when training models from real IMU data (R2R), from virtual IMU data (V2R), and from a mixture of both (Mix2R). Wilson score confidence intervals are shown. For Random Forest models, V2R achieves 98% of the R2R F1-score, while a Mix2R setup surpasses R2R by 12%.

(a) Random Forest (leave-one-subject-out)

R2R	V2R	Mix2R
0.5779±0.0025	0.5675±0.0025	0.6444±0.0024

(b) DeepConvLSTM (random single-subject hold-out)

R2R	V2R	Mix2R
0.7305±0.0073	0.5465±0.0082	0.7785±0.0068

of application set at either 0 and 0.5 depending on validation set result. We average over 3 runs initiated with different random seeds and report the mean F1-score.

In both cases, we report the highest test F1-score achieved using any amount of training data, along with the Wilson score interval with 95% confidence. We discuss the effect of training set size in Section 4.3. We reuse these settings throughout the paper unless stated otherwise.

Results. In Table 1a, we see convincing evidence that human activity classifiers can learn from virtual IMU data alone. When learning from virtual IMU data alone (V2R), the 8-class model achieves an F1-score of 0.57, which is within 2% of that achieved by learning from real IMU data (R2R). This result is remarkable as the difference in recognition performance of R2R and V2R is small notwithstanding the change in data source and the introduction of noise while going through our pipeline.

Furthermore, when we use a mixture of virtual and real IMU data to train the model, it is even able to surpass R2R performance with a significant relative performance gain of 12%, reaching an F1-score of 0.64. This showcases an additional potential of IMUTube – we can build activity classifiers using both virtual and real IMU data to push recognition capabilities beyond that achieved by either.

Our DeepConvLSTM results (evaluated on a random subject, Table 1b) offers another perspective into modeling virtual IMU data when deep learning models are used. Although learning from virtual IMU data alone is seen to pose more challenges (V2R achieves 75% of R2R), this is possibly related to the setup of learning directly from raw data, in contrast to processed features in the Random Forest case. As a consequence, DeepConvLSTM may be learning feature representations highly specific to the virtual IMU domain, which prevents immediate generalization to real IMU data. This issue is diminished when using a mixture of virtual and real IMU data for training, as Mix2R even outperforms R2R by 6.6%. We presume that the improvement is related to the complementary benefits of both real and virtual data, as well as the feature learning capabilities of deep learning models, which learn better when more data is available.

This set of results provide promising signs for IMUTube – we can learn capable activity classifiers with virtual IMU data alone, despite having only so far considered relatively straightforward techniques in extracting and modeling the virtual IMU data. We delve into these concerns about the quality of virtual IMU data in Section 4 to provide a more complete view.

3.2 Performance on Common Activity Recognition Datasets

We have achieved promising results under the controlled conditions of Realworld, which simultaneously gathers video and IMU together. We now seek to relax these conditions, and establish the viability of IMUTube when the exact actions performed in the video data and the real IMU data do not completely align. Imagine a scenario where we want to build a classifier for 'standing' vs. 'sitting'. Instead of collecting simultaneous video and real IMU data of people standing and sitting, we want to leverage existing videos of people standing and sitting and train the classifier using the derived virtual data.

Table 2. Recognition results (mean F1-score) on Opportunity dataset (4 classes) and locomotion activities found in PAMAP2 (8 classes) when using different training data. For Random Forest models, V2R achieves 95% of R2R F1-scores on average, while Mix2R outperforms R2R by 5% on average.

(a) Random Forest (leave-one-subject-out)

Dataset	R2R	V2R	Mix2R
Opportunity	0.8271±0.0034	0.7757±0.0037	0.8820±0.0029
PAMAP2 (8-class)	0.7029±0.0055	0.6728±0.0058	0.7284±0.0053

(b) DeepConvLSTM (random single-subject hold-out)

Dataset	R2R	V2R	Mix2R
Opportunity	0.8871±0.0074	0.7882±0.0096	0.8838±0.0075
PAMAP2 (8-class)	0.7002±0.0161	0.5524±0.0175	0.7020±0.0161

In the following, we test this scenario by re-using the video data from Realworld and learning models from its virtual data to test on two common HAR datasets, Opportunity and PAMAP2. These datasets are considered as they contain activity labels that roughly correspond to those in Realworld.

Data. We consider activities in Opportunity and PAMAP2 which are overlapping with those in Realworld, i.e., four classes (*stand, walk, sit, lie*) in Opportunity, and eight classes (*ascending stairs, descending stairs, rope jumping, lying, running, sitting, standing, walking*) in PAMAP2. We use 1-second sliding windows with 50% overlap.

For Opportunity, we re-extracted virtual data from the Realworld videos in eleven body positions (left and right feet, left shin and thigh, hip, back, left and right arms, left and right forearms), which resulted in $40k$ and $46k$ real and virtual IMU windows respectively. For DeepConvLSTM, we used random subject 3 for validation, 4 for test, and the rest for training.

For PAMAP2, the activities are slightly different from those in Realworld so we equated the labels with the closest meaning (e.g., using *jumping* Realworld videos as the source for *rope jumping* virtual IMU in PAMAP2. The PAMAP2 dataset specifies that sensors were placed in three locations (dominant wrist, dominant ankle, chest), which gives rise to a total of four possible combinations for arm and chest location when we extract virtual IMU data from a single video (i.e., left-left, right-right, left-right, right-left). We took advantage of this ambiguity and extracted 4× as much virtual IMU per video, resulting in $24k$ and $152k$ windows for real and virtual IMU respectively. For DeepConvLSTM, we followed the same setup as [29] and use subject 5 for validation, 6 for test, and the rest for training.

Results. Table 2a shows the classification performance for R2R, V2R and Mix2R. We observe encouraging results, where learning from virtual IMU data alone can recover high levels of R2R performance, despite data collection conditions not being held constant. A Random Forest classifier trained from virtual IMU data achieves 94% and 96% of R2R performance on Opportunity and PAMAP2 respectively. While this good performance might be related to the simplicity of the motions classified (mainly locomotive activities), we highlight that the conditions of data collection in Realworld and Opportunity are very different–subjects could be walking through the forest or city in Realworld, but all subjects perform activities inside a laboratory in Opportunity; Likewise for Realworld and PAMAP2–subjects could also be climbing down the streets of a city (a mixture of pavement and stairs) in Realworld whereas all subjects are climbing up the same building in PAMAP2. Thus, being able to utilize virtual data from one scenario and test it on another is not a trivial task. These results suggest that, on these two tasks, virtual IMU data can provide salient features that are generalizable and robust across testing scenarios.

Table 3. Recognition results (mean F1-score) on PAMAP2 (11-classes) when using different training data. The Random Forest model trained with virtual IMU data including YouTube videos (for four complex activities) recovered 80% of R2R model performance. For Mix2R, additional real IMU data helped the Random Forest model increase performance up to 98% of R2R model performance.

(a) Random Forest (leave-one-subject-out)

R2R	V2R	Mix2R
0.7225±0.0044	0.5792±0.0049	0.7111±0.0044

(b) DeepConvLSTM (random single-subject hold-out)

R2R	V2R	Mix2R
0.6977±0.0129	0.5326±0.0140	0.7095±0.0128

We also observe performance gains when training with a mixture of real and virtual IMU data, which exceeds R2R F1-scores by 5% on average. Not only does this observation solidify the argument that virtual and real IMU data can bring complementary benefits to activity recognition, but such performance gains are also a positive sign especially since the two types of data are collected under rather different circumstances. We argue that, adding virtual IMU data – in this case, virtual data generated from a related different scenario – can help expand the variety of motions seen by the classifier and as a result improve model generalization.

As before, we provide the performance by DeepConvLSTM on a random test subject as additional results in Table 2b, where V2R recovers 84% of R2R F1-scores, while Mix2R and R2R scores are statistically comparable.

Overall, this set of results presents strong evidence supporting the usefulness of virtual IMU data, either used standalone or in combination with real IMU data for activity recognition. Beyond this, these results also imply an encouraging view that aligns well with our vision for IMUTube – that virtual data, even when collected under vastly different settings, can be useful in building capable or even better models for activity recognition.

3.3 Virtual IMU Data for Complex Activity Recognition

Encouraged by the results so far, we now try to apply IMUTube onto activity recognition scenarios with even more challenging conditions and test its ability in building classifiers for complex activities. Our ultimate vision for IMUTube is to extract virtual data from any video, especially those freely available in large online repositories such as YouTube. To test the feasibility of doing so, we first need to curate a dataset with activity videos originating from the web. In the following, we attempt to source these videos for complex activities present in PAMAP2, and train classifiers with the extracted virtual IMU data.

