Mitigating Negative Transfer for Better Generalization and Efficiency in Transfer Learning

Zirui Wang

CMU-LTI-21-019

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15123
www.lti.cs.cmu.edu

Thesis Committee:

Yulia Tsvetkov (Co-Chair)	Carnegie Mellon University
Emma Strubell (Co-Chair)	Carnegie Mellon University
Graham Neubig	Carnegie Mellon University
Orhan Firat	Google Research

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
In Language and Information Technologies

Keywords: transfer learning, meta learning, multilingual models, multimodal models, few-shot learning, zero-shot learning

to my grand parents and Jaime

Abstract

The traditional machine learning paradigm of training a task-specific model on one single task has led to state-of-the-art performance in many fields (e.g. computer vision and natural language processing). To enable wider applicability of machine learning models, transfer learning aims to adapt knowledge learned from source task(s) to improve performance in other target task(s). However, existing transfer learning paradigm is still understudied, such that we have limited knowledge of its potential limitations, underlying mechanism and solutions for more intelligent transfer. In particular, when transferring knowledge from a less related source, it may inversely hurt the target performance, a phenomenon known as negative transfer. Nonetheless, the cause of negative transfer is ill-defined, and it is not clear how negative transfer affect models' generalization and sample-efficiency.

In this thesis, with the goal of thoroughly characterizing and addressing negative transfer in machine learning models, we carefully study negative transfer in popular vision and NLP setups, glean insights on its causes, and propose solutions that lead to improved generalization and sample-efficiency. This thesis consists of three parts. The first part conducts systematic analysis of negative transfer in state-of-the-art transfer learning models. We formally characterize its conditions in both domain adaptation and multilingual NLP models, and demonstrate the task conflict as a key factor of negative transfer. In the second part, we propose various alignment methods to enhance the generalization of transferable models by resolving the aforementioned task conflicts with better-aligned representations and gradients. Finally, in the third part, we explore sample-efficient transfer learning algorithms that mitigate negative transfer using less training and/or alignment data. The contributions of this thesis include new insights on addressing negative transfer in transfer learning and a series of practical methods and algorithms that improve model generalization and efficiency.

Acknowledgments

My journey of working as a PhD student has been quite unusual in many aspects: the unexpected loss of my advisor, the COVID-19 pandemic and many more. At some point, I had been pessimistic towards the finish line. Meeting my advisor Yulia Tsvetkov was a crucial moment. When I felt lost and abandoned by the whole world during the lock down, Yulia recognized me and gave me the courage to continue my career. Even though I did not start as her student in the first year, she provided me with every support I needed whether it is an logistic issue, a technical problem, or emotional concern. I felt incredibly fortunate to have her as my advisor and I definitely cannot make it without her support. Yulia, I cannot express how grateful I am to you. You are the nicest person I met who gives so much caring to your students. Thank you!

I would also like to thank my thesis committee members Emma Strubell, Graham Neubig and Orhan Firat. It is my honor to work with each one of you during my study at CMU. Thank you for your time and guidance for my thesis. Your works have been so impactful for the language field, and I look forward to continue learning from you in future.

Thank you to my friends and colleagues during my years at CMU for making my life colorful: Shaojie Bai, Ming Yang, Liyunshu Qian, Allan Wang, Tongtong Jiang, Yunwen Cai, Zhao Pan, Ted Li, Yiheng Zhou, Xiyang Zhang, Xinru Yang, Jiateng Xie, Qizhe Xie, Guokun Lai, Hieu Pham, Cindy Wang, Sanket Mehta, Adams Yu, Yuan Cao, Jiahui Yu, Yonghui Wu. And special thanks to Zihang Dai, who gave me so many suggestions about research and life.

Sadly, I lost my dear advisor Jaime Carbonell and grand parents unexpectedly during the pandemic. They will be deeply missed. And finally, I would like to thank my family members: my parents, my cat Goose and my fiancé Xinran Yuan. They all gave me unconditional support and love throughout my life. I love you all.

