On the Knowledge Transfer via Pretraining, Distillation and Federated Learning

by

Weituo Hao

Department of Electrical and Computer Engineering
Duke University

Date: _____

Approved:

_____
Lawrence Carin, Supervisor

_____
Ricardo Henao Giraldo

_____
Robert Calderbank

_____
Yiran Chen

_____
Rong Ge

Dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the Graduate School of
Duke University

2022

ABSTRACT

On the Knowledge Transfer via Pretraining, Distillation and Federated Learning

by

Weituo Hao

Department of Electrical and Computer Engineering
Duke University

Date: _____

Approved:

_____
Lawrence Carin, Supervisor

_____
Ricardo Henao Giraldo

_____
Robert Calderbank

_____
Yiran Chen

_____
Rong Ge

An abstract of a dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the Graduate School of
Duke University

2022

Abstract

Modern machine learning technology based on a revival of deep neural networks has been successfully applied in many pragmatic domains such as computer vision(CV) and natural language processing(NLP). The very standard paradigm is *pre-training*: a large model with billions of parameters is trained on a surrogate task and then adapted to the downstream task of interest via fine-tuning. Knowledge transfer is what makes the pre-training possible, but scale is what makes it powerful, which requires the availability of much more training data and computing resources.

Along with the great success of deep learning, fueled by larger datasets and more computation capability, however, come series of interesting research topics. First, most pre-trained models learn on one-modal(vision or text) dataset and are designed for single-step downstream task such as classification. Does pre-training for more complex tasks such as reinforcement learning still work? Second, pre-trained models obtain impressive empirical performance at the price of deployment challenges on low-resource(both memory and computation) platforms. How to compress the large models into smaller ones in an efficient way? Third, collecting sufficient training data is often expensive, time-consuming, or even unrealistic in many scenarios due to privacy constraints. Does it exist a training paradigm without data exchange?

For less explored questions mentioned above, I conducted several projects related but not limited to: *i*) large-scale pre-training based on multi-modal input for vision and language navigation, proofing the effectiveness of knowledge transfer across complex tasks via *pre-training*; *ii*) data augmentation for compressing large-scale language models, improving the efficiency of knowledge transfer in the teacher-student *distillation* framework; *iii*) weight factorization for model weights sharing in *Federated Learning*, achieving the trade-off between model performance and data privacy.

Contents

List of Tables

List of Figures

Acknowledgements

To pursue a PhD degree is never an easy decision, since it's like climbing a mountain of pressure, full of frustration and challenges. Fortunately, there are many people supporting me to complete the journey.

First of all, I would like to express my sincere thanks to my advisor Lawrence Carin, for his accepting me in the group, instructive research guidance and perhaps the most important thing, his experience in the correct way to communicate with the real world. Without his effort of the group management, I would never have such freedom on research topics, smart and creative group members, let alone this thesis work. Being lucky as Larry's last generation of PhD student at Duke, his influence on me will definitely guide me towards more success in the future. I would also like to extend my gratitude to Ricardo Henao, Robert Calderbank, Yiran Chen, Rong Ge, David Carlson, Hai Li, for their time and effort to server on my committee. Additionally, I must thank Mostafa El-Khamy for the Samsung Fellowship assistance to support my last two years of PhD life.

Another reason that I feel very proud of being part of Larry's group is the chance to work closely with rising stars in research. I do appreciate their help in every detail of my projects and learn much from the discussion with them. I would also like to thank my mentors and collaborators during my internships at Microsoft Research, Samsung, and TikTok.

Finally, I'm deeply grateful for my family. As their son and little brother, I get endless warm encouragement and understanding of whatever decisions I make in my life. Words express barely enough my appreciation for them but pursuing excellence will always be what I repay for their support.

Chapter 1

Introduction

Knowledge transfer aims at improving the performance of target learners on target domains by transferring the knowledge contained in different but related source domains, reducing the dependence on a large number of target domain data. In general perspective of knowledge transfer, *pre-training* is to adapt knowledge from single source to solve various downstream tasks, *distillation* is to convey information contained from one source to another and *federated learning* is to share common knowledge among multiple sources with privacy constraints. As a result, knowledge transfer for practical purpose becomes valuable and popular research topic.

The standard paradigm to apply knowledge transfer within deep learning is *pre-training*: a model is trained on a surrogate task and then adapted to the downstream task of interest via fine-tuning. For example, in the field of NLP, the Transformer [VSP+17] is trained to optimize the masked word prediction on the Wiki corpus and fine-tuned to solve different GLUE [WSM+19] tasks. Also, in the CV domain, ResNet [HZRS16] is used as the raw image feature extractor for various vision tasks such as image classification and object detection. However, most previous work only take single modal as the input and the downstream task is often a single step prediction such as classification. The pre-training with multi-modal as input and application on complex tasks such as reinforcement learning(RL), a typical step towards general artificial intelligence , is unfortunately less explored. Therefore, we choose to study the vision-language navigation, a reinforcement learning task requires the understanding of vision and language simultaneously, and verify the effectiveness

of pre-training paradigm in Chapter 2.

As pre-training has become the dominant approach in machine intelligence, the outcome model from pre-training is named as *foudation model* [BHA+21]. Foundation models usually come with large storage and computation cost, which are difficult to deploy on resource limited devices. Knowledge *distillation* (KD) [HVD15] is a popular model compression way by transferring valuable information from a cumbersome teacher network to a compact student network. Traditional distillation methods are designed to transfer knowledge from the intermediate feature maps or predictions of the teacher network, which requires large amount of original training data and limits the knowledge transfer when data is scarce. To improve the efficiency of distillation, we propose to adopt synthesized data to unveil more knowledge from teacher network to student network, which is presented in Chapter 3.

In domains where data are sensitive or private, there is great value in methods that can transfer knowledge in a distributed manner without the data ever leaving the local devices. In light of this need, *federated learning* [MMR+17] has emerged as a popular training paradigm that trades transmitting data for communicating updated weight parameters for each local device. However, many common federated approaches learn a single global model; while this may do well on average, performance degrades when data distribution is highly heterogeneous. To address the issues, we propose an approach that combines the Indian Buffet Process with a shared dictionary of weight factors for neural networks in Chapter 4.

An additional and very important topic is *fairness* as the field of artificial intelligence advances. It motivates related study not only in computer vision [MDH+20], natural language processing [LLZ+20] but also in federated learning domain [LSBS19]. The fairness issue is also broadly recognized as *social bias*, which denotes the unbal-

anced model behaviors with respect to some socially sensitive topics, such as gender, race, and religion. The main reasons for unfairness include but are not limited to: (*i*) the biased semantic information in the raw data [LLZ+20];(*ii*) the skewed distribution in the training data [MDH+20, LSBS19]. Note that unfairness does not disappear groundlessly during the process of knowledge transfer. Therefore, we introduce a zero-shot data augmentation technique for fair federated learning in Chapter 5 and a post-processing method based on contrastive learning to calibrate the language foundation models in Chapter 6.

1.1 Foundation Models via Pre-training

Large scale pre-training has first set off a revolution strongly in NLP since the transformer [VSP+17]. The attention based architectures free language models from low training efficiency by allowing parallel computing, which kicks of the era of foundation models. Text encoders such as BERT [DCLT19a], RoBERTa [LOG+19] have been successfully applied to various NLP benchmarks such as GLUE [WSM+18] and SuperGLUE [WPN+19]. Also, powerful text generators such as GPT-3 [BMR+20], 175 billion parameters compared to GPT-2's 1.5 billion [RNSS18], even allows a zero-shot solution to different downstream tasks. Recent work [GFC20] claims that the potential of language models are not fully developed, while providing a simple solution based on *prompt*. The selection of prompt and its influence on model's performance has been a popular research area not only in NLP but also in multi-modal domain.

In computer vision, extracting ResNet [HZRS16] feature from the raw images has been the standard pre-processing for a wide variety of downstream tasks. However, the huge success of transformer in NLP has begun to promote more attempts of pure attention based model. With the proposal of Vision Transformer(ViT) [DBK+20],

there occurs debate over the convolution-based and attention-based model architecture. To improve ViT, Swin Transformer [LLC+21] proposes attention mechanism based on hierarchical shifting window, a hybrid approach seemingly reintroducing the convolution prior. On the other hand, ConvNext [LMW+22] demonstrates convolution-based model's still valid superiority by redesigning the residual block in ResNet. Except for the change in model architecture choice, the training paradigm for the vision backbone models has also shifted from supervised learning to self-supervised learning such as contrastive learning [CKNH20, HFW+20] and masked image patches reconstruction [HCX+21].

In the increasing need to handling more complex tasks, we also see the foundation models across research communities in the form of multi-modal backbones [LJS+20, KSK21, CLTB21]. These models are trained on vision and language data and able to adapt to various tasks requiring multi-modal input understanding such as visual question answering, image captioning and vision-language navigation [SMV+19, TB19, LBPL19, ZPZ+20, SZC+19, LDF+19]. The existing approaches share two common grounds:*i* employ transformer-like architecture for the modality fusion;*ii* adopt masked text word/image pixels prediction as training objective.

1.2 Knowledge Distillation

In practice, after the pre-training stage, the resulting models can be fine-tuned to different downstream tasks. While these models have yielded impressive results, they typically have millions, if not billions, of parameters, and thus can be very expensive from storage and computaftional standpoints. Additionally, during deployment, such large models can require a lot of time to process even a single sample. In settings where computation may be limited (*e.g.* mobile, edge devices), such characteristics may preclude such powerful models from deployment entirely.

One promising strategy to compress and accelerate large-scale language models is knowledge distillation [ZGSZ19, TLL+19, SYS+20].The key idea is to train a smaller model (a "student") to mimic the behavior of the larger, stronger-performing, but perhaps less practical model (the "teacher"), thus achieving similar performance with a faster, lighter-weight model. A simple but powerful method of achieving this is to use the output probability logits produced by the teacher model as soft labels for training the student [HVD15]. With higher entropy than one-hot labels, these soft labels contain more information for the student model to learn from.

In formal, we query the teacher model $f(\cdot)$ with sampled training data $\mathbf{x}_i$, producing output prediction $f(\mathbf{x}_i)$. The student model $g(\cdot)$ is encouraged to imitate this prediction on the same input, by minimizing the objective:

$$\mathcal{L}_{\text{KD}} = d(f(\mathbf{x}_i), g(\mathbf{x}_i)) \tag{1.1}$$

where $d(\cdot, \cdot)$ is a distance metric for distillation, with temperature-adjusted cross-entropy. However, the training data of downstream task is often limited to unveil enough information of the teacher model, which reduces the efficiency of the distillation and fails to learn a student model with comparable performance.

1.3 Federated Learning

With the rise of the Internet of Things (IoT), the proliferation of smart phones, and the digitization of records, modern systems generate increasingly large quantities of data. These data provide rich information about each individual, opening the door to highly personalized intelligent applications, but this knowledge can also be sensitive: images of faces, typing histories, medical records, and survey responses are all examples of data that should be kept private. Federated learning [MMR+17] has been proposed as a possible solution to this problem. By keeping user data on

each local *client* device and only sharing model updates with the global *server*, federated learning represents a possible strategy for training machine learning models on heterogeneous, distributed networks in a privacy-preserving manner. A typical example illustrates how federated learning works for next word prediction in Figure 1.1 [LSTS20]. During the training process, multiple rounds of communication happen between the server and the clients, and a single communication round contains two stages:

- The clients download the model from the server and update it by the local dataset.
- The clients upload the model back to the server and the server aggregate the returned models.

Formally, consider N client devices, with the i^{th} device having data distribution $\mathcal{D}_i$, which may differ as a function of i. In many distributed learning settings, a single global model is learned and deployed to all N clients. Thus, the model weights θ is shared across all clients. To satisfy the global objective, θ is learned to minimize the loss on average across all clients. This is the approach of many federated learning approaches. For example, FedAvg [MMR$^+$17] minimizes the following objective:

$$\min_{\theta} \mathscr{L}(\theta) = \sum_{i=1}^{N} p_i \mathcal{L}_i(\theta) \tag{1.2}$$

where $\mathcal{L}_i(\theta) \coloneqq \mathbb{E}_{x_i \sim \mathcal{D}_i}[l_i(x_i; \theta)]$ is the local objective function, N is the number of clients, and $p_i \geq 0$ is the weight of each device i.

While demonstrating promise in such a paradigm, a number of challenges remain for federated learning such as statistical heterogeneity, personalization, security [LSTS19].

What is more concerning is that the accuracy loss due to statistical heterogeneity

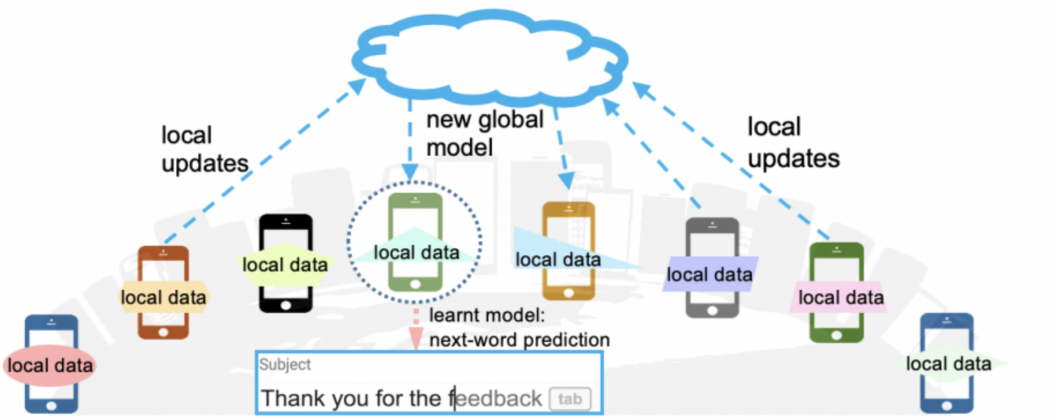


Figure 1.1: An example of federated learning applied on text auto-complete.

may be borne unequally among clients [LSBS19]. In populations with unequally sized subgroups, clients with less common classes tend to see worse performance [HML+20]. This may be, in part, due to catastrophic forgetting [MC89, SAK+19]: clients from outside a subpopulation have a tendency to forget features not found in their own data and, during aggregation, the less represented clients may have their learned features drowned out when the model weights are averaged. In the real world, these client characteristics may represent ethnicity [KBK+12], gender [AKV+20, Lea18], age [ABVK20], language [GHDL18], dialect, demographics, animal species, or disease trait. Therefore, the inability to cope with statistical heterogeneity may lead to potentially unfair algorithms.

Chapter 2

Knowledge Transfer for Navigation tasks via Multi-Modal Pre-training

2.1 Introduction

Learning to navigate in a photorealistic home environment based on natural language instructions has attracted increasing research interest [SCD+17, KMG+17, DDG+18, AWT+18, CSM+10], as it provides insight into core scientific questions about multimodal representations. It also takes a step toward real-world applications, such as personal assistants and in-home robots. Vision-and-language navigation (VLN) presents a challenging reasoning problem for agents, as the multimodal inputs are highly variable, inherently ambiguous, and often under-specified.

Most previous methods build on the sequence-to-sequence architecture [SVL14], where the instruction is encoded as a sequence of words, and the navigation trajectory is decoded as a sequence of actions, enhanced with attention mechanisms [AWT+18, WHC+19, MLW+19] and beam search [FHC+18]. While a number of methods [MLA17, MHGP17, WXWW18] have been proposed to improve language understanding, common to all existing work is that the agent learns to understand each instruction from scratch or in isolation, without collectively leveraging prior vision-grounded domain knowledge.

However, each instruction in practice only loosely aligns with the desired navigation path, making it imperfect for the existing paradigm of learning to understand the

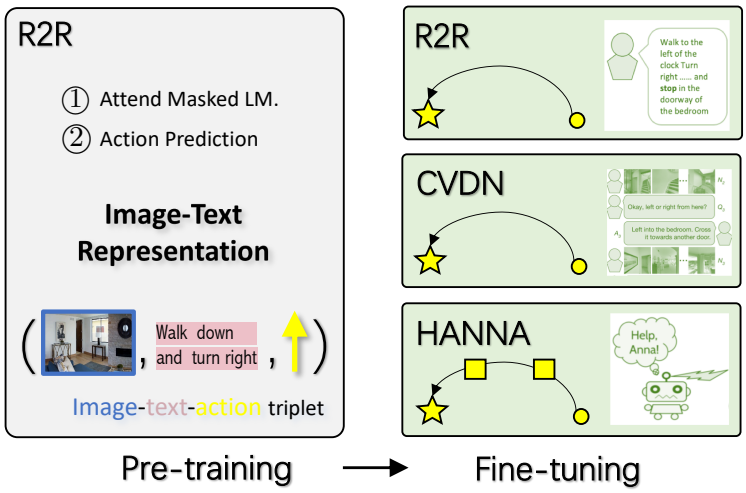


Figure 2.1: Illustration of the proposed pre-training and fine-tuning paradigm for VLN.

instruction from scratch. This is because (*i*) every instruction only partially characterizes the trajectory. It can be ambiguous to interpret the instructions, without grounding on the visual states. (*ii*) The objects in visual states and language instructions may share various common forms/relationships, and therefore it is natural to build an informative joint representation beforehand, and use this "common knowledge" for transfer learning in downstream tasks.

To address this natural ambiguity of instructions more effectively, we propose to pre-train an encoder to align language instructions and visual states for joint representations. The image-text-action triplets at each time step are independently fed into the model, which is trained to predict the masked word tokens and next actions, thus formulating the VLN pre-training in the self-learning paradigm. The complexity of VLN learning can then be reduced by eliminating language understandings which lack consensus from visual states. The pre-trained model plays the role of providing generic image-text representations, and is applicable to most existing approaches to VLN, leading to our agent PREVALENT. We consider three VLN scenarios as downstream tasks: Room-to-room (R2R) [AWT+18], cooperative vision-and-dialog

navigation (CVDN) [TMCZ19], and "Help, Anna!" (HANNA) [NDI19]. The overall pre-training and finetuning pipeline is shown in Figure 2.1.

Comprehensive experiments demonstrate strong empirical performance of PREVALENT. The proposed PREVALENT achieves a new state of the art on all three tasks [1]. Comparing with existing methods, it adapts faster, and generalizes better to unseen environments and new tasks.

Our code and pre-trained model is released on GitHub [2].

2.2 Related Work

Vision-language pre-training Vision-Language Pre-trainig (VLP) is a rapidly growing research area. The existing approaches employ BERT-like objectives [DCLT19b] to learn cross-modal representation for various vision-language problems, such as visual question-answering, image-text retrieval and image captioning *etc.* [SMV+19, TB19, LBPL19, ZPZ+20, SZC+19, LDF+19]. However, these VLP works focus on learning representations only for vision-language domains. This paper presents the first pre-trained models, grounding vision-language understanding with actions in a reinforcement learning setting. Further, existing VLP methods require faster R-CNN features as visual inputs [Gir15, AHB+18], which are not readily applicable to VLN. State-of-the-art VLN systems are based on panoramic views (*e.g.,* , 36 images per view for R2R), and therefore it is computationally infeasible to extract region features for all views and feed them into the agent.

Vision-and-language navigation Various methods have been proposed for learning to navigate based on vision-language cues. In [FHC+18] a panoramic action space

[1] Among *all* public results at the time of this submission.

[2] `https://github.com/weituo12321/PREVALENT`

and a "speaker" model were introduced for data augmentation. A novel neural decoding scheme was proposed in [KLB+19] with search, to balance global and local information. To improve the alignment of the instruction and visual scenes, a visual-textual co-grounding attention mechanism was proposed in [MLW+19], which is further improved with a progress monitor [MWA+19]. To improve the generalization of the learned policy to unseen environments, reinforcement learning has been considered, including planning [WXWW18], and exploration of unseen environments using a off-policy method [WHC+19]. An environment dropout was proposed [TYB19] to generate more environments based on the limited data, so that it can generalize well to unseen environments. These methods are specifically designed for particular tasks, and hard to generalize for new tasks. In this paper, we propose the first generic agent that is pre-trained to effectively understand vision-language inputs for a broad range of navigation tasks, and can quickly adapt to new tasks. The most related agent to ours is PreSS [LLX+19]. However, our work is different from [LLX+19] from two perspectives: (*i*) PreSS employs an off-the-shelf BERT [DCLT19b] model for language instruction understanding, while we pre-train a vision-language encoder from scratch, specifically for the navigation tasks. (*ii*) PreSS only focuses on the R2R task, while we verify the effectiveness of our pre-trained model on three tasks, including two out-of-domain navigation tasks.

2.3 Background

The VLN task can be formulated as a Partially Observable Markov Decision Process (POMDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P_s, r \rangle$, where $\mathcal{S}$ is the visual state space, $\mathcal{A}$ is a discrete action space, P_s is the unknown environment distribution from which we draw the next state, and $r \in \mathbb{R}$ is the reward function. At each time step t, the agent first observes an RGB image $\boldsymbol{s}_t \in \mathcal{S}$, and then takes an action $\boldsymbol{a}_t \in \mathcal{A}$. This leads the

simulator to generate a new image observation $\boldsymbol{s}_{t+1} \sim P_s(\cdot|\boldsymbol{s}_t, \boldsymbol{a}_t)$ as the next state.

The agent interacts with the environment sequentially, and generates a trajectory of length T. The episode ends when the agent selects the special `STOP` action, or when a pre-defined maximum trajectory length is reached. The navigation is successfully completed if the trajectory $\boldsymbol{\tau}$ terminates at the intended target location.

In a typical VLN setting, the instructions are represented as a set $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^{M}$, where M is the number of alternative instructions, and each instruction $\boldsymbol{x}_i$ consists of a sequence of L_i word tokens, $\boldsymbol{x}_i = [x_{i,1}, x_{i,2}, ..., x_{i,L_i}]$. The training dataset $\mathcal{D}_E = \{\boldsymbol{\tau}, \boldsymbol{x}\}$ consists of pairs of the instruction $\boldsymbol{x}$ together with its corresponding expert trajectory $\boldsymbol{\tau}$.

The agent then learns to navigate via performing maximum likelihood estimation (MLE) of the policy π, based on the individual sequences:

$$\max_{\boldsymbol{\theta}} \ \mathcal{L}_{\boldsymbol{\theta}}(\boldsymbol{\tau}, \boldsymbol{x}) = \log \pi_{\boldsymbol{\theta}}(\boldsymbol{\tau}|\boldsymbol{x}) = \sum_{t=1}^{T} \log \pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t, \boldsymbol{x}), \tag{2.1}$$

where $\boldsymbol{\theta}$ are the policy parameters.

The policy is usually parameterized as an attention-based Seq2Seq model [AWT+18, FHC+18], trained in the teacher-forcing fashion, *i.e.,* , the ground-truth states $\boldsymbol{s}_t$ are provided at every step in training. This allows reparameterization of the policy as an encoder-decoder architecture, by considering a function decomposition $\pi_{\boldsymbol{\theta}} = f_{\boldsymbol{\theta}_E} \circ f_{\boldsymbol{\theta}_D}$:

- A *vision-language encoder* $f_{\boldsymbol{\theta}_E} : \{\boldsymbol{s}_t, \boldsymbol{x}\} \rightarrow \boldsymbol{z}_t$, where a joint representation $\boldsymbol{z}_t$ at time step t is learned over the visual state $\boldsymbol{s}_t$ and the language instruction $\boldsymbol{x}$.
- An *action decoder* $f_{\boldsymbol{\theta}_D} : \{\boldsymbol{s}_t, \boldsymbol{z}_t\} \rightarrow \boldsymbol{a}_t$. For each joint representation $\boldsymbol{z}_t$, we ground it with $\boldsymbol{s}_t$ via neural attention, and decode into actions $\boldsymbol{a}_t$.

Successful navigation largely depends on precise joint understanding of natural language instructions and the visual states [TGB19]. We isolate the encoder stage, and focus on pre-training a generic vision-language encoder for various navigation tasks.

2.4 Pre-training Models

Our pre-training model aims to provide joint representations for image-text inputs in VLN.

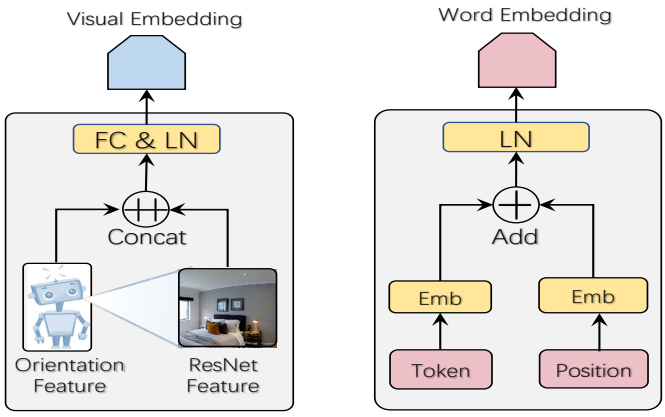
2.4.1 Input Embeddings

The input embedding layers convert the inputs (*i.e.,* , panoramic views and language instruction) into two sequences of features: image-level visual embeddings and word-level sentence embeddings.

Visual Embedding Following [FHC+18], we employ panoramic views as visual inputs to the agent. Each panoramic view consists of 36 images in total (12 angles, and 3 camera poses per angle): $\boldsymbol{s} = [s_1, \cdots, s_{36}]$. Each image is represented as a 2176-dimensional feature vector $s = [s_v, s_p]$, as a result of the concatenation of two vectors: (*i*) The 2048-dimensional visual feature s_v output by a Residual Network (ResNet) of the image [HZRS16]; (*ii*) the 128-dimensional orientation feature vector s_p that repeats $[\sin\psi; \cos\psi; \sin\omega; \cos\omega]$ 32 times, where ψ and ω are the heading and elevation poses, respectively [FHC+18]. The embedding for each image is:

$$\boldsymbol{h} = \text{Layer-Norm}(\mathbf{W}_e s + \boldsymbol{b}_e)) \tag{2.2}$$

where $\mathbf{W}_e \in \mathbb{R}^{d_h \times 2176}$ is a weight matrix, and $\boldsymbol{b}_e \in \mathbb{R}^{d_h}$ is the bias term; $d_h = 768$ in our experiments. Layer normalization (LN) [BKH16] is used on the output of this fully connected (FC) layer. An illustration of the visual embedding is shown in Figure 2.2(a).



(a) Visual embedding (b) Text embedding

Figure 2.2: Illustration for the representation procedure of (a) visual embedding and (b) text embedding.

Text Embedding The embedding layer for the language instruction follows the standard Transformer, where LN is applied to the summation of the token embedding and position embedding. An illustration of the text embedding is shown in Figure 2.2(b).

2.4.2 Encoder Architecture

Our backbone network has three principal modules: two single-modal encoders (one for each modality), followed by a cross-modal encoder. All modules are based on a multi-layer Transformer [VSP+17]. For the ℓ-th Transformer layer, its output is

$$\mathbf{H}_\ell = \mathcal{T}(\mathbf{H}_{\ell-1}, \mathbf{H}', \mathbf{M}) \tag{2.3}$$

where $\mathbf{H}_{l-1} \in \mathbb{R}^{L \times d_h}$ is the previous layer's features (L is the sequence length), $\mathbf{H}' \in \mathbb{R}^{L' \times d_h}$ is the feature matrix to attend, and $\mathbf{M} \in \mathbb{R}^{L \times L'}$ is the mask matrix, determining whether a pair of tokens can be attended to each other.

More specifically, in each Transformer block, the output vector is the concatenation of multiple attention heads $\mathbf{H}_\ell = [\mathbf{A}_{\ell,1}, \cdots, \mathbf{A}_{\ell,h}]$ (h is the number of heads). One

attention head $\mathbf{A}$ is computed via:

$$\mathbf{A}_\ell = \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} + \mathbf{M})\mathbf{V}, \tag{2.4}$$

$$\mathbf{M}_{ij} = \begin{cases} 0, & \text{allow to attend} \\ -\infty, & \text{not to attend} \end{cases} \tag{2.5}$$

$$\mathbf{Q} = \mathbf{W}_\ell^Q \mathbf{H}', \mathbf{K} = \mathbf{W}_\ell^K \mathbf{H}_{l-1}, \mathbf{V} = \mathbf{W}_\ell^V \mathbf{H}_{l-1} \tag{2.6}$$

where $\mathbf{H}_{l-1}$ and $\mathbf{H}'$ are linearly projected to a triple of queries, keys and values using parameter matrices $\mathbf{W}_\ell^Q, \mathbf{W}_\ell^K, \mathbf{W}_\ell^V \in \mathbb{R}^{d_h \times d_k}$, respectively; d_k is the projection dimension. In the following, we use different mask matrices $\mathbf{M}$ and attended feature matrices $\mathbf{H}'$ to construct the contextualized representation for each module.

Single-modal Encoder The standard self-attention layer is used in the single-modal encoder. All of the keys, values and queries come from the output of the previous layer in the encoder. Each position in the encoder can attend to all positions that belong to its own modality in the previous layer. Specifically, $\mathbf{M}$ is a full-zero matrix, and $\mathbf{H}' = \mathbf{H}_{l-1}$. Similar to the self-attention encoder module in the standard Transformer, the position-wise feed-forward network (FFN) is used.

Cross-modal Encoder To fuse the features from both modalities, a cross-attention layer is considered. The queries $\mathbf{H}'$ come from the previous layer of the other modality, and the memory keys and values come from the output $\mathbf{H}_{l-1}$ of the current modality. It allows every position in the encoder to attend over all positions in the different modality. This mimics the typical encoder-decoder attention mechanisms in the Transformer, but here we consider two different modalities, rather than input-output sequences. This cross-attention layer is followed by a self-attention layer and

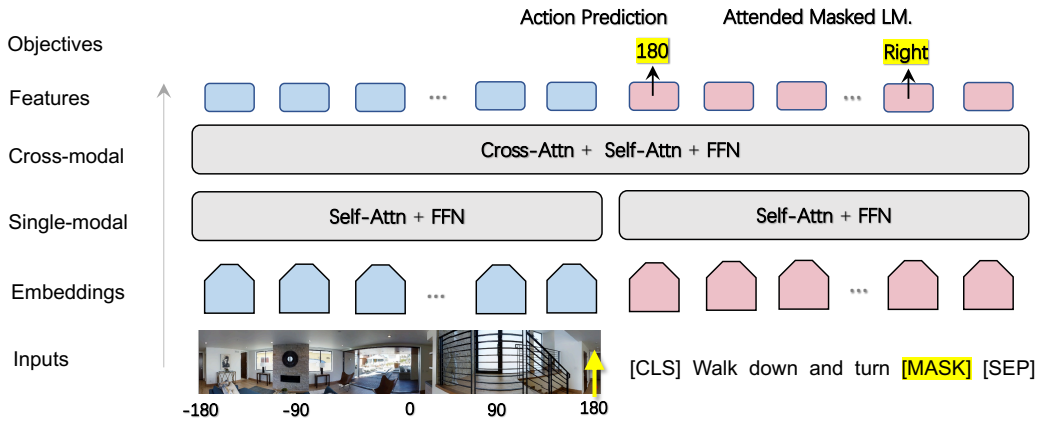


Figure 2.3: Illustration of model architecture. Two learning objectives are considered: image-attended masked language modeling and action prediction.

an FFN layer.

The overall model architecture is illustrated in Figure 2.3. Following [TB19], $L_{\text{text}} = 9$, $L_{\text{vision}} = 1$ and $L_{\text{cross}} = 3$. The last layer output of the encoder is denoted as $\boldsymbol{z} = \boldsymbol{h}_{L_{\text{cross}}}$, which is used as the features in the downstream tasks.

2.4.3 Pre-training Objectives

We introduce two main tasks to pre-train our model: Image-attended masked language modeling (MLM) and action prediction (AP). For an instruction-trajectory pair $\{\boldsymbol{x}, \boldsymbol{\tau}\}$ from the training dataset $\mathcal{D}_E$, we assume a state-action pair from the trajectory follows an independent identical distribution given the instruction in the pre-training stage: $(\boldsymbol{s}_t, \boldsymbol{a}_t) \overset{iid}{\sim} p(\boldsymbol{\tau})$.

Attended Masked Language Modeling We randomly mask out the input words with probability 15%, and replace the masked ones x_i with special token [`MASK`]. The goal is to predict these masked words based on the observation of their surrounding words $\boldsymbol{x}_{\setminus i}$ and all images $\boldsymbol{s}$ by minimizing the negative log-likelihood:

$$\mathcal{L}_{\text{MLM}} = -\mathbb{E}_{\boldsymbol{s}\sim p(\boldsymbol{\tau}),(\boldsymbol{\tau},\boldsymbol{x})\sim\mathcal{D}_E} \log p(x_i|\boldsymbol{x}_{\setminus i}, \boldsymbol{s}) \tag{2.7}$$

This is in analogy to the cloze task in BERT, where the masked word is recovered from surrounding words, but with additional image information to attend. It helps the learned word embeddings to be grounded in the context of visual states. This is particularly important for VLN tasks, where the agent is required to monitor the progress of completed instruction by understanding the visual images.

Action Prediction The output on the special token [CLS] indicates the fused representation of both modalities. We apply an FC layer on top of the encoder output of [CLS] to predict the action. It scores how well the agent can make the correct decision conditioned on the current visual image and the instruction, without referring to the trajectory history. During training, we sample a state-action pair $(\boldsymbol{s}, \boldsymbol{a})$ from the trajectory $\boldsymbol{\tau}$ at each step, and then apply a cross-entropy loss for optimization:

$$\mathcal{L}_{\text{AP}} = -\mathbb{E}_{(\boldsymbol{a},\boldsymbol{s})\sim p(\boldsymbol{\tau}),(\boldsymbol{\tau},\boldsymbol{x})\sim\mathcal{D}_E} \log p(\boldsymbol{a}|x_{[\texttt{CLS}]}, \boldsymbol{s}). \tag{2.8}$$

The full pre-training objective is:

$$\mathcal{L}_{\text{Pre-training}} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{AP}}. \tag{2.9}$$

Discussion Other loss designs can be considered for the pre-training objective. Our results on masked image modeling did not show better results, and thus are excluded in the experiments.

2.4.4 Pre-training Datasets

We construct our pre-training dataset based on the Matterport3D Simulator, a photo-realistic visual reinforcement learning (RL) simulation environment for the development of intelligent agents based on the Matterport3D dataset [CDF+17]. Specifically, it consists of two sets: (*i*) The training datasets of R2R, which has 104K image-text-action triplets; (*ii*) we employed the Speaker model in [FHC+18] to synthesize

1,020K instructions for the shortest-path trajectories on the training environments. This leads to 6,482K image-text-action triplets. Therefore, the pre-training dataset size is 6,582K.

2.5 Adapting to new tasks

We focus on three downstream VLN tasks that are based on the Matterport3D simulator. Each task poses a very different challenge to evaluate the agent. (*i*) The R2R task is used as an in-domain task; it can verify the agent's generalization capability to unseen environments. (*ii*) CVDN and HANNA are considered as out-of-domain tasks, to study the generalization ability of the agent to new tasks. More specifically, CVDN considers indirect instructions (*i.e.,* , dialog history), and HANNA is an interactive RL task.