Data. We curated a dataset of virtual data covering four complex activities present in PAMAP2, namely *vacuum cleaning, ironing, rope jumping* and *cycling*. To efficiently locate such videos, we extract annotated video segments from activity video datasets in the computer vision domain, including ActivityNet [9], Kinetics700 [12], HMDB51 [44], MPIIHPD [3], UCF101 [84], Charades [83], AVA [26], MSRdailyactivity3D [49], and NTU RGB-D [51]. The resulting video dataset consists of a mix of videos collected in experiment scenarios and in-the-wild (e.g., from YouTube). In total, we collected ~ 10 hours of virtual data from 7,255 videos. To extend our activity recognition task to as many classes in PAMAP2 as possible, we also reuse the other seven videos from Realworld (we do not use the *jumping* videos); this allows us to consider an 11-class activity recognition problem in PAMAP2. As mentioned for the PAMAP2 (8-class) task, we face an ambiguity in sensor location which led us to extract 4× virtual data per video. Using sliding windows of 1-second size and 50% overlap, resulted in 38k real and 390k virtual IMU windows in total.

Results. For these challenging conditions (using in-the-wild videos, learning complex activities), Table 3a shows that virtual IMU data can still be useful for training activity classifiers. With the Random Forest classifier, training from virtual IMU data alone achieves a 0.58 F1-score under V2R, which is 80% of that achieved with R2R (0.72

F1-score). This is a weaker result compared to those achieved on previous datasets (where V2R achieved 96% of R2R on average). However, this is because there is an even more drastic difference in the data sources and activity label interpretations between the real and virtual IMU data. Another likely factor causing the weaker performance is the quality of virtual IMU data that IMUTube is currently able to produce, which might be amplified by the complex activities introduced in this experiment. In Section 4.1 we will examine the fidelity of virtual IMU data, and provide directions to improve its quality in Section 6.2. Finally, one must consider as a factor the greater domain shift that is probably present between train and test scenarios, which we will discuss in Section 4.2.

When real IMU data is added to virtual IMU data for training, the Random Forest model gains 23% and achieves a F1-score of 0.71 in Mix2R versus 0.58 in V2R, though Mix2R is still 2% worse than R2R performance. We believe these results are related to the domain shift within the training data. To better cope with the scenario where we want to make use of both real and virtual IMU data, we investigate the effect of mixing data in Section 4.3 and investigate more sophisticated methods beyond simple mixing in Section 5. Our evaluation under DeepConvLSTM is shown in Table 3b, and results align with those of the Random Forest.

Through this set of experiments, we have demonstrated that, despite very challenging conditions–in-the-wild videos and complex activity recognition, it is still feasible to learn capable classifiers using virtual IMU data. The results overwhelmingly support that virtual IMU data generated via IMUTube are useful for even real-world instances of activity recognition. Demonstrated over a range of locomotion and more complex activities, virtual IMU data is seen to effectively capture motion information, such that classifiers can be trained from them alone and still perform well on real IMU data. In addition, mixing real and virtual IMU data for training is also shown to be a potential source of performance gain.

4 UNDERSTANDING VIRTUAL IMU DATA

Across multiple datasets, the model trained on the virtual IMU dataset (V2R) performed well on the real IMU test datasets. The V2R performance varies between 80% - 90% compared to R2R models, and only matches or outperforms R2R when trained alongside real IMU data (Mix2R). Although notably, for the experiment on PAMAP2 (11-class), the V2R model could not outperform R2R even when trained with the larger virtual dataset extracted from multiple video sources. Thus, in this section, we investigate the potential sources of such performance gaps in detail. First, we analyze the extracted virtual IMU data by inspecting the sample-level similarity in IMU signals using synced sequences of real and virtual IMU data. Then, at a distribution level, we investigate the effects of domain shift, along with the impact of our distribution mapping technique (Section 2.5). Finally, we investigate the mixing of real and virtual IMU data for model training (Mix2R), which was seen to give comparable, if not superior, performance relative to R2R in Section 3. Through our analysis, we aim to provide key insights into the IMUTube pipeline and the use of virtual IMU data for human activity recognition. All experiments presented in this section are carried out using Random Forest in the leave-on-subject-out setting unless otherwise specified.

4.1 Comparing Virtual and Real IMU

We do not expect IMUTube to function flawlessly. Given the complexity of the process, the translation from video to virtual IMU data will naturally contain errors. Despite this, we observe promising results of competitive V2R performance in the prior section. This seems to suggest that perfect sample-level realism in the virtual IMU data is not necessary to train capable human activity classifiers. In the following, we compare virtual and real IMU samples to better understand the limits of IMUTube and argue that, the focus, during virtual IMU generation, should be placed on capturing salient features useful for activity recognition.

Method. Sample-level comparison between the virtual and real IMU data requires a dataset with time-synchronized video and IMU sensor data. Although the Realworld dataset contains both accelerometer and video data, these modalities are not synchronized (as mentioned in Section 3.1). Therefore, in this experiment, we introduce the

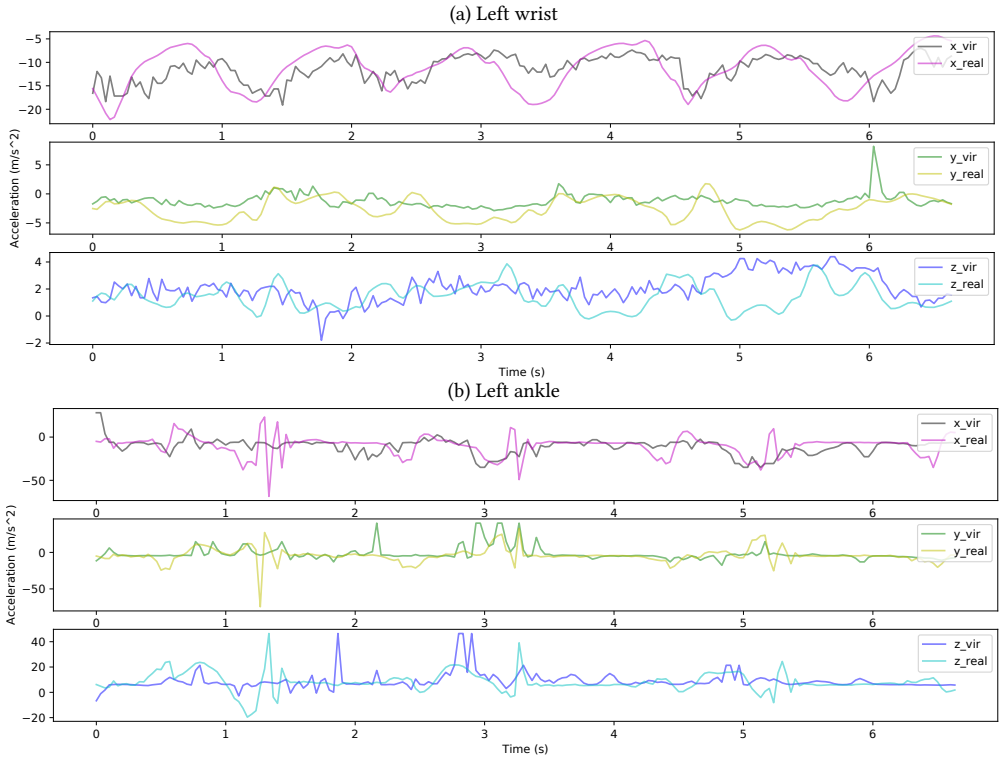

Fig. 4. Comparison between virtual and real IMU on the TotalCapture dataset. Distribution mapping has been applied to the virtual IMU data.

TotalCapture dataset [90] which contains time-synchronized (real) IMU data and video recordings (from which the virtual IMU data are extracted). As TotalCapture contains various scripted motions but not labels that are immediately useful for activity recognition-related tasks, we did not evaluate this dataset in Section 3.

Analysis. Figure 4 shows an example of the virtual and real IMU time-series data of a subject walking, with sensors placed on their wrist and ankle. Along the x-axis, virtual IMU readings are seen to reflect large movement changes also observed in the real IMU—one can almost see from the 'wrist' time-series (see Figure 4(a)) that the person is walking with periodic hand movements. Along the z-axis, the virtual IMU is also seen to capture any spikes in acceleration reasonably well, albeit with a noticeable time lag in the 'ankle' case. Virtual and real IMU data differ the most along the y-axis. We postulate that this is related to a dimensionality issue—we are trying to reconstruct 3-D information from a 2-D image time series. The y-axis here refers to the axis pointing perpendicular to the visual plane, which means any acceleration measured along this dimension cannot be easily deduced visually.

While generating realistic virtual IMU data is important, it is secondary to our main goal of producing virtual IMU data that captures useful information for HAR tasks. To achieve this, what is vital is the ability of the virtual IMU data to capture salient features of the activities that we need to recognize. We already see signs of this happening with the current IMUTube (e.g., the x-axis of the 'arm' while walking in Figure 4). This also offers a possible explanation for the better V2R performance seen in predicting locomotion-style activities in Section 3.

Table 4. Recognition results (mean F1-score, Random Forest) on the 4-class activity recognition task from Opportunity when using training data from other data sources (top rows) and from Opportunity itself (last row, provided for reference). Without distribution mapping, there is a significant drop in performance when using training data collected under different circumstances than the test case, regardless of whether the IMU data is real (66%) or virtual (64%). This is resolved when distribution mapping is applied.