Contents

Chapter 1

Introduction

Over the past decade, the development of deep neural networks (DNN) has largely pushed the performance of machine learning models on a wide range of tasks in both computer vision (CV) and natural language processing (NLP). With modern training systems using efficient accelerators such as graphics processing unit (GPU) and tensor processing unit (TPU), large-scale neural networks with billions of (Huang et al., 2019c; Radford et al., 2019) or even trillions of (Fedus et al., 2021) parameters obtain state-of-the-art performance on important tasks such as image classification (Pham et al., 2020) and machine translation (Lepikhin et al., 2020). However, DNNs often require a large amount of labeled data to train well-generalized models where the training data and testing data are assumed to be drawn from the same underlying distribution. In many cases, collecting large volumes of within-task labeled training data is expensive or even strictly prohibitive. Therefore, *transfer learning* (Pan and Yang, 2010; Weiss et al., 2016) has been developed to enable exploiting training data, supervised or unsupervised, from related task(s) to improve generalization in the target task(s). This paradigm of knowledge transfer has been shown success empirically on various kinds of tasks including core CV tasks such as image classification (Long et al., 2017; Tan and Le, 2019), object detection (He et al., 2017), and segmentation (Chen et al., 2017), and core NLP tasks such as natural language understanding (NLU) (Devlin et al., 2018; Liu et al., 2019; Yang et al., 2019) and cross-lingual transfer (Arivazhagan et al., 2019; Conneau et al., 2020; Lample and Conneau, 2019).

Despite the significant development and the exceptionally diverse range of applications, the success of transfer learning is not always guaranteed and its limitation is not well-understood. As a notable example, negative transfer has been observed empirically (Rosenstein et al., 2005) such that transferring from less related source tasks may hinder performance in the target tasks instead of improving. This poses several challenges towards more generalizable and efficient transfer learning as follows:

(1) The causes and underlying factors of negative transfer is unknown. While the notion of negative transfer has been well recognized within the transfer learning community (Pan and Yang, 2010; Weiss et al., 2016) and it has been observed empirically in different tasks ranging from simple binary classification problems (Rosenstein et al., 2005) to large-scale neural multilingual natural language understanding tasks (Arivazhagan et al., 2019), the root of such degenerative phenomena remains unclear. There is no clear formal definition for negative transfer nor systematic analysis of its conditions. For instance, we do not know whether negative transfer is model-dependent, or if it is determined by task relatedness only. Without such a thorough understanding of negative transfer, it is therefore particularly challenging to design methodologies to detect or prevent it, especially under settings where source-target divergence can be large.

(2) The generalization of transferable models is susceptible to negative transfer. The existence of negative transfer directly threatens the generalization of any transfer learning model. This is particularly true for NLP where pretrained models have become ubiquitous. For example, when a model is pretrained using corpus from a specific domain (such as wikipedia), it may not be applicable for other domains due to negative transfer (Raffel et al., 2019). On the other hand, a multilingual model can be trained on many languages at the same time, and negative interference among languages may also hinder its generalization (Arivazhagan et al., 2019). Thus, how to resolve negative transfer to extend applicabilities of existing NLP models has become an urgent issue.

(3) Negative transfer raises data requirements for efficient knowledge transfer. We human beings accumulate knowledge throughout our lifetime, and utilize most related past experience to efficiently learn new knowledge from only a few examples. However, existing transfer learning models are more likely to suffer from negative transfer when using few training examples (Pan and Yang, 2010), and therefore they require a large amount of training data, either from source or target distribution, to achieve reasonable performance. This would further increase the resources required to store extra training data and retrain transfer models to adapt to new domains, adding non-trivial cost and hinders transfer models' deployment in industry or other realistic setups where new tasks/domains continuously emerge.

1.1 Thesis Statement

The goal of this thesis is to thoroughly characterize and alleviate negative transfer in machine learning models. In particular, we aim to understand negative transfer's pre-conditions, investigate methods to mitigate it, and study the benefits of addressing it. To this end, we carefully study negative transfer in popular vision and NLP setups, glean insights on its causes, and propose solutions that lead to improved generalization and sample efficiency of transfer learning methods. In the first part, we conduct systematic analysis of negative transfer in state-of-the-art transfer learning models that (1) formally characterizes its conditions in both domain adaptation and multilingual NLP models and (2) demonstrates the task conflict as a key factor of negative transfer. In the second part, we present two methods to enhance the generalization of transferable models by resolving the aforementioned task conflicts with better-aligned representations and gradients. In the third part, we further show that mitigating data discrepancy among different tasks can also largely improve the sample efficiency of transfer learning methods, effectively reducing the requirement of human supervision.