2.5.1 Room-to-Room

In R2R, the goal is to navigate from a starting position to a target position with the minimal trajectory length, where the target is explicitly informed via language instruction. To use the pre-trained model for fine-tuning in R2R, the attended contextualized word embeddings are fed into an LSTM encoder-decoder framework, as in [FHC+18, LLX+19]. In prior work, random initialization is used in [FHC+18], and BERT is used in [LLX+19]. In contrast, our word embeddings are pre-trained from scratch with VLN data and tasks.

2.5.2 Cooperative Vision-and-Dialogue Navigation

In the CVDN environment, the Navigation from Dialog History (NDH) is defined, where the agent searches an environment for a goal location, based on the dialog history that consists of multiple turns of question-answering interactions between the

the agent and to its partner. The partner has privileged access to the best next steps that the agent should take according to a shortest path planner. CVDN is more challenging than R2R, in that the instructions from the dialog history are often ambiguous, under-specified, and indirect to the final target. The fine-tuning model architecture for CVDN is the same as R2R, except that CVND usually has much longer text input. We limit the sequence length to 300. Words that are longer than 300 in a dialog history are removed.

2.5.3 HANNA: Interactive Imitation Learning

HANNA simulates a scenario where a human requester asks an agent via language to find an object in an indoor environment, without specifying the process of how to complete the task. The only source of help the agent can leverage in the environment is the *assistant*, who helps the agent by giving subtasks in the form of (*i*) a natural language instruction that guides the agent to a specific location, and (*ii*) an image of the view at that location. When the help mode is triggered, we use our pre-trained model to encode the language instructions, and the features are used for the rest of their system.

2.5.4 Training details

Pre-training We pre-train the proposed model on eight V100 GPUs, and the batch size for each GPU is 96. The AdamW optimizer [KB14] is used, and the learning rate is 5×10^{-5}. The total number of training epochs is 20.

Fine-tuning The fine-tuning is performed on NVIDIA 1080Ti GPU. For the R2R task, we follow the same learning schedule as [TYB19]. When training the augmented listener, we use batch size 20. We continue to fine-tune the cross-attention encoder for 20k iterations, with the batch size 10 and learning rate 2×10^{-6}. For the NDH

task, we follow the same learning schedule as in [TMCZ19], and choose the batch size as 15 and learning rate as 5×10^{-4}. For HANNA, the training schedule is the same as [NDI19]. The batch size is 32 and learning rate is 1×10^{-4}.

2.5.5 Room-to-Room

Dataset The R2R dataset [AWT+18] consists of 10,800 panoramic views (each panoromic view has 36 images) and 7,189 trajectories. Each trajectory is paired with three natural language instructions. The R2R dataset consists of four splits: train, validation seen and validation unseen, test unseen. The challenge of R2R is to test the agent's generalization ability in unseen environments.

Evaluation Metrics The performance of different agents is evaluated using the following metrics:

- `TL` **Trajectory Length** measures the average length of the navigation trajectory.
- `NE` **Navigation Error** is the mean of the shortest path distance in meters between the agent's final location and the target location.
- `SR` **Success Rate** is the percentage of the agent's final location that is less than 3 meters away from the target location.
- `SPL` **Success weighted by Path Length [ACC+18]** trades-off `SR` against `TL`. A higher score represents more efficiency in navigation.

Among these metrics, `SPL` is the recommended primary metric, and other metrics are considered as auxiliary measures.

Baselines We compare our approach with *nine* recently published systems:

- RANDOM: an agent that randomly selects a direction and moves five step in that direction [AWT+18].

Table 2.1: **Bold** indicates the best value in a given setting. **S** indicates the single-instruction setting, **M** indicates the multiple-instruction setting.

		Validation Seen				Validation Unseen				Test Unseen			
	Agent	TL ↓	NE ↓	SR ↑	SPL ↑	TL ↓	NE ↓	SR ↑	SPL ↑	TL ↓	NE ↓	SR ↑	SPL ↑
Greedy, **S**	Random	9.58	9.45	16	-	9.77	9.23	16	-	9.93	9.77	13	12
	Seq2Seq	11.33	6.01	39	-	8.39	7.81	22	-	8.13	7.85	20	18
	RPA	-	5.56	43	-	-	7.65	25	-	9.15	7.53	25	23
	Speaker-Follower	-	3.36	66	-	-	6.62	35	-	14.82	6.62	35	28
	SMNA	-	-	-	-	-	-	-	-	18.04	5.67	48	35
	RCM+SIL(train)	10.65	3.53	67	-	11.46	6.09	43	-	11.97	6.12	43	38
	Regretful	-	3.23	69	63	-	5.32	50	41	13.69	5.69	48	40
	Fast	-	-	-	-	21.17	4.97	56	43	22.08	5.14	54	41
	EnvDrop	11.00	3.99	62	59	10.70	5.22	52	48	11.66	5.23	51	47
	Press	10.57	4.39	58	55	10.36	5.28	49	45	10.77	5.49	49	45
	Prevalent (ours)	10.32	3.67	69	**65**	10.19	**4.71**	**58**	**53**	10.51	5.30	**54**	**51**
M	Press	10.35	3.09	71	67	10.06	4.31	59	55	10.52	4.53	57	53
	Prevalent	**10.31**	3.31	67	63	**9.98**	**4.12**	**60**	**57**	**10.21**	**4.52**	**59**	**56**
	Human	-	-	-	-	-	-	-	-	11.85	1.61	86	76

- S2S-Anderson: a sequence-to-sequence model using a limited discrete action space [AWT+18].
- RPA [WXWW18]: an agent that combines model-free and model-based reinforcement learning, using a look-ahead module for planning.
- Speaker-Follower [FHC+18]: an agent trained with data augmentation from a speaker model on the panoramic action space.
- Smna [MLW+19]: an agent trained with a visual-textual co-grounding module and a progress monitor on the panoramic action space.
- RCM+SIL [WHC+19]: an agent trained with cross-modal grounding locally and globally via RL.
- Regretful [MWA+19]: an agent with a trained progress monitor heuristic for search that enables backtracking.
- Fast [KLB+19]: an agent that uses a fusion function to score and compare partial trajectories of different lengths, which enables the agent to efficiently backtrack after a mistake.
- EnvDrop [TYB19]: an agent is trained with environment dropout, which can

Figure 2.4: The percentage of pre-training datasets. The synthesized dataset occupies 98.4%.

generate more environments based on the limited seen environments.

- PRESS [LLX+19]: an agent is trained with pre-trained language models and stochastic sampling to generalize well in the unseen environment.

Comparison with SoTA Table 2.1 compares the performance of our agent against the existing published top systems.[3]. Our agent PREVALENT outperforms the existing models on SR and SPL by a large margin. On both validation seen and unseen environments, PREVALENT outperforms other agents on nearly all metrics.

In PRESS [LLX+19], multiple introductions are used. To have a fair comparison, we follow [LLX+19], and report PREVALENT results. We see that testing SPL is improved. Further, the gap between seen and unseen environments of PREVALENT is smaller than PRESS, meaning that image-attended language understanding is more effective to help the agent generalize better to an unseen environment. Results of adapting to new tasks and ablation studies are listed in Appendix.

2.6 Pre-training Dataset Preparation

We found that the largest VLN training dataset R2R contains only 104K samples, an order magnitude smaller than the pre-training datasets typically used in lan-

[3]The full list of leaderboard is publicly available: `https://evalai.cloudcv.org/web/challenges/challenge-page/97/leaderboard/270`

guage [DCLT19b] or vision-and-language pre-training [ZPZ+20]. This renders a case where pre-training can be degraded due to insufficient training data, while harvesting such samples with human annotations is expensive. Fortunately, we can resort to generative models to synthesize the samples. We first train an seq2seq auto-regressive model (*i.e.*, a speaker model [FHC+18]) that can produce language instructions conditioned on the agent trajectory (a sequence of actions and visual images) on R2R dataset; then collect a large number of shortest trajectories using the Matterport 3D Simulator, and synthesize their corresponding instructions using the speaker model. This leads to 6482K new training samples. The two datasets are compared in Figure 4(b). The agent is pre-trained on the combined dataset. Our results show that synthetic samples produced by generative models can be incorporated into the pre-training data and helps self-supervised learning.

2.7 Experiments

Table 2.2: Results on CVDN measured by Goal Progress. Bold indicates the best value in a given setting.

	Validation Unseen			Test Unseen		
Agent	**Oracle**	**Navigator**	**Mixed**	**Oracle**	**Navigator**	**Mixed**
Random	1.09	1.09	1.09	0.83	0.83	0.83
Seq2Seq	1.23	1.98	2.10	1.25	2.11	2.35
Prevalent (Ours)	**2.58**	**2.99**	**3.15**	**1.67**	**2.39**	**2.44**
Shortest Path Agent	8.36	7.99	9.58	8.06	8.48	9.76

2.7.1 Cooperative Vision-and-Dialogue Navigation

Dataset & Evaluation Metric The CVDN dataset has 2050 human-human navigation dialogs, comprising over 7K navigation trajectories punctuated by question-answer exchanges, across 83 MatterPort houses [CDF+17] . The metrics for R2R can be readily used for the CVDN dataset. Further, one new metric is proposed for the

NDH task:

GP **Goal Progress** measures the difference between completed distance and left distance to the goal. Larger values indicate a more efficient agent.

Three settings are considered, depending on which ground-truth action/path is employed [TMCZ19]. *Oracle* indicates the shortest path, and *Navigator* indicates the path taken by the navigator. The *Mixed* supervision path means to take the navigator path if available, otherwise the shortest path. The results are in Table 2.2. The proposed PREVALENT significantly outperforms the Seq2Seq baseline on both validation and testing unseen environments in all settings, leading to the top position on the leaderboard [4]. Note that our encoder is pre-trained on R2R dataset. We observe that it can provide significant improvement when used the new task built on the CVDN dataset. This shows that the pre-trained model can adapt well on new tasks, and yields better generalization.

Table 2.3: Results on test splits of HANNA.

		SEEN-ENV				UNSEEN-ALL			
	Agent	SR ↑	SPL ↑	NE ↓	#R ↓	SR ↑	SPL ↑	NE ↓	#R ↓
Rule	RANDOM WALK	0.54	0.33	15.38	0.0	0.46	0.23	15.34	0.0
	FORWARD 10	5.98	4.19	14.61	0.0	6.36	4.78	13.81	0.0
	NO ASSISTANCE	17.21	13.76	11.48	0.0	8.10	4.23	13.22	0.0
	ANNA	88.37	63.92	1.33	2.9	47.45	25.50	7.67	5.8
	PREVALENT (Ours)	83.82	59.38	1.47	3.4	**52.91**	**28.72**	**5.29**	6.6
Skyline	SHORTEST	100.00	100.00	0.00	0.0	100.00	100.00	0.00	0.0
	Perfect assistance	90.99	68.87	0.91	2.5	83.56	56.88	1.83	3.2

2.7.2 HANNA

Dataset & Evaluation Metric The HANNA dataset features 289 object types; the language instruction vocabulary contains 2,332 words. The numbers of locations

[4]The full list of leaderboard is publicly available: `https://evalai.cloudcv.org/web/challenges/challenge-page/463/leaderboard/1292`

on the shortest paths to the requested objects are restricted to be between 5 and 15. With an average edge length of 2.25 meters, the agent has to travel about 9 to 32 meters to reach its goals. Similar to R2R, SR, SPL and NE are used to evaluate the navigation. Further, one new metric is considered for this interactive task:

`#R` **Number of requests** measures how many helps are requested by the agent.

The results are shown in Table 2.3. Two rule-based methods and two skyline methods are reported as references; see [NDI19] for details. Our PREVALENT outperforms the baseline agent ANNA on the test unseen environments in terms of SR, SPL and NE, while requesting a slightly higher number of helps (#R). When measuring the performance gap between seen and unseen environments, we see that PREVALENT shows a significantly smaller difference than ANNA, *e.g.,* (59.38-28.72=30.66) vs (63.92-25.50=38.42) for SPL. This means that the pre-trained joint representation by PREVALENT can reduce over-fitting, and generalise better to unseen environments.

2.7.3 Ablation Studies

Table 2.4: Ablation study of the pre-training objectives on CVDN, measured by Goal Progress. Bold indicates the best value.

	Navigation QA			Oracle Answer			All		
Methods	**Oracle**	**Navigator**	**Mixed**	**Oracle**	**Navigator**	**Mixed**	**Oracle**	**Navigator**	**Mixed**
$\mathcal{L}_{\text{PA}} + \mathcal{L}_{\text{MLM}}$	**2.80**	**3.01**	**3.28**	2.78	**3.44**	**3.38**	**2.58**	**2.99**	**3.15**
$\mathcal{L}_{\text{MLM}}$	2.69	3.00	3.25	2.84	3.35	3.19	2.52	2.98	3.14
BERT pre-trainig	2.26	2.71	2.94	2.70	2.68	3.06	2.46	2.74	2.64
BERT fine-tuning	2.39	2.03	2.51	2.23	2.41	2.52	2.32	2.93	2.28

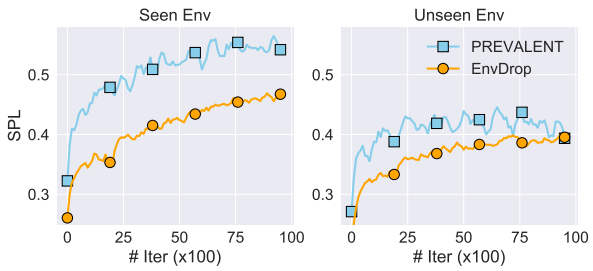
Table 2.5: Ablation study on R2R: feature-based vs fine-tuning. Bold indicates the better value.

	Validation Seen				Validation Unseen				Test Unseen			
Methods	`TL` ↓	`NE` ↓	`SR` ↑	`SPL` ↑	`TL` ↓	`NE` ↓	`SR` ↑	`SPL` ↑	`TL` ↓	`NE` ↓	`SR` ↑	`SPL` ↑
Two-stage	10.32	**3.67**	**0.69**	**0.66**	10.19	**4.71**	**0.58**	**0.53**	10.51	**5.30**	**0.54**	**0.51**
Feature-based	10.13	3.98	0.66	0.64	9.70	5.01	0.54	0.51	9.99	5.54	0.52	0.49

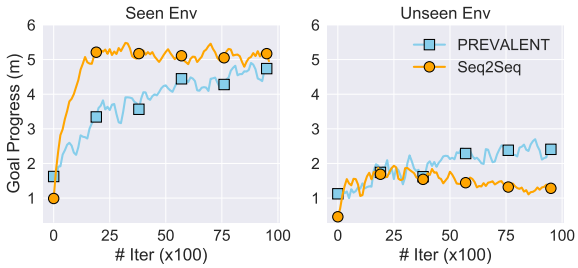
Is pre-training with actions helpful? Our pre-training objective in (2.9) includes two losses, $\mathcal{L}_{\text{PA}}$ and $\mathcal{L}_{\text{MLM}}$. To study the impact of each loss, we pre-train two model variants: one is based on the full objective $\mathcal{L}_{\text{PA}} + \mathcal{L}_{\text{MLM}}$, the other only uses $\mathcal{L}_{\text{MLM}}$. To verify its impact on new tasks, we consider CVDN first, and the results are shown in Table 2.4. Three types of text inputs are considered: Navigation QA, Orcale Answer, and All (a combination of both). More details are provided in the Appendix.

When $\mathcal{L}_{\text{PA}}$ is employed in the objective, we see consistent improvement on nearly all metrics and settings. Note that our MLM is different from BERT in that the attention over images is used in the cross-layer. To verify whether the image-attended learning is necessary, we consider BERT in two ways. (*i*) BERT pre-training: we apply the original MLM loss in BERT on our R2R pre-training dataset. The newly pre-trained BERT is used for fine-tuning on CVDN. (*ii*) BERT fine-tuning: we directly fine-tune off-the-shelf BERT on CVDN. Their performances are lower than the two variants of the proposed PREVALENT. This means our image-attended MLM is more effective for navigation tasks. More ablation studies on the pre-training objectives are conducted for HANNA, with results shown in the Appendix.

Feature-based vs Fine-tuning The pre-trained encoder can be used in two modes: (*i*) *fine-tuning* approach, where a task-specific layer is added to the pre-trained model, and all parameters are jointly updated on a downstream task. (*ii*) *feature-based* approach, where fixed features are extracted from the pre-trained model, and only the task-specific layer is updated. In this paper, all PREVALENT presented results generally have used the feature-based approach, as there are major computational benefits to pre-computing an expensive representation of the training data once, and then running many experiments with cheaper models on top of this representation. In the



(a) R2R



(b) CVDN

Figure 2.5: Learning curves on (a) R2R and (b) CVDN.

R2R dataset, we consider a *two-stage* scheme, where we fine-tune the cross-attention layers of the agent, after training via the feature-based approach. The results are reported in Table 2.5. We observe notable improvement with this two-stage scheme on nearly all metrics, expect the trajectory length.

How does pre-training help generalization? We plot the learning curves on the seen and unseen environments for R2R in Figure 2.5(a), and CVDN in Figure 2.5(b). Compared with the random initialized word embeddings in EnvDrop [TYB19], the pre-trained word embeddings can adapt faster (especially in the early stage), and converge to higher performance in unseen environments. This is demonstrated by the SPL values in the Figure 2.5(a). By comparing the learning curves in Figure 2.5(b), we see a much smaller gap between seen and unseen environments for PREVALENT than the Seq2Seq baseline [TMCZ19], meaning pre-training is an effective tool to help reduce over-fitting in learning.

Three types of inputs on CVDN We illustrate the naming of three types of text inputs on CVDN in Table 2.6.

Table 2.6: Three types of inputs on CVDN.

	V	t_0	A_i	Q_i	$Q_{1:i-1}\&A_{1:i-1}$
Oracle Answer	✓	✓	✓		
Navigation QA	✓	✓	✓	✓	
All	✓	✓	✓	✓	✓

Ablation Study Results on HANNA Table 2.7 shows the results with different pre-training objectives. We see that the $\mathcal{L}_{\text{PA}} + \mathcal{L}_{\text{MLM}}$ yields the best performance among all variants.

Table 2.7: Ablation study of pre-training objectives on test splits of HANNA.

	SEEN-ENV				UNSEEN-ALL			
Agent	SR ↑	SPL ↑	NE ↓	#R ↓	SR ↑	SPL ↑	NE ↓	#R ↓
PREVALENT ($\mathcal{L}_{\text{PA}} + \mathcal{L}_{\text{MLM}}$)	**83.82**	**59.38**	**1.47**	**3.4**	**52.91**	**28.72**	**5.29**	**6.6**
PREVALENT ($\mathcal{L}_{\text{MLM}}$)	78.75	54.68	1.82	4.3	44.29	24.27	6.33	8.1
BERT (feature-based)	57.54	34.33	4.71	3.9	24.12	11.50	9.55	11.3
BERT (fine-tuning)	80.75	57.46	1.97	4.0	26.36	12.66	9.1	8.3

2.8 Comparison with Related Work

Comparison with Press. The differences are summarized in Table 2.8 (a). Empirically, we show that (1) incorporating visual and action information into pre-training can improve navigation performance; (2) Pre-training can generalize across different new navigation tasks.

Comparison with vision-language pre-training (VLP). The differences are in Table 2.8 (b). Though the proposed methodology generally follows self supervised learning such as VLP or BERT, our research scope and problem setups are different, which renders existing pre-models are not readily applicable.

Table 2.8: Comparison with related works.

	Prevalent (Proposed)	**Press**
Dataset	Augmented R2R dataset	Generic language
Modality	Vision-language-action triplets	Language
Learning	Train from scratch	Off-the-shelf (BERT)
Downstream	Three navigation tasks	R2R

(a) PRESS

	Prevalent (Proposed)	**VLP**
Visual Input	Panoramic views (Size: 36 × 640 × 480)	Single image (Size: 640 ×480)
Visual Features	ResNet (View-level)	Fast RCNN (Object-level)
Objectives	Attentive MLM & Action Prediction	Masking on VL & Same-Pair Prediction
Downstream	RL: Navigation in sequential decision-making environments	Single-step prediction

(b) VLP

2.9 Conclusions

We present PREVALENT, a new pre-training and fine-tuning paradigm for vision-and-language navigation problems. This allows for more effective use of limited training data to improve generalization to previously unseen environments, and new tasks. The pre-trained encoder can be easily plugged into existing models to boost their performance. Empirical results demonstrate that PREVALENT significantly improves over existing methods, achieving new state-of-the-art performance.

Chapter 3

Efficient Knowledge Distillation for Large Scale Language Models

3.1 Introduction

Recent language models (LM) pre-trained on large-scale unlabeled text corpora in a self-supervised manner have significantly advanced the state of the art across a wide variety of natural language processing (NLP) tasks [DCLT19b, LOG+19, YDY+19, JCL+20, SWL+19, CLLM20, LLG+19, BDW+20]. After the LM pre-training stage, the resulting parameters can be fine-tuned to different downstream tasks. While these models have yielded impressive results, they typically have millions, if not billions, of parameters, and thus can be very expensive from storage and computaftional standpoints. Additionally, during deployment, such large models can require a lot of time to process even a single sample. In settings where computation may be limited (*e.g.* mobile, edge devices), such characteristics may preclude such powerful models from deployment entirely.

One promising strategy to compress and accelerate large-scale language models is knowledge distillation [ZGSZ19, TLL+19, SYS+20]. The key idea is to train a smaller model (a "student") to mimic the behavior of the larger, stronger-performing, but perhaps less practical model (the "teacher"), thus achieving similar performance with a faster, lighter-weight model. A simple but powerful method of achieving this is to use the output probability logits produced by the teacher model as soft labels for

training the student [HVD15]. With higher entropy than one-hot labels, these soft labels contain more information for the student model to learn from.

Previous efforts on distilling large-scale LMs mainly focus on designing better training objectives, such as matching intermediate representations [SCGL19b, MA19], learning multiple tasks together [LWL+19], or leveraging the distillation objective during the pre-training stage [JYS+19, SDCW19]. However, much less effort has been made to enrich task-specific data, a potentially vital component of the knowledge distillation procedure. In particular, tasks with fewer data samples provide less opportunity for the student model to learn from the teacher. Even with a well-designed training objective, the student model is still prone to overfitting, despite effectively mimicking the teacher network on the available data.

In response to these limitations, we propose improving the value of knowledge distillation by using data augmentation to generate additional samples from the available task-specific data. These augmented samples are further processed by the teacher network to produce additional soft labels, providing the student model more data to learn from a large-scale LM. Intuitively, this is akin to a student learning more from a teacher by asking more questions to further probe the teacher's answers and thoughts. In particular, we demonstrate that mixup [ZCDLP18] can significantly improve knowledge distillation's effectiveness, and we show with a theoretical framework why this is the case. We call our framework *MixKD*.

We conduct experiments on 6 GLUE datasets [WSM+19] across a variety of task types, demonstrating that *MixKD* significantly outperforms conventional knowledge distillation [HVD15] and other previous methods that compress large-scale language models. In particular, we show that our method is especially effective when the number of available task data samples is small, substantially improving the potency

of knowledge distillation. We also visualize representations learned with and without *MixKD* to show the value of interpolated distillation samples, perform a series of ablation and hyperparameter sensitivity studies, and demonstrate the superiority of *MixKD* over other BERT data augmentation strategies.

3.2 Related Work

3.2.1 Model Compression

Compressing large-scale language models, such as BERT, has attracted significant attention recently. Knowledge distillation has been demonstrated as an effective approach, which can be leveraged during both the pre-training and task-specific fine-tuning stages. Prior research efforts mainly focus on improving the training objectives to benefit the distillation process. Specifically, [TCLT19] advocate that task-specific knowledge distillation can be improved by first pre-training the student model. It is shown by [CLK$^+$19] that a multi-task BERT model can be learned by distilling from multiple single-task teachers. [LHCG19] propose learning a stronger student model by distilling knowledge from an ensemble of BERT models. Patient knowledge distillation (PKD), introduced by [SCGL19b], encourages the student model to mimic the teacher's intermediate layers in addition to output logits. DistilBERT [SDCW19] reduces the depth of BERT model by a factor of 2 via knowledge distillation during the pre-training stage. In this work, we evaluate *MixKD* on the case of task-specific knowledge distillation. Notably, it can be extended to the pre-training stage as well, which we leave for future work. Moreover, our method can be flexibly integrated with different KD training objectives (described above) to obtain even better results. However, we utilize the BERT-base model as the testbed in this paper without loss of generality.

3.2.2 Data Augmentation in NLP

Data augmentation (DA) has been studied extensively in computer vision as a powerful technique to incorporate prior knowledge of invariances and improve the robustness of learned models [SLDV98, SSP03, KSH12]. Recently, it has also been applied and shown effective on natural language data. Many approaches can be categorized as label-preserving transformations, which essentially produce neighbors around a training example that maintain its original label. For example, EDA [WZ19] propose using various rule-based operations such as synonym replacement, word insertion, swap or deletion to obtain augmented samples. Back-translation [YDL$^+$18, XDH$^+$19] is another popular approach belonging to this type, which relies on pre-trained translation models. Additionally, methods based on paraphrase generation have also been leveraged from the data augmentation perspective [KBBT19]. On the other hand, label-altering techniques like mixup [ZCDLP18] have also been proposed for language [GMZ19, CYY20], producing interpolated inputs and labels for the models predict. The proposed *MixKD* framework leverages the ability of mixup to facilitate the student learning more information from the teacher. It is worth noting that *MixKD* can be combined with arbitrary label-preserving DA modules. Back-translation is employed as a special case here, and we believe other advanced label-preserving transformations developed in the future can benefit the *MixKD* approach as well.

3.2.3 Mixup

Mixup [ZCDLP18] is a popular data augmentation strategy to increase model generalizability and robustness by training on convex combinations of pairs of inputs and

labels (x_i, y_i) and (x_j, y_j):

$$x' = \lambda x_i + (1 - \lambda) x_j \tag{3.1}$$

$$y' = \lambda y_i + (1 - \lambda) y_j \tag{3.2}$$

with $\lambda \in [0, 1]$ and (x', y') being the resulting virtual training example. This concept of interpolating samples was later generalized with Manifold mixup [VLB+19] and also found to be effective in semi-supervised learning settings [VLK+19, VQL+19, BCG+19, BCC+19]. Other strategies include mixing together samples resulting from chaining together other augmentation techniques [HMC+20], or replacing linear interpolation with the cutting and pasting of patches [YHO+19].

3.3 Methodology

3.3.1 Preliminaries

In NLP, an input sample i is often represented as a vector of tokens $\mathbf{w}_i = \{w_{i,1}, w_{i,2}, ..., w_{i,T}\}$, with each token $w_{i,t} \in \mathbb{R}^V$ a one-hot vector often representing words (but also possibly subwords, punctuation, or special tokens) and V being the vocabulary size. These discrete tokens are then mapped to word embeddings $\mathbf{x}_i = \{x_{i,1}, x_{i,2}, ..., x_{i,T}\}$, which serve as input to the machine learning model f. For supervised classification problems, a one-hot label $y_i \in \mathbb{R}^C$ indicates the ground-truth class of $\mathbf{x}_i$ out of C possible classes. The parameters θ of f are optimized with some form of stochastic gradient descent so that the output of the model $f(\mathbf{x}_i) \in \mathbb{R}^C$ is as close to y_i as possible, with cross-entropy as the most common loss function:

$$\mathcal{L}_{\text{MLE}} = -\frac{1}{n} \sum_{i}^{n} y_i \cdot \log(f(\mathbf{x}_i)) \tag{3.3}$$

where n is the number of samples, and $\cdot$ is the dot product.

3.3.2 Knowledge Distillation for BERT

Consider two models f and g parameterized by θ_T and θ_S, respectively, with $|\theta_T| \gg |\theta_S|$. Given enough training data and sufficient optimization, f is likely to yield better accuracy than g, due to higher modeling capacity, but may be too bulky or slow for certain applications. Being smaller in size, g is more likely to satisfy operational constraints, but its weaker performance can be seen as a disadvantage. To improve g, we can use the output prediction $f(\mathbf{x}_i)$ on input $\mathbf{x}_i$ as extra supervision for g to learn from, seeking to match $g(\mathbf{x}_i)$ with $f(\mathbf{x}_i)$. Given these roles, we refer to g as the student model and f as the teacher model.

While there are a number of recent large-scale language models driving the state of the art, we focus here on BERT [DCLT19b] models. Following [SCGL19b], we use the notation BERT_k to indicate a BERT model with k Transformer [VSP+17] layers. While powerful, BERT models also tend to be quite large; for example, the default `bert-base-uncased` (BERT_{12}) has ∼110M parameters. Reducing the number of layers (*e.g.* using BERT_3) makes such models significantly more portable and efficient, but at the expense of accuracy. With a knowledge distillation set-up, however, we aim to reduce this loss in performance.

3.3.3 Mixup Data Augmentation for Knowledge Distillation

While knowledge distillation can be a powerful technique, if the size of the available data is small, then the student has only limited opportunities to learn from the teacher. This may make it much harder for knowledge distillation to close the gap between student and teacher model performance. To correct this, we propose using data augmentation for knowledge distillation. While data augmentation [YDL+18, XDH+19, YHO+19, KBBT19, HMC+20, SZS+20, QSS+20] is a commonly used technique across machine learning for increasing training samples, robustness,

and overall performance, a limited modeling capacity constrains the representations the student is capable of learning on its own. Instead, we propose using the augmented samples to further query the teacher model, whose large size often allows it to learn more powerful features.

While many different data augmentation strategies have been proposed for NLP, we focus on mixup [ZCDLP18] for generating additional samples to learn from the teacher. Mixup's vicinal risk minimization tends to result in smoother decision boundaries and better generalization, while also being cheaper to compute than methods such as backtranslation [YDL+18, XDH+19]. Mixup was initially proposed for continuous data, where interpolations between data points remain in-domain; its efficacy was demonstrated primarily on image data, but examples in speech recognition and tabular data were also shown to demonstrate generality.

Directly applying mixup to NLP is not quite as straightforward as it is for images, as language commonly consists of sentences of variable length, each comprised of discrete word tokens. Since performing mixup directly on the word tokens doesn't result in valid language inputs, we instead perform mixup on the word embeddings at each time step $x_{i,t}$ [GMZ19]. This can be interpreted as a special case of Manifold mixup [VLB+19], where the mixing layer is set to the embedding layer. In other words, mixup samples are generated as:

$$x'_{i,t} = \lambda x_{i,t} + (1 - \lambda) x_{j,t} \quad \forall t \tag{3.4}$$

$$y'_i = \lambda y_i + (1 - \lambda) y_j \tag{3.5}$$

with $\lambda \in [0, 1]$; random sampling of λ from a Uniform or Beta distribution are common choices. Note that we index the augmented sample with i regardless of the value of λ. Sentence length variability can be mitigated by grouping mixup pairs by length. Alternatively, padding is a common technique for setting a consistent input

length across samples; thus, if $\mathbf{x}^{(i)}$ contains more word tokens than $\mathbf{x}^{(j)}$, then the extra word embeddings are mixed up with zero paddings. We find this approach to be effective, while also being much simpler to implement.

We query the teacher model with the generated mixup sample $\mathbf{x}'_i$, producing output prediction $f(\mathbf{x}'_i)$. The student is encouraged to imitate this prediction on the same input, by minimizing the objective:

$$\mathcal{L}_{\text{TMKD}} = d(f(\mathbf{x}'_i), g(\mathbf{x}'_i)) \tag{3.6}$$

where $d(\cdot, \cdot)$ is a distance metric for distillation, with temperature-adjusted cross-entropy and mean square error (MSE) being common choices.

Since we have the mixup samples already generated (with an easy-to-generate interpolated pseudolabel y'_i), we can also train the student model on these augmented data samples in the usual way, with a cross-entropy objective:

$$\mathcal{L}_{\text{SM}} = -\frac{1}{n}\sum_{i}^{n} y'_i \cdot \log(g(\mathbf{x}'_i)) \tag{3.7}$$

Our final objective for *MixKD* is a sum of the original data cross-entropy loss, student cross-entropy loss on the mixup samples, and knowledge distillation from the teacher on the mixup samples:

$$\mathcal{L} = \mathcal{L}_{\text{MLE}} + \alpha_{\text{SM}}\mathcal{L}_{\text{SM}} + \alpha_{\text{TMKD}}\mathcal{L}_{\text{TMKD}} \tag{3.8}$$

where α_{SM} and α_{TMKD} are hyperparameters weighting the loss terms.

3.3.4 Theoretical Analysis

We develop a theoretical foundation for the proposed framework. We wish to prove that by adopting data augmentation for knowledge distillation, one can achieve i) a

smaller gap between generalization error and empirical error, and *ii*) better generalization.

To this end, assume the original training data $\{\mathbf{x}_i\}_{i=1}^n$ are sampled i.i.d. from the true data distribution $p(\mathbf{x})$, and the augmented data distribution by mixup is denoted as $q(\mathbf{x})$ (apparently p and q are dependent). Let f be the teacher function, and $g \in \mathcal{G}$ be the learnable student function. Denote the loss function to learn g as $l(\cdot,\cdot)$[1]. The population risk w.r.t. $p(\mathbf{x})$ is defined as $\mathcal{R}(f,g,p) = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[l(f(\mathbf{x}),g(\mathbf{x}))\right]$, and the empirical risk as $\mathcal{R}_{emp}(f,g,\{\mathbf{x}_i\}_{i=1}^n) = \dfrac{1}{n}\sum_{i=1}^n l(f(\mathbf{x}_i),g(\mathbf{x}_i))$. A classic statement for generalization is the following: with at least $1-\delta$ probability, we have

$$\mathcal{R}(f,g_p,p) - \mathcal{R}_{emp}(f,g_p,\{\mathbf{x}_i\}_{i=1}^n) \le \epsilon, \tag{3.9}$$

where $\epsilon > 0$, and we have used g_p to indicate that the function is learned based on $p(\mathbf{x})$. Note different training data would correspond to a different error ϵ in (3.9). We use ϵ_p to denote the minimum value over all ϵ's satisfying (3.9). Similarly, we can replace p with q, and $\{\mathbf{x}_i\}_{i=1}^n$ with $\{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b$ in (3.9) in the data-augmentation case. In this case, the student function is learned based on both the training data and augmented data, which we denote as g^*. Similarly, we also have a corresponding minimum error, which we denote as ϵ^*. Consequently, our goal of better generalization corresponds to proving $\mathcal{R}(f,g^*,p) \le \mathcal{R}(f,g_p,p)$, and the goal of a smaller gap corresponds to proving $\epsilon^* \le \epsilon_p$. In our theoretical results, we will give conditions when these goals are achievable. First, we consider the following three cases about the joint data $\mathbf{X} \triangleq \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b$ and the function class $\mathcal{G}$:

- Case 1: There exists a distribution $\tilde{p}$ such that $\mathbf{X}$ are i.i.d. samples from it[2]; $\mathcal{G}$ is a finite set.