Train data source	Without mapping	With mapping
Virtual data	0.2949±0.0041	0.7757±0.0037
PAMAP2	0.2770±0.0040	0.6931±0.0041
Realworld	0.2828±0.0041	0.6637±0.0043
Opportunity	0.8272±0.0034	-

Perhaps IMUTube, in its current form, is best suited to capture information about simple motions (i.e., ones mostly characterized by movement in a 2D plane) of which there is still a wide variety, and to which existing HAR methods still struggle to generalize [46] (for additional qualitative observations see Section 6.2). To apply IMUTube to more complex activities, it may require improved techniques during virtual IMU data generation (also discussed further in Section 6.2).

4.2 Coping with Domain Shift

The last step of our pipeline performs a distribution mapping post-processing step between virtual and real data (Section 2.5). Applying some form of distribution mapping is necessary due to the presence of *domain shift* between training and testing data. This domain shift is not exclusive to extracting virtual sensor data from videos, but it is also present whenever data is taken from different tasks (or datasets) which result in dissimilar data distributions between training and testing [13].

Method. Our first experiment is to compare the recognition performance on the Opportunity dataset with models trained using data from sources other than Opportunity, with or without distribution mapping. Specifically, the train data can be *i)* real IMU data from PAMAP2, *ii)* real IMU data from Realworld, or *iii)* virtual IMU data from Realworld videos (Section 3.1). Without distribution mapping, we use all available data in the respective datasets that fall under the 4 Opportunity classes (stand, walk, sit, lie) for training, and test using the entire Opportunity dataset. With distribution mapping, we follow a leave-one-subject-out evaluation scheme; In each fold, we train the model using data distribution-mapped with data only from the corresponding train subjects in Opportunity, and evaluate on the remaining test subject.

In our second experiment, we focus on the virtual and real IMU used in the 4 datasets described in Section 3, i.e., Realworld, Opportunity, PAMAP2 8-class and 11-class. We aim to understand how much real IMU data is needed for distribution mapping on the virtual IMU data. To do so, we vary the amount of real IMU data used for distribution mapping and evaluate at what point do the virtual and real IMU data distribution become sufficiently similar. We report the similarity between each data distribution using the Frechet Inception Distance (FID), a metric commonly used in generative modeling to compare the real and generated datasets [33, 53]; lower FID scores indicate more similar data distributions. We also report the confidence interval as calculated by randomly sampling real IMU data with 10 different random seeds for distribution mapping.

Analysis. In Table 4, the effects of domain shift are demonstrated by the significant drop in the performance seen in the 'without mapping' column. When using training data not from Opportunity – despite having the same activity labels – even models trained with real IMU data (from PAMAP2 and Realworld) suffer a 66% drop in F1-scores. From this, it is clear that the domain shift issue is not exclusive to the shift present between virtual and

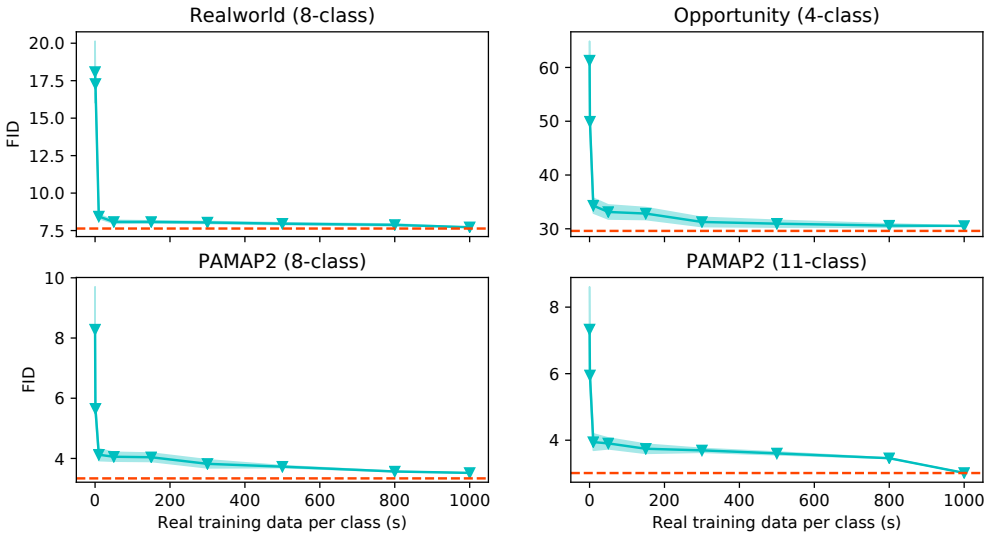

Fig. 5. Frechet Inception Distance (FID) and confidence interval (shaded area) between virtual IMU and real IMU data distributions after performing distribution mapping of the virtual IMU data with increasing amounts of real IMU samples. The dotted horizontal line is the FID score obtained when the entire real IMU dataset is used for distribution mapping.

real IMU domains. The drop in performance is resolved when we perform distribution mapping (Section 2.5); we even observe that training from virtual data outperforms training using other real IMU datasets. This hints that virtual data might have greater value than real IMU data in developing general HAR models. This conclusion was also supported by the results seen when performing the same analysis on the PAMAP2 and Realworld datasets.

Figure 5 shows how the virtual/real FID score varies with the quantity of real IMU data used for distribution mapping. In all four cases, an abrupt, significant drop in the FID score is seen with the use of under 100 seconds of real IMU samples. When using 10 minutes of real IMU data per class for distribution mapping, the FID scores are within 6% of the final FID score (when all real IMU is used for distribution mapping).

4.3 Varying the Mixture and Size of Training Data

Here, we inspect how varying the mixture and size of training data affects recognition performance on the four datasets considered in Section 3.

Method. Our first experiment compares the F1-scores achieved by models trained with a mixture of real and virtual IMU data (fixed at 1:1 ratio) as the amount of training data is varied. We repeat this on all 4 datasets and plot the respective learning curves to inspect if the performance gain by Mix2R over R2R is consistent. Our second experiment compares the F1-score achieved by models trained with a mixture of real and virtual IMU data, where the real IMU data is fixed at 300 seconds per class, but real-to-virtual data ratio is varied from 1:1 to 1:10.

Analysis. Figure 6 shows the learning curves on each dataset as the amount of training data is varied. For 3 out of 4 datasets, Mix2R outperforms R2R consistently by a clear margin at every inspected point of the learning curves. The greatest performance gain is observed throughout the Realworld learning curve, with an increase of at least 9% in F1-score by Mix2R over R2R. Similar trajectories are observed on Opportunity and PAMAP2 (8-class), with

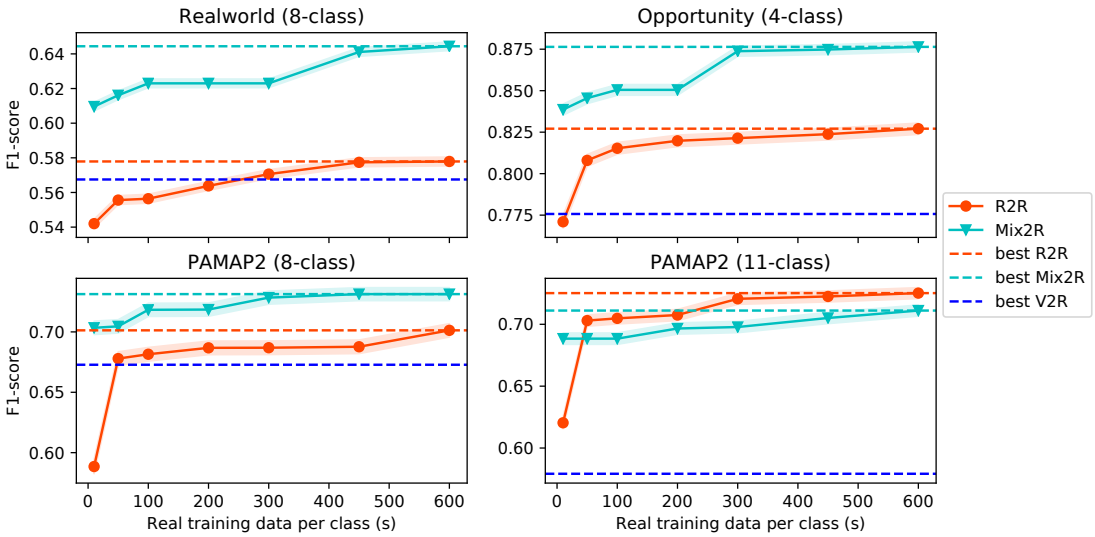

Fig. 6. Mix2R and R2R performance of a random forest model on 4 different HAR tasks when different amounts of real data per class (in seconds) are used for training. The ratio of virtual data and real data is kept at 1:1 at all datapoints.

the most significant difference between Mix2R and R2R occurring when very limited training data are available (under 100 seconds per class).

On PAMAP2 (11-class), Mix2R outperforms R2R when there are only 10 samples per class, and both Mix2R and R2R curves plateau when there are more data available. Given that PAMAP2 (11-class) is also the case where we predict complex activities under the most dissimilar settings, the plot highlights the difficulty of the classification task for both real and virtual IMU data.