1.2 Thesis Overview

This thesis is a collection of diverse case studies centered around the effect of negative transfer and the above three related challenges. The goal is to thoroughly investigate task conflicts in transfer learning models that could lead to negative transfer and resolve them to achieve more generalizable and sample-efficient knowledge transfer. There are three major parts, each corresponding to one problem mentioned above, detailed as follows:

Part I Measuring and Characterizing Conflicts in Transfer Learning In this part, we reexamine negative transfer for neural models in popular CV and NLP setups. Despite its popularity, there is no formal definition for negative transfer nor consistent metric to measure it thereby making it hard to compare transfer algorithms and detect negative transfer. We first revisit negative transfer in standard domain adaptation benchmarks and utilize its theoretical properties to define a negative transfer gap (NTG) as a practically measurable metric (chapter 3). The definition further reveals task conflict as the root of negative transfer and two other underlying factors, which are confirmed with empirical results. Next in (chapter 4), we extend our analysis beyond synthetic settings to real-world multilingual models where hundreds of languages are trained jointly in a single model. We specifically test four hypotheses of causes for negative transfer, and find gradient conflicts and task-specific parameters to be the main factors. In both cases, we report empirical evidence that resolving such conflicts may mitigate negative transfer.

Part II Enhancing Transfer Generalization by Addressing Task Conflicts From the previous part, we have learned that negative transfer exists in current transfer learning models and affects their generalization when transferred to new tasks/domains. The main cause of negative transfer is the inter-task conflict occurring during the training process or post-hoc adaptation. In this part, we are therefore particularly interested in addressing task conflicts through explicit alignment of source-target distributions. Specifically, we propose two different types of alignment for multilingual transfer models. In (chapter 5), we start with aligning representations of different languages after they are learned, both contextualized and non-contextualized. We compare two popular representation alignment paradigms and further propose a unified framework to combine them in a systematic way. Next we explore alignment during the optimization process (chapter 6) in a multilingual neural machine translation (NMT) model. We find that gradient similarity measured along the optimization trajectory is an important signal for the overall model performance. And thus we propose a scalable method to align gradients, named Gradient Vaccine (GRADVAC), which encourages more geometrically aligned parameter updates for close tasks. These proposed methods mitigate task conflicts and thus improve models' generalization in target tasks.

Part III Improving Transfer Efficiency with Less Supervision With recent advances of large-scale pre-trained models, efficiently adapting and reusing them for other tasks has become crucial for the practical application of these models. In this part, we show that alleviating data discrepancy and negative transfer can improve the sample efficiency of transferable models. We start with an empirical analysis of negative transfer in lifelong learning settings using pre-trained language models (chapter 7), and observe that the model requires much more examples to adapt robustly to new tasks when negative transfer occurs. Consequently, we propose a meta-learning method that explicitly learns to conduct local adaption robustly and effectively improve its sample efficiency such that it can outperform prior methods using 100 times less training data. Next in (chapter 8), we present a simple visual language model framework (SimVLM) for vision-language pretraining. Unlike prior methods, SimVLM only uses weakly aligned image-text data and enables zero-shot learning in visual understanding tasks, relieving the reliance on human annotation. This improves the sample efficiency for both pretraining and finetuning in vision-language transfer. Finally, we show that it is possible to attain strong transfer performance without utilizing any labeled data in (chapter 9). Specifically, we propose unsupervised data generation (UDG) using giant language models pretrained on a large scale, which enables zero-label language learning on a wide range of natural language understanding tasks and obtains the first super-human performance on the SuperGLUE benchmark (Wang et al., 2019a).

Chapter 2

Background and Literature Review

From a broad perspective, machine learning aims to train a generalizable model that outputs a label y for a given input x (the x, y pair can be in an arbitrary form, such as an image and a corresponding object class label) by utilizing a set of training data in the form of (x, y) pairs. The goal is being able to generalize to unseen test examples in a wide variety of settings. While the traditional machine learning training paradigm assumes both training and testing data come from the same underlying distribution, we often want to utilize training data from a related but different distribution when it is hard to directly collect training examples from the testing distribution. For example, if we do not have enough labeled images captured by professional photographers and we still want to train a model for image classification for this domain, we can utilize online pictures to improve the model quality. This is the key motivation for transfer learning and in this chapter, we formalize this setting for the rest of this thesis.