[1]This is essentially the same as $\mathcal{L}$ in (3.8). We use a different notation $l(f(\mathbf{x}),g(\mathbf{x}))$ to explicitly spell out the two data-wise arguments $f(\mathbf{x})$ and $g(\mathbf{x})$.

[2]We make such an assumption because $\mathbf{x}_i$ and $\mathbf{x}'_i$ are dependent, thus existence of $\tilde{p}$ is unknown.

- Case 2: There exists $\tilde{p}$ such that $\mathbf{X}$ are i.i.d. samples from it; $\mathcal{G}$ is an infinite set.
- Case 3: There does not exist a distribution $\tilde{p}$ such that $\mathbf{X}$ are i.i.d. samples from it.

Our theoretical results are summarized in Theorems 3.3.1-3.3.3, which state that with enough augmented data, our method can achieve smaller generalization errors. Proofs are given in the Section 3.5.

Theorem 3.3.1. *Assume the loss function $l(\cdot, \cdot)$ is upper bounded by $M > 0$. Under Case 1, there exists a constant $c > 0$ such that if*

$$b \geq \frac{M^2 \log(|\mathcal{G}|/\delta)}{c} - a$$

then

$$\epsilon^* \leq \epsilon_p$$

where ϵ^* and ϵ_p denote the minimal generalization gaps one can achieve with or without augmented data, with at least $1 - \delta$ probability. If further assuming a better empirical risk with data augmentation (which is usually the case in practice), i.e., $\mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}_i'\}_{i=1}^b) \leq \mathcal{R}_{emp}(f, g_p, \{\mathbf{x}_i\}_{i=1}^n)$, we have

$$\mathcal{R}(f, g^*, p) \leq \mathcal{R}(f, g_p, p)$$

Theorem 3.3.2. *Assume the loss function $l(\cdot, \cdot)$ is upper bounded by $M > 0$ and Lipschitz continuous. Fix the probability parameter δ. Under Case 2, there exists a constant $c > 0$ such that if*

$$b \geq \frac{M^2 \log(1/\delta)}{c} - a$$

then

$$\epsilon^* \leq \epsilon_p$$

where ϵ^* *and* ϵ_p *denote the minimal generalization gaps one can achieve with or without augmented data, with at least* $1-\delta$ *probability. If further assuming a better empirical risk with data augmentation, i.e.,* $\mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b) \leq \mathcal{R}_{emp}(f, g_p, \{\mathbf{x}_i\}_{i=1}^n)$*, we have*

$$\mathcal{R}(f, g^*, p) \leq \mathcal{R}(f, g_p, p)$$

A more interesting setting is Case 3. Our result is based on [Bax00], which studies learning from different and possibly correlated distributions.

Theorem 3.3.3. *Assume the loss function* $l(\cdot, \cdot)$ *is upper bounded. Under Case 3, there exists constants* $c_1, c_2, c_3 > 0$ *such that if*

$$b \geq \frac{a \log(4/\delta)}{c_1 a - c_2} \text{ and } a \geq c_3$$

then

$$\epsilon^* \leq \epsilon_p$$

where ϵ^* *and* ϵ_p *denote the minimal generalization gaps one can achieve with or without augmented data, with at least* $1-\delta$ *probability. If further assuming a better empirical risk with data augmentation, i.e.,* $\mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b) \leq \mathcal{R}_{emp}(f, g_p, \{\mathbf{x}_i\}_{i=1}^n)$*, we have*

$$\mathcal{R}(f, g^*, p) \leq \mathcal{R}(f, g_p, p)$$

Remark. *For Theorem 3.3.3 to hold, based on [Bax00], it is enough to ensure* $\{\mathbf{x}_i, \mathbf{x}'_i\}$ *and* $\{\mathbf{x}_j, \mathbf{x}'_j\}$ *to be independent for* $i \neq j$*. We achieve this by constructing* $\mathbf{x}'_i$ *with* $\mathbf{x}_i$

and an extra random sample from the training data. Since all $(\mathbf{x}_i, \mathbf{x}_j)$ *and the extra random samples are independent, the resulting concatenation will also be independent.*

3.4 Experiments

We demonstrate the effectiveness of MixKD on a number of GLUE [WSM+19] dataset tasks: Stanford Sentiment Treebank (SST-2) [SPW+13], Microsoft Research Paraphrase Corpus (MRPC) [DB05], Quora Question Pairs (QQP)[3], Multi-Genre Natural Language Inference (MNLI) [WNB18], Question Natural Language Inference (QNLI) [RZLL16], and Recognizing Textual Entailment (RTE) [DGM05, HDD+06, GMDD07, BCDG09]. Note that MNLI contains both an in-domain (MNLI-m) and cross-domain (MNLI-mm) evaluation set. These datasets span sentiment analysis, paraphrase similarity matching, and natural language inference types of tasks. We use the Hugging Face Transformers[4] implementation of BERT for our experiments.

3.4.1 Glue Dataset Evaluation

We first analyze the contributions of each component of our method, evaluating on the dev set of the GLUE datasets. For the teacher model, we fine-tune a separate 12 Transformer-layer `bert-base-uncased` (BERT_{12}) for each task. We use the smaller BERT_3 and BERT_6 as the student model. We find that initializing the embeddings and Transformer layers of the student model from the first k layers of the teacher model provides a significant boost to final performance. We use MSE as the knowledge distillation distance metric $d(\cdot, \cdot)$. We generate one mixup sample for each original sample in each minibatch (mixup ratio of 1), with $\lambda \sim \text{Beta}(0.4, 0.4)$. We set hyperparameters weighting the components in the loss term in (3.8) as

[3] `data.quora.com/First-Quora-Dataset-Release-Question-Pairs`

[4] `https://huggingface.co/transformers/`

Table 3.1: GLUE dev set results. We report the results of our BERT_{12} teacher model, the 6-layer DistilBERT, and 3- and 6-layer student models.

Model	SST-2	MRPC	QQP	MNLI-m	QNLI	RTE
BERT_{12}	92.20	90.53/86.52	88.21/91.25	84.12	91.32	77.98
DistilBERT_6	91.3	87.5/82.4	—-/88.5	82.2	**89.2**	59.9
BERT_6-FT	90.94	88.54/83.82	87.16/90.43	81.28	88.25	66.43
BERT_6-TMKD	91.63	88.93/83.82	86.60/90.27	81.49	88.71	65.34
BERT_6-SM+TMKD	91.17	89.30/84.31	87.19/90.56	82.02	88.63	65.34
BERT_6-FT+BT	91.74	**89.60/84.80**	87.06/90.39	82.10	87.68	67.51
BERT_6-TMKD+BT	91.86	89.52/84.56	87.15/90.59	82.17	88.38	**69.98**
BERT_6-SM+TMKD+BT	**92.09**	89.22/84.07	**87.57/90.78**	**82.53**	**88.82**	67.87
BERT_3-FT	87.16	81.68/71.08	84.99/88.65	75.55	83.98	58.48
BERT_3-TMKD	88.76	81.62/71.08	83.27/87.80	75.73	84.26	58.48
BERT_3-SM+TMKD	88.99	81.73/71.08	84.47/88.37	75.52	84.24	59.57
BERT_3-FT+BT	88.88	83.36/74.26	85.31/88.81	76.88	83.67	59.21
BERT_3-TMKD+BT	89.79	**84.46/75.74**	85.17/89.00	77.19	84.68	**62.82**
BERT_3-SM+TMKD+BT	**90.37**	84.14/75.25	**85.56/89.09**	**77.52**	**84.83**	60.65

$\alpha_{\text{SM}} = \alpha_{\text{TMKD}} = 1$.

As a baseline, we fine-tune the student model on the task dataset without any distillation or augmentation, which we denote as BERT_k-FT. We compare this against *MixKD*, with both knowledge distillation on the teacher's predictions ($\mathcal{L}_{\text{TMKD}}$) and mixup for the student ($\mathcal{L}_{\text{SM}}$), which we call BERT_k-SM+TMKD. We also evaluate an ablated version without the student mixup loss (BERT_k-TMKD) to highlight the knowledge distillation component specifically. We note that our method can also easily be combined with other forms of data augmentation. For example, backtranslation (translating an input sequence to the data space of another language and then translating back to the original language) tends to generate varied but semantically similar sequences; these sentences also tend to be of higher quality than masking or word-dropping approaches. We show that our method has an additive effect with other techniques by also testing our method with the dataset augmented with German backtranslation, using the `fairseq` [OEB+19] neural machine translation codebase to generate these additional samples. We also compare all of the aforementioned

Table 3.2: Computation cost comparison of teacher and student models on SST-2 with batch size of 16 on a Nvidia TITAN X GPU.

Model	Inference Speed (samples/second)	# of Parameters
BERT_{12} Teacher	115	109,483,778
BERT_6 Student	252	66,956,546
BERT_3 Student	397	45,692,930

variants with backtranslation samples augmenting the data; we denote these variants with an additional +BT.

We report the model accuracy (and F_1 score, for MRPC and QQP) in Table 3.1. We also show the performance of the full-scale teacher model (BERT_{12}) and DistilBERT [SDCW19], which performs basic knowledge distillation during BERT pre-training to a 6-layer model. For our method, we observe that a combination of data augmentation and knowledge distillation leads to significant gains in performance, with the best variant often being the combination of teacher mixup knowledge distillation, student mixup, and backtranslation. In the case of SST-2, for example, BERT_6-SM+TMKD+BT is able to capture 99.88% of the performance of the teacher model, closing 91.27% of the gap between the fine-tuned student model and the teacher, despite using far fewer parameters and having a much faster inference speed (Table 3.2).

After analyzing the contributions of the components of our model on the dev set, we find the SM+TMKD+BT variant to have the best performance overall and thus focus on this variant. We submit this version of *MixKD* to the GLUE test server, reporting its results in comparison with fine-tuning (FT), vanilla knowledge distillation (KD) [HVD15], and patient knowledge distillation (PKD) [SCGL19b] in Table 3.3. Once again, we observe that our model outperforms the baseline methods on most tasks.

Table 3.3: GLUE test server results. We show results for the full variants of the 3- and 6-layer student models.

Model	SST-2	MRPC	QQP	MNLI-m	MNLI-mm	QNLI	RTE
$BERT_{12}$	93.5	88.9/84.8	71.2/89.2	84.6	83.4	90.5	66.4
$BERT_6$-FT	90.7	85.9/80.2	69.2/88.2	80.4	79.7	86.7	63.6
$BERT_6$-KD	91.5	86.2/80.6	70.1/88.8	80.2	79.8	88.3	64.7
$BERT_6$-PKD	92.0	85.0/79.9	**70.7**/88.9	81.5	81.0	**89.0**	65.5
$BERT_6$-*MixKD*	**92.5**	**86.4/81.9**	70.5/**89.1**	**82.2**	**81.2**	88.2	**68.3**
$BERT_3$-FT	86.4	80.5/72.6	65.8/86.9	74.8	74.3	84.3	55.2
$BERT_3$-KD	86.9	79.5/71.1	67.3/87.6	75.4	74.8	84.0	56.2
$BERT_3$-PKD	87.5	80.7/72.5	**68.1/87.8**	76.7	76.3	**84.7**	58.2
$BERT_3$-*MixKD*	**89.5**	**83.3/75.2**	67.2/87.4	**77.2**	**76.8**	84.4	**62.0**

3.4.2 Limited-Data Settings

One of the primary motivations for using data augmentation for knowledge distillation is to give the student more opportunities to query the teacher model. For datasets with a large enough number of samples relative to the task's complexity, the original dataset may provide enough chances to learn from the teacher, reducing the relative value of data augmentation.

As such, we also evaluate *MixKD* with a $BERT_3$ student on downsampled versions of QQP, MNLI (matched and mismatched), and QNLI in Figure 3.1. We randomly select 10% and 1% of the data from these datasets to train both the teacher and student models, using the same subset for all experiments for fair comparison. In this data limited setting, we observe substantial gains from *MixKD* over the fine-tuned model for QQP (+2.0%, +3.0%), MNLI-m (+3.9%, +3.4%), MNLI-mm (+4.4%, +3.3%), and QNLI (+2.4%, +4.1%) for 10% and 1% of the training data.

3.4.3 Embeddings Visualization

We perform a qualitative examination of the effect of the proposed *MixKD* by visualizing the latent space between positive and negative samples as encoded by the

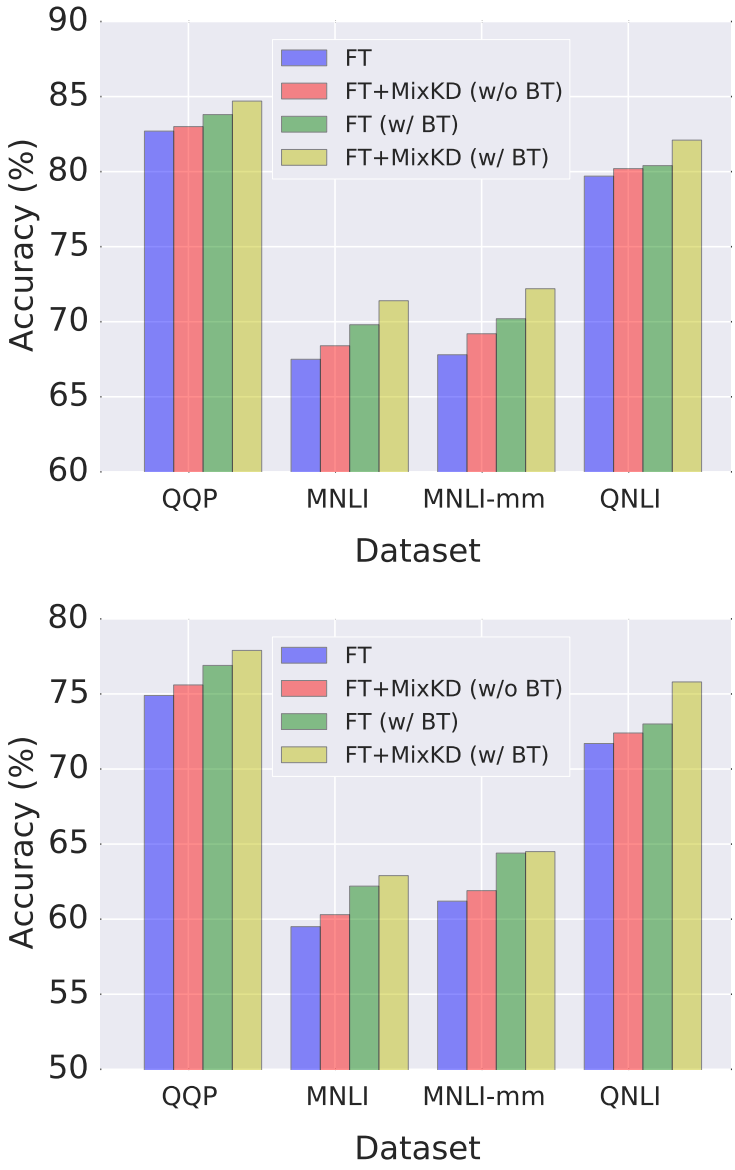
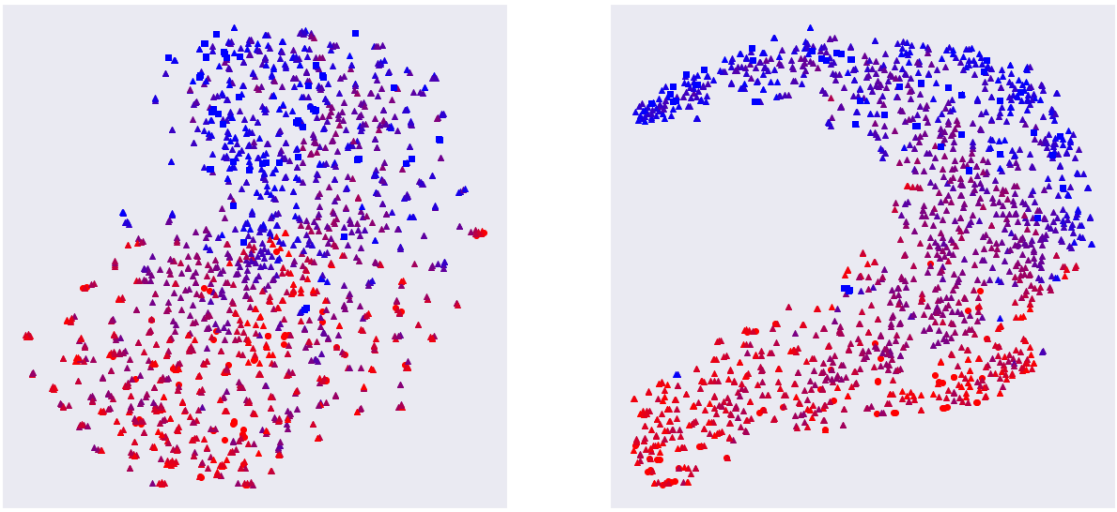


Figure 3.1: Results of limited data case, where both the teacher and student models are learned with only 10% (up) or 1% of the training data (down).

student model with t-SNE plots [MH08]. In Figure 3.2, we show the shift of the transformer features at the [CLS] token position, with and without mixup data augmentation from the teacher. We randomly select a batch of 100 sentences from the SST-2 dataset, of which 50 are positive sentiment (blue square) and 50 are negative sentiment (red circle). The intermediate mixup neighbours are indicated by triangles



(a) (b)

Figure 3.2: Latent space of randomly sampled training data and their mixup neighbours encoded by student model (a) standard fine-tuning (b) *MixKD*.

with color determined by the closeness to the positive group or negative group. From Figure 3.2a to Figure 3.2b, *MixKD* forces the linearly interpolated samples to be aligned with the manifold formed by the real training data and leads the student model to explore meaningful regions of the feature space effectively.

3.4.4 Hyperparameter Sensitivity & Further Analysis

Loss Hyperparameters Our final objective in equation 3.8 has hyperparameters α_{SM} and α_{TMKD}, which control the weight of the student model's cross-entropy loss for the mixup samples and the knowledge distillation loss with the teacher's predictions on the mixup samples, respectively. We demonstrate that the model is fairly stable over a wide range by sweeping both α_{SM} and α_{TMKD} over the range $\{0.1, 0.5, 1.0, 2.0, 10.0\}$. We do this for a BERT_3 student and BERT_{12} teacher, with SST-2 as the task; we show the results of this sensitivity study, both with and without German backtranslation, in Figure 3.3. Given the overall consistency, we observe

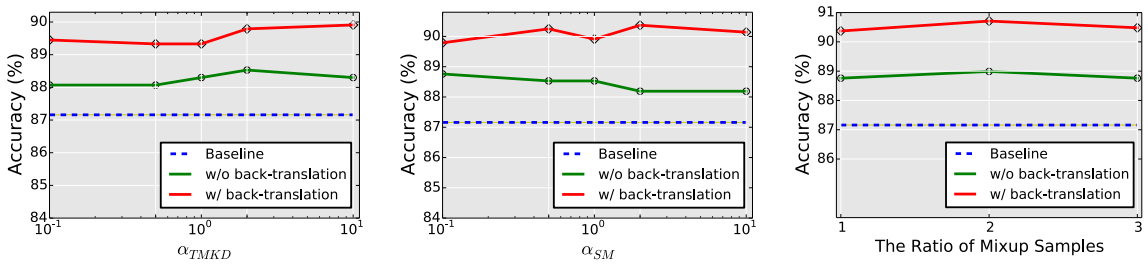


Figure 3.3: Hyperparameter analysis regarding *MixKD*, with different $\alpha_{\text{TMKD}}, \alpha_{\text{SM}}$ and the ratio of mixup samples (*w.r.t.* the original training data).

Table 3.4: We compare our approach with the data augmentation module proposed by TinyBert [JYS+19].

Methods	MNLI	SST-2
BERT_6	81.3	90.9
BERT_6 + TinyBERT DA module	81.5	91.3
BERT_6 + *MixKD*	**82.5**	**92.1**

that our method is stable over a wide range of settings.

Mixup Ratio We also investigate the effect of the mixup ratio: the number of mixup samples generated for each sample in a minibatch. We run a smaller sweep of α_{SM} and α_{TMKD} over the range $\{0.5, 1.0, 2.0\}$ for mixup ratios of 2 and 3 for a BERT_3 student SST-2, with and without German backtranslation, in Figure 3.3. We conclude that the mixup ratio does not have a strong effect on overall performance. Given that higher mixup ratio requires more computation (due to more samples over which to compute the forward and backward pass), we find a mixup ratio of 1 to be enough.

Comparing with TinyBERT's DA module TinyBERT [JYS+19] also utilizes data augmentation for knowledge distillation. Specifically, they adopt a conditional BERT contextual augmentation [WLZ+19] strategy. To further verify the effectiveness of our approach, we use TinyBERT's released codebase[5] to generate augmented

[5] `https://github.com/huawei-noah/Pretrained-Language-Model/tree/master/TinyBERT`

samples and make a direct comparison with *MixKD*. As shown in Table 3.4, our approach exhibits much stronger results for distilling a 6-layer BERT model (on both MNLI and SST-2 datasets). Notably, TinyBERT's data augmentation module is much less efficient than mixup's simple operation, generating 20 times the original data as augmented samples, thus leading to massive computation overhead.

3.5 Proofs

Proof of Theorem 3.3.1. First of all, $\{\mathbf{x}_i\}_{i=1}^{a} \cup \{\mathbf{x}'_i\}_{i=1}^{b}$ can be regarded as drawn from distribution $r(\mathbf{x}) = \dfrac{ap(\mathbf{x}) + bq(\mathbf{x})}{a+b}$.

Given $\mathcal{G}$ is finite, we have the following theorem

Theorem 3.5.1. *[MRT18] Let l be a bounded loss function, hypothesis set $\mathcal{G}$ is finite. Then for any $\delta > 0$, with probability at least $1-\delta$, the following inequality holds for all $g \in \mathcal{G}$:*

$$\mathcal{R}(f, g, p) - \mathcal{R}_{emp}(f, g, \{\mathbf{x}_i\}_{i=1}^{n}) \leq M\sqrt{\frac{\log(|\mathcal{G}|/\delta)}{2n}}$$

Thus we have in our case:

$$\mathcal{R}(f, g_p, p) - \mathcal{R}_{emp}(f, g_p, \{\mathbf{x}_i\}_{i=1}^{n}) \leq \epsilon_p \leq M\sqrt{\frac{\log(|\mathcal{G}|/\delta)}{2n}}$$

and

$$\begin{aligned}
&\mathcal{R}(f,g^*,p)-\mathcal{R}_{emp}(f,g^*,\{\mathbf{x}_i\}_{i=1}^a\cup\{\mathbf{x}'_i\}_{i=1}^b)\\
=&\mathcal{R}(f,g^*,r)-\mathcal{R}_{emp}(f,g^*,\{\mathbf{x}_i\}_{i=1}^a\cup\{\mathbf{x}'_i\}_{i=1}^b)+\int l(f(\mathbf{x}),g^*(\mathbf{x}))(p(\mathbf{x})-r(\mathbf{x}))d\,\mathbf{x}\\
=&\mathcal{R}(f,g^*,r)-\mathcal{R}_{emp}(f,g^*,\{\mathbf{x}_i\}_{i=1}^a\cup\{\mathbf{x}'_i\}_{i=1}^b)+\frac{b}{a+b}\int l(f(\mathbf{x}),g^*(\mathbf{x}))(p(\mathbf{x})-q(\mathbf{x}))d\,\mathbf{x}\\
\leq&\mathcal{R}(f,g^*,r)-\mathcal{R}_{emp}(f,g^*,\{\mathbf{x}_i\}_{i=1}^a\cup\{\mathbf{x}'_i\}_{i=1}^b)+\int l(f(\mathbf{x}),g^*(\mathbf{x}))(p(\mathbf{x})-q(\mathbf{x}))d\,\mathbf{x}\\
\leq&M\sqrt{\frac{\log(|\mathcal{G}|/\delta)}{2(a+b)}}+\triangle
\end{aligned}\tag{3.10}$$

where $\triangle=\int l(f(\mathbf{x}),g^*(\mathbf{x}))(p(\mathbf{x})-q(\mathbf{x}))d\,\mathbf{x}$. If

$$b\geq\frac{M^2\log(|\mathcal{G}|/\delta)}{2(\epsilon_p-\triangle)^2}-a$$

then

$$2(a+b)\geq\frac{M^2\log(|\mathcal{G}|/\delta)}{(\epsilon_p-\triangle)^2}$$

$$(\epsilon_p-\triangle)^2\geq\frac{M^2\log(|\mathcal{G}|/\delta)}{2(a+b)}$$

$$\epsilon_p\geq M\sqrt{\frac{\log(|\mathcal{G}|/\delta)}{2(a+b)}}+\triangle$$

Substitute into (3.10), we have

$$\mathcal{R}(f,g^*,p)-\mathcal{R}_{emp}(f,g^*,\{\mathbf{x}_i\}_{i=1}^a\cup\{\mathbf{x}'_i\}_{i=1}^b)\leq\epsilon_p$$

Recall the definition of ϵ^*, which is the minimum value of all possible ϵ satisfying

$$\mathcal{R}(f, g^*, p) - \mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b) \leq \epsilon$$

we know that $\epsilon^* \leq \epsilon_p$. Let $c = 2(\epsilon_p - \triangle)^2$, we can conclude the theorem.

□

Proof of Theorem 3.3.2. First of all, $\{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b$ can be regarded as drawn from distribution $r(\mathbf{x}) = \dfrac{ap(\mathbf{x}) + bq(\mathbf{x})}{a+b}$.

Theorem 3.5.2. *[MRT18] Let l be a non-negative loss function upper bounded by $M > 0$, and for any fixed $\mathbf{y}$, $l(\mathbf{y}, \mathbf{y}')$ is L-Lipschitz for some $L > 0$, then with probability at least $1 - \delta$,*

$$\mathcal{R}(f, g, p) - \mathcal{R}_{emp}(f, g, \{\mathbf{x}_i\}_{i=1}^n) \leq 2L\mathfrak{R}_p(\mathcal{G}) + M\sqrt{\frac{log(1/\delta)}{2n}}$$

Thus we have

$$\mathcal{R}(f, g, p) - \mathcal{R}_{emp}(f, g, \{\mathbf{x}_i\}_{i=1}^n) \leq \epsilon_p \leq 2L\mathfrak{R}_p(\mathcal{G}) + M\sqrt{\frac{log(1/\delta)}{2n}}$$

where $\mathfrak{R}_p(\mathcal{G})$ are Rademacher complexity over all samples of size n samples from $p(\mathbf{x})$.

We also have

$$\begin{aligned}
&\mathcal{R}(f, g^*, p) - \mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b) \\
=&\mathcal{R}(f, g^*, r) - \mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b) + \int l(f(\mathbf{x}), g^*(\mathbf{x}))(p(\mathbf{x}) - r(\mathbf{x}))d\,\mathbf{x} \\
=&\mathcal{R}(f, g^*, r) - \mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b) + \frac{b}{a+b}\int l(f(\mathbf{x}), g^*(\mathbf{x}))(p(\mathbf{x}) - q(\mathbf{x}))d\,\mathbf{x} \\
\leq&\mathcal{R}(f, g^*, r) - \mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b) + \int l(f(\mathbf{x}), g^*(\mathbf{x}))(p(\mathbf{x}) - q(\mathbf{x}))d\,\mathbf{x} \\
\leq&2L\mathfrak{R}_r(\mathcal{G}) + M\sqrt{\frac{log(1/\delta)}{2(a+b)}} + \triangle
\end{aligned} \tag{3.11}$$

where $\triangle = \int l(f(\mathbf{x}), g^*(\mathbf{x}))(p(\mathbf{x}) - q(\mathbf{x}))d\,\mathbf{x}$. $\mathfrak{R}_r(\mathcal{G})$ are Rademacher complexity over all samples of size $(a+b)$ samples from $r(\mathbf{x}) = \frac{ap(\mathbf{x}) + bq(\mathbf{x})}{a+b}$.

If

$$b \geq \frac{M^2 \log(1/\delta)}{2(\epsilon_p - \triangle - 2L\mathfrak{R}_r(\mathcal{G}))^2} - a$$

then:

$$2(a+b) \geq \frac{M^2 \log(1/\delta)}{(\epsilon_p - \triangle - 2L\mathfrak{R}_r(\mathcal{G}))^2}$$

$$\epsilon_p - \triangle - 2L\mathfrak{R}_r(\mathcal{G}) \geq M\sqrt{\frac{\log(1/\delta)}{2(a+b)}}$$

$$\epsilon_p \geq M\sqrt{\frac{\log(1/\delta)}{2(a+b)}} + \triangle + 2L\mathfrak{R}_r(\mathcal{G})$$

Substitute into (3.11), we have:

$$\mathcal{R}(f, g^*, p) - \mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b) \leq \epsilon_p$$

Recall the definition of ϵ^*, which is the minimum value of all possible ϵ satisfying

$$\mathcal{R}(f, g^*, p) - \mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b) \leq \epsilon$$

we know that $\epsilon^* \leq \epsilon_p$. Let $c = 2(\epsilon_p - \triangle - 2L\mathfrak{R}_r(\mathcal{G}))^2$, we can conclude the theorem. $\square$

Proof of Theorem 3.3.3. Similar to previous theorems, we write

$$\begin{aligned}
&\mathcal{R}(f, g^*, p) - \mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b) \\
=&\mathcal{R}(f, g^*, \frac{ap+bq}{a+b}) - \mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b) + \int l(f(\mathbf{x}), g^*(\mathbf{x}))(p(\mathbf{x}) - \frac{ap(\mathbf{x})+bq(\mathbf{x})}{a+b}) d\,\mathbf{x} \\
=&\mathcal{R}(f, g^*, \frac{ap+bq}{a+b}) - \mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b) + \frac{b}{a+b}\int l(f(\mathbf{x}), g^*(\mathbf{x}))(p(\mathbf{x}) - q(\mathbf{x})) d\,\mathbf{x} \\
\leq&\mathcal{R}(f, g^*, \frac{ap+bq}{a+b}) - \mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b) + \triangle \qquad (3.12)
\end{aligned}$$

where $\triangle = \int l(f(\mathbf{x}), g^*(\mathbf{x}))(p(\mathbf{x}) - q(\mathbf{x})) d\,\mathbf{x}$. For notation consistency, we write $\mathcal{R}(f, g^*, \frac{ap+bq}{a+b}) = \int l(f(\mathbf{x}) - g(\mathbf{x})) \frac{ap(\mathbf{x})+bq(\mathbf{x})}{a+b} d\,\mathbf{x}$. However, $\{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b$ are not drawn from the same distribution (which is $r(\mathbf{x}) = \frac{ap(\mathbf{x})+bq(\mathbf{x})}{a+b}$ in previous cases).

Let $\gamma = \lfloor \frac{a+b}{a} \rfloor$, we split $\{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b$ into γ parts that don't overlap with each other. The first part is $\{\mathbf{x}_i\}_{i=1}^a$, all the other parts has at least a elements from $\{\mathbf{x}'_i\}_{i=1}^b$.

Let

$$\lambda = \sqrt{\frac{64}{b}\log(4/\delta) + \frac{64}{a}\log C(\mathcal{G})}$$

where $C(\mathcal{G})$ is space capacity defined in Definition 4 in [Bax00], which depends on ϵ^* and $\mathcal{G}$.

By Theorem 4 in [Bax00],

$$\left[\mathcal{R}(f, g^*, \frac{ap+bq}{a+b}) - \mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b)\right]^2 \leq \max\{\frac{64}{\gamma a}\log(\frac{4C(\mathcal{G}^\gamma)}{\delta}), \frac{16}{a}\}$$

By Theorem 5 in [Bax00],

$$\frac{64}{\gamma a}\log(\frac{4C(\mathcal{G}^\gamma)}{\delta}) = \frac{64}{\gamma a}(\log(\frac{4}{\delta}) + \log(C(\mathcal{G}^\gamma))) \leq \frac{64}{\gamma a}(\log(\frac{4}{\delta}) + \gamma\log(C(\mathcal{G}))) \leq \lambda^2$$

The last inequality comes from $b \leq \gamma a$, which is because of $\gamma = \lfloor\frac{a+b}{a}\rfloor$. Then we have

$$\left[\mathcal{R}(f, g^*, \frac{ap+bq}{a+b}) - \mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b)\right]^2 \leq \max\{\frac{64}{\gamma a}\log(\frac{4C(\mathcal{G}^\gamma)}{\delta}), \frac{16}{a}\} \leq \max\{\lambda^2, \frac{16}{a}\} \quad (3.13)$$

If

$$b \geq \frac{64\log(4/\delta)}{(\epsilon_p - \triangle)^2 - 64\log C(\mathcal{G})/a}$$

Then

$$\lambda^2 \leq \frac{64}{a}\log C(\mathcal{G}) + 64\log(\frac{4}{\delta})\frac{(\epsilon_p - \triangle)^2 - 64\log C(\mathcal{G})/a}{64\log(4/\delta)}$$

$$\lambda^2 \leq (\epsilon_p - \triangle)^2 \quad (3.14)$$

If

$$\frac{16}{(\epsilon_p - \triangle)^2} \leq a$$

then

$$\frac{16}{a} \leq (\epsilon_p - \triangle)^2 \quad (3.15)$$

Combine (3.14) and (3.15), we have

$$\mathcal{R}(f, g^*, \frac{ap + bq}{a + b}) - \mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b) \leq \epsilon_p - \triangle$$

Substitute into (3.12), we have:

$$\mathcal{R}(f, g^*, p) - \mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b) \leq \epsilon_p$$

Recall the definition of ϵ^*, which is the minimum value of all possible ϵ satisfying

$$\mathcal{R}(f, g^*, p) - \mathcal{R}_{emp}(f, g^*, \{\mathbf{x}_i\}_{i=1}^a \cup \{\mathbf{x}'_i\}_{i=1}^b) \leq \epsilon$$

we know that $\epsilon^* \leq \epsilon_p$.

□

3.6 Variance Analysis

For the purpose of getting a sense of variance, we run experiments with additional random seeds on MRPC and RTE, which are relatively smaller datasets, and MNLI and QNLI, which are relatively larger datasets. Mean and standard deviation on the dev set of these GLUE datasets are reported in Table 3.5. We observe the variance of the same model's performance to be small, especially on the relatively larger datasets.

Table 3.5: Mean and variance reported for variants of BERT_6 and BERT_3.