Although it may appear that the learning performance saturates with relatively small amount of data per class (600 seconds per class, for instance) – we highlight that this has been commonly observed in the literature for the HAR datasets we used (Opportunity, PAMAP2, e.g., [31, 73, 99]).

Next, we evaluate the performance achieved when varying virtual and real IMU data mixtures, as presented in Table 5. When compared to the F1-scores of models only trained from real data, adding virtual data to training is seen to give a better or comparable performance at all considered ratios on Realworld, Opportunity, and PAMAP2 (8-class). On Realworld, the greatest gain is seen at 1:5, where the F1-score is increased by 9% over that at 1:0; At the ratio 1:1, both Opportunity and PAMAP2 (8-class) see improvements of 6%. The performance however does not increase monotonically with the addition of more virtual data. This shows that the effect of mixing virtual and real data is not straightforward. It is possible that as more virtual IMU data is used, the domain shift issue becomes severe and the Random Forest classifier starts to overfit to the virtual IMU data. Adding virtual data has a detrimental effect on PAMAP2 (11-class). This follows our observations in Figure 6 and can be similarly explained by PAMAP2 (11-class) containing complex activities under the most dissimilar settings in comparison to the virtual IMU data.

Hence, we suggest finding the right balance between the amount of real and virtual IMU data for a model to learn the target activity pattern coexisting in both real and virtual IMU data, before overfitting to the virtual IMU

Table 5. Recognition results (mean F1-score, Random Forest classifier) on all datasets. Different amounts of virtual data are added to a constant amount of real data, given in seconds per class.

Real Data	Virtual Data	Real : Virtual	RealWorld	Opportunity	PAMAP2 (8-class)	PAMAP2 (11-class)
300	0	1:0	0.5706±0.0025	0.8214±0.0034	0.6869±0.0056	0.7206±0.0044
300	300	1:1	0.6230±0.0025	0.8738±0.0029	0.7284±0.0053	0.6978±0.0045
300	600	1:2	0.6146±0.0025	0.8637±0.0030	0.7006±0.0055	0.7051±0.0045
300	1500	1:5	0.6247±0.0024	0.8503±0.0032	0.6926±0.0055	0.6898±0.0045
300	3000	1:10	0.6061±0.0025	0.8396±0.0031	0.6792±0.0056	0.6824±0.0046

data. We also anticipate that as the quality of virtual IMU improves in future versions of IMUTube, that larger amounts of it will be able to be successfully integrated during HAR training.

5 TRANSFER LEARNING WITH VIRTUAL IMU DATA FOR HAR CLASSIFIERS

In the previous two sections, we have demonstrated that sensor-based human activity classifiers can learn from virtual IMU data, although limitations still exist. So far, we have assumed that labeled virtual and real IMU datasets for target activities are always available. In practice, such a scenario may not always be possible. For example, curating video datasets for virtual IMU data could be challenging, as titles or descriptions of videos can be arbitrarily ambiguous.

Here, we explore two additional cases for utilizing virtual IMU data: *i)* when the virtual IMU dataset contains a subset of target activity labels; *ii)* when labels for virtual IMU are not available at all. To do so, we leverage two transfer learning setups, *supervised* and *unsupervised*, respectively. The analysis in this section represents our first attempts in utilizing more sophisticated modeling techniques from deep learning to extend the contribution of IMUTube. Our results are a first step towards handling realistic issues in label collection, as we do not yet incorporate any automated video labeling or search mechanisms. All experiments follow the same hold-out evaluation protocol detailed in Section 3.

5.1 Supervised Transfer Learning

With supervised transfer learning, we pre-train a model using labeled virtual IMU data and fine-tune it using labeled real IMU data. Importantly, the labels for pre-training and fine-tuning need not match. We first explore the setup where virtual and real IMU data share the same set of activity labels – Imagine we have already curated video and virtual IMU data for some targeted activities, and also collected a small amount of real IMU data; instead of waiting until sufficient amounts of real IMU data is collected, we can first train a model on the virtual IMU data and fine-tune on the small-scale real IMU data. By studying this scenario, we can also gauge if pre-training with virtual data might provide any benefits to activity recognition performance.

Next, we consider a scenario where the virtual IMU data only contains a subset of the real IMU data activity classes. To examine this, we pre-train a model on the virtual PAMAP locomotion (8-classes) task and fine-tune it on the complex activities (11-classes) tasks.

Method. We compare the recognition performance of DeepConvLSTM models *i)* trained only with real data and *ii)* pre-trained on virtual and fine-tuned on real IMU data. The former is the same as the R2R case in Section 3. For *ii)*, we randomly split the virtual IMU data into train/validation/test (80%/10%/10%) and pre-train the network using the virtual IMU training data. During fine-tuning, all model weights are updated and we report the performance on the real IMU test dataset; In the case where real and virtual IMU activity labels do not match, we replace the last layer of the pre-trained model with the target number of classes (thereby going from 8 to 11 activity classes

Table 6. Recognition results (mean F1-score) of transfer learning setups when evaluated on different HAR tasks. R2R is the baseline trained on real data from scratch. Transfer learning (TL) results show the performance of the models fine-tuned on real data.

		DeepConvLSTM		CAE+RF	
Pre-training	Fine-tuning	Supervised R2R	Supervised TL	Unsupervised R2R	Unsupervised TL
Realworld	Realworld	0.7305±0.0073	0.8337±0.0061	0.7923±0.0067	0.7718±0.0069
Opportunity	Opportunity	0.8871±0.0074	0.9100±0.0067	0.8896±0.0074	0.8477±0.0084
PAMAP2 (8-class)	PAMAP2 (8-class)	0.7002±0.0161	0.7137±0.0159	0.6471±0.0168	0.6809±0.0164
PAMAP2 (11-class)	PAMAP2 (11-class)	0.6977±0.0129	0.7023±0.0129	0.7004±0.0129	0.6989±0.0129
PAMAP2 (8-class)	PAMAP2 (11-class)	-	0.7071±0.0129	-	-

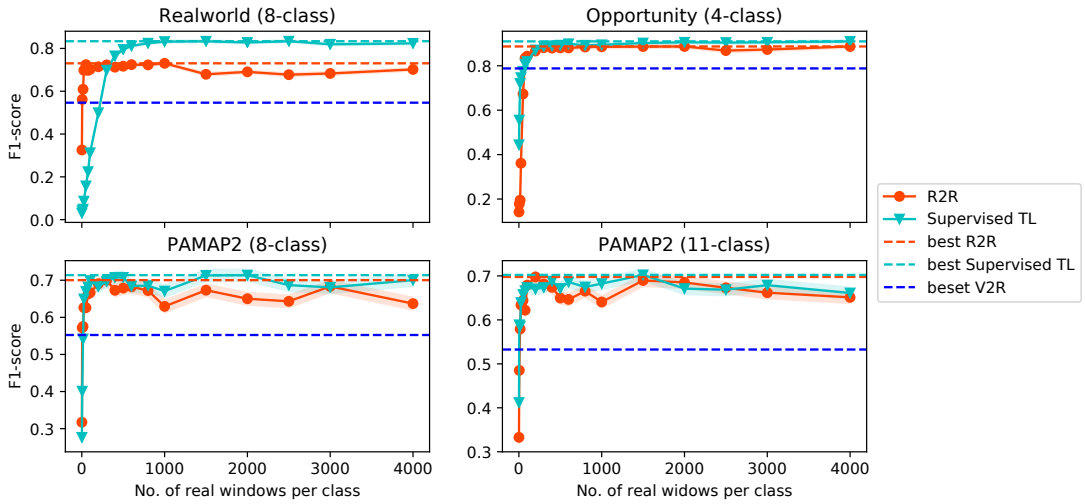

Fig. 7. Transfer learning vs. amount of real data used for training.

for PAMAP2) and update all the network weights. We present results on all 4 datasets and follow the training and evaluation protocols described in Section 3.

Results. The left of Table 6 shows the differences in F1-scores achieved by DeepConvLSTM models with (Supervised TL) and without pre-training (Supervised R2R) when evaluated on a random test subject. With pre-training, statistically significant performance gains are observed on Realworld and Opportunity, at 14% and 3% respectively. We further present the effects of using different amounts of real IMU data for fine-tuning the selected base model in Figure 7. We see the most obvious difference in learning trajectories on the Realworld dataset, where only a small amount of real data is needed to fine-tune the base model such that it surpasses R2R performance. In the last row of Table 6, we also show results for transfer learning in the case where virtual and real IMU data labels do not match (PAMAP2 8-class for pre-training, PAMAP2 11-class for fine-tuning). In this case, the model with pre-training achieves an F1-score of 0.71 on PAMAP2 (11-class), which is statistically comparable to the result in the R2R setting.