2.1 Terminology and Definitions

In machine learning, the typical setup is to train a model to approximate a function (defined through the form of model parameters) for a specific task. A task, in its simplest form, is composed of two measurable spaces $\mathcal{X}$ and $\mathcal{Y}$ and their joint underlying distribution $P(\mathcal{X}, \mathcal{Y})$. Here, $\mathcal{X}$ represents the input space and $\mathcal{Y}$ represents the label space. Then, a finite set of training examples $\{(x^i, y^i)\}_{i=1}^n$ are sampled/collected based on the joint distribution $P(\mathcal{X}, \mathcal{Y})$, and we train the model/hypothesis $h(\cdot)$ to minimize the expected risk as:

$$R_P(h) \coloneqq \mathbb{E}_{x,y\sim P}\left[\ell(h(x), y)\right], \tag{2.1}$$

where ℓ is some loss function defined for this task. In practice, since directly minimizing the expected risk is prohibited, we usually minimize the empirical risk instead as:

$$\tilde{R}_P(h) \coloneqq \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i). \tag{2.2}$$

Transfer Learning. The above training protocol does not work well when the training size n is small in a complex task. Transfer learning then aims to improve the model's generalization in this target task by utilizing training data from other task(s). Formally, let $P_S(\mathcal{X}, \mathcal{Y})$ and $P_T(\mathcal{X}, \mathcal{Y})$ denote the the joint distribution in the source and the target domain. Without loss of generality, given a labeled source set $\mathcal{S} = \{(x_s^i, y_s^i)\}_{i=1}^{n_s}$ sampled from the source joint $P_S(\mathcal{X}, \mathcal{Y})$ and a target set $\mathcal{T} = (\mathcal{T}_l, \mathcal{T}_u)$ where $\mathcal{T}_l = \{(x_l^j, y_l^j)\}_{j=1}^{n_l}$ is a labeled target set drawn from the target joint $P_T(\mathcal{X}, \mathcal{Y})$, and $\mathcal{T}_u = \{x_u^k\}_{k=1}^{n_u}$ is an unlabeled target set from the target marginal $P_T(\mathcal{X})$, transfer learning aims at designing an algorithm A, which takes both the source and target domain data $\mathcal{S}, \mathcal{T}$ as input, and outputs a hypothesis (model) $h = A(\mathcal{S}, \mathcal{T})$, to minimize the expected risk as

$$R_{P_T}(h) \coloneqq \mathbb{E}_{x,y \sim P_T} \left[\ell(h(x), y)\right]. \tag{2.3}$$

To make the setting meaningful, it is often assumed that $n_s \gg n_l$. This definition of transfer learning can be easily adapted to different settings to include multiple sources/targets such as domain adaptation and multilingual models. It can also be extended to related settings such as few-shot learning and lifelong learning.

Negative Transfer. Although the goal of transfer learning is to improve target task performance through knowledge transfer from source tasks, the success of such paradigm is not always guaranteed. In particular, since we typically have limited amount of labeled data in the target task, it becomes hard to accurately approximate the expected risk in Eq. (2.3). Consequently, it is challenging to guarantee performance when transferring to the target, hurting the model's **generalization**. On the other hand, this also raises the issue of **sample-efficiency**, as it often requires more training samples to mitigate negative transfer. As we will see later in this thesis, a key factor of these two problems is **negative transfer**, which is closely related to task conflict and transfer efficiency. The notion of negative transfer lacks a rigorous definition. A widely accepted description of negative transfer (Pan and Yang, 2010; Weiss et al., 2016) is stated as "*transferring knowledge from the source can have a negative impact on target learner*". More broadly, this can be interpreted as utilizing training data from a different distribution is less effective, compared to the standard machine learning training protocol such as the one in Eq. (2.2).

We will redefine this term formally in chapter 3 and provide in-depth analysis. In addition, we will also study negative transfer in transfer paradigm with multiple sources/targets, which we refer to as **negative interference**.