Model	MRPC	MNLI-m	QNLI	RTE
BERT_6-TMKD+BT	**89.79**±0.27/**85.04**±0.48	82.05±0.11	88.42±0.06	**69.37**±0.50
BERT_6-SM+TMKD+BT	89.64±0.38/84.43±0.36	**82.41**±0.12	**88.76**±0.15	68.02±0.11
BERT_3-TMKD+BT	**84.79**±0.33/**75.82**±0.48	77.16±0.03	84.60±0.07	**62.47**±0.36
BERT_3-SM+TMKD+BT	84.53±0.39/75.85±0.60	**77.42**±0.11	**84.88**±0.06	60.83±0.18

3.7 Conclusions

We introduce *MixKD*, a method that uses data augmentation to significantly increase the value of knowledge distillation for compressing large-scale language models. Intuitively, *MixKD* allows the student model additional queries to the teacher model, granting it more opportunities to absorb the latter's richer representations. We analyze *MixKD* from a theoretical standpoint, proving that our approach results in a smaller gap between generalization error and empirical error, as well as better generalization, under appropriate conditions. Our approach's success on a variety of GLUE tasks demonstrates its broad applicability, with a thorough set of experiments for validation. We also believe that the *MixKD* framework can further reduce the gap between student and teacher models with the incorporation of more recent mixup and knowledge distillation techniques [LLY20, WLWG20, MFL$^+$19], and we leave this to future work.

Chapter 4

Partial Knowledge Share in Federated Learning via Weight Factorization

4.1 Introduction

With the rise of the Internet of Things (IoT), the proliferation of smart phones, and the digitization of records, modern systems generate increasingly large quantities of data. These data provide rich information about each individual, opening the door to highly personalized intelligent applications, but this knowledge can also be sensitive: images of faces, typing histories, medical records, and survey responses are all examples of data that should be kept private. Federated learning [MMR+17] has been proposed as a possible solution to this problem. By keeping user data on each local *client* device and only sharing model updates with the global *server*, federated learning represents a possible strategy for training machine learning models on heterogeneous, distributed networks in a privacy-preserving manner. While demonstrating promise in such a paradigm, a number of challenges remain for federated learning [LSTS19].

As with centralized distributed learning settings [DCM+12], many federated learning algorithms focus on learning a single global model. However, due to variation in user characteristics, personal data are likely to exhibit significant *statistical heterogeneity*. To simulate this, federated learning algorithms are commonly tested in non-i.i.d. settings [MMR+17, SCST17, LW19, PKM19], but data are often equally represented

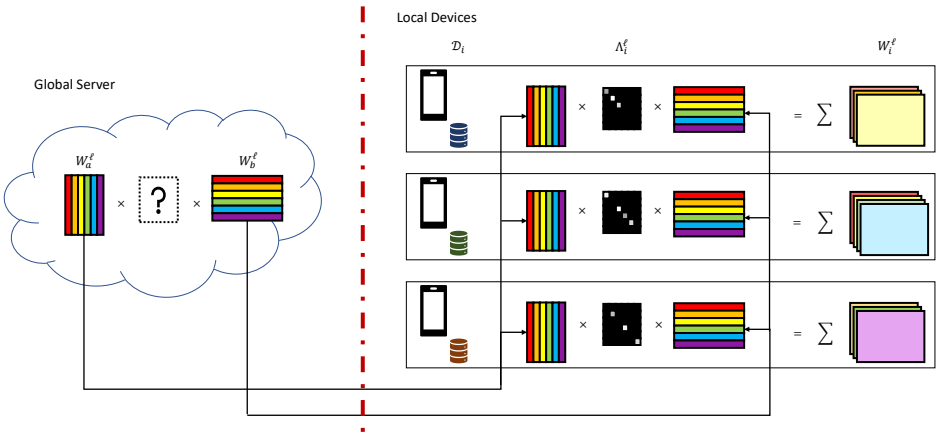


Figure 4.1: The clients only share weight factors $\{W_a^\ell, W_b^\ell\}$.Each client uses a sparse diagonal matrix Λ_i^ℓ constitute its own personalized model.

across clients and ultimately a single global model is typically learned. As is usually the case for one-size-fits-all solutions, while the model may perform acceptably on average for many users, some clients may see poor performance. Questions of fairness [MSS19, LSBS19] may arise if performance is compromised for individuals in the minority in favor of the majority.

Another challenge for federated learning is security. Data privacy is the primary motivation for keeping user data local on each device, rather than gathering it in a centralized location for training. In traditional distributed learning systems, data are exposed to additional vulnerabilities while being transmitted to and while residing in the central data repository. In lieu of the data, many federated learning approaches require clients to send weight updates to train the aggregated model. However, the threat of membership inference attacks [SSSS17, NSH19] or model inversion [FJR15, ZLH19] mean that private data on each device can still be compromised if federated learning updates are intercepted or if the central server is breached.

We propose **W**eight **A**nonymized **F**actorization for **F**ederated **Le**arning (WAFFLe), leveraging Bayesian nonparametrics and neural network weight factorization to ad-

dress these issues. Rather than learning a single global model, we learn a dictionary of rank-1 weight factor matrices. By selecting and weighting these factors, each local device can have a model customized to its unique data distribution, while sharing the learning burden of the weight factors across devices. We employ the Indian Buffet Process [GG06] as a prior to encourage factor sparsity and reuse of factors, performing variational inference to infer the distribution of factors for each client. While updates to the dictionary of factors are transmitted to the server, the distribution capturing which factors a client uses are kept local. This adds an extra insulating layer of security by obfuscating which factors a client is using, hindering an adversary's ability to perform membership inference attacks or dataset reconstruction.Finally, individually customized models represent in more fairness.

We perform experiments on MNIST [LBBH98], FMNIST [XRV17], and CIFAR-10 [Kri09] in settings exhibiting strong statistical heterogeneity. We observe that the model customization central to WAFFLe's design leads to higher performance for each client's local distribution, while also being significantly fairer across all clients. Finally, we perform membership inference [SSSS17] and model inversion [FJR15] attacks on WAFFLe, showing that it is much harder to expose user data than with FedAvg [MMR+17].

4.2 Methodology

4.2.1 Shared Dictionary of Weight Factors

Single Global Model Consider N client devices, with the i^{th} device having data distribution $\mathcal{D}_i$, which may differ as a function of i. In many distributed learning settings, a single global model is learned and deployed to all N clients. Thus, assuming a multilayer perceptron (MLP) architecture[1] with layers $\ell = 1, ..., L$, the set

[1]While we restrict our discussion to fully connected layers here for simplicity, this can be generalized

of weights $\theta = \{W^\ell\}_{\ell=1}^L$ is shared across all clients. To satisfy the global objective, θ is learned to minimize the loss on average across all clients. This is the approach of many federated learning approaches. For example, FedAvg [MMR+17] minimizes the following objective:

$$\min_\theta \mathscr{L}(\theta) = \sum_{i=1}^N p_i \mathcal{L}_i(\theta) \tag{4.1}$$

where $\mathcal{L}_i(\theta) \coloneqq \mathbb{E}_{x_i \sim \mathcal{D}_i}[l_i(x_i; \theta)]$ is the local objective function, N is the number of clients, and $p_i \geq 0$ is the weight of each device i. However, given statistical heterogeneity, such a one-size-fits-all approach may lead to the global model underfitting on certain clients; often this translates to how close a particular client's local distribution is to the population distribution. As a result, this model may be viewed as less fair to these clients with less common traits.

Individual Local Models On the other extreme, we may alternatively consider learning N local models $\theta_i = \{W_i^\ell\}_{\ell=1}^L$, each only trained on $\mathcal{D}_i$. In this case, each set of weights θ_i is maximally specific to the data distribution of each client i. However, each client typically has limited data, which may be insufficient for training a full model without overfitting; the total number of parameters that must be learned across all clients scales with N. Additionally, learning N separate models does not leverage similarities between client data distributions or the shared learning task.

Shared Weight Factors To make more efficient use of data, we instead propose a compromise between a single global model and N individual local models. Specifically, we allow each client's model to be personalized to the client's local distribution, but with all models sharing a dictionary of jointly learned components. Using a layer-wise decomposition [MLVC21], we construct each weight matrix with the following

to other types of layers as well. See Appendix 4.5 for 2D *convolutional* layers.

factorization:

$$W_i^\ell = W_a^\ell \Lambda_i^\ell W_b^\ell, \quad \Lambda_i^\ell = \mathrm{diag}(\boldsymbol{\lambda}_i^\ell) \tag{4.2}$$

where $W_a^\ell \in \mathbb{R}^{J \times F}$ and $W_b^\ell \in \mathbb{R}^{F \times M}$ are global parameters shared across clients and $\boldsymbol{\lambda}_i^\ell \in \mathbb{R}^F$ is a client-specific vector. This factorization can be equivalently expressed as

$$W_i^\ell = \sum_{k=1}^{F} \lambda_{i,k}^\ell \left(\boldsymbol{w}_{a,k}^\ell \otimes \boldsymbol{w}_{b,k}^\ell \right) \tag{4.3}$$

where $\boldsymbol{w}_{a,k}^\ell$ is the k^{th} column of W_a^ℓ, $\boldsymbol{w}_{b,k}^\ell$ is the k^{th} row of W_b^ℓ, and $\otimes$ represents an outer product. Written in this way, the interpretation of the corresponding pairs of columns and rows $\boldsymbol{w}_{a,k}^\ell$ and $\boldsymbol{w}_{b,k}^\ell$ as weight *factors* is more apparent: W_a^ℓ and W_b^ℓ together comprise a global dictionary of the weight factors, and $\boldsymbol{\lambda}_i^\ell$ can be viewed as the factor *scores* of client i used to select the corresponding rank-1 matrices formed using weight factors. Differences in $\boldsymbol{\lambda}_i^\ell$ between clients allows for customization of the model to each client's data distribution (see Figure 4.1), while sharing of the underlying factors W_a^ℓ and W_b^ℓ enables learning from the data of all clients.

We constitute each of the client's factor scores $\boldsymbol{\lambda}_i^\ell$ as the element-wise product:

$$\boldsymbol{\lambda}_i^\ell = \boldsymbol{r}^\ell \odot \boldsymbol{b}_i^\ell \tag{4.4}$$

where $\boldsymbol{r}^\ell \in \mathbb{R}^F$ indicates the strength of each factor and $\boldsymbol{b}_i^\ell \in \{0,1\}^F$ is a binary vector indicating the active factors. As explained below, $\boldsymbol{b}_i^\ell$ is typically sparse, so in general each client only uses a small subset of the available weight factors. Throughout this work, we use the absence of the ℓ superscript (*e.g.*, $\boldsymbol{\lambda}_i$) to refer to the entire collection across all layers for which this factorization is done. We learn a point-estimate for W_a, W_b and $\boldsymbol{r}$.

Algorithm 1 Updating Scheme in Each Communication

Input: local training epochs E, learning rate η
Server randomly selects subset $\mathcal{S}_t$ of clients
Server sends $\{W_a, \boldsymbol{r}, W_b\}$ to $\mathcal{S}_t$
for client $i \in \mathcal{S}_t$ **in parallel do**
    $W_a, \boldsymbol{r}, W_b, \boldsymbol{\pi}_i \boldsymbol{c}_i, \boldsymbol{d}_i \leftarrow \textsc{ClientStep}(W_a, \boldsymbol{r}, W_b, \boldsymbol{\pi}_i, \boldsymbol{c}_i, \boldsymbol{d}_i)$
    Send $\{W_a, \boldsymbol{r}, W_b\}$ to the server.
end for
Server aggregates and averages updates $\{W_a, \boldsymbol{r}, W_b\}$

Algorithm 2 Client Updating Function

function $\textsc{ClientStep}(W_a, \boldsymbol{r}, W_b, \boldsymbol{\pi}_i, \boldsymbol{c}_i, \boldsymbol{d}_i)$
    for $e = 1, \cdots, E$ **do**
        for minibatch $b \in \mathcal{D}_i$ **do**
            Update $\{W_a, \boldsymbol{r}, W_b, \boldsymbol{\pi}_i, \boldsymbol{c}_i, \boldsymbol{d}_i\}$ by minimizing (4.11)
        end for
    end for
    return $W_a, \boldsymbol{r}, W_b, \boldsymbol{\pi}_i, \boldsymbol{c}_i, \boldsymbol{d}_i$
end function

4.2.2 The Indian Buffet Process

Desiderata Within the context of federated learning with statistical heterogeneity, there are a number of desirable properties we wish the client factor scores to have collectively. Firstly, $\boldsymbol{\lambda}_i$ should be *sparse*, which encourages consolidation of related knowledge while minimizing interference: client A should be able to update the global factors during training without destroying client B's ability to perform its own task. This encourages *fairness*, as in settings with multiple subpopulations, this interference is most likely to be at the smaller groups' expense. On the other hand, we would also like factors to be reused among clients. While data may be non-i.i.d. across clients, there are often some similarities; thus, *shared* factors distribute learning across all clients' data, avoiding the N independent model's scenario. Finally, in the distributed settings considered in federated learning, the total number of nodes is rarely pre-defined. Therefore, there needs to be a way to gracefully *expand* to accommodate new clients to the system without re-initializing the whole model. This includes both

increasing server-side capacity if necessary and initializing new clients.

Prior Given these desiderata, the Indian Buffet Process (IBP) [GG06] is a natural choice. As a prior, the IBP regularizes client factors to be sparse, and new factors are introduced but at a harmonic rate, preferring reusing factors as much as possible over initializing new ones. This Bayesian nonparametric approach allows the data to dictate client factor assignment, factor reuse, and server-side model expansion. We use the stick-breaking construction of the IBP [TGG07] as a prior for the factor selection:

$$v_{i,\kappa}^{\ell} \sim \text{Beta}(\alpha, 1) \tag{4.5}$$

$$\pi_{i,k}^{\ell} = \prod_{\kappa=1}^{k} v_{i,\kappa}^{\ell} \tag{4.6}$$

$$b_{i,k}^{\ell} \sim \text{Bernoulli}(\pi_{i,k}^{\ell}) \tag{4.7}$$

where k indexes the factor, $\pi_{i,k}^{\ell}$ denotes the probability of the k^{th} factor being active, and α is a hyperparameter controlling the expected number of active factors and the rate of new factors being incorporated. Note that in the stick-breaking construction, $\pi_{i,k}^{\ell}$ is generated using a cumulative product of Beta random variables ($v_{i,\kappa}^{\ell}$).

Inference We learn the posterior distribution for the random variables $\boldsymbol{\phi}_i = \{\boldsymbol{b}_i, \boldsymbol{v}_i\}$. Exact inference of the posterior is intractable, so we employ variational inference with mean-field approximation to determine the active factors for each client device, using the variational distributions:

$$q(\boldsymbol{b}_i^{\ell}, \boldsymbol{v}_i^{\ell}) = q(\boldsymbol{b}_i^{\ell}) q(\boldsymbol{v}_i^{\ell}) \tag{4.8}$$

$$\boldsymbol{b}_i^{\ell} \sim \text{Bernoulli}(\boldsymbol{\pi}_i^{\ell}) \tag{4.9}$$

$$\boldsymbol{v}_i^{\ell} \sim \text{Kumaraswamy}(\boldsymbol{c}_i^{\ell}, \boldsymbol{d}_i^{\ell}) \tag{4.10}$$

learning the variational parameters $\{\boldsymbol{\pi}_i, \boldsymbol{c}_i, \boldsymbol{d}_i\}$ for each queried client using Bayes

by Backprop [BCKW15]. Needing a differentiable parameterization, we use the Kumaraswamy distribution [Kum80] as a replacement for the Beta distribution of $\boldsymbol{v}_i$ and utilize a soft relaxation of the Bernouilli distribution [MMT17]. The objective for each client is to maximize the variational lower bound:

$$\begin{aligned}\mathcal{L}_i(\theta) &= \sum_{n=1}^{|\mathcal{D}_i|} \mathbb{E}_q \log p\left(y_i^{(n)} | \boldsymbol{\phi}_i, x_i^{(n)}, W_a, W_b, \boldsymbol{r}\right) \\ &\quad - \underbrace{\mathrm{KL}\left(q\left(\boldsymbol{\phi}_i\right) || p\left(\boldsymbol{\phi}_i\right)\right)}_{\mathscr{R}} \end{aligned} \tag{4.11}$$

$$\mathscr{R} = \sum_{\ell=1}^{L} \mathbb{E}_{q(\boldsymbol{v}_i^\ell)}\left[\mathrm{KL}\left(q(\boldsymbol{b}_i^\ell) || p(\boldsymbol{b}_i^\ell | \boldsymbol{v}_i^\ell)\right)\right] + \mathrm{KL}\left(q(\boldsymbol{v}_i^\ell) || p(\boldsymbol{v}_i^\ell)\right)$$

where $\theta = \{W_a, W_b, \boldsymbol{r}, \boldsymbol{b}_i\}$ and $|\mathcal{D}_i|$ is the number of training examples at client i. Note that in (4.11) the first term provides label supervision and the second term ($\mathscr{R}$) regularizes the posterior with the IBP prior. The KL divergence in $\mathscr{R}$ is approximated by sampling from the posterior distribution.

4.2.3 Client-Server Communication

Training Before the training begins, the global weight factors $\{W_a, W_b\}$ and the factor strengths $\boldsymbol{r}$ are initialized by the server. Once initialized, each training round begins with $\{W_a, W_b, \boldsymbol{r}\}$ being sent to the selected subset of clients. Each sampled client then trains the model on their own private dataset $\mathcal{D}_i$ for E epochs, updating not only the weight factor dictionary $\{W_a, W_b\}$ and the factor strengths r, but also its also own variational parameters $\{\boldsymbol{\pi}_i, \boldsymbol{c}_i, \boldsymbol{d}_i\}$, which controls which factors it uses. Once local training is finished, each client sends $\{W_a, W_b, \boldsymbol{r}\}$ back to the server, but not $\{\boldsymbol{\pi}_i, \boldsymbol{c}_i, \boldsymbol{d}_i\}$, which remain with the client with data $\mathcal{D}_i$. After the server has received back updates from all clients, the various new values for $\{W_a, W_b, \boldsymbol{r}\}$ are aggregated with a simple averaging step. The process then repeats, with the server selecting a new subset of clients to query, sending the new updated set of global

parameters, until the desired number of communication rounds have passed. This process is summarized in Algorithm 1 and 2.

Evaluation When a client enters the evaluation mode, it requests the current version of global parameters $\{W_a, W_b, \boldsymbol{r}\}$ from the server. If the client has been previously queried for federated training, the local model consists of the aggregated global parameters and the factor score vector generated by its own local variational parameters $\{\boldsymbol{\pi}_i\}$. Otherwise, the client uses only the aggregated $\{W_a, W_b, \boldsymbol{r}\}$. Note that if a client has been previously queried, the most recently cached copy of the global parameters is an option if a network connection is unavailable or too expensive; in our experiments, we assume clients are able to request the most up-to-date parameters.

Security Data security is one of the central tenets of federated learning. Simpler, more standard methods of training a model could be utilized if all data were first aggregated at a central server. However, sensitive client data being intercepted during transmission or the server's data repository being breached by an attacker are major concerns, motivating federated learning's approach of keeping the data on the local device. On the other hand, keeping the data client-side may not be sufficient. Just as data can be compromised in transit or at the central database in non-federated settings, federated training updates are similarly vulnerable. In methods like FedAvg, this update is the entirety of the model's parameters. Effectively, this means that FedAvg trades yielding the data immediately for surrendering whitebox access to the model, which opens the model to a wide range of malicious activities [SZS+14, FJR15, SSSS17, WSZ+19, ZLH19], including, critically, exposing the very data that federated learning aims to protect. With WAFFLe, clients transmit back the entire dictionary of weight factors $\{W_a, W_b\}$ and $\boldsymbol{r}$, but not $\{\boldsymbol{\pi}_i, \boldsymbol{c}_i, \boldsymbol{d}_i\}$. As such, the knowledge of which specific factors that a particular client uses is kept local. Therefore, even if messages are intercepted, an adversary cannot completely reconstruct the model,

hampering their ability to perform attacks to recover the data.

4.3 Related Work

4.3.1 Statistical Heterogeneity

Statistical heterogeneity of the data distributions of client devices has long been recognized as a challenge for federated learning. Despite acknowledging statistical heterogeneity, many federated learning algorithms focus on learning a single global model [MMR+17]; such an approach often suffers from model divergence, as local models may vary significantly from each other. To address this, a number of works break away from the single-global-model formulation. Several [SCST17, CB19] have cast federated learning as a multi-task learning problem, with each client treated as a separate task. FedProx [LSZ+18] adds a proximal term to account for statistical heterogeneity by limiting the impact of local updates. Others study federated learning within a model-agnostic meta-learning framework [JKRK19, KBT19]. [ZLL+18] recognize performance degradation from non-i.i.d. data and propose global sharing of a small subset of data, which while effective, may compromise privacy. In settings of high statistical heterogeneity, fairness is also a natural question. AFL [MSS19] and q-FFL [LSBS19] propose methods of focusing the optimization objective on the clients with the worst performance, though they do not change the network itself to model different data distributions.

4.3.2 Preserving Data Safety

While much progress has been made in machine learning with public datasets [LBBH98, Kri09, DDS+09], in real-world settings, data are often sensitive, potentially for propriety [**?**], security [LHG+18], or privacy [RHU+18] reasons. Protecting user data is one of the primary motivations for federated learning in the first place.

Approaches include releasing artificial data [TF20, GT20], homomorphic encryption [HHIL+17], or differential privacy [DMNS06, ACG+16, MDDC15]. However, artificial data can still strongly resemble the original data, and sharing the model architecture and its parameters presents risks associated with whitebox access, leaving the data vulnerable to attacks such as membership inference [SSSS17] or model inversion [FJR15, WSZ+19, ZLH19].

4.3.3 Bayesian Nonparametric Federated Learning

Several previous works have applied Bayesian nonparameterics to federated learning, primarily as a means for parameter matching during aggregation. Instead of averaging the parameters weight-wise without considering the meaning of each parameter, past works have proposed using the Beta-Bernouilli Process [TJ07] for matching parameters, first with fully connected layers [YAG+19], but later also extended by [WYS+20] to convolutions and LSTMs [HS97a]. In contrast, our method utilizes Bayesian nonparametrics for modeling rank-1 factors for multitask learning, instead of the aggregation stage.

4.3.4 Personalized Federated Learning

Personalized FL models has become a recent focus. One approach is to mix the global and local model parameters during optimization [HR20]. However, this requires meta-features from each client, which partially violates the goal of privacy in FL. Another commonly used strategy is splitting neural networks [AASC19, YZH+21]: the model is divided into two parts, the feature extractor and the personalized layers. The feature extractor is aggregated and shared by the server, and both parts are trained for a personalized model. Recent work also explore meta-learning, particularly model-agnostic meta-learning (MAML) [FAL17]. For example, Per-FedAvg [FMO20] builds

a meta-model initialization that is then updated by a gradient step for a personalized model. However, meta-optimization often requires computing second-order derivatives, which can be computationally prohibitive for FL. pFedMe [DTN20] proposes decoupling the process of optimizing personalized models from learning the global model. pFedMe keeps the learning process of FedAvg while optimizing the personalized model in parallel, showing performance superiority over Per-FedAvg [FMO20].

4.4 Experiments

4.4.1 Experimental Set-up

Statistical Heterogeneity

Settings with higher statistical heterogeneity are more challenging for federated learning than when data are i.i.d. across clients, as well as more representative of the real-world, so we focus our experiments on the former. We consider two forms of statistical heterogeneity.

Unimodal non-i.i.d. We first consider the non-i.i.d. setting introduced by [MMR+17]. This is a widely used evaluation setting, commonly referred to as "non-i.i.d." or "heterogeneous" in other federated learning works, to distinguish it from completely i.i.d. data splits. We refer to this as *unimodal non-i.i.d.* to distinguish it from our second setting, which is also non-i.i.d. The primary purpose of such a partition is to investigate the behavior of federated average algorithms when each client has data from only a subset (Z) of classes.

This type of partition begins by sorting all data by class. Given N client devices, the samples from each class are evenly divided into shards of data, each consisting of a single class, resulting in NZ shards across all classes. These shards are then randomly distributed to the N clients such that each receives Z shards. The data in

the Z shards for each client is then shuffled together and split into a local training and test set. This ensures that the local test set for each client is representative of its own private data distribution.While this setting can be challenging, it has the property that the classes present in every client's data is equally represented in the global data distribution.As a result, a single global model may perform reasonably uniformly across all clients.

Multimodal non-i.i.d. While the *unimodal non-i.i.d.* partition does explore the non-i.i.d. nature of class distribution among clients, it does not adequately characterize the tendency for subpopulations to exist, with some being more prevalent than others. We propose a new non-i.i.d. setting to capture this, which we call *multimodal non-i.i.d.*, as each subpopulation group can be thought of as a mode of the overall distribution. In the real world, the mode can correspond to age, gender, ethnicity, wealth, or a number of other demographic factors. The number of subpopulation group is arbitrary, but we choose two for simplicity, creating "majority" and "minority" subpopulations. In our experiments, the two modes are odd digits ($N_1 = 100$) versus even digits ($N_2 = 20$) for MNIST [LBBH98], footwear and shirts ($N_2 = 20$) versus everything else ($N_1 = 90$) for FMNIST [XRV17], and animals ($N_1 = 90$) versus vehicles ($N_2 = 20$) for CIFAR-10 [Kri09], where N_1 and N_2 are the number of clients in the majority and minority subpopulations, respectively.

Once the classes have been separated by group, the process proceeds similarly to the unimodal i.i.d. partition process, with the data being divided into shards and then randomly allocated to clients within each subpopulation. We make the shards equal in size both within and across modes, so in instances where there are more data shards available than there are clients, we discard the unallocated data. Just as for unimodal non-i.i.d., local training and test sets are created for each client from its allocated data. An example multimodal non-i.i.d for MNIST is shown in Figure 4.2.

Compared with unimodal non-i.i.d, the difference is that there is now a $5:1$ ratio of odd to even digits in the total population, resulting in the clients with only even digits being in the minority of the global population.

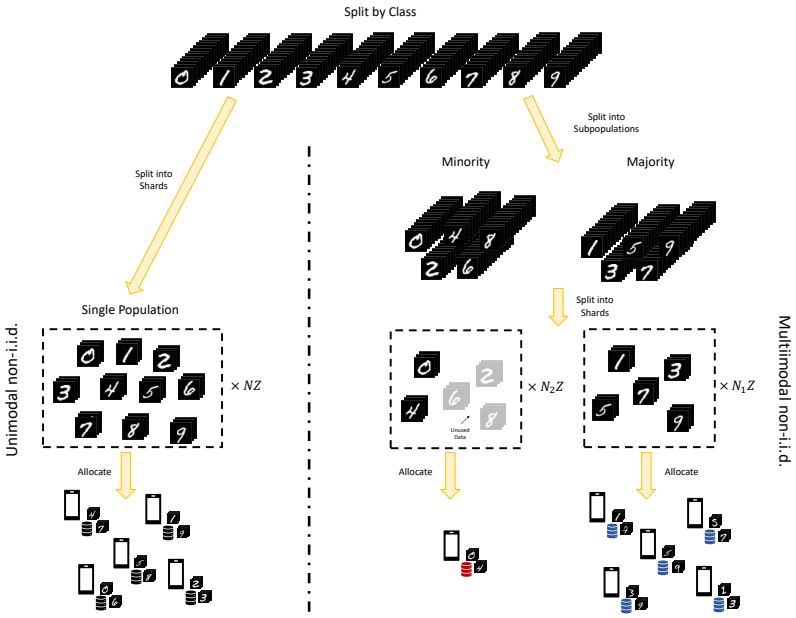


Figure 4.2: Example data allocation process to N clients for MNIST and $Z = 2$ in the unimodal i.i.d. (left) and multimodal i.i.d. (right) settings.

Model Architecture and Training Setting

For MNIST [LBBH98] digit recognition, we use a multilayer perceptron with 1-hidden layer with 200 units using ReLU activations [NH10]. Based on this model, we constructed WAFFLe with $F = 120$ factors. The traditional 60K training examples are partitioned into local training and test sets as described in Section 4.4.1. Stochastic gradient descent (SGD) with learning rate $\eta = 0.04$ is employed for all methods.

For FMNIST [XRV17] fashion recognition, we use a convolutional network consisting of two 5×5 convolution layers with 16 and 32 output channels respectively. Each convolution layer is followed by a 2×2 maxpooling operation with ReLU activations. A fully connected layer with a softmax is added for the output. Based on this model,

we construct WAFFLe by only factorizing the convolution layers, with $F = 25$ factors. As with MNIST, the traditional 60K training examples are used to form the two local sets. SGD with learning rate $\eta = 0.02$ is used as the optimizer for all methods.

For CIFAR-10 [Kri09], we use we use a convolutional network consisting of two 3×3 convolution layers with 16 and 16 output channels respectively. Each convolution layer is followed by a 2×2 maxpooling operation with ReLU activations. These two convolutions are followed by two fully-connected layers with hidden size 80 and 60, with a softmax applied for the final output probabilities. To construct WAFFLe, we set the number of factors $F = 10$ for the two convolution layers, $F = 80$ for the first fully connected layer, and $F = 40$ for the second fully connected layer. The 50K training examples are used for constructing the local train and test sets. SGD with learning rate $\eta = 0.02$ is utilized for all methods.

For all federated learning methods, the server selects a fraction $C = 0.1$ of clients during each communication round, with $T = 100$ total rounds for all methods. Each selected client trains their own model for $E = 5$ local epochs with mini-batch size $B = 10$.

For FedProx [LSZ+18], the proximal parameter μ is set to 1.0. For q-FFL [LSBS19], we searched $q \in \{0.001, 0.005, 0.01, 0.1, 1, 3, 5\}$ and found $q = 0.001$ as the best setting, matching the settings of [LSBS19] on more complex data.

4.4.2 Local Test Performance

We compare WAFFLe with FedAvg [MMR+17], the fairness-oriented q-FFL [LSBS19], and FedProx [LSZ+18], which augments FedAvg with a proximal term designed for high statistical heterogeneity. We record local test performance averaged across all clients for both types of non-i.i.d. data allocation in Table 4.2, along with the total

Table 4.1: Sub-population Local Test Performance Analysis

	Method	MNIST	FMNIST	CIFAR-10
Major.	FedAvg	**96.63**±0.70	89.75±1.76	51.98±1.69
	FedProx	96.43±0.67	89.95±1.73	51.26±1.44
	q-FFL	94.93±0.31	88.73±0.17	42.00±0.29
	WAFFLe	95.93±0.16	88.91±2.07	**68.37**±1.01
Minor.	FedAvg	67.40±11.26	68.05±4.43	16.83±4.42
	FedProx	68.60±9.44	67.50±4.50	16.56±3.32
	q-FFL	54.20±7.37	69.40±1.48	18.14±3.05
	WAFFLe	**93.87**±0.66	**79.67**±1.52	**55.00**±6.00
Gap	FedAvg	29.23±11.79	21.70±4.21	35.15±4.12
	FedProx	27.83±10.03	22.45±4.38	32.70±6.99
	q-FFL	40.73±7.55	19.33±1.43	23.87±3.00
	WAFFLe	**2.07**±0.77	**9.25**±0.61	**13.37**±2.61
Var.	FedAvg	199±106	231±35	338±59
	FedProx	186±92	233±42	318±36
	q-FFL	355±117	212±19	220±17
	WAFFLe	**26**±6	**145**±27	**182**±27

Table 4.2: Local Test Performance for $Z = 2$

	MNIST			FMNIST			CIFAR-10		
Method	Param. ↓	Unimodal ↑	Multimodal ↑	Param. ↓	Unimodal ↑	Multimodal ↑	Param. ↓	Unimodal ↑	Multimodal ↑
FedAvg	155,800	94.46±0.84	91.57±1.42	28,880	83.96±0.91	83.43±2.27	61,770	52.54±0.14	45.46±1.69
FedProx	155,800	94.44±1.15	91.53±1.05	28,800	84.19±0.99	83.59±2.30	61,770	52.36±0.11	44.95±1.17
q-FFL	155,800	91.46±1.07	88.42±1.24	28,800	83.10±0.36	85.73±0.21	61,770	43.82±0.52	38.25±1.12
WAFFLe	**120,200**	**96.23**±0.31	**95.41**±0.36	**18,155**	**87.12**±0.89	**86.09**±0.92	**42,780**	**71.30**±0.92	**66.35**±0.72

number of learnable parameters. WAFFLe performs well despite strong statistical heterogeneity, as each client can learn a personalized model by selecting different factors from $\{W_a, W_b\}$; having a model specific to each data distribution results in higher local test performance than the baselines. This advantage is especially apparent when the data are distributed multimodal non-i.i.d., mainly because WAFFLe more effectively models underrepresented clients.

Interestingly, we find that WAFFLe outperforms the baselines particularly significantly for CIFAR-10, the most challenging of the tested datasets, with WAFFLe's

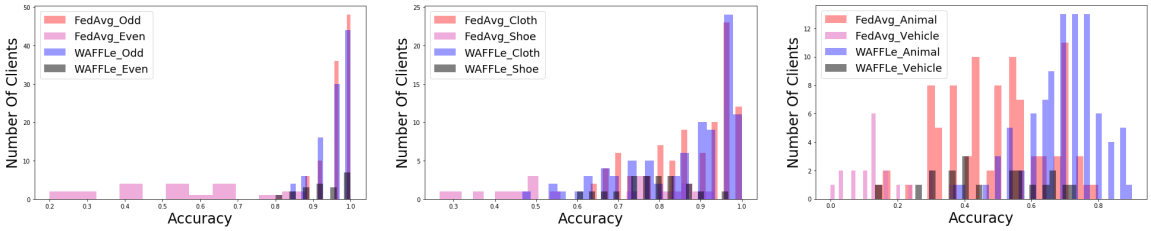


Figure 4.3: Performance distribution across clients in the multimodal non-i.i.d. setting for (a) MNIST, (b) FMNIST and (c) CIFAR-10.

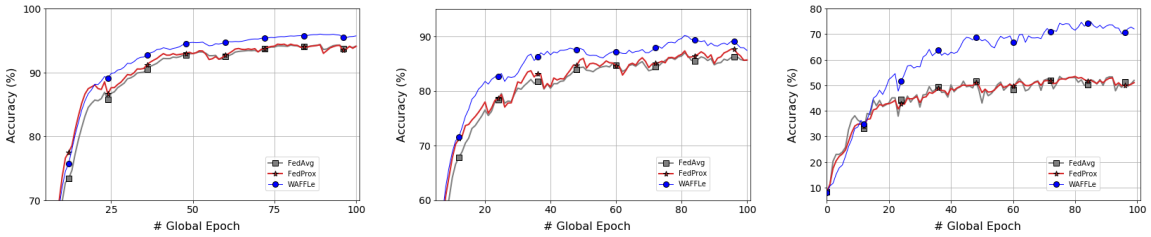


Figure 4.4: Local test performance for unimodal non-*i.i.d.* degree $Z = 2$. (a) MNIST; (b) FMNIST; (c) CIFAR-10.

local test performance outstripping the other methods by 18.8% and 20.9% for unimodal and multimodal settings, respectively. This demonstrates WAFFLe's ability to scale to complex tasks beyond MNIST, a common federated learning test bed. Notably, even though WAFFLe effectively learns a different model for each client, this does not lead to the computation or memory costs typically associated with independent models. WAFFLe's number of communication rounds is largely the same, and by sharing rank-1 factors, each weight factor can be represented compactly, resulting in a total number of parameters that is *fewer* than the single model used by the baselines, despite using the same architecture.