5.2 Unsupervised Transfer Learning

Unsupervised transfer learning considers the scenario where we extract the virtual IMU data from a large body of videos without labels. Curating a collection of unlabeled videos is easier relative to obtaining labeled videos, particularly in scenarios where the video descriptions/labels may be unreliable. Without a set of specific target activities in mind, any videos with humans can be utilized. Validating the feasibility of this approach represents a first step towards curating a large collection of virtual IMU data, consisting of very diverse movements and activities from which a model can learn generic representations.

Method. Our unsupervised transfer learning setup consists of two stages. The first stage pre-trains a convolutional autoencoder (CAE) to learn feature representations. The second stage extracts from real IMU data the learned representations, which are then used to train a random forest classifier. We compare the recognition performance achieved by the entire CAE-RF setup when: *i)* virtual IMU data is used for training the CAE (Unsupervised TL); and *ii)* when real IMU data is used for training the CAE (Unsupervised R2R).

We use Haresamudram et al.'s architecture, where the encoder contains four convolutional blocks, leading to the bottleneck layer [31]. Each block contains two 3x3 convolutional layers followed by 2x2 max-pooling. Batch normalization is applied after each layer [37]. The output from the last convolutional block is flattened before being connected to the bottleneck layer. The decoder inverts the encoder by performing convolution, interpolation, and padding to match the sizes of the corresponding encoder blocks [58]. ReLU activation [57] is used throughout, except the output, where the hyperbolic tangent function is used instead. We follow the evaluation protocol used in the DeepConvLSTM case.

Results. The right part of Table 6, shows results for using virtual IMU for pre-training with varying performance relative to models trained on real IMU data. On Realworld, Opportunity and PAMAP2 (11-class), unsupervised TL using virtual IMU data reached up to 95% – 100% of R2R F1-scores. On PAMAP2 (8-class), we even see an increase of 5% over the R2R protocol. These results demonstrate the feasibility in utilizing virtual IMU data even in scenarios where video labels are completely absent.

6 DISCUSSION

In this section, we discuss the implications of the results presented, limitations in our approach, and highlight opportunities which this work opens up.

6.1 Demonstrating Feasibility

We have presented a processing pipeline and a series of validation studies to support our thesis that an automated pipeline from video to virtual IMU data can replace the labor-intensive practice of collecting labeled datasets from real on-body IMU devices. IMUtube shows how a full three-axis virtual accelerometer sensor derived from arbitrary videos can be utilized for human activity recognition. The automated pipeline provides the opportunity to collect much larger labeled data sets, which in turn can improve classifiers for human activity recognition.

Our validation experiments explored ways to model virtual IMU data, either standalone or in conjunction with real IMU data. On three different datasets (Realworld, Opportunity and PAMAP2 8-class), training from virtual IMU alone led to competitive results compared to those from real IMU data (recovering up to 90%), and a simple mixing of data from two sources brought considerable gains (4%-12% increase) to recognition performance.

PAMAP2 11-class is a special case as the activity recognition task extends to complex, non-locomotion human activities, such as vacuum cleaning. It is also different because we have utilized a diverse range of video data collected over multiple visual datasets. Our results show that we can indeed still learn from virtual data under such settings, and our V2R results still reach at least 80% when compared to R2R (Table 3a). However, what is still missing is that we have not seen an improvement over R2R results through mixing (2% decrease) or

transfer learning (insignificant change). While this suggests that modeling complex activities and mixed data sources remain issues, we believe that modelling complex activities, and by extension, the merit of using more accelerometry data (be it virtual or real) still warrants further investigation. For example, is there simply an upper bound to predicting these complex activities using motion-based data alone?

6.2 Limitations and Extensions of Current Approach

We have utilized a series of off-the-shelf techniques at every step of the proposed pipeline in Section 2. While this supports reproducibility of our results, it does result in limitations that impact the overall quality of labeled data for HAR. We discuss the known limitations of each step of the pipeline and present steps forward to advance this line of research.

6.2.1 From Vision To Pose. Accurate recovery of the human skeleton pose from videos has known limitations arising from the movement of both the subjects as well as the camera.

The 2D pose estimator used in IMUTube, *OpenPose*, has previously known failure scenarios including partial detection of joints, swapping between left and right for rare poses, self-occlusion from the camera viewpoints, and partially visible bodies [10]. Such errors in the estimated 2D pose propagate to the 3D pose estimation, which itself is a challenging problem due to the inherent uncertainty of the added third dimension [63]. Erroneous 2D and 3D poses may distort corresponding perspective projections (PnP) between the poses leading to wrong 3D pose calibration [38]. Depth map or camera ego-motion estimation for a dynamic scene can be imprecise when having occlusions or motion blur between foreground and background objects, or when light condition changes [25]. Therefore, the current pipeline can result in distorted 3D motion due to the accumulated errors, since these challenges are common for videos in the wild.

For the recovery of the human skeleton pose, we would expect improvements based on solutions that leverage more sophisticated pose tracking techniques that are more robust to vigorous movement, a change of scenery, the presence of multiple people, and occlusion. Also, camera movements relative to the people captured in the videos could come from the instability of the camera (e.g., for hand-held cameras) as well as video filming techniques (e.g., panning shots). Specialized video stabilization strategies or camera ego-motion techniques can address these issues [80, 96, 100]. We believe that the application of these techniques (and others not yet mentioned or even developed) will further improve pose extraction quality and expand the variety of videos that can be treated as input to our pipeline.

6.2.2 From Pose To Accelerometry. Our current approach assumes an equivalence between acceleration measured by a device on the wearer's body with that measured at the nearest body joint. This view discounts any consideration of factors such as body mass, device movement and skin friction. To better model the on-body location of IMU devices, utilizing techniques from body mesh modeling is a straightforward solution to increase realism to the pipeline. We foresee that investigating the use of body mesh might also bring up the possibilities of synthesizing credible accelerometry data from people of different body shapes from the movement of a single human skeleton pose [39, 52, 68]. In addition, while we have only considered the generation of virtual accelerometry data in this work, we can adapt most parts of the pipeline to generate the full set of IMU signals, including gyroscope and magnetometer readings.

6.2.3 From Accelerometry to Virtual IMU. Real IMU data, which have been the basis of building HAR classifiers, are not free of noise. Sensor noise may come from factors such as drift, hysteresis and device calibration. To carry over such characteristic sensor noise on our virtual data, domain adaptation techniques can be deployed as well as more sophisticated techniques like Generative Adversarial Networks. [24, 75]

6.2.4 Learning from Virtual Data. A domain shift exists when a machine learning model is trained from virtual data and tested on real IMU data. Again, domain adaptation strategies to the input of the machine learning model is a solution. Alternatively, it will be promising to investigate domain-invariant features learned from virtual and real data, which could potentially lead to performance gains in HAR.

6.2.5 Which Videos are Currently Suitable for IMUTube? Our qualitative inspection of the IMUTube output and the per-class V2R results suggests that certain video features may contribute to poorer recognition performance on virtual IMU data: large ego-motions, multiple moving objects and people, and occlusion.

Large ego-motions can be found in the Realworld 'running' videos in which the video-taker was also running, leading to significant vertical shaking motion from the camera itself. It is possible that such vertical motions end up producing features that are very similar to those from a 'jumping' motion, which may explain a higher class confusion observed between 'running' and 'jumping' on the Realworld and PAMAP2 (8-class) tasks. An additional factor might have come from the presence of multiple moving objects and people in the background (e.g. pedestrians) in the 'running' videos.

We also found that V2R models struggle more in classifying activities with similar poses which have more subtle differences in limb movements, e.g. 'standing' vs. 'vacuum cleaning', 'sitting' vs. 'ironing', as in PAMAP2 (11-class). In many 'vacuum cleaning' and 'ironing' videos, the subject's arm movements are occluded by objects in the scene, e.g. clothes or home furniture.

On the other hand, videos with fewer or without such motion artifacts tend to produce virtual IMU data that are well-classified under V2R. Moreover, videos featuring activities with distinctive poses and motions, e.g. cycling, are well-classified under the V2R setting. There are also many existing techniques that will allow us to further tackle motion blur ([1, 81]) and occlusion ([23, 71]). It is also possible that the future curation of video data can automatically rank videos by the presence of these undesirable features to arrive at a suitable dataset for virtual IMU data extraction.

6.3 The Road Ahead

Our primary goal in this paper was to motivate the HAR community with a promising approach that overcomes the main impediment to progress—lacking large labeled data sets of IMU data. While technical challenges remain, we have validated this approach and provide a processing pipeline that the community can collectively develop. Here we highlight the most compelling research opportunities.

6.3.1 Large-scale Data Collection. The ultimate goal, as suggested by the name of IMUTube for our initial tool, is to develop a fully automated pipeline that begins with the retrieval of videos representing particular human activities from readily available sources (e.g., YouTube) and converts that video data to labeled IMU data. Since it is much more common to have video evidence of the wide variety of human behaviors, this is an obvious advantage over past labor-intensive and small-scale efforts to produce such HAR datasets. We have shown great promise with this direction, and above listed some known limitations that can be addressed by different vision, signal processing, and machine learning techniques. The reader will note that the videos used for our validation studies were also curated, meaning there was a significant effort in selecting appropriate video examples. The hope is that this curation effort can also be reduced and ultimately eliminated because the sheer number of relevant videos will overcome the deficiencies of less useful video data.