2.2 Background

2.2.1 Transfer Learning

Transfer learning Pan and Yang (2010); Yang et al. (2013) aims to transfer knowledge learned in the source domain to assist learning in the target domain. There are many forms of knowledge transfer, such as parameter transfer (Ganin et al., 2016) or regularization transfer (Huang et al., 2007). Pioneer work of transfer learning study the simple setting of transferring from a single source domain to a single target domain, where the input/output spaces are also assumed to be the same (a.k.a. homogeneous domain adaptation). Early methods (Caruana, 1997) exploit conventional statistical techniques such as weighting source domain instances (Huang et al., 2007) and mapping domain features to be similar between the source and the target (Pan et al., 2011; Uguroglu and Carbonell, 2011). Compared to these earlier approaches, deep transfer networks achieve better results in discovering domain invariant factors (Yosinski et al., 2014). Some deep learning methods (Long et al., 2015; Sun and Saenko, 2016) transfer via distribution (mis)match measurements such as Maximum Mean Discrepancy (MMD) (Huang et al., 2007). More recent work(Cao et al., 2018a; Ganin et al., 2016; Sankaranarayanan et al., 2018; Tzeng et al., 2015) exploit generative adversarial networks (GANs) (Goodfellow et al., 2014) and add a subnetwork as a domain discriminator, obtaining strong results on computer vision tasks (Sankaranarayanan et al., 2018). More recently for natural language processing tasks, language model pre-training has become a successful transfer learning approach to effectively reduce the requirement for task-specific labeled data (Brown et al., 2020; Devlin et al., 2018; Liu et al., 2019; Radford et al., 2019; Raffel et al., 2019; Yang et al., 2019). Via training on unsupervised large-scale text corpus, bi-directional language models such as BERT and XLNet are able to learn contextualized text representations that can then be fine-tuned on downstream tasks with small training data sizes, which have pushed the state of the art on a variety of natural language understanding benchmarks. These results raise increasing interest in transfer learning, which has been extended to challenging problem settings such as multilingual NLP, meta learning and lifelong learning.

Multilingual NLP attempts to enable knowledge transfer among languages and build multilingual systems that can process many languages at the same time. Early methods try to capture the cross-lingual mapping of word embeddings with cross-lingual supervision, including bilingual dictionaries (Artetxe et al., 2016; Duong et al., 2016; Faruqui and Dyer, 2014; Gouws and Søgaard, 2015; Joulin et al., 2018; Mikolov et al., 2013a; Xing et al., 2015), sentence-aligned corpora (Gouws et al., 2015; Hermann and Blunsom, 2014; Kočiskỳ et al., 2014) and document-aligned corpora (Søgaard et al., 2015; Vulić and Moens, 2016). More recently, multilingual models train multiple languages jointly (Aharoni et al., 2019; Arivazhagan et al., 2019; Conneau et al., 2020; Devlin et al., 2018; Firat et al., 2016; Johnson et al., 2017; Lample and Conneau, 2019). Follow-up work study the cross-lingual ability of these models and what contributes to it (Artetxe et al., 2019b; Karthikeyan et al., 2020; Kudugunta et al., 2019; Pires et al., 2019; Wu and Dredze, 2019; Wu et al., 2020), the limitation of such training paradigm (Arivazhagan et al., 2019; Wang et al., 2020c), and how to further improve it by utilizing post-hoc alignment (Cao et al., 2020; Wang et al., 2020d), data balancing (Jean et al., 2019; Wang et al., 2020b), or calibrated training signal (Huang et al., 2019a; Mulcaire et al., 2019).

Meta learning studies the problem of transferring knowledge from known tasks to facilitate learning in the unseen tasks, a.k.a. learning to learn. In (Finn et al., 2017), it is defined as *to train a model on a variety of learning tasks, such that it can solve new learning tasks using only a small number of training samples*. A popular setting is few-shot learning that aims to learn how to perform fast adaptation on new tasks by utilizing past experience (Finn et al., 2017; Flennerhag et al., 2019; Gu et al., 2018). More broadly, meta-level learning can be also applied to learn how to perform data selection on unseen tasks (Wang et al., 2020b) or choose hyperparameters (Baydin et al., 2018). This form of transferring combines well with existing transfer learning approach (Pham et al., 2020).