4.4.3 Training Efficiency Comparison

We plot local test accuracy against the global epoch for FedAvg, FedProx and WAFFLe on MNIST, FMNIST, and CIFAR-10 averaged over all clients for unimodal non-*i.i.d.* data in Figure 4.4. A similar comparison is made between FedAvg and

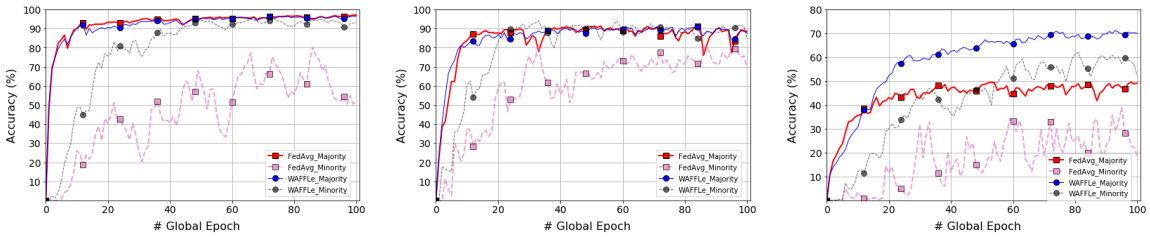


Figure 4.5: Local test performance for multimodal non-*i.i.d.* degree $Z = 2$. (a) MNIST; (b) FMNIST; (c) CIFAR-10.

WAFFLe for multimodal non-*i.i.d.* data in Figure 4.5, with the majority and minority learning curves separately shown. For both cases, the clear gap between curves shows that WAFFLe achieves better performance throughout training. Notably, WAFFLe converges at a similar rate as FedAvg with respect to the global epoch number; this is important as the number of communication rounds is often considered one of the primary bottlenecks in federated learning.

In the multimodal non-*i.i.d.* case, the difference is especially stark for the minority subpopulation, which lags significantly behind the majority when modeled with FedAvg's one-size-fits-all approach. Interesting, in addition to having lower value, the FedAvg minority's training curve is not as smooth, with large dips and spikes, especially when compared with the majority subpopulation's curve. We hypothesize that this may be due to the smaller subpopulation being more vulnerable to being unrepresented during client sampling, which may lead to catastrophic forgetting [SAK+19]. We find this to be an interesting future direction of research. In comparison, the WAFFLe minority, with its separate set of customized weight factors, has a much smoother training trajectory.

4.4.4 Fairness

Average performance over all clients as in Table 4.2 is a commonly reported metric, but we argue that it does reveal the full story. We report subpopulation mean

performance and overall population variance in Table 4.1. We observe that FedAvg, which learns a single global model, focuses on minimizing mean error across the population, resulting in stronger performance for the clients in the majority. However, as a result, clients in the minority are severely compromised, as evidenced by the large difference ("Gap") between majority (Major.) and minority (Minor.) values in Table 4.1; for example, FedAvg's performance for the "evens" group of clients is almost 30% lower than that of the "odds" group. This is gap is especially clear when visualizing the distributions of final local test performance for each client in the majority and minority groups (Figure 4.3). This underfitting can also be seen to exist throughout training from the "FedAvg_Minority" curve in Figure 4.5, which lags far below the "FedAvg_Majority" in all three datasets. On the other hand, because of WAFFLe's shared weight factor dictionary design (Equation 4.3), different knowledge can be encoded in separate weight factors, which can be used by different parts of the population. As a result, despite certain classes being underrepresented (both in terms of clients, and total samples) in the training set, WAFFLe is able to successfully model them, with performances on par with the overall population. Notably, we achieve this without explicitly enforcing fairness through client sampling during training [MSS19, LSBS19], which can be incorporated to further encourage uniform performance across clients.

4.4.5 Data Safety

A primary objective of federated learning is to keep data safe. However, as mentioned in Section 4.2.3, the predominant federated learning strategy of each client sending their entire updated model's weights still leaves the client's data vulnerable. We demonstrate this with both membership inference and model inversion attacks.

Membership inference attacks (MIAs) [SSSS17, NSH19] can be used to infer whether

Table 4.3: Membership Inference Attacks

Methods	Accuracy	F1-score
FedAvg	83.85± 1.62	83.72 ± 2.19
WAFFLe	**56.20** ± 1.40	**54.39** ± 1.85

a given data query was used for model training, leveraging the tendency of machine learning to overfit or memorize training data. As such, a successful MIA can be used by an attacker to surmise the content of a client's private data from the model. We compare a LeNet [LBBH98] FedAvg [MMR+17] model with an analogous WAFFLe model, training both on 1000 CIFAR-10 samples per client. We attack both with a MIA inspired by [SSSS17], using a small ensemble of 3 "shadow" models. As shown in Table 4.3, this simple attack achieves a high success rate at identifying a FedAvg client's training data, as intercepting the training update gives the full model. On the other hand, WAFFLe's training update only send partial model information, as the identity of the active factors is kept private. As a result, MIA success rate on WAFFLe is only moderately higher than random chance (50%). This means it is significantly harder to identify the private training data for WAFFLe, relative to FedAvg.

We also perform a model inversion attack [FJR15, WSZ+19] on both FedAvg and WAFFLe. Unlike MIAs, which must start from a query data input, model inversion attacks seek to reconstruct the inputs used to train a model from the trained model itself; successful inversion attacks pose a significant risk from a data security perspective. We perform a model inversion attack on FedAvg and WAFFLe models trained on FMNIST, showing randomly selected results in Figure 4.6 recovered from an individual user. Importantly, reconstructions on FedAvg are significantly sharper, with the class identity far clearer than for WAFFLe, meaning FedAvg is more vulnerable to model inversion attacks.

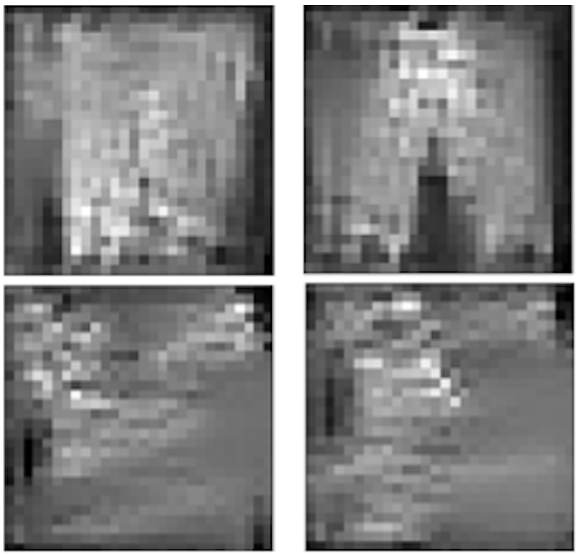


Figure 4.6: FMNIST model inversion attacks. Top row is the attack against FedAvg on T-shirt and pants sample. Botttom row is the WAFFLe result.

Table 4.4: Data Safety Comparison on FMNIST

Methods	PSNR	Attack Acc
FedAvg	12.46	60.63
WAFFLe	**10.22**	**40.78**

We report two quantitative metrics[ZJP+20] to evaluate model inversion attack in Table 4.4. *i)* Peak Signal-to-Noise Ratio (PSNR) is the ratio of an image's maximum squared pixel fluctuation over the mean squared error between the target image and the reconstructed image. The higher the PSNR, the better the quality of the reconstructed image. However, clear reconstruction images reveal the identity information of the client's data. In other words, the lower the PSNR, the more secure the system. For each class, for example T-shirt, we compute the PSNR between the reconstructed T-shirt and the average image of randomly selected T-shirt from the training data. The average PSNR of all classes of FedAvg and WAFFLe is reported. *ii)* Attack Accuracy (Attack Acc) is the accuracy of the input reconstructed image by an evaluation classifier that is trained separately. If the evaluation classifier achieves high

Table 4.5: Personalization Comparison on CIFAR-10

Methods	Unimodal	Multimodal
FedPer	67.90± 0.36	64.12 ±0.93
pFedMe	68.10± 0.2	57.16 ± 0.1
WAFFLe	**71.30** ± 0.92	**66.35** ± 0.72

accuracy, the reconstructed image is considered to expose identity information about the target label. We obtain an evaluation classifier with accuracy 96.67%. This evaluation classifier is used to classify the images reconstructed by FedAvg and WAFFLe. The average attack accuracy is reported. In Table 4.4, WAFFLe shows superior performance over FedAvg on both PSNR and attack accuracy, proofing more secure against model-inversion attack.

4.4.6 Personalization

We further conduct experiments to compare against two personalized FL methods FedPer [YZH+21] and pFedMe [DTN20] based on CIFAR-10 under both unimodal and multimodal settings for $Z = 2$. The local test performance is reported in Table 4.5. WAFFLe outperforms FedPer by offering personalization for multiple layers while FedPer only focuses on the last layer of the neural networks. Also, WAFFLe and FedPer outperforms pFedMe by 9.19% and 6.96% under multimodal setting respectively, highlighting that the methods based on model splitting are more effective than regularization-based methods for complicated non-i.i.d. settings.

4.4.7 Ablation Studies

Statistical Heterogeneity (Z)

WAFFLe is specifically designed for statistical heterogeneity, as each client can select different weight factors, effectively learning personalized models. WAFFLe was shown

to excel when $Z = 2$, as this is a strongly non-i.i.d. setting: as each client only has samples from two classes.We also did experiments in unimodal settings with less statistical heterogeneity, for $Z = \{3, 4\}$. Although it takes longer to converge in these cases, WAFFLe still outperforms FedAvg by 7.20% and 2.74%, respectively. The learning curve comparison when $Z = 3$ is shown in the Figure 4.7.

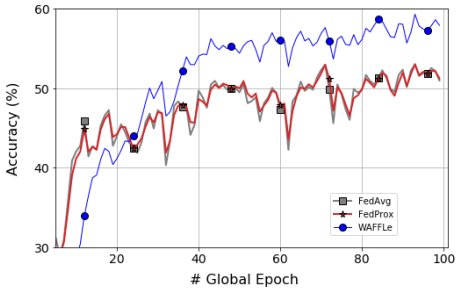


Figure 4.7: Learning efficiency comparison when $Z = 3$

Local epochs (E)

Training client devices for more local epochs allows each server to collect a bigger update from each device, increasing local computation in exchange for fewer total communication rounds. This is often a desirable trade-off, as communication costs are commonly viewed as the primary bottleneck for federated learning. However, too many local epochs can lead to divergence during the aggregation step. We study the influence of local epochs E for unimodal non-i.i.d. in Table 4.6 and for multimodal non-i.i.d. in Table 4.7 using the same settings as in Section 4.4.1 except for reducing the global training epochs T to 50 and the learning rate η to 0.02 for all methods in multimodal non-i.i.d scenario. We observe that WAFFLe can handle increased number of local epochs, improving performance for all three datasets.

Table 4.6: Unimodal Local Test Accuracy vs Local Epochs

Dataset	Method	E=10	E=20	E=30
MNIST	FedAvg	92.95	93.36	93.55
	WAFFLe	95.10	96.32	96.43
FMNIST	FedAvg	85.32	85.13	85.14
	WAFFLe	87.52	87.07	89.25
CIFAR-10	FedAvg	47.40	47.60	55.39
	WAFFLe	64.18	71.92	74.50

Table 4.7: Multimodal Local Test Accuracy vs Local Epochs

Dataset	Method	E=10	E=20	E=30
MNIST	FedAvg	88.70	89.27	89.03
	WAFFLe	95.37	94.87	95.07
FMNIST	FedAvg	86.21	86.58	86.47
	WAFFLe	87.03	89.15	91.33
CIFAR-10	FedAvg	40.91	42.09	42.00
	WAFFLe	58.79	57.00	62.61

IBP Sparsity (α) and Number of Factors (F)

At the cost of more parameters, an increasing number factors F and higher IBP parameter α gives client more expressivity for modeling its local distribution.

We study the influence of α and F for an MLP architecture on MNIST partitioned in multimodal non-i.i.d. settings in Tables 4.9 As expected, the higher α and F are, the better performance we observe, though in practice we prefer lower α and F for efficiency. On the other hand, the overall difference in local test accuracy does not vary drastically, meaning that WAFFLe is fairly robust to both hyperparameters.

To empirically demonstrate the value of an IBP prior, we also considered an alternative non-Bayesian version of our model. Specifically, we replace WAFFLe's inferred per-client weight factors with per-client weight factors optimized by standard gradient descent, and use an L1 sparsity constraint on factor usage as a replacement

Table 4.8: Sparsity Comparison on MNIST

Methods	Local Test Accuracy
FedAvg	94.46
FedProx	94.44
q-FFL	91.46
WAFFLe(without L1 norm)	94.96
WAFFLe(with L1 norm, weight=0.1)	94.59
WAFFLe(with L1 norm, weight=1.0)	96.03
WAFFLe(with L1 norm, weight=2.0)	95.92
WAFFLe(with L1 norm, weight=10.0)	94.24
WAFFLE	**96.23**

Table 4.9: Multimodal Local Test Accuracy vs α and F

	F=80	F=100	F=150
$\alpha/F = 0.4$	91.83	92.70	93.23
$\alpha/F = 0.6$	94.23	94.48	95.26
$\alpha/F = 0.8$	94.76	95.15	95.70
$\alpha/F = 1.0$	94.70	94.93	95.93

for the sparsity induced by the IBP. We list the test accuracy under Unimodal on MNIST in the Table 4.8.Note that our Bayesian formulation (WAFFLe) outperforms the non-Bayesian version while also avoiding the hyperparameter tuning of the weight of the sparsity term, which the non-Bayesian version is somewhat sensitive to.

4.5 Generalizing Weight Factorization to Convolutional Kernels

While introducing WAFFLe's formulation in Section 4.2.1, we assumed a multilayer perceptron (MLP) model, as illustrating our proposed shared dictionary with the 2D weight matrices composing fully connected layers is made especially clearer. While MLPs are sufficient for simple datasets such as MNIST, more challenging datasets require more complex architectures to achieve the most competitive results. For

computer vision, for example, this often means convolutional layers, whose kernels are 4D. While 4D tensors can be similarly decomposed into rank-1 factors with tensor rank decomposition, such an approach would result in a large increase in the number of parameters in the weight factor dictionary due to the low spatial dimensions of the convolutional kernels (*e.g.*, 3×3) in most commonly used architectures. Instead, we reshape the 4D convolutional kernels into 2D matrices by combining the three input dimensions (number of input channels, kernel width, and kernel height) into a single input dimension. We then proceed with the formulation in (4.2). Similar approaches can be taken to generalize our formulation to other types of layers.

4.6 Conclusion

We have introduced WAFFLe, a Bayesian nonparametric framework for federated learning, employing shared rank-1 weight factors. This approach allows for learning individual models for each client's specific data distributions while still sharing the underlying learning problem in a parameter-efficient manner. Our experiments demonstrate that this model customizability makes WAFFLe successful at improving local test performance and, more importantly, significantly improves fairness in model performance when the data distribution among clients is multimodal. Furthermore, we are able to scale our results to CIFAR-10 and convolutional networks, where we observe the biggest improvements. We also show that by keeping the active factors selected by each model private on each device along with the data, WAFFLe's communication rounds only send partial model information, making it significantly harder to perform membership inference or gradient-based model inversion attacks on the private data.

Chapter 5

Improving Fairness in Federated Learning

5.1 Introduction

Major advances in deep learning over the last decade have in large part been possible due to the increasing availability of data. With the proliferation of personal computers, smart phones, and edge devices, data are being generated and collected at unprecedented rates, providing the large datasets needed to train the machine learning that power "intelligent" services that are becoming increasingly common in daily life. However, the rich content in these data that enables such smart behavior may also be revealing of personal information. Traditional learning methods pool the data into a central repository for training, which makes personal data vulnerable to breaches or interception.

Federated learning [MMR+17] has emerged as an alternative strategy, with an emphasis on user data privacy. In the federated learning paradigm, learning takes place on the client devices themselves, which means that the user's personal data never leaves the local device. In place of the data, the updated model itself is sent to a coordinating server, which then aggregates the updates and distributes the new model to the clients.

While federated learning has demonstrated promise for user data privacy, a major challenge is statistical heterogeneity [LSTS20, LSZ+18, AASC19, HML+20]: data distributions between clients may exhibit significant differences. These differences

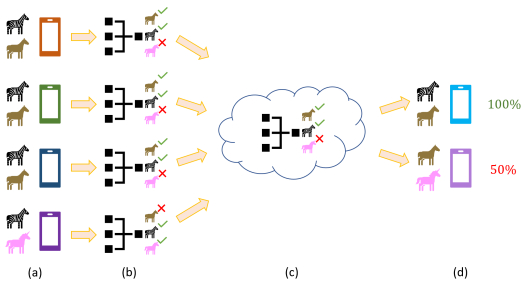


Figure 5.1: (a) Statistical heterogeneity (b) Model characteristics (c) Model aggregation. (d) One-size-fits-all global model poorly on minority.

may lead to variance in learned local models after training on each client's local data. Additionally, the formulation in [MMR+17] is fundamentally a one-size-fits-all solution, meaning the learned global model may perform worse for some clients, as shown in Figure 5.1. As a result of these factors, federated learning methods tend to perform poorer when the data are not independent and identically distributed (*i.i.d.*) among clients [MMR+17, ZLL+18].

What is more concerning, however, is that the accuracy loss due to statistical heterogeneity may be borne unequally among clients [LSBS19]. In populations with unequally sized subgroups, clients with less common classes tend to see worse performance [HML+20]. This may be, in part, due to catastrophic forgetting [MC89, SAK+19]: clients from outside a subpopulation have a tendency to forget features not found in their own data and, during aggregation, the less represented clients may have their learned features drowned out when the model weights are averaged. In the real world, these client characteristics may represent ethnicity [KBK+12], gender [AKV+20, Lea18], age [ABVK20], language [GHDL18], dialect, demographics, animal species, or disease trait. Therefore, the inability to cope with statistical heterogeneity my lead to potentially unfair algorithms, that provide in- accurate classifications based on certain characteristics of their input data. A popular and effective strategy for preventing forgetting is replay [LPR17, SLKK17]: storing a small buffer

of samples for rehearsal. In federated learning, however, clients do not have access to data from parts of the distribution that are not well-represented in their own data. This is, in part, by design, as client data are kept private and local to the device.

In this work, we propose a federated learning system with zero-shot data augmentation (Fed-ZDA) to generate pseudo-exemplars of unseen classes, without having access to the private data. Such a strategy preserves the model's ability on previously sampled client data when learning the local client update. This makes the model less likely to lose representational ability for parts of the distribution that are rarer. We explore two strategies for using zero-shot data augmentation for federated learning, one in which synthetic samples are generated at the client (Fed-ZDAC), and another where they are generated at the server (Fed-ZDAS). Both methods are illustrated in Figure 5.2. Differential privacy analysis shows that our proposed approach satisfies $(0, \delta)$ differential privacy. Finally, experiments on MNIST [LBBH98], FMNIST [XRV17], and CIFAR-10 [Kri09] show that both Fed-ZDAC and Fed-ZDAS result in more equitable model performance.

5.2 Related Work

5.2.1 Federated Learning

Statistical Heterogeneity Statistical heterogeneity of the data distributions of client devices has long been recognized as a challenge for federated learning [ZLL+18]. Despite acknowledging statistical heterogeneity, many federated learning algorithms still focus on learning a single global model [MMR+17]; such an approach often suffers from divergence of the model, as local models may vary significantly from each other. To address this challenge, a number of works break away from the single-global-model formulation. Several [SCST17, CB19] have cast federated learning as a multi-task learning problem, with each client treated as a separate task. FedProx [LSZ+18]

adds a proximal term to account for statistical heterogeneity by limiting the impact of local updates. In [ZLL+18] performance degradation from skewed data is recognized, proposing global sharing of a small subset of data which, while effective, may compromise privacy.

Fairness There has been rising interest in developing fair methods for machine learning [YAAMB20]. However, such concerns have been less addressed in federated learning. A commonly used fairness definition has been proposed in [ZVRG17]. However, it forces the accuracy to be identical on each device across hundreds to millions of clients, given the significant variability of data in the network. Recent work [LSBS19] has taken a step towards addressing this by introducing uniformity to describe the fairness in federated learning, in which the goal is instead to ensure that the underfit groups are assigned more weight in the global learning objective. However, the proposed objective causes a performance drop in clients who could have better results under traditional federated average objective, which may reduce these clients' incentive to participate the federated learning process. The work in [HML+20] proposed rank-one factorization on model parameters to ensure consistent model performance across clients, by leaving factors locally. However, this Bayesian approach usually costs more training time, and development of client-specific models is beyond the single-global-model focus of this paper.

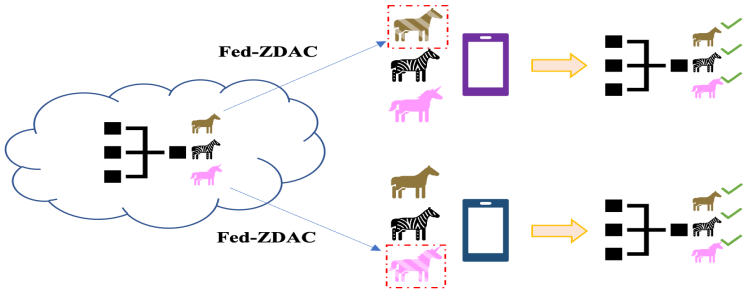
5.2.2 Zero-Shot Data Augmentation

Deep learning performance is highly dependent on the quantity of data available [HNP09, SSSG17]. Data augmentation, which inflates the size of a dataset without necessitating further data collection, has proven effective in a wide range of settings [LBBH98, KSH12, ZCDLP18, LHS+21], improving machine learning model generalization. However, most data augmentations apply transformations to the existing data, thus mak-

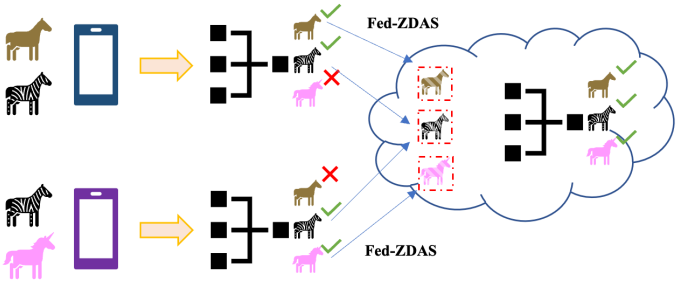
ing the implicit assumption that at least some data is available. These techniques are thus difficult to apply when no data is available. Consequently, *DeepInversion* [YMA+20] proposes a data-free knowledge transfer based on synthesizing data, effectively providing more teacher behavior for a student to learn. Also, [CYD+20] proposes a similar method for network quantization, by updating random input to match stored batch norm layer statistics. In our work, since the server has no access to the local data, synthesizing a reasonable amount of fake data for deficient classes would encourage a more fair global model. Also, unlike the work [JOK+18, ZLL+18] which violates the rule that clients should never share data to other clients or the server, zero-shot data augmentation synthesize data based on the model information only. Note that using synthesized samples for data augmentation differs from related works like [GT20], which take an approach similar to dataset distillation [WZTE18] to synthesize data for the purpose of compressing model updates for communication efficiency purposes.

5.2.3 Differentially Private Federated Learning

With the increasing awareness of data security and confidential user information, privacy has become an important topic for machine learning systems and algorithms. In order to solve this issue, differential privacy has been proposed to prevent revealing training data [ACG+16]. Even though federated learning enables local training without sharing the data to the server, it is still possible for an adversary to infer the private information to some extent, by analyzing the model parameters after local training [WSZ+19, MLD+20]. Therefore, combining differential privacy with federated learning has been studied in many previous works. To ensure federated learning approaches satisfy differential privacy, the work in [GKN17] proposed a client level perspective by adding Gaussian noise to the model update, which can prevent the



(a)



(b)

Figure 5.2: Illustration of Fed-ZDAC (a) and Fed-ZDAS (b).

leakage of private information and achieve good privacy performance. In [TBA+19], a combination of differential privacy and secure multiparty computation was proposed to block differential attacks. However, previous approaches based on adding noise to model parameters struggle to capture the appropriate trade-off between the model performance and privacy budget. Our proposed zero-shot data augmentation can be interpreted as a new randomization mechanism different from adding Gaussian noise, satisfying differential privacy without hurting model performance.

5.3 Federated Learning with Zero-Shot Data Augmentation

We propose a federated learning method with zero-shot data augmentation (Fed-ZDA), for the purpose of improving the robustness and fairness of federated learning. To improve the fairness of the global model, Fed-ZDA introduces new synthetic data,

generated either at the server or at the client nodes, to supplement training with underrepresented samples. Notably, these samples are generated without access to user data, but rather from shared models post-local update. We start by reviewing standard federated learning, which Fed-ZDA builds on. We then describe the zero-shot data-augmentation method we use for Fed-ZDA. We describe two deployments of Fed-ZDA, Fed-ZDAC and Fed-ZDAS, where the zero-shot data-augmentation is done at the client nodes and at the server node, respectively.

5.3.1 Federated Learning

In its most basic form, the federated learning objective is commonly expressed as the following:

$$\min_{w} f(w) = \sum_{i=1}^{Z} p_i F_i(w) \tag{5.1}$$

where $F_i(w) \coloneqq \mathbb{E}_{\boldsymbol{x}_i \sim \mathcal{D}_i}[f_i(w; \boldsymbol{x}_i)]$ is the local objective function of the i^{th} client, Z is the number of devices or clients, and $p_i \geq 0$ is a weight assigned to the i^{th} client.

Standard federated learning aims to aggregate, at the centralized server, a federated global model from the client models, typically by averaging them. In this scenario, the clients only share their trained models with the server, and do not share the datasets on which their models have been trained. The server and the client communicate for T rounds to update the global model M. A single communication round contains three main steps:

1. The server randomly samples a subset of clients and distributes the model to the sampled clients.

2. Each sampled client updates the model by training it with their local training data.

3. Each client sends their updated model back to the server and the server aggregates the received client models into a new global model.

Typically, learning aggregates the models by federated averaging (FedAvg), in which the federated global model $M_{\mathcal{G},t}$ aggregated at the t^{th} communication round is simply a weighted average of all the client models received at this round. Let $M_{i,t}$ be the model trained by the i^{th} client C_i at the t^{th} communication round, and $\mathcal{S}_t$ is the set of indices of the sampled clients at the t^{th} round.

$$M_{\mathcal{G},t} = \sum_{i \in \mathcal{S}_t} w_i M_{i,t} \tag{5.2}$$

Different weights $0 \leq w_i \leq 1$ can be assigned to the clients depending on different factors, such as the amount of data they have been trained on, if such information is known at the server. Otherwise, a simple arithmetic mean is adopted.

Ideally, after sufficient communication rounds, the global model should converge to a solution that has learned using the data from all clients. However, heterogeneous data distributions across clients may cause inconsistent model performance. In particular, if the dataset distributions of the different clients are skewed towards a majority group of classes, FedAvg may result in a model with a large variance in accuracy across classes, resulting in a large variance in the global model accuracy on the data of different clients. Hence, standard federated learning suffers from the notion of unfairness towards the under-represented clients, providing poor accuracy on their data.

5.3.2 Zero-Shot Data Generation

Data augmentation has proven effective in many machine learning settings, such as when there is data scarcity or class imbalance. Commonly used techniques include

performing transforms (*e.g.* rotations, flips, crops, added noise) based on the original true data, combinations in feature space, and synthesizing data by generative models. However, these techniques require access to training data or at least a few data sample seeds. In federated learning, these are not available, as data never leaves the individual clients, making conventional augmentation techniques challenging. In this work, we propose zero-shot data generation (ZSDG), to generate labeled synthetic data for data-augmentation at the clients, without having any access to any training data. This approach utilizes trained models (either the global model pre-update, or the local models post-update) to generate synthetic data of the desired classes without access to *any* non-local data.

One way to generate synthetic data whose statistics match those of the original training data is to find the data that results in similar statistics as those stored in the batch normalization (BN) layers of the pretrained model. However, without assigning class labels to this data, one cannot use this data in a data-augmentation regime for supervised training. For data augmentation with N possible classes, we generate data for each class $1 \leq n \leq N$, represented by its corresponding one-hot vector $\bar{y}(n)$, which has 1 at the n^{th} index and zero otherwise. Let model M be a neural network with L layers. For simplicity of notation, assume the model has L batch normalization (BN) layers and denote the activation before the ℓ^{th} BN layer to be z_ℓ. The ℓ^{th} BN layer is parameterized by a mean μ_ℓ and variance σ_ℓ calculated from the input feature maps when the model was being trained. During the forward propagation, z_ℓ is normalized with the parameters of the BN layer. Note that given a pretrained model M, batch norm statistics of all BN layers are stored and accessible. Given a target class $\bar{y}(n)$ the ZSDG reduces to the optimization problem that finds the input data $\bar{x}$ that result in the batch norm statistics matching those stored in the BN layers of the pretrained model, and are classified by the pretrained model

as having label $\bar{y}(n)$. Given the pretrained model M, with BN statistics μ_ℓ and σ_ℓ stored in its layers $1 \leq \ell \leq L$, the ZSDG optimization problem to generate synthetic labeled data $(\bar{x}(n), \bar{y}(n))$ for $n \in \{1, 2, \cdots, N\}$ can be expressed as:

$$\bar{x}(n) = \arg\min_{\bar{x}} \sum_{\ell=1}^{L} \|\bar{\mu}_\ell - \mu_\ell\|_2^2 + \|\bar{\sigma}_\ell - \sigma_\ell\|_2^2 + \mathcal{H}(M(\bar{x}), \bar{y}(n)), \tag{5.3}$$

where $\bar{\mu}_\ell$, and $\bar{\sigma}_\ell$ are, respectively, the mean and standard deviation evaluated at layer ℓ with the generated input data, $M(\bar{x})$ denotes the model classification output when the input is $\bar{x}$, and $\mathcal{H}$ is the cross entropy loss function to learn the class labels. To solve Equation 5.3 for a selected class $\bar{y}(n)$, an input is initialized randomly from a normal distribution and, then, updated using gradient descent, while fixing the model parameters during back-propagation. The ZSDG is described in Algorithm 3.

Algorithm 3 Zero-Shot Data Generation (ZSDG)

Input: Model M with L batch normalization layers
Output: A batch of labeled fake data: $(\bar{x}, \bar{y})$
Get μ_ℓ, σ_ℓ from Batch Normalization layers of M, $\ell \in \{1, 2, \cdots, L\}$
for $n = 1, 2, \cdots, N$ **do**
    Generate $\bar{x}(n)$ randomly from a Gaussian distribution, assign it a label $\bar{y}(n)$
end for
for $j = 1, 2, \cdots$ **do**
    Forward propagate $M(\bar{x}(n))$ for all n
    Gather intermediate activations $\bar{z}_\ell$, $\ell \in \{1, 2, ..., L\}$
    Gather BN statistics: $\bar{\mu}_\ell$ and $\bar{\sigma}_\ell$ induced by intermediate activations $\bar{z}_\ell$, $\ell \in \{1, 2, ..., L\}$
    Compute the loss based on Equation 5.3
    Backward propagate and update the input $\bar{x}(n)$ only
end for
Return $(\bar{x}, \bar{y}) = \cup_{n \in \{1,2,\cdots,N\}} (\bar{x}(n), \bar{y}(n))$

Algorithm 4 Fed-ZDAC: Federated Learning with Zero-Shot Data Augmentation at Clients

Input: Communication rounds T, global model M
for $t = 1, \cdots, T$ **do**
    Server randomly selects subset $\mathcal{S}_t$ of clients
    Server sends $M_{\mathcal{G},t-1}$ to $\mathcal{S}_t$
    for Clients C_i, $i \in \mathcal{S}_t$ **in parallel do**
        Generate labeled fake data $(\bar{x}_i, \bar{y}_i)_t$ by ZSDG from the global model $M_{\mathcal{G},t-1}$
        Client C_i produces the model $M_{i,t}$ by updating the model $M_{\mathcal{G},t-1}$ with the mix of real local data available at round t, and the fake ZSDG data: $\{(x_i, y_i)_t, (\bar{x}_i, \bar{y}_i)_t\}$
        Send the updated client model $M_{i,t}$ to the server.
    end for
    Server aggregates all client models $M_{i,t}$, $i \in \mathcal{S}_t$, e.g. by Equation 5.2, to obtain the updated $M_{\mathcal{G},t}$
end for

5.3.3 Zero-Shot Data Augmentation at Clients

It is common to have statistical heterogeneity in the training data across clients. To address the deficiency of their training data in some classes, and promote the global model fairness, clients are instructed to augment their training data with fake data using ZSDG, before updating the received global model. Let the i^{th} client at the t^{th} communication round have the real local training data $(x_i, y_i)_t$ with input and label pairs. Let $(\bar{x}_i, \bar{y}_i)_t$ be the synthetic (fake) data generated using ZSDG over all classes from the received global model $M_{\mathcal{G},t-1}$. Then the procedure for federated learning with zero-shot data augmentation at the clients (Fed-ZDAC) is described by Algorithm 4.

5.3.4 Zero-Shot Data Augmentation at Server

In Section 5.3.3, we discussed federated learning with data augmentation at the client nodes. In practice, clients may be mobile computing devices that are limited in their computing resources and storage capacity, which may restrict their capacity

for data augmentation. Clients may also not care about fairness of the global model towards other clients, and would like to train the best model for their classes of interest only. It is also in the best interest of the server to produce a fair and accurate model, that does not ignore data classes of the under-represented clients. In addition, if the global model is fair, and each client updates the global model from the same fair initialization, federated learning can convergence faster to a fair solution. Consequently, we propose federated learning with zero-shot data augmentation at the server (Fed-ZDAS). We use the same notation as described in Section 5.3.3. In more detail, the server distributes its global model to a subset of clients, Each of these clients update this global model with their local training data (x, y) and send it back to the server. In strive for fairness, the server will generate equal amount of fake data from each received client model, and combines all fake client data into a balanced synthetic dataset. The server aggregates all received client models into a single model, and then trains the single model by the combined synthetic dataset. To our knowledge, this is the first federated learning protocol which involves training at the server, since in general the server is assumed not to have any data. Fed-ZDAS is described in Algorithm 5.