6.3.2 Deep Learning. Deep Neural Networks have transformed recognition rates in other fields [32, 60], but HAR has lagged behind, again due to the lack of large corpora of labeled data. While we expect that IMUTube is a significant advance towards that goal, having the data alone is not the end goal. We have not yet produced a large-scale HAR dataset, and until we do so we can only hope that deep learning techniques will take over. We then fully expect HAR to inform deep learning techniques.

6.3.3 Extending the Field of HAR. An important advantage to generating virtual IMU data is that we can place the virtual sensor in a wide variety of places on the human body. While some of the standard datasets we used in this work have subjects wearing multiple IMUs, there are limits to how many devices one can wear and still perform activities naturally. IMUTube removes that limitation. Now, for any given activity, we can experimentally determine where to place one IMU (or multiple IMUs) to best recognize that activity. For a set of activities, which place optimizes the recognition of all of the activities in that set. We have never had the ability to contemplate that kind of question. We also need not limit to IMUs placed directly on the body. Models of how clothing responds on a body might be used to generate virtual IMU data for objects that are loosely connected to the body [2, 67]. HAR can now inform clothing manufacturers of where in the material for a shirt, for example, one would want to integrate IMU data collection to predict the activities of the person wearing the shirt, or any other piece of clothing for that matter [41, 56].

6.3.4 Real IMU as 'Seeds' to Our Pipeline. While IMUTube is about generating lots of virtual IMU data, our results show the value for the more traditional curated datasets from real IMU data. The real IMU data provides a seed that the virtual data grows into more sophisticated HAR models. Now the efforts in real IMU data collection can be focused on producing very high quality labeled data from a wide enough variety of subjects performing key activities. It may even be the case that this real IMU seed data is the treasured commodity that companies can use to provide the best seeds for IMUTube-generated virtual IMU data and the models grown from them.

7 RELATED WORK

The proposed method details a pipeline towards opportunistically extracting virtual sensor data from a potentially very large body of publicly available videos. This is in contrast to current wearable sensor data collection protocols, which involve user studies and human participants, as well as other approaches that generate sensor data from motion capture (mocap) settings. In what follows, we first discuss approaches to data collection for sensor-based human activity recognition as well as mocap based techniques. These approaches represent the state-of-the-art in the field that are based on dedicated data recording protocols. Subsequently, we detail prior work on training classifiers with limited labeled data, thereby focusing on data augmentation techniques and transfer learning.

7.1 Sensor Data Collection in HAR

Sensor data collection for human activity recognition is often performed by conducting user studies [14, 72, 97]. Typically, the participants in a study are asked to perform activities in laboratory settings while wearing a sensing platform. The advantage of data recording in a lab setting is that in addition to sensor data typically video data is recorded that is subsequently used for manual data annotation. For this purpose, the sensor and video data streams need to be synchronized [65], and human annotators need to be trained for consistency in annotation. The laboratory is designed to resemble a real-world environment, and user activities are either scripted or naturalistic. These include various gesture and locomotion level activities. However, designing a lab study to capture realistic natural behaviors is difficult. The protocol of such studies makes it challenging to collect large scale datasets. Furthermore, the annotation of activities is costly and error-prone and therefore prohibitive towards creating large datasets as they are required for deriving complex machine learning models.

Recently, Ecological Momentary Assessment (EMA) based approaches have been employed to record and especially annotate real-world activity data [35, 47, 88]. The sensing apparatus (containing sensors such as accelerometers or full-fledged IMUs) is worn on-body, and users self-report the activity labels when they are asked to do so through direct notification. Although these methods may lose sample-precise annotation of the activities, they encourage the collection of larger-scale datasets. While limited to gesture-based activities, Laput and Harrison [47] have shown that larger numbers (83) of fine-grained hand activities can be reliably recorded

and annotated. Both in-lab and EMA based collection protocols directly involve human participants to collect movement data using body-worn sensors.

Other approaches have explored alternative data collection methods that do not directly involve human participants. Kang *et al.* render a 3D human model on computer graphics software and simulate human activities [40]. The sensor data is extracted from the simulated human motion, and subsequently used to train the recognition models. However, it is very difficult to realistically simulate and design complex human activities. Therefore, such methods typically only explore simple gestures and locomotion activities. Alternatively, [87, 92] extract sensory data from public, large-scale motion capture (mocap) datasets [45, 54, 59], which contain a variety of motions and poses for human activity recognition. Although these datasets cover hundreds of subjects and thousands of poses and motions, they rarely include everyday activities. The majority of such mocap datasets include dancing, quick locomotion transitions, and martial arts, which are less relevant to recognizing daily human activities.

Most related to our work, Rey *et al.* [74] also proposed to collect virtual sensor data from online videos and demonstrated the effectiveness of the virtual sensor data for recognizing fitness activities. Their approach computes the 2D pose motion for a single person in the video with a fixed camera viewpoint. A regressor is trained for a target real sensor with the synced video and accelerometer recordings, which transfers the changes in joint locations from the 2D scene to the norm of the three-axis accelerometer. In contrast, our work can generate data from the full IMU (three-axis accelerometer, gyroscope, and simulated magnetometer). Further, we perform 3D motion estimation from videos with multiple people and scenes in the wild using camera motion tracking. We do not require synced video and wearable recordings as the virtual sensor can be adapted to any real sensor with our efficient distribution mapping method.

We leverage the availability of large scale video datasets that cover real-world activities to extract sensory data. These videos are recorded in-the-wild and contain a wide range of activities, including everyday activities, which makes them very attractive for deriving realistic and robust human activity recognition systems.

7.2 Tackling the Sparse Data Problem

Many publicly available datasets for human activity recognition contain imbalanced classes. For example, approximately 75% of the Opportunity dataset (which has 18 classes in total) [14] consists of the null class [27], making it challenging to design classifiers. The activities being studied also impact the class imbalance to some extent. In the PAMAP2 dataset, the skipping rope class constitutes approximately 2.5% of data, relative to other activities which constitute around 9% on average [27]. This follows reason as, unless your name is Rocky Balboa, it is harder for subjects to perform rope skipping for longer durations of time, in contrast to walking or lying down. This resulting class imbalance poses a challenge for the design and training of classifiers, which may find it easier to simply predict the majority class. Furthermore, the relatively small size of labeled datasets results in models quickly overfitting and does not allow the application of complex model architectures. It is also difficult to apply potentially alleviating techniques such as transfer learning, which rely on large datasets for knowledge transfer. As a result, [89] have noticed that the adoption of deep learning methods in human activity recognition has not yet translated to the pronounced accuracy gains seen in other domains.

As a way to overcome the problem of small, class-imbalanced datasets, data augmentation techniques have been applied previously to prevent overfitting, improve generalizability and increase variability in the datasets. They involve techniques that systematically transform the data during the training process in order to make classifiers more robust to noise and other variations [55]. They artificially inflate the training data by utilizing methods, which perform data warping, or oversampling [82]. Data warping includes geometric transformations such as rotations, and cropping, as well as adversarial training. For time series classification, the data warping techniques include window slicing, window warping, rotations, permutations and dynamic time warping [20, 48]. Several of these transformations can be combined to further improve the performance over a single method.

Um *et al.* demonstrate that combining three basic methods (permutation, rotation and time warping) yields better performance than using a single method [91]. In [70], construction equipment activity recognition is also improved by combining simple transformations.

Recently, data generation using either oversampling or generative adversarial networks (GANs [82]) have also been successfully introduced to sensor-based human activity recognition [93]. However, in contrast to other domains such as computer vision, performance improvements remain moderate, most likely due to non-trivial challenges inherent to generating realistic yet novel timeseries data. Oversampling based methods include synthetic minority oversampling technique (SMOTE) [22]. GANs have been used to, for example, augment biosignals [30] or in IoT [93]. Extending the conventional GAN approach, in [69], a data augmentation technique for time series data with irregular sampling is proposed utilizing conditional GANs. It is shown to outperform data warping techniques such as window slicing and time warping. Augmentation for wearable sensor data has been explored for monitoring Parkinson's disease in [91]. In this paper, seven transformations, including jittering, scaling, rotation and warping are detailed and their effects relative to no augmentation is studied. Further, the authors observed that combining multiple transformations results in higher performance. In [85], augmentation is performed on IMU spectrogram features to improve the activity recognition performance.

Another approach to deal with small labeled datasets includes transfer learning. Here, a base classifier (typically a neural network) is first trained on a base dataset and task. Subsequently, the learned features are re-purposed, or *transferred*, to a second target network to be trained on the target dataset and task. In particular, if the target dataset is significantly smaller compared to the base dataset, transfer learning enables training a large target network without overfitting [94], and typically results in improved performance. In [77], the authors propose a self-supervision pretext task and demonstrate its effectiveness for unsupervised transfer learning on other datasets with little labeled data. A more extreme example of having very small labeled datasets includes one-shot and few-shot learning, which contain very few labeled samples per class [21].