Since the motivation of federated learning is protecting client data privacy, we also prove that our proposed method satisfies client-level differential privacy (DP), a local differential privacy adopted as [WLD$^+$20]. Intuitively, before clients send updated model parameters back to the server, we seek for a randomized perturbation on these model parameters such that the server can not distinguish if certain client has been involved in the current communication round. A standard way to satisfy differential privacy is adding Gaussian noise to model parameters with trade-off between model's performance and privacy budget. [ACG$^+$16, WLD$^+$20]. In contrast, our method can be considered as a kind of perturbation to model parameters with useful information

Algorithm 5 Fed-ZDAS: Federated Learning with Zero-Shot Data Augmentation at the Server

Input: Communication rounds T, global model M
for $t = 1, \cdots, T$ **do**
    Server randomly selects subset $\mathcal{S}_t$ of clients
    Server sends $M_{\mathcal{G},t-1}$ to $\mathcal{S}_t$
    for Clients C_i, $i \in \mathcal{S}_t$ **in parallel do**
        Client C_i produces the model $M_{i,t}$ by updating the model $M_{\mathcal{G},t-1}$ with its real local data available at round t $(x_i, y_i)_t$ and sends the updated client model $M_{i,t}$ to the server.
    end for
    Server generates a class-balanced fake labeled data $(\bar{x}_i, \bar{y}_i)_t$ by ZSDG with each received client model $M_{i,t}$, $i \in \mathcal{S}_t$.
    Server combines the fake data generated with the different client models into a combined balanced dataset $(\bar{x}, \bar{y})_t = \cup_{i \in \mathcal{S}_t} (\bar{x}_i, \bar{y}_i)_t$.
    Server aggregates all client models $M_{i,t}$, $i \in \mathcal{S}_t$, e.g. by Equation 5.2, to obtain an interim global model $\tilde{M}_{\mathcal{G},t}$
    Server trains $\tilde{M}_{\mathcal{G},t}$ using the combined fake dataset $(\bar{x}, \bar{y})_t$ to produce the updated global model $M_{\mathcal{G},t}$
end for

as opposed to pure random noise. As a result, we show that our proposed method satisfies $(0, \delta)$ differential privacy. For more details about the proof, please check Section 5.5.

5.4 Experiments

5.4.1 Datasets and Settings

Task and Datasets We conduct experiments on three standard datasets: MNIST [LBBH98], FMNIST [XRV17], and CIFAR-10 [Kri09]. Following [MMR+17] for the federated learning setting, the server selects a proportion $\gamma = 0.1$ of 100 clients during each communication round, with $T = 100$ total rounds for all methods. Each selected client trains their own model for $E = 5$ local epochs with mini-batch size $B = 10$. For the data partition, we focus on the non-*i.i.d.* setting, which is typically more challenging and realistic for federated learning. We divide the 60k images into a

training set of 50k images and external test set of 10k images, then the training set is distributed to the clients, such that each client only has a subset Z of the classes, and divide their local data set as local training set and local testing set.

Following [HML+20], we study two data splits, each representing different types of statistical heterogeneity. The first is unimodal non-*i.i.d.* which is identical to the data partition introduced by [MMR+17]. The second is multimodal non-*i.i.d.*, in which there exists subpopulations, with some being more prevalent than others. Each subpopulation group can be thought of as a mode of the overall distribution. In other words, the classes are imbalanced in the data set aggregating from all clients' data.

Table 5.1: Local test performance and client level fairness.

		Unimodal		Multimodal	
Dataset	Method	Mean Accuracy ↑	Variance ↓	Mean Accuracy↑	Variance↓
MNIST	FedAvg	97.98±0.01	6.70±1.21	96.67±0.73	47±27
	FedProx	97.93±0.01	6.33±1.25	91.98±0.80	72±6
	q-FFL	95.84 ±0.45	17.00±9.20	94.81±7.55	78±20
	Fed-ZDAC	**98.23**±0.22	**3.54**±0.85	**97.07**±0.56	**27**±12
	Fed-ZDAS	97.34±0.61	6.22±0.33	95.49±0.99	49±22
FMNIST	FedAvg	85.30±2.67	368±222	83.43±2.28	245±41
	FedProx	85.64±2.19	360±215	83.37±2.04	237±38
	q-FFL	83.09±0.36	283±45	85.97±0.18	175±10
	Fed-ZDAC	84.65±2.81	280±112	**86.00**±0.07	161±40
	Fed-ZDAS	**86.23**±2.09	**188**±67	85.66±0.85	**135**±11
CIFAR-10	FedAvg	**50.30**±0.91	417±190	45.53±1.30	288±98
	FedProx	49.92±0.55	416±186	**45.88**±1.44	266±100
	q-FFL	41.72±3.00	**285**±115	38.25±1.12	**243**±49
	Fed-ZDAC	47.18±1.55	337±155	43.92±1.66	244±70
	Fed-ZDAS	47.78±1.02	325±145	42.18±0.81	**243**±64

Model Architecture Our zero-shot data augmentation requires the model to contain batch normalization layers. For both MNIST and FMNIST, we use a convolutional network consisting of two 5×5 convolution layers with 16 and 32 output channels, respectively. Each convolution layer is followed by a batch normalization layer and a 2×2 max-pooling operation with ReLU activations. A fully connected

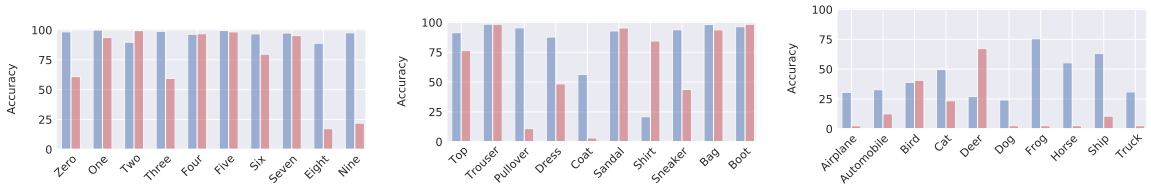
Table 5.2: Global Test Performance and class level fairness.

Dataset	Method	Unimodal		Multimodal	
		External Accuracy ↑	Variance ↓	External Accuracy↑	Variance↓
MNIST	FedAvg	98.02±0.14	3.69±0.55	93.54±2.38	78±48
	FedProx	98.05±0.15	3.69±0.60	93.62±2.38	75±58
	q-FFL	95.76 ±0.56	7.40±2.02	92.56±0.29	63±2
	Fed-ZDAC	**98.21**±0.08	**1.71**±0.21	**95.66**±0.72	**22**±7
	Fed-ZDAS	97.66±0.08	2.11±0.39	94.10±0.75	40±16
FMNIST	FedAvg	**85.03**±1.54	435±296	79.18±2.0	779±46
	FedProx	84.94±1.19	426±274	79.13±1.80	794±33
	q-FFL	80.99±1.23	558±192	81.24±0.43	673±12
	Fed-ZDAC	83.13±2.56	263±101	**83.41**±0.26	483±84
	Fed-ZDAS	83.90±1.56	**260**±76	83.27±0.25	**313**±68
CIFAR-10	FedAvg	48.89±1.04	473±195	**41.74**±4.30	361±154
	FedProx	48.83±0.89	258±13	37.06±0.62	480±50
	q-FFL	34.01±4.46	370 ±135	32.83 ±0.89	**218** ±38
	Fed-ZDAC	**49.50**±0.27	378 ±108	40.18±2.59	288±19
	Fed-ZDAS	48.26±1.02	**200**±69	39.07±1.85	295±98

layer with a softmax is added for the output. For CIFAR-10, we use a convolutional network consisting of two 3×3 convolution layers with 16 filters each. Each convolutional layer is followed by a batch normalization layer and a 2×2 max-pooling operation with ReLU activations. These two convolutions are followed by two fully-connected layers with hidden size 80 and 60, with a softmax applied for the final output probabilities. We utilize SGD as the optimizer and set the learning rate as 0.02 for all methods. We compare our methods with three baselines: FedAvg [MMR+17], FedProx [LSZ+18] and q-FFL [LSBS19].

5.4.2 Local Test and Client-Level Fairness

Local test performance is a metric to evaluate the aggregated model on each client's local test set, that is usually class imbalanced. It is an important metric to demonstrate the personalization ability of the aggregated model. As with [LSBS19], the variance of local test performance across all clients is taken as the fairness metric. Lower variance means the learned model does not lean towards subpopulations who



(a) MNIST (b) FMNIST (c) CIFAR-10

Figure 5.3: The Blue bars are the trained model's ability.The red bars are the accuracy from oracle classifiers.

share prevalent data distributions, which is a more fair solution. This metric can be considered as fairness on clients level. We test all methods under both unimodal non-*i.i.d.* and multimodal non-*i.i.d.* The results are listed in Table 5.1. The mean accuracy is the average local test accuracy over all clients and the variance is the client level fairness metric. The standard deviation values are calculated based on the results of different trials by changing random seeds. For MNIST and FMNIST, the proposed method not only achieves the best mean accuracy, but also improves the fairness over all baselines. For CIFAR-10, our method achieves better accuracy than q-FFL and more fairness than FedAvg and FedProx.

5.4.3 Global Test and Class-Level Fairness

Global test performance is a metric to evaluate the aggregated model on an external test set, that is usually class balanced. This is an important criterion to justify the efficiency of the federated learning mechanism and the model's performance on newly coming clients. However, it is still a metric based on average which cannot fully capture whether the model is biased towards, if exists, any prevalent class distribution. We report the variance of accuracy across classes as an extra fairness metric on class level. In Table 5.2, the external accuracy is the accuracy of the federated model on the held out test set, and the variance is class level fairness metric. We observe better performance on MNIST and FMNIST and comparable results on

CIFAR-10. Similarly, all the standard deviation values are calculated based on the results of different trials by changing random seeds.

5.4.4 The Analysis of Augmented Data

The augmented data are generated conditioned on the given label. To study the quality of the synthesized data, we separately trained three classifiers of the same architecture using the optimizer and learning rate described in Section 5.4.1, but in a centralized way for MNIST, FMNIST, and CIFAR-10. After training, each classifier achieves test accuracy 99.06%, 89.79% and 67.32%, respectively. These classifiers are taken as the standalone oracle to evaluate the augmented data. To obtain the synthesized data for test, we run ZSDG based on the model trained in federated learning under multimodal non-*i.i.d.* setting. For each class, we generate 64 images as the test data. The test results for augmented images are shown in Figure 5.3. We also list the trained model's ability to recognize each class as comparison. In general, the accuracy of synthesized data reflects the ability to 'fool' the oracle classifier, *i.e.* the ability to reduce the local data distribution divergence among clients. Since each client owns at most two classes in our experimental setting, the statistical heterogeneity can be mitigated, as long as deficient class of images is synthesized by ZSDG.

5.4.5 The Influence of Client Data Distribution

As mentioned in Section 5.4.4, the quality of synthetic data depends on the performance of the model we invert, and the model performance is highly affected by the client data distribution. To further study the influence of the client data distribution on data augmentation quality, we compare the models f_a, f_b, and f_c learned by three different algorithms:

- f_a: the model trained by a regular machine learning process on aggregated dataset
- f_b : the model trained by federated learning framework on distributed dataset following an $i.i.d$ setting
- f_c: the model trained by federated learning framework on distributed dataset following non-$i.i.d.$ setting.Each client has at most 3 out of 10 classes of the images

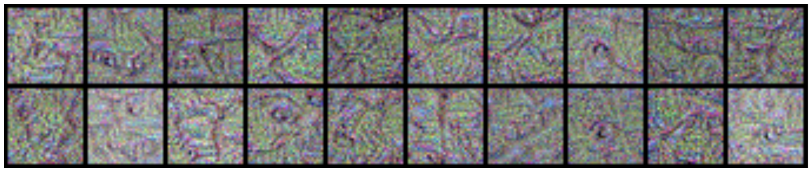
We utilize the standard ResNet34 architecture and train it on CIFAR-10 dataset. The models' performance on test dataset for f_a,f_b, and f_c are 95.20%,73.96% and 58.10%, respectively. In other words, the model's performance decreases as more constraints put in the learning process, which is expected. Consequently, we invert the models and observe the quality of the synthetic images of the same target labels decreases as shown in Figure 5.4. This result not only validates that the quality of the synthetic data depends on the base model's performance but also suggests a burn-in stage before model inversion in the federated learning framework which is studied in Section 5.4.6.

5.4.6 When to start data augmentation

It is important to choose the starting point for when the data augmentation is triggered, as the quality of reconstructed data highly depends on the model performance, which increases as the communication rounds between the server and clients climbs. High-quality augmented data help shrink the divergence of local data distribution among clients and improves privacy, therefore increasing the difficulty for the adversary to tell if certain clients have participated in training. Bad augmented data, such as random noise, can also help maintain privacy, but is likely to erase the useful information in learned model. We study the influence of starting epoch when data

(a) Synthetic images from f_a

(b) Synthetic images from f_b



(c) Synthetic images from f_c

Figure 5.4: The images recovered from models learned by three different learning algorithms.

augmentation happens. We compare the Fed-ZDAC's fairness performance under the multimodal non-*i.i.d.* setting when the data augmentation starts from global epoch 80, 90, and 95, with the results shown in Figure 5.5. In federated learning, usually longer training epoch leads to solutions with better performance. As a result, the augmented data with higher quality make each client's local data distribution more similar, and contribute to reduce the variance more.

5.5 Differential Privacy Analysis

We analyze the differential privacy of our proposed methods, adopting the same definition as [WLD+20] for differential privacy in randomized mechanisms. We show that our proposed method satisfies $(0, \delta)$ differential privacy or $(0, \delta)$-DP for short.

Definition 5.5.1 ((ϵ, δ)-*DP*)**.** *A randomized mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{R}$ with domain $\mathcal{X}$ and range $\mathcal{R}$ satisfies (ϵ, δ)-DP if for all measurable sets $\mathcal{S} \subset \mathcal{R}$ and for any two*

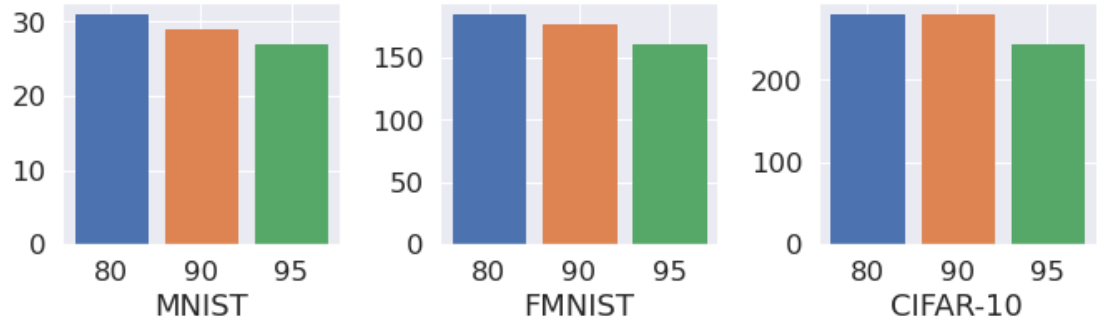


Figure 5.5: The influence of data augmentation starting point. The horizontal axis is the start global epoch and the vertical axis is the variance.

adjacent databases $\mathcal{C}$ *and* $\mathcal{C}' \in \mathcal{X}$,

$$P(\mathcal{M}(\mathcal{C}) \in \mathcal{S}) \leq e^{\epsilon} P(\mathcal{M}(\mathcal{C}') \in \mathcal{S}) + \delta$$

Since we focus on the client level perspective, the databases $\mathcal{C}$ and $\mathcal{C}'$ here are the sets of clients, which differ on one client only, c and c', *i.e.*,

$$\begin{aligned} \mathcal{C} &= c \cup \mathcal{C}_0, \\ \mathcal{C}' &= c' \cup \mathcal{C}_0. \end{aligned} \tag{5.4}$$

Here, we denote the distributions of the datasets D and D' of the two client sets $\mathcal{C}$ and $\mathcal{C}'$ as $P_D(X)$ and $P_{D'}(X)$. Assume both clients start training their models, on their local datasets, starting from the same initial parameter W, e.g. the global model. If their datasets having different distributions, both clients will obtain two different models after local training, which have different parameter distributions. We denote the two parameter distributions as $P_{\mathcal{C}}(W)$ and $P_{\mathcal{C}'}(W)$. For simplicity, we assume the model training is a stochastic process estimating the following posterior distribution according to the Bayes' rule,

$$P(W|X) \propto P(X|W)P_0(W),$$

where $P_0(W)$ is the prior distribution of W. Since each client trains on the same model architecture, the likelihood model $P(W|X)$ will be the same for all clients. It is also reasonable to use the same prior distribution for every client.

Assumption 5.5.1. *The total variation distance (TV) between the distributions of any two different augmented client datasets are less than δ: $TV(P_D(X), P_{D'}(X)) \leq \delta$.*

To verify the assumption 5.5.1, we denote the distribution of generated data as G, and the i-th client's dataset is the union of the generated data and the raw data, and the distribution of this combined dataset is denoted as P_i. According to the definition of TV distance and its triangle inequality, given an arbitrary δ, we can always generate large enough samples such that $TV(G, P_i)$ is smaller than $\delta/2$. Thus for any two clients, we have $TV(P_j, P_i) \leq TV(P_j, G) + TV(P_i, G) \leq \delta/2 + \delta/2 = \delta$. As a result, the assumption 5.5.1 is reasonable. With the above assumption, we use the data processing inequality stated in Lemma 5.5.1 to derive the TV distance between $P_{\mathcal{C}}(W)$ and $P_{\mathcal{C}'}(W)$.

Lemma 5.5.1. *(Theorem 6.2 in [AMK+17]) Consider a channel that produces Y given X based on the law $P_{Y|X}$ (illustrated in Figure 5.6). If P_Y is the distribution of Y when X is generated by P_X and Q_Y is the distribution of Y when X is generated by Q_X, then for any f-divergence $D_f(\cdot\|\cdot)$,*

$$D_f(P_Y\|Q_Y) \leq D_f(P_X\|Q_X)$$

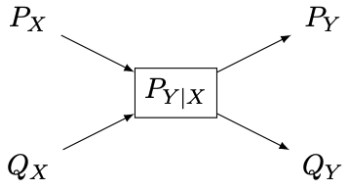


Figure 5.6: Data processing inequality

Theorem 5.5.2. *Federated learning with zero-shot data augmentation satisfies the differential privacy* $(0, \delta)$*-DP.*

Proof. Since the total variation distance is an instance of f-divergence [AMK+17], applying Lemma 5.5.1, we obtain

$$TV(P_{\mathcal{C}}(W), P_{\mathcal{C}'}(W)) \leq TV(P_D(X), P_{D'}(X)) \leq \delta.$$

In federated learning, we perform model aggregation, denoted as W_{agg}, as

$$W_{agg} = \frac{1}{n} W + \frac{n-1}{n} W_0$$

where W_0 is the parameter aggregated on the set of other clients $\mathcal{C}_0$ (as defined in Eq. 5.4) and n is the number of clients in $\mathcal{C}$. We denote the two different distributions of W_{agg} in the two models as $P_{\mathcal{C}}(W_{agg})$ and $P_{\mathcal{C}'}(W_{agg})$. Similarly, we can also use the Lemma 5.5.1 to derive that,

$$TV(P_{\mathcal{C}}(W_{agg}), P_{\mathcal{C}'}(W_{agg})) \leq TV(P_{\mathcal{C}}(W), P_{\mathcal{C}'}(W)) \leq \delta$$

Based on the definition of total variation distance, we have

$$\sup_{S \subset R} |P_{\mathcal{C}}(W_{agg} \in S) - P_{\mathcal{C}'}(W_{agg} \in S)| \leq \delta$$

Define the stochastic mechanism M as the projection from the client set to any model parameter $W_{agg} \in \mathcal{R}$. Then the distribution of $M(\mathcal{C})$ and $M(\mathcal{C}')$ are the distributions of W_{agg} and W'_{agg}, respectively. Hence, for any $S \subset R$:

$$P(M(\mathcal{C}) \in S) \leq P(M(\mathcal{C}') \in S) + \delta \ ,$$

which finishes the proof that Fed-ZDA satisfies $(0, \delta)$-DP. □

5.6 Conclusions

To promote fairness and robustness in federated learning, we propose a federated learning system with zero-shot data augmentation, with possible deployments at the server (Fed-ZDAS), or at the clients (Fed-ZDAC). We provide a differential privacy analysis. We note that such methods only utilize the statistics of the shared models to generate fake data. Empirical results demonstrate our method achieves both better performance and fairness over commonly used federated learning baselines. For future research, we would like to investigate the combining of Fed-ZDAS and Fed-ZDAC in the same communication round, or at alternate rounds. Similarly, for clarity of the analysis in this paper, we assumed that Fed-ZDAC and Fed-ZDAS are deployed on top of the FedAvg with a simple arithmetic mean aggregation at the server. For future research, we would like to study the effect of deploying ZSDG on top of more complex aggregation schemes.

Chapter 6

Reducing Bias in Large Scale Language Models

6.1 Introduction

Text encoders, which map raw-text data into low-dimensional embeddings, have become one of the fundamental tools for extensive tasks in natural language processing [KZS+15b, CMS+20]. With the development of deep learning, large-scale neural sentence encoders pretrained on massive text corpora, such as Infersent [CKS+17a], ELMo [PNI+18], BERT [DCLT19a], and GPT [RNSS18], have become the mainstream to extract the sentence-level text representations, and have shown desirable performance on many NLP downstream tasks [MYCG19, SLW+19, ZKW+19]. Although these pretrained models have been studied comprehensively from many perspectives, such as performance [JCL+20], efficiency [SDCW19], and robustness [LOG+19], the *fairness* of pretrained text encoders has not received significant research attention.

The fairness issue is also broadly recognized as *social bias*, which denotes the unbalanced model behaviors with respect to some socially sensitive topics, such as gender, race, and religion [LLZ+20]. For data-driven NLP models, social bias is an intrinsic problem mainly caused by the unbalanced data of text corpora [BCZ+16]. To quantitatively measure the bias degree of models, prior work proposed several statistical tests [CBN17, CM19, BAHAZ19], mostly focusing on word-level embedding models.

To evaluate the sentence-level bias in the embedding space, [MWB+19] extended the Word Embedding Association Test (WEAT) [CBN17] into a Sentence Encoder Association Test (SEAT). Based on the SEAT test, [MWB+19] claimed the existence of social bias in the pretrained sentence encoders.

Although related works have discussed the measurement of social bias in sentence embeddings, debiasing pretrained sentence encoders remains a challenge. Previous word embedding debiasing methods [BCZ+16, KB19, MLTB19] have limited assistance to sentence-level debiasing, because even if the social bias is eliminated at the word level, the sentence-level bias can still be caused by the unbalanced combination of words in the training text. Besides, retraining a state-of-the-art sentence encoder for debiasing requires a massive amount of computational resources, especially for large-scale deep models like BERT [DCLT19a] and GPT [RNSS18]. To the best of our knowledge, [LLZ+20] proposed the only sentence-level debiasing method (Sent-Debias) for pretrained text encoders, in which the embeddings are revised by subtracting the latent biased direction vectors learned by Principal Component Analysis (PCA) [WEG87]. However, Sent-Debias makes a strong assumption on the linearity of the bias in the sentence embedding space. Further, the calculation of bias directions depends highly on the embeddings extracted from the training data and the number of principal components, preventing the method from adequate generalization.

In this chapter, we proposed the first neural debiasing method for pretrained sentence encoders. For a given pretrained encoder, our method learns a fair filter (FairFil) network, whose inputs are the original embeddings of the encoder, and outputs are the debiased embeddings. Inspired by the multi-view contrastive learning [CKNH20], for each training sentence, we first generate an augmentation that has the same semantic meaning but in a different potential bias direction. We contrastively train our FairFil by maximizing the mutual information between the debiased embeddings of

the original sentences and corresponding augmentations. To further eliminate bias from sensitive words in sentences, we introduce a debiasing regularizer, which minimizes the mutual information between debiased embeddings and the sensitive words' embeddings. In the experiments, our FairFil outperforms Sent-Debias [LLZ+20] in terms of the fairness and the representativeness of debiased embeddings, indicating our FairFil not only effectively reduces the social bias in the sentence embeddings, but also successfully preserves the rich semantic meaning of input text.

6.2 Method

Suppose $E(\cdot)$ is a pretrained sentence encoder, which can encode a sentence $\boldsymbol{x}$ into low-dimensional embedding $\boldsymbol{z} = E(\boldsymbol{x})$. Each sentence $\boldsymbol{x} = (w^1, w^2, \ldots, w^L)$ is a sequence of words. The embedding space of $\boldsymbol{z}$ has been recognized to have social bias in a series of studies [MWB+19, KVP+19, LLZ+20]. To eliminate the social bias in the embedding space, we aim to learn a fair filter network $f(\cdot)$ on top of the sentence encoder $E(\cdot)$, such that the output embedding of our fair filter $\boldsymbol{d} = f(\boldsymbol{z})$ can be debiased. To train the fair filter, we design a multi-view contrastive learning framework, which consists of three steps. First, for each input sentence $\boldsymbol{x}$, we generate an augmented sentence $\boldsymbol{x}'$ that has the same semantic meaning as $\boldsymbol{x}$ but in a different potential bias direction. Then, we maximize the mutual information between the original embedding $\boldsymbol{z} = f(\boldsymbol{x})$ and the augmented embedding $\boldsymbol{z}' = f(\boldsymbol{x}')$ with the InfoNCE [OLV18] contrastive loss. Further, we design a debiasing regularizer to minimize the mutual information between $\boldsymbol{d}$ and sensitive attribute words in $\boldsymbol{x}$. In the following, we discuss these three steps in detail.

6.2.1 Data Augmentations with Sensitive Attributes

We first describe the sentence data augmentation process for our FairFil contrastive learning. Denote a social sensitive topic as $\mathcal{T} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K\}$, where $\mathcal{D}_k$ ($k = 1, \ldots, K$) is one of the potential bias directions under the topic. For example, if $\mathcal{T}$ represents the sensitive topic *"gender"*, then $\mathcal{T}$ consists two potential bias directions $\{\mathcal{D}_1, \mathcal{D}_2\} = \{$ *"male"*, *"female"*$\}$. Similarly, if $\mathcal{T}$ is set as the major *"religions"* of the world, then $\mathcal{T}$ could contain $\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4\} = \{$ *"Christianity"*, *"Islam"*, *"Judaism"*, *"Buddhism"*$\}$ as four components.

For a given social sensitive topic $\mathcal{T} = \{\mathcal{D}_1, \ldots \mathcal{D}_K\}$, if a word w is related to one of the potential bias direction $\mathcal{D}_k$ (denote as $w \in \mathcal{D}_k$), we call w a *sensitive attribute word* of $\mathcal{D}_k$ (also called bias attribute word in [LLZ+20]). For a sensitive attribute word $w \in \mathcal{D}_k$, suppose we can always find another sensitive attribute word $u \in \mathcal{D}_j$, such that w and u has the equivalent semantic meaning but in a different bias direction. Then we call u as a *replaceable word* of w in direction $\mathcal{D}_j$, and denote as $u = r_j(w)$. For the topic *"gender"* $= \{$ *"male"*, *"female"*$\}$, the word $w =$ "boy" is in the potential bias direction $\mathcal{D}_1 =$ *"male"*; a replaceable word of "boy" in *"female"* direction is $r_2(w) =$ "girl" $\in \mathcal{D}_2$.

With the above definitions, for each sentence $\boldsymbol{x}$, we generate an augmented sentence $\boldsymbol{x}'$ such that $\boldsymbol{x}'$ has the same semantic meaning as $\boldsymbol{x}$ but in a different potential bias direction. More specifically, for a sentence $\boldsymbol{x} = (w^1, w^2, \ldots, w^L)$, we first find the sensitive word positions as an index set $\mathcal{P}$, such that each w^p ($p \in \mathcal{P}$) is a sensitive attribute words in direction $\mathcal{D}_k$. We further make a reasonable assumption that the embedding bias of direction $\mathcal{D}_k$ is only caused by the sensitive words $\{w^p\}_{p \in \mathcal{P}}$ in $\boldsymbol{x}$. To sample an augmentation to $\boldsymbol{x}$, we first select another potential bias direction $\mathcal{D}_j$, and then replace all sensitive attribute words by their replaceable words in the

Table 6.1: Examples of generating an augmentation sentence under the sensitive topic *"gender"*.

	Bias direction	Sensitive Attribute words	Text content
Original	male	he, his	{He} is good at playing {his} basketball.
Augmented	female	she, her	{She} is good at playing {her} basketball.

direction $\mathcal{D}_j$. That is, $\boldsymbol{x}' = \{v^1, v^2, \ldots, v^L\}$, where $v^l = w^l$ if $l \notin \mathcal{P}$, and $v^l = r_j(w^l)$ if $l \in \mathcal{P}$. In Table 6.1, we provide an example for sentence augmentation under the *"gender"* topic.

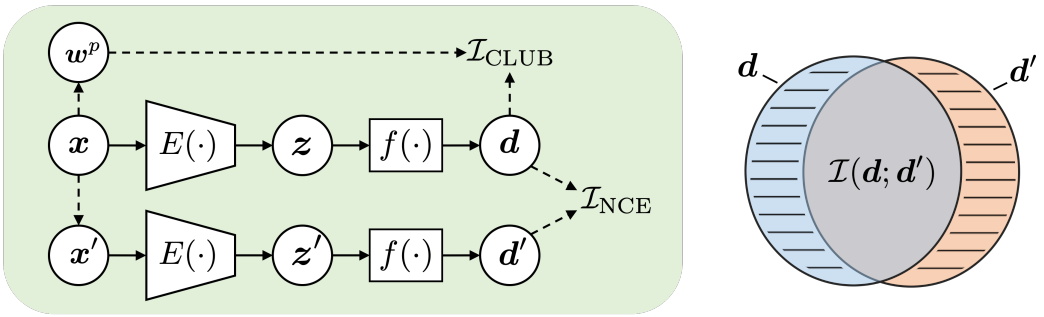
6.2.2 Contrastive Learning Framework

After obtaining the sentence pair $(\boldsymbol{x}, \boldsymbol{x}')$ with the augmentation strategy from Section 6.2.1, we construct a contrastive learning framework to learn our debiasing fair filter $f(\cdot)$. As shown in the Figure 6.1(a), our framework consists of the following two steps:

(1) We encode sentences $(\boldsymbol{x}, \boldsymbol{x}')$ into embeddings $(\boldsymbol{z}, \boldsymbol{z}')$ with the pretrained encoder $E(\cdot)$. Since $\boldsymbol{x}$ and $\boldsymbol{x}'$ have the same meaning but different potential bias directions, the embeddings $(\boldsymbol{z}, \boldsymbol{z}')$ will have different bias directions, which are caused by the sensitive attributed words in $\boldsymbol{x}$ and $\boldsymbol{x}'$.

(2) We then feed the sentence embeddings $(\boldsymbol{z}, \boldsymbol{z}')$ through our fair filter $f(\cdot)$ to obtain the debiased embedding outputs $(\boldsymbol{d}, \boldsymbol{d}')$. Ideally, $\boldsymbol{d}$ and $\boldsymbol{d}'$ should represent the same semantic meaning without social bias. Inspired by SimCLR [CKNH20], we encourage the overlapped semantic information between $\boldsymbol{d}$ and $\boldsymbol{d}'$ by maximizing their mutual information $\mathcal{I}(\boldsymbol{d}; \boldsymbol{d}')$.

However, the calculation of $\mathcal{I}(\boldsymbol{d}; \boldsymbol{d}')$ is practically difficult because only embedding samples of $\boldsymbol{d}$ and $\boldsymbol{d}'$ are available. Therefore, we use the InfoNCE mutual information estimator [OLV18] to minimize the lower bound of $\mathcal{I}(\boldsymbol{d}; \boldsymbol{d}')$ instead. Based on a



(a) (b)

Figure 6.1: (a) Contrastive learning framework of FairFil; (b) Illustration of information in $\boldsymbol{d}$ and $\boldsymbol{d}'$.

learnable score function $g(\cdot, \cdot)$, the contrastive InfoNCE estimator is calculated within a batch of samples $\{(\boldsymbol{d}_i, \boldsymbol{d}'_i)\}_{i=1}^N$:

$$\mathcal{I}_{\text{NCE}} = \frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(g(\boldsymbol{d}_i, \boldsymbol{d}'_i))}{\frac{1}{N} \sum_{j=1}^{N} \exp(g(\boldsymbol{d}_i, \boldsymbol{d}'_j))}. \tag{6.1}$$

By maximize $\mathcal{I}_{\text{NCE}}$, we encourage the difference between the positive pair score $g(\boldsymbol{d}_i, \boldsymbol{d}'_i)$ and the negative pair score $g(\boldsymbol{d}_i, \boldsymbol{d}'_j)$, so that $\boldsymbol{d}_i$ can share more semantic information with $\boldsymbol{d}'_i$ than other embeddings $\boldsymbol{d}'_{j \neq i}$.

6.2.3 Debiasing Regularizer

Practically, the contrastive learning framework in Section 6.2.2 can already show encouraging debiasing performance (as shown in the Experiments). However, the embedding $\boldsymbol{d}$ can contain extra biased information from $\boldsymbol{z}$, that only maximizing $\mathcal{I}(\boldsymbol{d}; \boldsymbol{d}')$ fails to eliminate. To encourage no extra bias in $\boldsymbol{d}$, we introduce a debiasing regularizer which minimizes the mutual information between embedding $\boldsymbol{d}$ and the potential bias from embedding $\boldsymbol{z}$. As discussed in Section 6.2.1, in our framework the potential bias of $\boldsymbol{z}$ is assumed to come from the sensitive attribute words in $\boldsymbol{x}$. Therefore, we should reduce the bias word information from the debiased representation $\boldsymbol{d}$.

Let $\boldsymbol{w}^p$ be the embedding of a sensitive attribute word w^p in sentence $\boldsymbol{x}$. The word embedding $\boldsymbol{w}^p$ can always be obtained from the pretrained text encoders [BB19]. We then minimize the mutual information $\mathcal{I}(\boldsymbol{w}^p; \boldsymbol{d})$, using the CLUB mutual information upper bound [CHD+20] to estimate $\mathcal{I}(\boldsymbol{w}^p; \boldsymbol{d})$ with embedding samples. Given a batch of embedding pairs $\{(\boldsymbol{d}_i, \boldsymbol{w}^p)\}_{i=1}^N$, we can calculate the debiasing regularizer as:

$$\mathcal{I}_{\text{CLUB}} = \frac{1}{N}\sum_{i=1}^{N}\Big[\log q_\theta(\boldsymbol{w}_i^p|\boldsymbol{d}_i) - \frac{1}{N}\sum_{j=1}^{N}\log q_\theta(\boldsymbol{w}_j^p|\boldsymbol{d}_i)\Big], \tag{6.2}$$

where q_θ is a variational approximation to ground-truth conditional distribution $p(\boldsymbol{w}|\boldsymbol{d})$. We parameterize q_θ with another neural network. As proved in [CHD+20], the better $q_\theta(\boldsymbol{w}|\boldsymbol{d})$ approximates $p(\boldsymbol{w}|\boldsymbol{d})$, the more accurate $\mathcal{I}_{\text{CLUB}}$ serves as the mutual information upper bound. Therefore, besides the loss in (6.2), we also maximize the log-likelihood of $q_\theta(\boldsymbol{w}|\boldsymbol{d})$ with samples $\{(\boldsymbol{d}_i, \boldsymbol{w}_i^p)\}_{i=1}^N$.

Based on the above sections, the overall learning scheme of our fair filter (FairFil) is described in Algorithm 6. Also, we provide an intuitive explanation to the two loss terms in our framework. In Figure 6.1(b), the blue and red circles represent $\boldsymbol{d}$ and $\boldsymbol{d}'$, respectively, in the embedding space. The intersection $\mathcal{I}(\boldsymbol{d}; \boldsymbol{d}')$ is the common semantic information extracted from sentences $\boldsymbol{x}$ and $\boldsymbol{x}'$, while the two shadow parts are the extra bias. Note that the perfect debiased embeddings lead to coincident circles. By maximizing $\mathcal{I}_{\text{NCE}}$ term, we enlarge the overlapped area of $\boldsymbol{d}$ and $\boldsymbol{d}'$; by minimizing $\mathcal{I}_{\text{CLUB}}$, we shrink the biased shadow parts.

6.3 Related Work

6.3.1 Bias in Natural Language Processing

Social bias has recently been recognized as an important issue in natural language processing (NLP) systems. The studies on bias in NLP are mainly delineated into two

Algorithm 6 Updating the FairFil with a sample batch

Begin with the pretrained text encoder $E(\cdot)$, and a batch of sentences $\{\boldsymbol{x}_i\}_{i=1}^N$.
Find the sensitive attribute words $\{w^p\}$ and corresponding embeddings $\{\boldsymbol{w}^p\}$.
Generate augmentation $\boldsymbol{x}'_i$ from $\boldsymbol{x}_i$, by replacing $\{w^p\}$ with $\{r_j(w^p)\}$.
Encode $(\boldsymbol{x}_i, \boldsymbol{x}'_i)$ into embeddings $\boldsymbol{d}_i = f(E(\boldsymbol{x}_i)), \boldsymbol{d}'_i = f(E(\boldsymbol{x}'_i))$.
Calculate $\mathcal{I}_{\text{NCE}}$ with $\{(\boldsymbol{d}_i, \boldsymbol{d}'_i)\}_{i=1}^N$ and score function g.
if adding debiasing regularizer **then**
    Update the variational approximation $q_\theta(\boldsymbol{w}|\boldsymbol{d})$ by maximizing log-likelihood with $\{(\boldsymbol{d}_i, \boldsymbol{w}_i^p)\}$
    Calculate $\mathcal{I}_{\text{CLUB}}$ with $q_\theta(\boldsymbol{w}|\boldsymbol{d})$ and $\{(\boldsymbol{d}_i, \boldsymbol{w}_i^p)\}_{i=1}^N$.
    Learning loss $\mathcal{L} = -\mathcal{I}_{\text{NCE}} + \beta\mathcal{I}_{\text{CLUB}}$.
else
    Learning loss $\mathcal{L} = -\mathcal{I}_{\text{NCE}}$.
end if
Update FairFil f and score function g by gradient descent with respect to $\mathcal{L}$.

categories: bias in the embedding spaces, and bias in downstream tasks [BBDIW20]. For bias in downstream tasks, the analyses cover comprehensive topics, including machine translation [SSZ19], language modeling [BB19], sentiment analysis [KM18b] and toxicity detection [DLS+18]. The social bias in embedding spaces has been studied from two important perspectives: bias measurements and and debiasing methods. To measure the bias in an embedding space, [CBN17] proposed a Word Embedding Association Test (WEAT), which compares the similarity between two sets of target words and two sets of attribute words. [MWB+19] further extended the WEAT to a Sentence Encoder Association Test (SEAT), which replaces the word embeddings by sentence embeddings encoded from pre-defined biased sentence templates. For debiasing methods, most of the prior works focus on word-level representations [BCZ+16, BB19]. The only sentence-level debiasing method is proposed by [LLZ+20], which learns bias directions by PCA and subtracts them in the embedding space.

6.3.2 Contrastive Learning

Contrastive learning is a broad class of training strategies that learns meaningful representations by making positive and negative embedding pairs more distinguishable. Usually, contrastive learning requires a pairwise embedding critic as a similarity/distance of data pairs. Then the learning objective is constructed by maximizing the margin between the critic values of positive data pairs and negative data pairs. Previously contrastive learning has shown encouraging performance in many tasks, including metric learning [WBS06, DKJ+07], word representation learning [MCCD13], graph learning [TQW+15, GL16], *etc.* Recently, contrastive learning has been applied to the unsupervised visual representation learning task, and significantly reduced the performance gap between supervised and unsupervised learning [HFW+20, CKNH20, QMG+20]. Among these unsupervised methods, [CKNH20] proposed a simple multi-view contrastive learning framework (SimCLR). For each image data, SimCLR generates two augmented images, and then the mutual information of the two augmentation embeddings is maximized within a batch of training data.

6.4 Experiments

We first describe the experimental setup in detail, including the pretrained encoders, the training of FairFil, and the downstream tasks. The results of our FairFil are reported and analyzed, along with the previous Sent-Debias method. In general, we evaluate our neural debiasing method from two perspectives: (1) fairness: we compare the bias degree of the original and debiased sentence embeddings for debiasing performance; and (2) representativeness: we apply the debiased embeddings into downstream tasks, and compare the performance with original embeddings.

6.4.1 Bias Evaluation Metric

To evaluate the bias in sentence embeddings, we use the Sentence Encoder Association Test (SEAT) [MWB+19], which is an extension of the Word Embedding Association Test (WEAT) [CBN17]. The WEAT test measures the bias in word embeddings by comparing the distances of two sets of target words to two sets of attribute words. More specifically, denote $\mathcal{X}$ and $\mathcal{Y}$ as two sets of target word embeddings (*e.g.*, $\mathcal{X}$ includes *"male"* words such as "boy" and "man"; $\mathcal{Y}$ contains *"female"* words like "girl" and "woman"). The attribute sets $\mathcal{A}$ and $\mathcal{B}$ are selected from some social concepts that should be "equal" to $\mathcal{X}$ and $\mathcal{Y}$ (*e.g.*, career or personality words). Then the bias degree *w.r.t* attributes $(\mathcal{A}, \mathcal{B})$ of each word embedding $\boldsymbol{t}$ is defined as:

$$s(\boldsymbol{t}, \mathcal{A}, \mathcal{B}) = \text{mean}_{\boldsymbol{a}\in\mathcal{A}} \cos(\boldsymbol{t}, \boldsymbol{a}) - \text{mean}_{\boldsymbol{b}\in\mathcal{B}} \cos(\boldsymbol{t}, \boldsymbol{b}), \tag{6.3}$$

where $\cos(\cdot, \cdot)$ is the cosine similarity. Based on (6.3), the normalized WEAT effect size is:

$$d_{\text{WEAT}} = \frac{\text{mean}_{\boldsymbol{x}\in\mathcal{X}} s(\boldsymbol{x}, \mathcal{A}, \mathcal{B}) - \text{mean}_{\boldsymbol{y}\in\mathcal{Y}} s(\boldsymbol{y}, \mathcal{A}, \mathcal{B})}{\text{std}_{\boldsymbol{t}\in\mathcal{X}\cup\mathcal{Y}} s(\boldsymbol{t}, \mathcal{A}, \mathcal{B})}. \tag{6.4}$$

The SEAT test extends WEAT by replacing the word embeddings with sentence embeddings. Both target words and attribute words are converted into sentences with several semantically bleached sentence templates (*e.g.*, "This is <word>"). Then the SEAT statistic is similarly calculated with (6.4) based on the embeddings of converted sentences. The closer the effect size is to zero, the more fair the embeddings are. Therefore, we report the absolute effect size as the bias measure.

6.4.2 Pretrained Encoders

We test our neural debiasing method on BERT [DCLT19a]. Since the pretrained BERT requires the additional fine-tuning process for downstream tasks, we report

the performance of our FairFil under two scenarios: (1) pretrained BERT: we directly learn our FairFil network based on pretrained BERT without any additional fine-tuning; and (2) BERT post tasks: we fix the parameters of the FairFil network learned on pretrained BERT, and then fine-tune the BERT+FairFil together on task-specific data. Note that when fine-tuning, our FairFil will no longer update, which satisfies a fair comparison to Sent-Debias [LLZ+20].

For the downstream tasks of BERT, we follow the setup from Sent-Debias [LLZ+20] and conduct experiments on the following three downstream tasks: (1) SST-2: A sentiment classification task on the Stanford Sentiment Treebank (SST-2) dataset [SPW+13], on which sentence embeddings are used to predict the corresponding sentiment labels; (2) CoLA: Another sentiment classification task on the Corpus of Linguistic Acceptability (CoLA) grammatical acceptability judgment [WSB19]; and (3) QNLI: A binary question answering task on the Question Natural Language Inference (QNLI) dataset [WSM+18].

6.4.3 Training of FairFil

We parameterize the fair filter network with one-layer fully-connected neural networks with the ReLU activation function. The score function g in the InfoNCE estimator is set to a two-layer fully-connected network with one-dimensional output. The variational approximation q_θ in CLUB estimator is parameterized by a multivariate Gaussian distribution $q_\theta(\boldsymbol{w}|\boldsymbol{d}) = N(\boldsymbol{\mu}(\boldsymbol{d}), \boldsymbol{\sigma}^2(\boldsymbol{d}))$, where $\boldsymbol{\mu}(\cdot)$ and $\boldsymbol{\sigma}(\cdot)$ are also two-layer fully-connected neural nets. The batch size is set to 128. The learning rate is 1×10^{-5}. We train the fair filter for 10 epochs.

For an appropriate comparison, we follow the setup of Sent-Debias [LLZ+20] and select the same training data for the training of FairFil. The training corpora consist 183,060 sentences from the following five datasets: WikiText-2 [MXBS1y], Stan-

Table 6.2: Performance of debiased embeddings on Pretrained BERT and BERT post SST-2.

	Pretrained BERT				BERT post SST-2			
	Origin	Sent-D	FairF$^-$	FairF	Origin	Sent-D	FairF$^-$	FairF
Names, Career/Family	0.477	**0.096**	0.218	0.182	0.036	**0.109**	0.237	0.218
Terms, Career/Family	0.108	0.437	0.086	**0.076**	0.010	**0.057**	0.376	0.377
Terms, Math/Arts	0.253	0.194	0.133	**0.124**	0.219	**0.221**	0.301	0.263
Names, Math/Arts	0.254	0.194	0.101	**0.082**	1.153	0.755	**0.084**	0.099
Terms, Science/Arts	0.399	**0.075**	0.218	0.204	0.103	**0.081**	0.133	0.127
Names, Science/Arts	0.636	0.540	0.320	**0.235**	0.222	0.047	0.017	**0.005**
Avg. Abs. Effect Size	0.354	0.256	0.179	**0.150**	0.291	0.212	0.191	**0.182**
Classification Acc.	-	-	-	-	92.7	89.1	**91.7**	91.6

Table 6.3: Performance of debiased embeddings on BERT post CoLA and BERT post QNLI.

	BERT post CoLA				BERT post QNLI			
	Origin	Sent-D	FairF$^-$	FairF	Origin	Sent-D	FairF$^-$	FairF
Names, Career/Family	0.009	0.149	0.273	**0.034**	0.261	**0.054**	0.196	0.103
Terms, Career/Family	0.199	0.186	0.156	**0.119**	0.155	**0.004**	0.050	0.206
Terms, Math/Arts	0.268	0.311	**0.008**	0.092	0.584	**0.083**	0.306	0.323
Names, Math/Arts	0.150	0.308	**0.060**	0.101	0.581	0.629	**0.168**	0.288
Terms, Science/Arts	0.425	**0.163**	0.245	0.249	0.087	0.716	0.500	**0.245**
Names, Science/Arts	0.032	0.192	**0.102**	0.127	0.521	0.443	0.378	**0.167**
Avg. Abs. Effect Size	0.181	0.217	0.141	**0.120**	0.365	0.321	0.266	**0.222**
Classification Acc.	57.6	55.4	**56.5**	**56.5**	91.3	90.6	**91.0**	90.8

ford Sentiment Treebank [SPW+13], Reddit [VPSS17], MELD [PHM+19] and POM [PSC+14]. Following [LLZ+20], we mainly select *"gender"* as the sensitive topic $\mathcal{T}$, and use the same pre-defined word sets of sensitive attribute words and their replaceable words as Sent-Debias did. The word embeddings for training the debiasing regularizer is selected from the token embedding of the pretrained BERT.

6.4.4 Debiasing Results

In Tables 6.2 and 6.3 we report the evaluation results of debiased embeddings on both the absolute SEAT effect size and the downstream classification accuracy. For the SEAT test, we follow the setup in [LLZ+20], and test the sentence templates of

Table 6.4: Comparison of average debiasing performance on pretrained BERT

Method	Bias Degree
BERT origin [DCLT19a]	0.354
FastText [BGJM17]	0.565
BERT word [BCZ+16]	0.861
BERT simple [MWB+19]	0.298
Sent-Debias [LLZ+20]	0.256
FairFil$^-$ (Ours)	0.179
FairFil (Ours)	**0.150**

Terms/Names under different domains designed by [CBN17]. The column name Origin refers to the original BERT results, and Sent-D is short for Sent-Debias [LLZ+20]. FairFil$^-$ and FairFil (as FairF$^-$ and FairF in the tables) are our method without/with the debiasing regularizer in Section 6.2.3. The best results of effect size (the lower the better) and classification accuracy (the higher the better) are bold among Sent-D, FairFil$^-$, and FairFil. Since the pretrained BERT does not correspond to any downstream task, the classification accuracy is not reported for it.

From the SEAT test results, our contrastive learning framework effectively reduces the gender bias for both pretrained BERT and fine-tuned BERT under most test scenarios. Comparing with Sent-Debias, our FairFil reaches a lower bias degree on the majority of the individual SEAT tests. Considering the average of absolute effect size, our FairFil is distinguished by a significant margin to Sent-Debias. Moreover, our FairFil achieves higher downstream classification accuracy than Sent-Debias, which indicates learning neural filter networks can preserve more semantic meaning than subtracting bias directions learned from PCA.

For the ablation study, we also report the results of FairFil without the debiasing regularizer, as in FairF$^-$. Only with the contrastive learning framework, FairF$^-$ already reduces the bias effectively and even achieves better effect size than the

FairF on some of the SEAT tests. With the debiasing regularizer, FairF has better average SEAT effect sizes but slightly loses in terms of the downstream performance. However, the overall performance of FairF and FairF$^-$ shows a trade-off between fairness and representativeness of the filter network.

We also compare the debiasing performance on a broader class of baselines, including word-level debiasing methods, and report the average absolute SEAT effect size on the pretrained BERT encoder. Both FairF$^-$ and FairF achieve a lower bias degree than other baselines. The word-level debiasing methods (FastText [BGJM17] and BERT word [BCZ$^+$16]) have the worst debiasing performance, which validates our observation that the word-level debiasing methods cannot reduce sentence-level social bias in NLP models.

6.4.5 Analysis

To test the influence of data proportion on the model's debiasing performance, we select WikiText-2 with 13,750 sentences as the training corpora following the setup in [LLZ$^+$20]. Then we randomly divide the training data into 5 equal-sized partitions. We evaluate the bias degree of the sentence debiasing methods on different combinations of the partitions, specifically with training data proportions (20%, 40%, 60%, 80%, 100%). Under each data proportion, we repeat the training 5 times to obtain the mean and variance of the absolute SEAT effect size. In Figure 6.2, we plot the bias degree of BERT post tasks with different training data proportions. In general, both Sent-Debias and FairFil achieve better performance and smaller variance when the proportion of training data is larger. Under a 20% training proportion, our FairFil can better remove bias in text encoder, which shows FairFil has better data efficiency with the contrastive learning framework.

To further study output debiased sentence embedding, we visualize the relative dis-

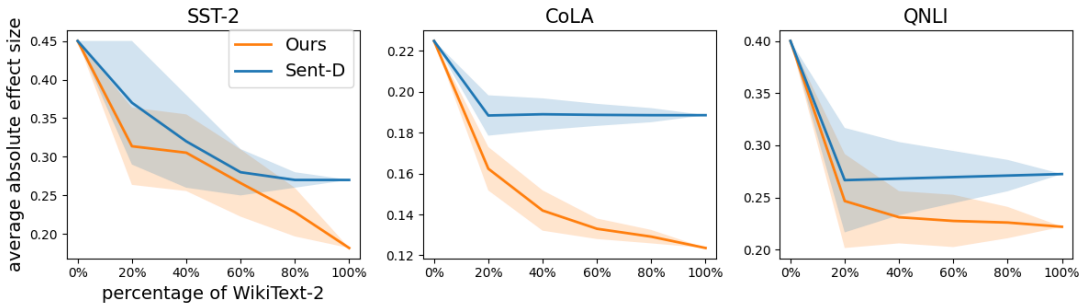


Figure 6.2: Influence of the training data proportion to debias degree of BERT.

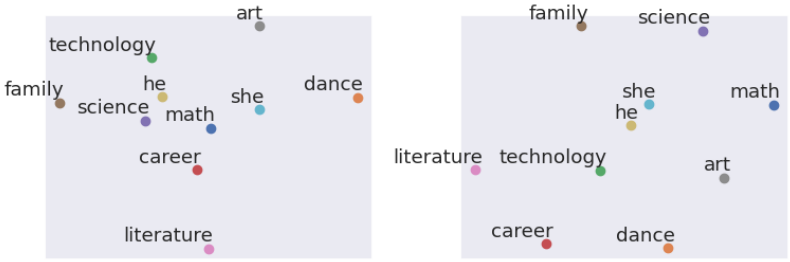


Figure 6.3: T-SNE plots of each words contextualized in templates. Left-hand side: the original pretrained BERT; right-hand side: FairFil.

tances of attributes and targets of SEAT before/after our debiasing process. We choose the target words as "he" and "she.". We first contextualize the selected words into sentence templates as described in Section 6.4.1. We then average the original/debiased embeddings of these sentence template and plot the t-SNE [MH08] in Figure 6.3. From the t-SNE, the debiased encoder provides more balanced distances from gender targets "he/she" to the attribute concepts.

6.5 Conclusions

This chapter has developed a novel debiasing method for large-scale pretrained text encoder. We proposed a fair filter (FairFil), which takes the original sentence embeddings as input and outputs the debiased sentence embeddings. To train the fair filter, we constructed a multi-view contrast learning framework, which maximizes the mutual information between each sentence and its augmentation. The augmented

sentence is generated by replacing sensitive words in the original sentence with words in a similar semantic but different bias directions. Further, we designed a debiasing regularizer that minimizes the mutual information between the debiased embeddings and the corresponding sensitive words in sentences. This *post hoc* method does not require access to the training corpora, or any retraining process of the text encoder, which enhances its applicability.

Chapter 7

Conclusions

Large scale pre-training has become the dominant approach within deep learning during recent years. Strategies in knowledge transfer for constructing models such as feature transformation and parameter sharing lead to successful application of foundation models.

In this dissertation, I first explore the pre-training on a large amount of image-text-action triplets in a self-supervised learning manner, and achieve the state of the art results on three different vision-and-language navigation tasks, which validates the effectiveness of pre-training for complex tasks. To compress large-scale pre-trained models, I propose a data-agnostic distillation framework that leverages Mixup, a simple yet efficient data augmentation approach, to endow the resulting model with stronger generalization ability. To avoid a single global modal in federated learning, I propose anonymized weight factorization, an approach that combines the Indian Buffet Process with a shared dictionary of weight factors for neural networks, and achieves significant improvement to local test performance and fairness while simultaneously providing an extra layer of security. Lastly, I propose to filter out the large scale language model's bias by a contrastive learning method and calibrate the skewed data distribution by a zero-shot data augmentation technique for federated learning.

In sum, this work explores possible solution to the scale problems raised in foundation model era. Broad application prospects of large-scale models and constantly updated data collection constraints will definitely motivate further exploration.

Bibliography

[AASC19] Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

[ABC+14] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 81–91, 2014.

[ABVK20] Vítor Albiero, Kevin Bowyer, Kushal Vangara, and Michael King. Does face recognition accuracy get better with age? deep face matchers say no. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 261–269, 2020.

[ACC+18] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir Zamir. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

[ACG+16] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[ADSFS19] Jayadev Acharya, Chris De Sa, Dylan Foster, and Karthik Sridharan. Distributed learning with sublinear communication. volume 97 of *Proceedings of Machine Learning Research*, pages 40–50, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[AFDM17] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

[AHB+18] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[AHD+19] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *CVPR*, 2019.

[AKB+17] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *CoRR*, abs/1608.04207, 2017.

[AKV+20] Vítor Albiero, Krishnapriya KS, Kushal Vangara, Kai Zhang, Michael C King, and Kevin W Bowyer. Analysis of gender inequality in face recognition accuracy. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops*, pages 81–89, 2020.

[Alt92] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[AMK+17] Ganesh Ajjanagadde, Anuran Makur, Jason Klusowski, Sheng Xu, et al. Lecture notes on information theory. 2017.

[ASZAA07] Mehdi Aghagolzadeh, Hamid Soltanian-Zadeh, B Araabi, and Ali Aghagolzadeh. A hierarchical clustering based on mutual information maximization. In *2007 IEEE International Conference on Image Processing*, volume 1, pages I–277. IEEE, 2007.

[AWT+18] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2018.

[BA03] David Barber and Felix V Agakov. The im algorithm: a variational approach to information maximization. In *Advances in neural information processing systems*, page None, 2003.

[BAHAZ19] Marc-Etienne Brunet, Colleen Alkalay-Houlihan, Ashton Anderson, and Richard Zemel. Understanding the origins of bias in word embeddings. In *International Conference on Machine Learning*, pages 803–811, 2019.

[BAPM15a] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *EMNLP*, 2015.

[BAPM15b] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical*

Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2015.

[BAPM15c] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[Bax00] J. Baxter. A Model of Inductive Bias Learning. *Journal of Artificial Intelligence Research*, 2000.

[BB19] Shikha Bordia and Samuel R. Bowman. Identifying and reducing gender bias in word-level language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 7–15, Minneapolis, Minnesota, 2019.

[BBDIW20] Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. Language (technology) is power: A critical survey of" bias" in nlp. *arXiv preprint arXiv:2005.14050*, 2020.

[BBR+18] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Devon Hjelm, and Aaron Courville. Mutual information neural estimation. In *International Conference on Machine Learning*, pages 530–539, 2018.

[BC94] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, pages 359–370. Seattle, WA, 1994.

[BCC+19] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring. *International Conference on Learning Representations*, 2019.

[BCDG09] Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The Fifth PASCAL Recognizing Textual Entailment Challenge. *TAC*, 2009.

[BCG+19] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. MixMatch: A Holistic Approach to Semi-Supervised Learning. *Neural Information Processing Systems*, 2019.

[BCKW15] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight Uncertainty in Neural Networks. *International Conference on Machine Learning*, 2015.

[BCZ+16] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016.

[BDW+20] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, et al. UniLMv2: Pseudo-masked Language Models for Unified Language Model Pre-training. *arXiv preprint arXiv:2002.12804*, 2020.

[BGJM17] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[BHA+21] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[BHP+18] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*, 2018.

[BKH16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[BM98] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*. ACM, 1998.

[BMR+20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[BTS+16] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *NeurIPS*, 2016.

[CA07] James A Coan and John JB Allen. *Handbook of emotion elicitation and assessment*. Oxford university press, 2007.

[CB02] George Casella and Roger L Berger. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.

[CB19] Luca Corinzia and Joachim M Buhmann. Variational Federated Multi-Task Learning. *arXiv preprint arXiv:1906.06268*, 2019.

[CBK+10] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka, and Tom M Mitchell. Toward an architecture for never-ending language learning. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[CBN17] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

[CCEKL20] Yoojin Choi, Jihwan Choi, Mostafa El-Khamy, and Jungwon Lee. Data-free network quantization with adversarial knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 710–711, 2020.

[CCK+18] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[CcYyLsL18] Juchieh Chou, Cheng chieh Yeh, Hung yi Lee, and Lin shan Lee. Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations. In *Proc. Interspeech 2018*, pages 501–505, 2018.

[CDF+17] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[CDH+16] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

[CDLH18] Fei Chen, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated Meta-learning for Recommendation. *arXiv preprint arXiv:1802.07876*, 2018.

[CG97] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[CH+67] Thomas M Cover, Peter E Hart, et al. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[Cha12] Santosh V Chapaneri. Spoken digits recognition using weighted mfcc and improved features for dynamic time warping. *International Journal of Computer Applications*, 40(3):6–12, 2012.

[CHD+20] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. Club: A contrastive log-ratio upper bound of mutual information. In *Proceedings of the 37th international conference on Machine learning*, 2020.

[CK18] Alexis Conneau and Douwe Kiela. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*, 2018.

[CKMT18] Sebastian Caldas, Jakub Konečny, H Brendan McMahan, and Ameet Talwalkar. Expanding the Reach of Federated Learning by Reducing Client Resource Requirements. *arXiv preprint arXiv:1812.07210*, 2018.

[CKNH20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th international conference on Machine learning*, 2020.

[CKS+17a] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*, 2017.

[CKS+17b] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[CKS+17c] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*, 2017.

[CL19] Ju-chieh Chou and Hung-Yi Lee. One-shot voice conversion by separating speaker and content representations with instance normalization. *Proc. Interspeech 2019*, pages 664–668, 2019.

[CLGD18] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620, 2018.

[CLK+19] Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D Manning, and Quoc V Le. BAM! Born-again Multi-task Networks for Natural Language Understanding. *arXiv preprint arXiv:1907.04829*, 2019.

[CLKM15] Benjamin Charrow, Sikang Liu, Vijay Kumar, and Nathan Michael. Information-theoretic mapping using cauchy-schwarz quadratic mutual information. In *ICRA*, 2015.

[CLLM20] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. ELECTRA: Pre-training Text Encoders as Discriminators Rather than Generators. *arXiv preprint arXiv:2003.10555*, 2020.

[CLTB21] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*, pages 1931–1942. PMLR, 2021.

[CLY+17] Dongdong Chen, Jing Liao, Lu Yuan, Nenghai Yu, and Gang Hua. Coherent online video style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1105–1114, 2017.

[CLZ+20] Pengyu Cheng, Yitong Li, Xinyuan Zhang, Liqun Cheng, David Carlson, and Lawrence Carin. Dynamic embedding on textual networks via a gaussian process. In *AAAI*, 2020.

[CM19] Kaytlin Chaloner and Alfredo Maldonado. Measuring gender bias in word embeddings across domains and discovering new gender bias word categories. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 25–32, 2019.

[CMG17] Anmol Chachra, Pulkit Mehndiratta, and Mohit Gupta. Sentiment analysis of text using deep convolution neural networks. In *2017 Tenth International Conference on Contemporary Computing (IC3)*, pages 1–6. IEEE, 2017.

[CMS18] Ting Chen, Martin Renqiang Min, and Yizhou Sun. Learning k-way d-dimensional discrete codes for compact embedding representations. *arXiv preprint arXiv:1806.09464*, 2018.

[CMS+20] Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. Improving

disentangled text representation learning with information-theoretic guidance. *arXiv preprint arXiv:2006.00693*, 2020.

[CPO19] Kai-Wei Chang, Vinod Prabhakaran, and Vicente Ordonez. Bias and fairness in natural language processing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): Tutorial Abstracts*, 2019.

[CPR15] Miguel A Carreira-Perpinán and Ramin Raziperchikolaei. Hashing with binary autoencoders. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 557–566, 2015.

[CSM+10] Howard Chen, Alane Shur, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. *CVPR*, 2010.

[CT12] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[CWT+19] Liqun Chen, Guoyin Wang, Chenyang Tao, Dinghan Shen, Pengyu Cheng, Xinyuan Zhang, Wenlin Wang, Yizhe Zhang, and Lawrence Carin. Improving textual network embedding with global attention via optimal transport. In *ACL*, 2019.

[CYD+20] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13169–13178, 2020.

[CYY20] Jiaao Chen, Zichao Yang, and Diyi Yang. MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification. *Association for Computational Linguistics*, July 2020.

[CYyK+18] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder. *CoRR*, abs/1803.11175, 2018.

[D+17] Emily L Denton et al. Unsupervised learning of disentangled representations from video. In *Advances in neural information processing systems*, pages 4414–4423, 2017.

[DB05] William B Dolan and Chris Brockett. Automatically Constructing a Corpus of Sentential Paraphrases. *International Workshop on Paraphrasing*, 2005.

[DBK+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[DBV16] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[DCLT19a] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, 2019.

[DCLT19b] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL*, 2019.

[DCM+12] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, et al. Large Scale Distributed Deep Networks. *Neural Information Processing Systems*, 2012.

[DCZ+19] Shuyang Dai, Yu Cheng, Yizhe Zhang, Zhe Gan, Jingjing Liu, and Lawrence Carin. Contrastively smoothed class alignment for unsupervised domain adaptation. *arXiv preprint arXiv:1909.05288*, 2019.

[DDG+18] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *CVPR*, 2018.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A Large-scale Hierarchical Image Database. *Computer Vision and Pattern Recognition*, 2009.

[DFKE08] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image restoration by sparse 3d transform-domain collaborative filtering. In *Image Processing: Algorithms and Systems VI*, volume 6812, page 681207. International Society for Optics and Photonics, 2008.

[DGK07] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(11):1944–1957, 2007.

[DGK+17] Bo Dai, Ruiqi Guo, Sanjiv Kumar, Niao He, and Le Song. Stochastic generative hashing. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 913–922. JMLR. org, 2017.

[DGM05] Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL Recognising Textual Entailment Challenge. *Machine Learning Challenges Workshop*, 2005.

[DKJ+07] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216, 2007.

[DL15] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087, 2015.

[DLS+18] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73, 2018.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating Noise to Sensitivity in Private Data Analysis. *Theory of Cryptography Conference*, pages 265–284, 2006.

[DNP13] Shivanker Dev Dhingra, Geeta Nijhawan, and Poonam Pandit. Isolated speech recognition using mfcc and dtw. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 2(8):4085–4092, 2013.

[DS98] Kathleen Dahlgren and Edward Stabler. Natural language understanding system, August 11 1998. US Patent 5,794,050.

[DTN20] Canh T Dinh, Nguyen H Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. *arXiv preprint arXiv:2006.08848*, 2020.

[DV99] Georges A Darbellay and Igor Vajda. Estimation of the information by an adaptive partitioning of the observation space. *IEEE Transactions on Information Theory*, 1999.

[EKM+19] Hubert Eichner, Tomer Koren, H Brendan McMahan, Nathan Srebro, and Kunal Talwar. Semi-cyclic Stochastic Gradient Descent. *arXiv preprint arXiv:1904.10120*, 2019.

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[FDA17] Carlos Florensa, Yan Duan, and Pieter Abbeel. Stochastic neural networks for hierarchical reinforcement learning. *arXiv preprint arXiv:1704.03012*, 2017.

[FHC+18] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, pages 3314–3325, 2018.

[FJR15] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. *ACM SIGSAC Conference on Computer and Communications Security*, 2015.

[FL03] Beat Fasel and Juergen Luettin. Automatic facial expression analysis: a survey. *Pattern recognition*, 36(1):259–275, 2003.

[FMO20] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.

[GB10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[GBR+12] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[GCL+20] Prakhar Ganesh, Yao Chen, Xin Lou, Mohammad Ali Khan, Yin Yang, Deming Chen, Marianne Winslett, Hassan Sajjad, and Preslav Nakov. Compressing Large-scale Transformer-based Models: A Case Study on BERT. *arXiv preprint arXiv:2002.11985*, 2020.

[Gea35] Roy C Geary. The ratio of the mean deviation to the standard deviation as a test of normality. *Biometrika*, 27(3/4):310–332, 1935.

[GEB16] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[GFC20] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020.

[GG06] Zoubin Ghahramani and Thomas L Griffiths. Infinite Latent Feature Models and the Indian Buffet Process. *Neural Information Processing Systems*, 2006.

[GH10] 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.

[GHDL18] Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor OK Li. Universal neural machine translation for extremely low resource languages. *arXiv preprint arXiv:1802.05368*, 2018.

[Gir15] Ross Girshick. Fast R-CNN. In *CVPR*, 2015.

[GKN17] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[GL94] Clive Granger and Jin-Lung Lin. Using the mutual information coefficient to identify lags in nonlinear models. *Journal of time series analysis*, 1994.

[GL16] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[GMDD07] Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The Third PASCAL Recognizing Textual Entailment Challenge. *ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, 2007.

[GMZ19] Hongyu Guo, Yongyi Mao, and Richong Zhang. Augmenting Data with Mixup for Sentence Classification: An Empirical Study. *arXiv preprint arXiv:1905.08941*, 2019.

[GPH+17] Zhe Gan, Yunchen Pu, Ricardo Henao, Chunyuan Li, Xiaodong He, and Lawrence Carin. Learning generic sentence representations using convolutional neural networks. In *EMNLP*, 2017.

[GPL+19] Sahaj Garg, Vincent Perot, Nicole Limtiaco, Ankur Taly, Ed H Chi, and Alex Beutel. Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 219–226, 2019.

[GRC11] Elizabeth Godoy, Olivier Rosec, and Thierry Chonavel. Voice conversion using dynamic frequency warping with amplitude scaling, for parallel or nonparallel corpora. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(4):1313–1323, 2011.

[GSR+18] Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Unsupervised multi-target domain adaptation: An information theoretic approach. *arXiv preprint arXiv:1810.11547*, 2018.

[GSR+20] Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Unsupervised multi-target domain adaptation: An information theoretic approach. *IEEE Transactions on Image Processing*, 29:3993–4002, 2020.

[GT20] Jack Goetz and Ambuj Tewari. Federated learning via synthetic data. *arXiv preprint arXiv:2008.04489*, 2020.

[GUA+16] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016.

[HB17] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[HCK16] Felix Hill, Kyunghyun Cho, and Anna Korhonen. Learning distributed representations of sentences from unlabelled data. In *HLT-NAACL*, 2016.

[HCX+21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[HDD+06] R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The Second PASCAL Recognising Textual Entailment Challenge. *PASCAL Challenges Workshop on Recognising Textual Entailment*, 2006.

[HFLM+18] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2018.

[HFW+20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[HHIL+17] Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. Private Federated Learning on Vertically Partitioned Data via Entity Resolution and Additively Homomorphic Encryption. *arXiv preprint arXiv:1711.10677*, 2017.

[HHW+16] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang. Voice conversion from non-parallel corpora using variational auto-encoder. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–6. IEEE, 2016.

[Hin12] G Hinton. Neural networks for machine learning. coursera,[video lectures], 2012.

[HKGV11] Jihun Hamm, Christian G Kohler, Ruben C Gur, and Ragini Verma. Automated facial action coding system for dynamic analysis of facial expressions in neuropsychiatric disorders. *Journal of neuroscience methods*, 200(2):237–256, 2011.

[HLH+18] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li F Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. In *Advances in Neural Information Processing Systems*, pages 517–526, 2018.

[HLY19] Wei Hu, Zhiyuan Li, and Dingli Yu. Understanding generalization of deep neural networks trained with noisy labels. *arXiv preprint arXiv:1905.11368*, 2019.

[HMC+20] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A Simple Method to

Improve Robustness and Uncertainty under Data Shift. *International Conference on Learning Representations*, 2020.