While the data augmentation techniques do improve the classification performance, they, ultimately, produce perturbed training samples. Therefore, they are unable to provide for the variety in human movements that is obtained by collecting data from a large number of subjects. On the other hand, the GAN based techniques perform augmentation by sampling from the dataset distribution. However, they require substantial amounts of data to train, and may suffer from training instability and non-convergence [93]. Furthermore, there is limited prior work studying data augmentation by GANs for wearable sensor data and their actual suitability for sensor-based human activity recognition remains to be shown. This makes it challenging to readily apply these generative networks to create more data.

We tackle the problem of having small labeled datasets with a different approach – by generating large quantities of virtual IMU data from videos. As we can leverage a large body of videos, containing many individuals, we generate datasets containing more diverse movements and potentially much larger datasets of realistic data, which is in stark contrast to existing methods that try to combat the sparse data problem.

8 CONCLUSION

In this paper we developed a framework for generating virtual IMU data based on automated extraction from video as a means to collect large-scale labeled datasets to support research in human activity recognition (HAR). We designed and validated our framework, IMUTube, that integrates a collection of techniques from computer vision, signal processing, and machine learning. Our initial findings show great promise for this technique to extend the capabilities for HAR, at a minimum for simple activities whose main IMU characteristics are confined to expression in 2D.

The greater promise of this work requires a collective approach by computer vision, signal processing, and activity recognition communities (who have already been greatly united through the advances of deep learning)

to advance the underlying agenda. Computer vision researchers can clearly build upon the IMUTube pipeline to address a variety of current limitations, further automating the pipeline and reducing the need for human curation of online videos. Signal processing advances can further manipulate the virtually-generated data to better condition the virtual data and represent the features and distributions of real IMU data. Activity recognition researchers can apply known modern learning techniques to this new class of labeled data for HAR and develop more effective ways to model, both with and without a mixture of real IMU data. Within a few years, we expect this collective effort to result in HAR as yet another success story for large-data-inspired learning techniques.

REFERENCES

[1] S. Alireza Golestaneh and L. Karam. 2017. Spatially-varying blur detection based on multiscale fused and sorted transform coefficients of gradient magnitudes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5800–5809.

[2] T. Alldieck, M. Magnor, B. Bhatnagar, C. Theobalt, and G. Pons-Moll. 2019. Learning to reconstruct people in clothing from a single RGB camera. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1175–1186.

[3] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2014. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[4] P. Asare, R. Dickerson, X. Wu, J. Lach, and J. Stankovic. 2013. BodySim: A Multi-Domain Modeling and Simulation Framework for Body Sensor Networks Research and Design. In *International Conference on Body Area Networks (BODYNETS)*. ICST.

[5] M. Bächlin, M. Plotnik, and G. Tröster. 2010. Wearable assistant for Parkinson's disease patients with the freezing of gait symptom. *IEEE Trans. Inf. Technol. Biomed.* 14, 2 (2010), 436–446.

[6] P.J. Besl and N. McKay. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 2 (February 1992), 239–256.

[7] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. 2016. Simple online and realtime tracking. In *IEEE International Conference on Image Processing (ICIP)*. 3464–3468.

[8] O. Bogdan, V. Eckstein, F. Rameau, and J. Bazin. 2018. DeepCalib: a deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. In *Proceedings of the ACM SIGGRAPH European Conference on Visual Media Production*. ACM, 6:1–6:10.

[9] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 961–970.

[10] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7291–7299.

[11] B. Caprile and V. Torre. 1990. Using vanishing points for camera calibration. *International Journal of Computer Vision* 4, 2 (March 1990), 127–139.

[12] J. Carreira, E. Noland, C. Hillier, and A. Zisserman. 2019. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987* (2019).

[13] Y. Chang, A. Mathur, A. Isopoussu, J. Song, and F. Kawsar. 2020. A Systematic Study of Unsupervised Domain Adaptation for Robust Human-Activity Recognition. 4, 1, Article 39 (March 2020), 30 pages.

[14] R. Chavarriaga, H. Sagha, and D. Roggen. 2013. The Opportunity challenge: A benchmark database for on-body sensor-based activity recognition. *Pattern Recognition Letter* 34, 15 (2013), 2033–2042.

[15] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005* (2013).

[16] Blender Online Community. 2018. *Blender - a 3D modelling and rendering package.* Blender Foundation, Stichting Blender Foundation, Amsterdam. http://www.blender.org

[17] W. Conover and R. Iman. 1981. Rank transformations as a bridge between parametric and nonparametric statistics. *The American Statistician* 35, 3 (1981), 124–129.

[18] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 248–255.

[19] J. Devlin, M. Chang, K. Lee, and K. Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* 1 (2019), 4171–4186.

[20] H. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. Muller. 2018. Data augmentation using synthetic data for time series classification with deep residual networks. *arXiv preprint arXiv:1808.02455* (2018).

[21] S. Feng and M. Duarte. 2019. Few-shot learning-based human activity recognition. *Expert Systems with Applications* 138 (2019), 112782.

[22] A. Fernández, S. Garcia, F. Herrera, and N. Chawla. 2018. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research* 61 (2018), 863–905.

[23] R. Girshick. 2015. Fast r-cnn. In *IEEE International Conference on Computer Vision (ICCV)*. 1440–1448.
[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. 2014. Generative adversarial nets. 2672–2680.
[25] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova. 2019. Depth From Videos in the Wild: Unsupervised Monocular Depth Learning From Unknown Cameras. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE.
[26] C. Gu, C. Sun, D. Ross, C. Vondrick, C. Pantofaru, Y. Li, S. Vijayanarasimhan, G. Toderici, S. Ricco, R. Sukthankar, C. Schmid, and J. Malik. 2018. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 6047–6056.
[27] Y. Guan and T. Plötz. 2017. Ensembles of deep lstm learners for activity recognition using wearables. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies (IMWUT)* 1, 2 (2017), 1–28.
[28] N. Hammerla, R. Kirkham, P. Andras, and T. Ploetz. 2013. On preserving statistical characteristics of accelerometry data using their empirical cumulative distribution. In *Proceedings of the ACM International Symposium on Wearable Computers*. 65–68.
[29] N. Y. Hammerla, S. Halloran, and T. Plötz. 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables.. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. AAAI Press, 1533–1540.
[30] S. Haradal, H. Hayashi, and S. Uchida. 2018. Biosignal data augmentation based on generative adversarial networks. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 368–371.
[31] H. Haresamudram, D. Anderson, and T. Plötz. 2019. On the role of features in human activity recognition. In *Proceedings of the ACM International Symposium on Wearable Computers*. 78–88.
[32] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.
[33] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*. 6626–6637.
[34] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine* 29, 6 (2012), 82–97.
[35] K. Hovsepian, M. Al'Absi, E. Ertin, T. Kamarck, M. Nakajima, and S. Kumar. 2015. cStress: towards a gold standard for continuous stress assessment in the mobile environment. In *Proceedings of the ACM international joint conference on pervasive and ubiquitous computing*. 493–504.
[36] Y. Huang, M. Kaufmann, E. Aksan, M. Black, O. Hilliges, and G. Pons-Moll. 2018. Deep inertial poser: learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–15.
[37] S. Ioffe and C. Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).
[38] I. Joel, A.and Stergios. 2011. A Direct Least-Squares (DLS) method for PnP. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE.
[39] A. Kanazawa, M. Black, D. Jacobs, and J. Malik. 2018. End-to-end recovery of human shape and pose. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7122–7131.
[40] C. Kang, H. Jung, and Y. Lee. 2019. Towards Machine Learning with Zero Real-World Data. In *The ACM Workshop on Wearable Systems and Applications*. 41–46.
[41] S. Kang, H. Choi, H. Park, B. Choi, H. Im, D. Shin, Y. Jung, J. Lee, H. Park, S. Park, and J. Roh. 2017. The development of an IMU integrated clothes for postural monitoring using conductive yarn and interconnecting technology. *Sensors* 17, 11 (2017), 2560.
[42] P. Karlsson, B. Lo, and G. Z. Yang. 2014. Inertial sensing simulations using modified motion capture data. In *Proceedings of the International Conference on Wearable and Implantable Body Sensor Networks (BSN)*. 16–19.
[43] D. Kingma and J. Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[44] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. 2011. HMDB: a large video database for human motion recognition. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2556–2563.
[45] Carnegie Mellon Graphics Lab. 2008. *Carnegie Mellon Motion Capture Database.* http://mocap.cs.cmu.edu/
[46] N. Lane, Y. Xu, H. Lu, S. Hu, T. Choudhury, A. Campbell, and F. Zhao. 2011. Enabling Large-Scale Human Activity Inference on Smartphones Using Community Similarity Networks. In *Proceedings of the International Conference on Ubiquitous Computing*. ACM, 355–364.
[47] G. Laput and C. Harrison. 2019. Sensing Fine-Grained Hand Activity with Smartwatches. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.
[48] A. Le Guennec, S. Malinowski, and R. Tavenard. 2016. Data Augmentation for Time Series Classification using Convolutional Neural Networks. In *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*.
[49] W. Li, Z. Zhang, and Z. Liu. 2010. Action recognition based on a bag of 3D points. In *The IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 9–14.