[HML+20] Weituo Hao, Nikhil Mehta, Kevin J Liang, Pengyu Cheng, Mostafa El-Khamy, and Lawrence Carin. WAFFLe: Weight Anonymized Factorization for Federated Learning. *arXiv preprint arXiv:2008.05687*, 2020.

[HMP+16] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2016.

[HMSW04] Wolfgang Karl Härdle, Marlene Müller, Stefan Sperlich, and Axel Werwatz. *Nonparametric and Semiparametric Models*. Springer Science & Business Media, 2004.

[HMT+17] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *ICML*, 2017.

[HNP09] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 2009.

[HPR+17] Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1480–1490. JMLR. org, 2017.

[HR20] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020.

[HS97a] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 1997.

[HS97b] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[HSK+12] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[HVD15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015.

[HVG11] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

[HWL+17] Haozhi Huang, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, and Wei Liu. Real-time neural style transfer for videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 783–791, 2017.

[HYF+18] 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.

[HYL17a] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[HYL+17b] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org, 2017.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[JBP+02] S. V. Jones, S. Brown, T. Parker, V. Stubblefield, E. Kössler, C. Simmons, and J. Wittner. Anomalous indeterminate amorphous transitory gradations observed in certain specific nongeneralizable locii attendant to unique fleeting unpredictable phenomena. *Journal of Nonspecific Research*, 88:665–703, 2002.

[JBS17] Yacine Jernite, Samuel R. Bowman, and David A Sontag. Discourse-based objectives for fast unsupervised sentence representation learning. *CoRR*, abs/1705.00557, 2017.

[JBvdM+18] Sarthak Jain, Edward Banner, Jan-Willem van de Meent, Iain J Marshall, and Byron C Wallace. Learning disentangled representations of texts with application to biomedical abstracts. *arXiv preprint arXiv:1804.07212*, 2018.

[JCL+20] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Association for Computational Linguistics*, 2020.

[JGBM16] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[JGP16] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[JKR14] Brian J Julian, Sertac Karaman, and Daniela Rus. On mutual information-based control of range sensing robots for mapping applications. *The International Journal of Robotics Research*, 2014.

[JKRK19] Yihan Jiang, Jakub Konečnỳ, Keith Rush, and Sreeram Kannan. Improving Federated Learning Personalization via Model Agnostic Meta Learning. *arXiv preprint arXiv:1909.12488*, 2019.

[JMBV19a] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. Disentangled representation learning for non-parallel text style transfer. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[JMBV19b] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. Disentangled representation learning for non-parallel text style transfer. In *ACL*, 2019.

[Joa96] Thorsten Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. Technical report, Carnegie-mellon univ pittsburgh pa dept of computer science, 1996.

[JOK+18] 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.

[JYL15] Bo Jiang, Chao Ye, and Jun S Liu. Nonparametric k-sample tests via dynamic slicing. *Journal of the American Statistical Association*, 2015.

[JYS+19] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for Natural Language Understanding. *arXiv preprint arXiv:1909.10351*, 2019.

[KAS11] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. Fairness-aware learning through regularization approach. In *IEEE 11th International Conference on Data Mining Workshops*, 2011.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KB19] Masahiro Kaneko and Danushka Bollegala. Gender-preserving debiasing for pre-trained word embeddings. *arXiv preprint arXiv:1906.00742*, 2019.

[KBBT19] Ashutosh Kumar, Satwik Bhattamishra, Manik Bhandari, and Partha Talukdar. Submodular Optimization-based Diverse Paraphrasing and Its Effectiveness in Data Augmentation. *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[KBK+12] Brendan F Klare, Mark J Burge, Joshua C Klontz, Richard W Vorder Bruegge, and Anil K Jain. Face recognition performance: Role of demographic information. *IEEE Transactions on Information Forensics and Security*, 7(6):1789–1801, 2012.

[KBT19] Mikhail Khodak, Maria-Florina F Balcan, and Ameet S Talwalkar. Adaptive Gradient-based Meta-learning Methods. *Neural Information Processing Systems*, 2019.

[KC18] Jamie Kiros and William Chan. Inferlite: Simple universal sentence representations from natural language inference data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4868–4874, 2018.

[KCT00] Takeo Kanade, Jeffrey F Cohn, and Yingli Tian. Comprehensive database for facial expression analysis. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pages 46–53. IEEE, 2000.

[KD18] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.

[KK18] Takuhiro Kaneko and Hirokazu Kameoka. Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2100–2104. IEEE, 2018.

[KKTH18] Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo. Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 266–273. IEEE, 2018.

[KL51] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[KLB+19] Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha Srinivasa. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. *CVPR*, 2019.

[KM18a] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *ICML*, 2018.

[KM18b] Svetlana Kiritchenko and Saif Mohammad. Examining gender and race bias in two hundred sentiment analysis systems. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 43–53, 2018.

[KMG+17] Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An interactive 3D environment for visual AI. *arXiv preprint arXiv:1712.05474*, 2017.

[KMY+16] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated Learning: Strategies for Improving Communication Efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[Kri09] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. 2009.

[KSAG05] Alexander Kraskov, Harald Stögbauer, Ralph G Andrzejak, and Peter Grassberger. Hierarchical clustering using mutual information. *EPL (Europhysics Letters)*, 70(2):278, 2005.

[KSC76] Peter Ewen King-Smith and D Carden. Luminance and opponent-color contributions to visual detection and adaptation and to temporal and spatial integration. *JOSA*, 66(7):709–717, 1976.

[KSC16] Meina Kan, Shiguang Shan, and Xilin Chen. Multi-view deep network for cross-view classification. In *CVPR*, 2016.

[KSG04] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 2004.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*, 2012.

[KSK21] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021.

[KST+18] Hadi Kazemi, Sobhan Soleymani, Fariborz Taherkhani, Seyed Iranmanesh, and Nasser Nasrabadi. Unsupervised image-to-image translation using domain-specific variational information bound. In *NeurIPS*, 2018.

[Kub93] Robert Kubichek. Mel-cepstral distance measure for objective speech quality assessment. In *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, volume 1, pages 125–128. IEEE, 1993.

[Kul97] Solomon Kullback. *Information theory and statistics*. Courier Corporation, 1997.

[Kum80] Ponnambalam Kumaraswamy. A Generalized Probability Density Function for Double-Bounded Random Processes. *Journal of Hydrology*, 1980.

[KVAMR18] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4281–4289, 2018.

[KVP+19] Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. Measuring bias in contextualized word representations. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 166–172, 2019.

[KW13] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[KW16] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[KZS+15a] Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Skip-thought vectors. In *NIPS*, 2015.

[KZS+15b] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

[LB18] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA, 2018. Association for Computational Linguistics.

[LBB+98] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[LBL+19] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124, 2019.

[LBPL19] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. VilBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *NIPS*, 2019.

[LCK+10] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 94–101. IEEE, 2010.

[LCL+09] Patrick Lucey, Jeffrey Cohn, Simon Lucey, Iain Matthews, Sridha Sridharan, and Kenneth M Prkachin. Automatically detecting pain using facial actions. In *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, pages 1–8. IEEE, 2009.

[LCWJ15] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

[LDF+19] Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. Unicoder-VL: A universal encoder for vision and language by cross-modal pre-training. *arXiv preprint arXiv:1908.06066*, 2019.

[Lea18] Susan Leavy. Gender bias in artificial intelligence: The need for diversity and gender theory in machine learning. In *Proceedings of the 1st international workshop on gender equality in software engineering*, pages 14–16, 2018.

[LFS+17] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*, 2017.

[LGLC16] Alexander Lachmann, Federico M Giorgi, Gonzalo Lopez, and Andrea Califano. Aracne-ap: gene network reverse engineering through adaptive partitioning inference of mutual information. *Bioinformatics*, 2016.

[LHCG19] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task Deep Neural Networks for Natural Language Understanding. *arXiv preprint arXiv:1901.11504*, 2019.

[LHG+18] Kevin J Liang, Geert Heilmann, Christopher Gregory, Souleymane O. Diallo, David Carlson, Gregory P. Spell, John B. Sigman, Kris Roe, and Lawrence Carin. Automatic Threat Recognition of Prohibited Items at Aviation Checkpoints with X-ray Imaging: A Deep Learning Approach. *SPIE Anomaly Detection and Imaging with X-Rays (ADIX) III*, 2018.

[LHS+21] Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. MixKD: Towards Efficient Distillation of Large-scale Language Models. *International Conference on Learning Representations*, 2021.

[LJS+20] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020.

[LL18] Lajanugen Logeswaran and Honglak Lee. An efficient framework for learning sentence representations. *ICLR*, 2018.

[LLC+21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[LLG+19] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[LLX+19] Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah Smith, and Yejin Choi. Robust navigation with language pretraining and stochastic sampling. *arXiv preprint arXiv:1909.02244*, 2019.

[LLY20] Saehyung Lee, Hyungyu Lee, and Sungroh Yoon. Adversarial Vertex Mixup: Toward Better Adversarially Robust Generalization. *Computer Vision and Pattern Recognition*, 2020.

[LLZ+20] Paul Pu Liang, Irene Mengze Li, Emily Zheng, Yao Chong Lim, Ruslan Salakhutdinov, and Louis-Philippe Morency. Towards debiasing sentence representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5502–5515, 2020.

[LMW+22] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022.

[LOG+19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019.

[LPR17] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Neural Information Processing Systems*, 2017.

[LPSB17] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4990–4998, 2017.

[LSBS19] Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

[LSS+19] Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. Multiple-attribute text rewriting. In *International Conference on Learning Representations*, 2019.

[LSTS19] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated Learning: Challenges, Methods, and Future Directions. *arXiv preprint arXiv:1908.07873*, 2019.

[LSTS20] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[LSZ+18] 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.

[LTH+18] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 35–51, 2018.

[LW06] Chung-Han Lee and Chung-Hsien Wu. Map-based adaptation for speech conversion using adaptation data selection and non-parallel training. In *Ninth International Conference on Spoken Language Processing*, 2006.

[LW19] Daliang Li and Junpu Wang. FedMD: Heterogenous Federated Learning via Model Distillation. *arXiv preprint arXiv:1910.03581*, 2019.

[LWL+17] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017.

[LWL+19] Linqing Liu, Huan Wang, Jimmy Lin, Richard Socher, and Caiming Xiong. Attentive Student Meets Multi-task Teacher: Improved Knowledge Distillation for Pretrained Models. *arXiv preprint arXiv:1911.03588*, 2019.

[LYF+18] Yen-Cheng Liu, Yu-Ying Yeh, Tzu-Chien Fu, Sheng-De Wang, Wei-Chen Chiu, and Yu-Chiang Frank Wang. Detach and adapt: Learning cross-domain disentangled deep representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8867–8876, 2018.

[LZZ+02] Stan Z Li, Long Zhu, ZhenQiu Zhang, Andrew Blake, HongJiang Zhang, and Harry Shum. Statistical learning of multi-view face detection. In *ECCV*. Springer, 2002.

[MA19] Subhabrata Mukherjee and Ahmed Hassan Awadallah. Distilling Transformers into Simple Neural Networks with Unlabeled Transfer Data. *arXiv preprint arXiv:1910.01769*, 2019.

[MBE10] Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc)

and dynamic time warping (dtw) techniques. *arXiv preprint arXiv:1003.4083*, 2010.

[MC89] Michael McCloskey and Neal J Cohen. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *The Psychology of Learning and Motivation*, 1989.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[MDDC15] Luca Melis, George Danezis, and Emiliano De Cristofaro. Efficient Private Statistics with Succinct Sketches. *arXiv preprint arXiv:1508.06110*, 2015.

[MDH+20] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 2437–2445, 2020.

[Mei07] Marina Meilă. Comparing clusterings—an information based distance. *Journal of multivariate analysis*, 98(5):873–895, 2007.

[MF11] Markku Makitalo and Alessandro Foi. Optimal inversion of the anscombe transformation in low-count poisson image denoising. *IEEE transactions on Image Processing*, 20(1):99–109, 2011.

[MFL+19] Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved Knowledge Distillation via Teacher Assistant. *arXiv preprint arXiv:1902.03393*, 2019.

[MH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 2008.

[MH14] Kevin R Moon and Alfred O Hero. Ensemble estimation of multivariate f-divergence. In *2014 IEEE International Symposium on Information Theory*, pages 356–360. IEEE, 2014.

[MHGP17] Will Monroe, Robert XD Hawkins, Noah D Goodman, and Christopher Potts. Colors in context: A pragmatic neural model for grounded language understanding. *TACL*, 2017.

[MIA99] Malik Magdon-Ismail and Amir F Atiya. Neural networks for density estimation. In *NeurIPS*, 1999.

[MK17] Seyed Hamidreza Mohammadi and Alexander Kain. An overview of voice conversion systems. *Speech Communication*, 88:65–82, 2017.

[MLA17] Dipendra Misra, John Langford, and Yoav Artzi. Mapping instructions and visual observations to actions with reinforcement learning. *EMNLP*, 2017.

[MLD+20] Chuan Ma, Jun Li, Ming Ding, Howard H Yang, Feng Shu, Tony QS Quek, and H Vincent Poor. On safeguarding privacy and security in the framework of federated learning. *IEEE Network*, 2020.

[MLTB19] Thomas Manzini, Yao Chong Lim, Yulia Tsvetkov, and Alan W Black. Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. *arXiv preprint arXiv:1904.04047*, 2019.

[MLVC21] Nikhil Mehta, Kevin J Liang, Vinay K Verma, and Lawrence Carin. Continual Learning using a Bayesian Nonparametric Dictionary of Weight Factors. *Artificial Intelligence and Statistics*, 2021.

[MLW+19] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. *ICLR*, 2019.

[MMR+17] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient Learning of Deep Networks from Decentralized Data. *Artificial Intelligence and Statistics*, 2017.

[MMT17] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. *International Conference on Learning Representations*, 2017.

[MQ10] Anderson F Machado and Marcelo Queiroz. Voice conversion: A critical survey. 2010.

[MRT18] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. 2018.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[MSG+18] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In *CVPR*, 2018.

[MSS19] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic Federated Learning. *International Conference on Machine Learning*, 2019.

[MW17] Willem Marais and Rebecca Willett. Proximal-gradient methods for poisson image reconstruction with bm3d-based regularization. In *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 1–5. IEEE, 2017.

[MWA+19] Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The regretful agent: Heuristic-aided navigation through progress estimation. *CVPR*, 2019.

[MWB+19] Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. On measuring social biases in sentence encoders. 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)*, pages 622–628, 2019.

[MXBS1y] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *ICLR*, 201y.

[MYCG19] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. Cedr: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1101–1104, 2019.

[Naw28] B. Nawahi. Acoustic production in liquid filled ceramic environments. *J. Chem. Phys.*, 108:9893–9904, 1928.

[NBG17] Allen Nie, Erin D. Bennett, and Noah D. Goodman. Dissent: Sentence representation learning from explicit discourse relations. *CoRR*, abs/1710.04334, 2017.

[NCZ17] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017.

[NDI19] Khanh Nguyen and Hal Daumé III. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. *EMNLP*, 2019.

[NH10] Vinod Nair and Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. *International Conference on Machine Learning*, 2010.

[NSH19] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks Against Centralized and Federated Learning. *IEEE Symposium on Security and Privacy*, 2019.

[NTSS06] Keigo Nakamura, Tomoki Toda, Hiroshi Saruwatari, and Kiyohiro Shikano. Speaking aid system for total laryngectomees using voice conversion of body transmitted artificial speech. In *Ninth International Conference on Spoken Language Processing*, 2006.

[NWJ10] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 2010.

[ODZ+16] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[OEB+19] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. *North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, 2019.

[OLV18] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[OS19] Samet Oymak and Mahdi Soltanolkotabi. Overparameterized nonlinear learning: Gradient descent takes the shortest path? In *ICML*, 2019.

[PARS14] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[PCPK15] 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.

[PGJ18] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. In *NAACL-HLT*, 2018.

[Pha16] Phatpiglet. phatpiglet/autocorrect, Aug 2016.

[PHM+19] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536, 2019.

[PHZS19] Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. Federated Adversarial Domain Adaptation. *arXiv preprint arXiv:1911.02054*, 2019.

[PKM19] Daniel Peterson, Pallika Kanani, and Virendra J Marathe. Private Federated Learning with Domain Adaptation. *arXiv preprint arXiv:1912.06733*, 2019.

[PNI+18] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237, 2018.

[POVDO+19] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180, 2019.

[PRWZ02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[PSC+14] Sunghyun Park, Han Suk Shim, Moitreya Chatterjee, Kenji Sagae, and Louis-Philippe Morency. Computational analysis of persuasiveness in social multimedia: A novel dataset and multimodal prediction approach. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 50–57, 2014.

[PSF18] Ji Ho Park, Jamin Shin, and Pascale Fung. Reducing gender bias in abusive language detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2799–2804, 2018.

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[QMG+20] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *arXiv preprint arXiv:2008.03800*, 2020.

[QSS+20] Yanru Qu, Dinghan Shen, Yelong Shen, Sandra Sajeev, Jiawei Han, and Weizhu Chen. Coda: Contrast-enhanced and diversity-promoting data augmentation for natural language understanding. *arXiv preprint arXiv:2010.08670*, 2020.

[QYQ+19] Chen Qu, Liu Yang, Minghui Qiu, W Bruce Croft, Yongfeng Zhang, and Mohit Iyyer. Bert with history answer embedding for conversational question answering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1133–1136, 2019.

[QZC+19] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. Autovc: Zero-shot voice style transfer with only autoencoder loss. In *International Conference on Machine Learning*, pages 5210–5219, 2019.

[RCL+09] Andrew Ryan, Jeffery F Cohn, Simon Lucey, Jason Saragih, Patrick Lucey, Fernando De la Torre, and Adam Rossi. Automated facial expression recognition system. In *43rd annual 2009 international carnahan conference on security technology*, pages 172–177. IEEE, 2009.

[RH18] Sebastian Ruder and Jeremy Howard. Universal language model fine-tuning for text classification. In *ACL*, 2018.

[RHU+18] Dezsö Ribli, Anna Horváth, Zsuzsa Unger, Péter Pollner, and István Csabai. Detecting and Classifying Lesions in Mammograms with Deep Learning. *Nature Scientific Reports*, 8, 2018.

[RK18] Sujith Ravi and Zornitsa Kozareva. Self-governing neural networks for on-device short text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 887–893, 2018.

[RM15] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.

[RMM+17] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, 2017.

[RN16] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[RNSS18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

[RTC17] Aaditya Ramdas, Nicolás Trillos, and Marco Cuturi. On Wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2):47, 2017.

[RZLL16] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *Empirical Methods in Natural Language Processing*, 2016.

[SAK+19] Neta Shoham, Tomer Avidor, Aviv Keren, Nadav Israel, Daniel Benditkis, Liron Mor-Yosef, and Itai Zeitak. Overcoming forgetting in federated learning on non-iid data. *arXiv preprint arXiv:1910.07796*, 2019.

[San05] F. Sanford. Oxidized ferrous materials. *Journal of Junk*, 5(4):324–345, 2005.

[SCD+17] Manolis Savva, Angel X Chang, Alexey Dosovitskiy, Thomas Funkhouser, and Vladlen Koltun. MINOS: Multimodal indoor simulator for navigation in complex environments. *arXiv preprint arXiv:1712.03931*, 2017.

[SCGL19a] S. Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient Knowledge Distillation for BERT Model Compression. *Empirical Methods in Natural Language Processing*, 2019.

[SCGL19b] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient Knowledge Distillation for BERT Model Compression. *arXiv preprint arXiv:1908.09355*, 2019.

[SCM98] Yannis Stylianou, Olivier Cappé, and Eric Moulines. Continuous probabilistic transform for voice conversion. *IEEE Transactions on speech and audio processing*, 6(2):131–142, 1998.

[SCST17] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[SDCW19] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, A Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[SGC16] Jiaming Song, Zhe Gan, and Lawrence Carin. Factored Temporal Sigmoid Belief Networks for Sequence Learning. *International Conference on Machine Learning*, 2016.

[SH70] M. Smith and I. A. Hall. *Handbook of Interstellar Travel*. Dover, New York, 7th edition, 1970.

[SH09] Ruslan Salakhutdinov and Geoffrey Hinton. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978, 2009.

[SHH+19] Kurt Shuster, Samuel Humeau, Hexiang Hu, Antoine Bordes, and Jason Weston. Engaging image captioning via personality. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12516–12526, 2019.

[SINT18] Yuki Saito, Yusuke Ijima, Kyosuke Nishida, and Shinnosuke Takamichi. Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5274–5278. IEEE, 2018.

[SK08] Karthik Sridharan and Sham M Kakade. An information theoretic framework for multi-view learning. *COLT*, 2008.

[SKG+19] Jiaming Song, Pratyusha Kalluri, Aditya Grover, Shengjia Zhao, and Stefano Ermon. Learning controllable fair representations. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2164–2173, 2019.

[SLBJ17] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. In *Advances in neural information processing systems*, pages 6830–6841, 2017.

[SLDV98] Patrice Y Simard, Yann A LeCun, John S Denker, and Bernard Victorri. Transformation Invariance in Pattern Recognition–Tangent Distance and Tangent Propagation. *Neural Networks: Tricks of the Trade*, 1998.

[SLDV17] Rachit Singh, Jeffrey Ling, and Finale Doshi-Velez. Structured Variational Autoencoders for the Beta-Bernoulli Process. *Neural Information Processing Systems Workshop on Advances in Approximate Bayesian Inference*, 2017.

[SLKK17] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual Learning with Deep Generative Replay. *Neural Information Processing Systems*, 2017.

[SLS+18] Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. Multiple-attribute text style transfer. *arXiv preprint arXiv:1811.00552*, 2018.

[SLW+19] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1441–1450, 2019.

[SMV+19] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. VideoBERT: A joint model for video and language representation learning. *ICCV*, 2019.

[SN17] Raphael Shu and Hideki Nakayama. Compressing word embeddings via deep compositional code learning. *arXiv preprint arXiv:1711.01068*, 2017.

[SNB+08] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

[SPP19] Joan Serrà, Santiago Pascual, and Carlos Segura Perales. Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion. In *Advances in Neural Information Processing Systems*, pages 6790–6800, 2019.

[SPW+13] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank. *Empirical Methods in Natural Language Processing*, 2013.

[SPW+18] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[SSC+18] Dinghan Shen, Qinliang Su, Paidamoyo Chapfuwa, Wenlin Wang, Guoyin Wang, Lawrence Carin, and Ricardo Henao. Nash: Toward end-to-end neural architecture for generative semantic hashing. In *ACL*, 2018.

[SSG+13] Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, Kenji Fukumizu, et al. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5):2263–2291, 2013.

[SSP03] PY Simard, D Steinkraus, and JC Platt. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. *International Conference on Document Analysis and Recognition*, 2003.

[SSSG17] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. *International Conference on Computer Vision*, 2017.

[SSSK08] Taiji Suzuki, Masashi Sugiyama, Jun Sese, and Takafumi Kanamori. Approximating mutual information by maximum likelihood density ratio estimation. In *New challenges for feature selection in data mining and knowledge discovery*, 2008.

[SSSS17] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. *IEEE Symposium on Security and Privacy (SP)*, 2017.

[SSZ19] Gabriel Stanovsky, Noah A Smith, and Luke Zettlemoyer. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, 2019.

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[SWL+19] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. ERNIE: Enhanced Representation through Knowledge Integration. *arXiv preprint arXiv:1904.09223*, 2019.

[SWUH18] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018.

[SYKM17] Ananda Theertha Suresh, Felix X Yu, Sanjiv Kumar, and H Brendan McMahan. Distributed Mean Estimation with Limited Communication. *International Conference on Machine Learning*, 2017.

[SYS+20] Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. MobileBERT: A Compact Task-agnostic BERT for Resource-limited Devices. *arXiv preprint arXiv:2004.02984*, 2020.

[SZC+19] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.

[SZL+18] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[SZS+14] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing Properties of Neural Networks. *International Conference on Learning Representations*, 2014.

[SZS+18] Berrak Sisman, Mingyang Zhang, Sakriani Sakti, Haizhou Li, and Satoshi Nakamura. Adaptive wavenet vocoder for residual compensation in gan-based voice conversion. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 282–289. IEEE, 2018.

[SZS+20] Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. *arXiv preprint arXiv:2009.13818*, 2020.

[TB19] Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. *EMNLP*, 2019.

[TBA+19] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pages 1–11, 2019.

[TC19] Yi Chern Tan and L Elisa Celis. Assessing social and intersectional biases in contextualized word representations. In *Advances in Neural Information Processing Systems*, pages 13230–13241, 2019.

[TCLT19] Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read Students Learn Better: On the Importance of Pre-training Compact Models. *arXiv preprint arXiv:1908.08962*, 2019.

[TdS18] Shuai Tang and Virginia R de Sa. Improving sentence representations with multi-view frameworks. *arXiv preprint arXiv:1810.01064*, 2018.

[TF20] Aleksei Triastcyn and Boi Faltings. Federated Generative Privacy. *IEEE Intelligent Systems*, 2020.

[TGB19] Jesse Thomason, Daniel Gordon, and Yonatan Bisk. Shifting the baseline: Single modality performance on visual navigation & qa. In *NAACL*, 2019.

[TGG07] Yee Whye Teh, Dilan Grür, and Zoubin Ghahramani. Stick-breaking construction for the indian buffet process. In *Artificial Intelligence and Statistics*, pages 556–563, 2007.

[TH09] Graham W Taylor and Geoffrey E Hinton. Factored Conditional Restricted Boltzmann Machines for Modeling Motion Style. *International Conference on Machine Learning*, 2009.

[THG19] Julien Tissier, Amaury Habrard, and Christophe Gravier. Near-lossless binarization of word embeddings. *AAAI*, 2019.

[TJ07] Romain Thibaux and Michael I Jordan. Hierarchical Beta Processes and the Indian Buffet Process. *Artificial Intelligence and Statistics*, 2007.

[TLJ07] Yan Tong, Wenhui Liao, and Qiang Ji. Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE transactions on pattern analysis and machine intelligence*, 29(10):1683–1699, 2007.

[TLL+19] Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling Task-specific Knowledge from BERT into Simple Neural Networks. *arXiv preprint arXiv:1903.12136*, 2019.

[TMCZ19] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. *CoRL*, 2019.

[TPB00] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[TQW+15] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.

[TYB19] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. *EMNLP*, 2019.

[TYL17] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1415–1424, 2017.

[UVL16] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[VB10] Fernando Villavicencio and Jordi Bonada. Applying voice conversion to concatenative singing-voice synthesis. In *Eleventh annual conference of the international speech communication association*, 2010.

[VÇE+08] Esra Vural, Müjdat Çetin, Aytül Erçil, Gwen Littlewort, Marian Bartlett, and Javier Movellan. Automated drowsiness detection for improved driving safety. 2008.

[VDL10] Benjamin Van Durme and Ashwin Lall. Online generation of locality sensitive hash signatures. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 231–235. Association for Computational Linguistics, 2010.

[vdMH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing high-dimensional data using t-SNE. *JMLR*, 2008.

[VFH+18] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.

[VLB+19] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold Mixup: Better Representations by Interpolating Hidden States. *International Conference on Machine Learning*, 2019.

[VLK+19] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation Consistency Training for Semi-supervised Learning. *International Joint Conference on Artificial Intelligence*, 2019.

[VNG99] Gertjan Van Noord and Dale Gerdemann. An extendible regular expression compiler for finite-state approaches in natural language processing. In *International Workshop on Implementing Automata*, pages 122–139. Springer, 1999.

[VPSS17] Michael V"olske, Martin Potthast, Shahbaz Syed, and Benno Stein. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, 2017.

[VQL+19] Vikas Verma, Meng Qu, Alex Lamb, Yoshua Bengio, Juho Kannala, and Jian Tang. GraphMix: Improved Training of GNNs for Semi-Supervised Learning. *arXiv preprint arXiv:1909.11715*, 2019.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. *Neural Information Processing Systems*, 2017.

[WALB15] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *ICML*, 2015.

[WBGL16] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Towards universal paraphrastic sentence embeddings. *CoRR*, abs/1511.08198, 2016.

[WBS06] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2006.

[WEG87] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[WG18] John Wieting and Kevin Gimpel. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *ACL*, 2018.

[WHC+19] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. *CVPR*, 2019.

[Wil92] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 1992.

[WK18] John Wieting and Douwe Kiela. No training required: Exploring random encoders for sentence classification. *CoRR*, abs/1901.10444, 2018.

[WLD+20] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 2020.

[WLG10] Kuansan Wang, Xiaolong Li, and Jianfeng Gao. Multi-style language model for web scale information retrieval. In *SIGIR*. ACM, 2010.

[WLWG20] Dongdong Wang, Yandong Li, Liqiang Wang, and Boqing Gong. Neural Networks Are More Productive Teachers Than Human Raters: Active Mixup for Data-Efficient Knowledge Distillation from a Blackbox Model. *Computer Vision and Pattern Recognition*, 2020.

[WLZ+19] Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. Conditional BERT Contextual Augmentation. *International Conference on Computational Science*, 2019.

[WNB17] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

[WNB18] Adina Williams, Nikita Nangia, and Samuel R Bowman. A Broad-coverage Challenge Corpus for Sentence Understanding through Inference. *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018.

[WPN+19] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

[WSB19] Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

[WSM+18] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, 2018.

[WSM+19] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. GLUE: A Multi-task Benchmark and Analysis Platform for Natural Language Understanding. *International Conference on Learning Representations*, 2019.

[WSSJ14] Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji. Hashing for similarity search: A survey. *arXiv preprint arXiv:1408.2927*, 2014.

[WSZ+19] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2512–2520. IEEE, 2019.

[WVHC08] Michael M Wolf, Frank Verstraete, Matthew B Hastings, and J Ignacio Cirac. Area laws in quantum systems: mutual information and correlations. *Physical review letters*, 100(7):070502, 2008.

[WWPM18] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE, 2018.

[WWY16] Mirjam Wester, Zhizheng Wu, and Junichi Yamagishi. Analysis of the voice conversion challenge 2016 evaluation results. In *Interspeech*, pages 1637–1641, 2016.

[WXWW18] Xin Wang, Wenhan Xiong, Hongmin Wang, and William Yang Wang. Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. *ECCV*, 2018.

[WYS+20] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated Learning with Matched Averaging. *arXiv preprint arXiv:2002.06440*, 2020.

[WZ19] Jason Wei and Kai Zou. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. *arXiv preprint arXiv:1901.11196*, 2019.

[WZTE18] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset Distillation. *arXiv preprint arXiv:1811.10959*, 2018.

[XDH+19] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised Data Augmentation for Consistency Training. *arXiv preprint arXiv:1904.12848*, 2019.

[XRV17] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[XTX13] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

[XWT+15] Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. Convolutional neural networks for text hashing. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[YAAMB20] Seyma Yucer, Samet Akçay, Noura Al-Moubayed, and Toby P Breckon. Exploring racial bias within face recognition via per-subject adversarially-enabled data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 18–19, 2020.

[YAG+19] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. Bayesian Nonparametric Federated Learning of Neural Networks. *arXiv preprint arXiv:1905.12022*, 2019.

[YDL+18] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. QANet: Combining Local Convolution with Global Self-attention for Reading Comprehension. *arXiv preprint arXiv:1804.09541*, 2018.

[YDY+19] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *Neural Information Processing Systems*, 2019.

[YHD+18] Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems*, pages 7287–7298, 2018.

[YHO+19] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. *International Conference on Computer Vision*, 2019.

[YLZ+15] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information. In *IJCAI*, volume 2015, pages 2111–2117, 2015.

[YM18] Li Yingzhen and Stephan Mandt. Disentangled sequential autoencoder. In *International Conference on Machine Learning*, pages 5656–5665, 2018.

[YMA+20] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020.

[YVM+19] Junichi Yamagishi, Christophe Veaux, Kirsten MacDonald, et al. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92). 2019.

[YZH+21] Qian Yang, Jianyi Zhang, Weituo Hao, Gregory Spell, and Lawrence Carin. Flop: Federated learning on medical datasets using partial networks. *arXiv preprint arXiv:2102.05218*, 2021.

[ZANMB16] Diego J Zea, Diego Anfossi, Morten Nielsen, and Cristina Marino-Buslje. Mitos. jl: mutual information tools for protein sequence analysis in the julia language. *Bioinformatics*, 2016.

[ZCDLP18] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. *International Conference on Learning Representations*, 2018.

[ZCG+20] Ruiyi Zhang, Changyou Chen, Zhe Gan, Wenlin Wang, Dinghan Shen, Guoyin Wang, Zheng Wen, and Lawrence Carin. Improving adversarial text generation by modeling the distant future. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[ZGSZ19] Sanqiang Zhao, Raghav Gupta, Yang Song, and Denny Zhou. Extreme Language Model Compression with Optimal Subwords and Shared Projections. *arXiv preprint arXiv:1909.11687*, 2019.

[ZJP+20] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 253–261, 2020.

[ZJZ10] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52, 2010.

[ZKW+19] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[ZKZ+18] Junbo Zhao, Yoon Kim, Kelly Zhang, Alexander Rush, and Yann LeCun. Adversarially regularized autoencoders. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5902–5911, 2018.

[ZLdM18] Xunjie Zhu, Tingfeng Li, and Gerard de Melo. Exploring semantic properties of sentence embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 632–637, 2018.

[ZLH19] Ligeng Zhu, Zhijian Liu, and Song Han. Deep Leakage from Gradients. *Neural Information Processing Systems*, 2019.

[ZLL+18] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated Learning with Non-IID Data. *arXiv preprint arXiv:1806.00582*, 2018.

[ZLL+19] Hang Zhou, Yu Liu, Ziwei Liu, Ping Luo, and Xiaogang Wang. Talking face generation by adversarially disentangled audio-visual representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9299–9306, 2019.

[ZPRH09] Zhihong Zeng, Maja Pantic, Glenn I Roisman, and Thomas S Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE transactions on pattern analysis and machine intelligence*, 31(1):39–58, 2009.

[ZPZ+20] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and VQA. *AAAI*, 2020.

[ZS18] James Zou and Londa Schiebinger. Ai can be sexist and racist—it's time to make it fair. *Nature Publishing Group*, 2018.

[ZVRG17] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970, 2017.

[ZWCL10] Dell Zhang, Jun Wang, Deng Cai, and Jinsong Lu. Self-taught hashing for fast similarity search. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 18–25. ACM, 2010.

[ZWY+19] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. Gender bias in contextualized word embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 629–634, 2019.

[ZXW+20] Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. Incorporating bert into neural machine translation. *arXiv preprint arXiv:2002.06823*, 2020.

[ZYS+19] Ruiyi Zhang, Tong Yu, Yilin Shen, Hongxia Jin, and Changyou Chen. Text-based interactive recommendation via constraint-augmented reinforcement learning. In *Advances in neural information processing systems*, pages 15214–15224, 2019.

[ZZC+17] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

Biography

Weituo Hao is a recipient of Samsung Fellowship from 2020 to 2021.