[50] D. Liaqat, M. Abdalla, Pegah Abed-Esfahani, Moshe Gabel, Tatiana Son, Robert Wu, Andrea Gershon, Frank Rudzicz, and Eyal De Lara. 2019. WearBreathing: Real World Respiratory Rate Monitoring Using Smartwatches. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies (IMWUT)* 3, 2 (2019), 1–22.

[51] J. Liu, A. Shahroudy, M. Perez, G. Wang, L. Duan, and A. Kot. 2019. NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[52] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. Black. 2015. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 1–16.

[53] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. 2018. Are gans created equal? a large-scale study. In *Advances in neural information processing systems*. 700–709.

[54] N. Mahmood, N. Ghorbani, N. Troje, G. Pons-Moll, and M. Black. 2019. AMASS: Archive of motion capture as surface shapes. In *IEEE International Conference on Computer Vision (ICCV)*. 5442–5451.

[55] A. Mathur, T. Zhang, S. Bhattacharya, P. Velickovic, L. Joffe, N. Lane, F. Kawsar, and P. Lió. 2018. Using deep data augmentation training to address software and hardware heterogeneities in wearable and smartphone sensing devices. In *IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 200–211.

[56] A. Muhammad Sayem, S. Hon Teay, H. Shahariar, P. Fink, and A. Albarbar. 2020. Review on Smart Electro-Clothing Systems (SeCSs). *Sensors* 20, 3 (2020), 587.

[57] V. Nair and G. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the international conference on machine learning (ICML)*. 807–814.

[58] A. Odena, V. Dumoulin, and C. Olah. 2016. Deconvolution and checkerboard artifacts. *Distill* 1, 10 (2016), e3.

[59] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. 2013. Berkeley mhad: A comprehensive multimodal human action database. In *IEEE Workshop on Applications of Computer Vision (WACV)*. IEEE, 53–60.

[60] A. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).

[61] F. J. Ordóñez and D. Roggen. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* 16, 1 (2016), 115.

[62] J. Park, Q. Zhou, and V. Koltun. 2017. Colored Point Cloud Registration Revisited. In *IEEE International Conference on Computer Vision (ICCV)*. 143–152.

[63] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli. 2019. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7753–7762.

[64] T. Phạm and Y. Suh. 2018. Spline Function Simulation Data Generation for Walking Motion Using Foot-Mounted Inertial Sensors. In *Sensors*. MDPI, 199–210.

[65] T. Plötz, C. Chen, N. Hammerla, and G. Abowd. 2012. Automatic synchronization of wearable sensors and video-cameras for ground truth annotation–a practical approach. In *Proceedings of the ACM International Symposium on Wearable Computers*. IEEE, 100–103.

[66] F. Pomerleau, F. Colas, and R. Siegwart. 2015. A Review of Point Cloud Registration Algorithms for Mobile Robotics. *Found. Trends Robot* 4, 1 (May 2015), 1âĂŞ104.

[67] G. Pons-Moll, S. Pujades, S. Hu, and M. Black. 2017. ClothCap: Seamless 4D clothing capture and retargeting. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–15.

[68] G. Pons-Moll, J. Romero, N. Mahmood, and M Black. 2015. Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–14.

[69] G. Ramponi, P. Protopapas, M. Brambilla, and R. Janssen. 2018. T-cgan: Conditional generative adversarial network for data augmentation in noisy time series with irregular sampling. *arXiv preprint arXiv:1811.08295* (2018).

[70] K. Rashid and J. Louis. 2019. Times-series data augmentation and deep learning for construction equipment activity recognition. *Advanced Engineering Informatics* 42 (2019), 100944.

[71] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. 2016. You only look once: Unified, real-time object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 779–788.

[72] A. Reiss and D. Stricker. 2012. Introducing a new benchmarked dataset for activity monitoring. In *Proceedings of the ACM International Symposium on Wearable Computers*. IEEE, 108–109.

[73] A. Reiss and D. Stricker. 2013. Personalized mobile physical activity recognition. In *Proceedings of the ACM International Symposium on Wearable Computers*. 25–28.

[74] V. Rey, P. Hevesi, O. Kovalenko, and P. Lukowicz. 2019. Let there be IMU data: generating training data for wearable, motion sensor based activity recognition from monocular RGB videos. In *Adjunct Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the ACM International Symposium on Wearable Computers*. 699–708.

[75] M. Rosca, B. Lakshminarayanan, and S. Mohamed. 2018. Distribution matching in variational inference. *arXiv preprint arXiv:1802.06847* (2018).

[76] S. Rusinkiewicz and M. Levoy. 2001. Efficient variants of the ICP algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. IEEE.
[77] A. Saeed, T. Ozcelebi, and J. Lukkien. 2019. Multi-task Self-Supervised Learning for Human Activity Detection. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies (IMWUT)* 3, 2 (2019), 1–30.
[78] P. M. Scholl, M. Wille, and K. Van Laerhoven. 2015. Wearables in the wet lab: a laboratory system for capturing and guiding experiments. In *Proceedings of the International Conference on Ubiquitous Computing*. ACM, 589–599.
[79] S. Shah and J.K. Aggarwal. 1996. Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation. *Pattern Recognition* 29, 11 (November 1996), 1775–1788.
[80] Z. Shen, W. Wang, X. Lu, J. Shen, H. Ling, T. Xu, and L. Shao. 2019. Human-Aware Motion Deblurring. In *IEEE International Conference on Computer Vision (ICCV)*. 5572–5581.
[81] J. Shi, L. Xu, and J. Jia. 2014. Discriminative blur detection features. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2965–2972.
[82] C. Shorten and T. Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data* 6, 1 (2019), 60.
[83] G. Sigurdsson, G. Varol, X. Wang, I. Laptev, A. Farhadi, and A. Gupta. 2016. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. *arXiv preprint arXiv:1604.01753* (2016).
[84] K. Soomro, A. Zamir, and M. Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).
[85] O. Steven Eyobu and D. Han. 2018. Feature representation and data augmentation for human activity classification based on wearable IMU sensor data using a deep LSTM neural network. *Sensors* 18, 9 (2018), 2892.
[86] T. Sztyler and H. Stuckenschmidt. 2016. On-body localization of wearable devices: An investigation of position-aware activity recognition. In *IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 1–9.
[87] S. Takeda, T. Okita, P. Lago, and S. Inoue. 2018. A multi-sensor setting activity recognition simulation tool. In *Proceedings of the ACM International Joint Conference and International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. 1444–1448.
[88] E. Thomaz, I. Essa, and G. Abowd. 2015. A practical approach for recognizing eating moments with wrist-mounted inertial sensing. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 1029–1040.
[89] C. Tong, S. Tailor, and N. Lane. 2020. Are Accelerometers for Activity Recognition a Dead-End?. In *Proceedings of the International Workshop on Mobile Computing Systems and Applications*. ACM, 39–44.
[90] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, and J. Collomosse. 2017. Total Capture: 3D Human Pose Estimation Fusing Video and Inertial Sensors. In *British Machine Vision Conference (BMVC)*.
[91] T. Um, F. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulić. 2017. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the ACM International Conference on Multimodal Interaction*. 216–220.
[92] F. Xiao, L. Pei, L. Chu, D. Zou, W. Yu, Y. Zhu, and T. Li. 2020. A Deep Learning Method for Complex Human Activity Recognition Using Virtual Wearable Sensors. *arXiv preprint arXiv:2003.01874* (2020).
[93] S. Yao, Y. Zhao, H. Shao, C. Zhang, A. Zhang, S. Hu, D. Liu, S. Liu, Lu Su, and T. Abdelzaher. 2018. Sensegan: Enabling deep learning for internet of things with a semi-supervised framework. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies (IMWUT)* 2, 3 (2018), 1–21.
[94] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. 2014. How transferable are features in deep neural networks?. In *Advances in neural information processing systems*. 3320–3328.
[95] A. Young, M. Ling, and D. Arvind. 2011. IMUSim: A simulation environment for inertial sensing algorithm design and evaluation. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 199–210.
[96] J. Yu and R. Ramamoorthi. 2019. Robust Video Stabilization by Optimization in CNN Weight Space. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3800–3808.
[97] M. Zhang and A. A. Sawchuk. 2012. USC-HAD: a daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the International Conference on Ubiquitous Computing*.
[98] Q. Zhang and R. Pless. 2004. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE.
[99] Z. Zhao, Y. Chen, J. Liu, Z. Shen, and M. Liu. 2011. Cross-people mobile-phone based activity recognition. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.
[100] T. Zhou, M. Brown, Noah S., and D. Lowe. 2017. Unsupervised learning of depth and ego-motion from video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1851–1858.
[101] H. Zhuang. 1995. A self-calibration approach to extrinsic parameter estimation of stereo cameras. *Robotics and Autonomous Systems* 15, 3 (August 1995), 189–197.