Lifelong learning transfers knowledge continuously in a model's lifetime. The model is required to quickly adapt to new environments and acquire new skills by leveraging past experiences, while retaining old skills and continuously accumulating knowledge (Parisi et al., 2019). There is a surge of research interest in the lifelong learning by applying regularization or expanding model sizes (Chaudhry et al., 2019; Kirkpatrick et al., 2017; Lopez-Paz and Ranzato, 2017; Rusu et al., 2016; Sprechmann et al., 2018; Yoon et al., 2018; Zenke et al., 2017). One successful approach to achieving lifelong learning has been augmenting the learning model with an episodic memory module (Sprechmann et al., 2018). The underlying idea is to first store previously seen training examples in memory, and later use them to perform experience replay (Rolnick et al., 2019) or to derive optimization constraints (Chaudhry et al., 2019; Lopez-Paz and Ranzato, 2017) while training on new tasks. The key is to prevent catastrophic forgetting when trained on new tasks (Yogatama et al., 2019), such that the model may forget learned knowledge on old ones.

2.2.2 Negative Transfer

Negative transfer occurs when transferring knowledge from source tasks hurts the performance in the target tasks instead of assisting. It is hypothesized that the cause of this negative impact is the difference between the two underlying distributions. Early work that noted negative transfer (Rosenstein et al., 2005) was targeted at simple classifiers such as hierarchical Naive Bayes. They observe degenerative performance in the target domain when transfer learning is applied. Later, similar negative effects have also been observed in various settings including multi-source transfer learning (Duan et al., 2012), imbalanced distributions (Ge et al., 2014) and partial transfer learning (Cao et al., 2018a). While the importance of detecting and avoiding negative transfer has raised increasing attention (Weiss et al., 2016), the literature lacks in-depth analysis. Besides, similar effects of negative transfer are observed in other settings such as multilingual models and lifelong learning.

Multilingual models are multi-task in nature and thus interference among tasks may exist. This is particularly true since language input spaces are heterogeneous, with different vocabularies, morphosyntactic rules, and different pragmatics across cultures. Prior work have discovered that multilingual models are not equally beneficial for all languages. Conneau et al. (2020) demonstrated that including more languages in a single model can improve performance for low-resource languages but hurt performance for high-resource languages. Similarly, recent work (Aharoni et al., 2019; Arivazhagan et al., 2019; Johnson et al., 2017; Tan et al., 2019) in multilingual neural machine translation (NMT) also observed performance degradation on high-resource language pairs. This phenomenon is known as *negative interference* (Ruder, 2017), where training multiple tasks jointly hinders the performance on individual tasks.

In lifelong learning, similar negative impact among tasks has also been reported as *catastrophic forgetting* (McCloskey and Cohen, 1989). This is the case when learning new tasks hurts the performance in the old ones. On the other hand, negative transfer also occurs when transferring from the old tasks hurts performance in the new ones. These can be seen as negative transfer in two directions, backward and forward respectively. However, human beings suffer less from these due to the complementary learning systems (CLS) theory (McClelland et al., 1995). It states that humans rely on episodic memory to store past experiences and conduct experience rehearsal, and have two learning phases that iteratively learn new knowledge efficiently but remember old ones robustly for a longer period of time. Therefore, it is expected that by addressing negative transfer, we can also train machine learning models that are able to conduct transfer learning efficiently and robustly.

Part I

Measuring and Characterizing Conflicts in Transfer Learning

Chapter 3

Negative Transfer in Domain Adaptation

In this chapter and the next chapter, we present the first part of the thesis to study negative transfer in transfer learning. We aim to systematically investigate its definition and underlying factors, which will serve as important building blocks for later chapters. This chapter first study negative transfer in domain adaptation, a popular setup for traditional transfer learning. Despite its pervasiveness, negative transfer is usually described in an informal manner, lacking rigorous definition, careful analysis, or systematic treatment. In this chapter, we propose a formal definition of negative transfer and analyze three important aspects thereof. Stemming from this analysis, a novel technique is proposed to circumvent negative transfer by filtering out unrelated source data. Based on adversarial networks, the technique is highly generic and can be applied to a wide range of transfer learning algorithms. The proposed approach is evaluated on six state-of-the-art deep transfer methods via experiments on four benchmark datasets with varying levels of difficulty. Empirically, the proposed method consistently improves the performance of all baseline methods and largely avoids negative transfer, even when the source data is degenerate.

3.1 Introduction

The development of deep neural networks (DNNs) has improved the state-of-the-art performance on a wide range of machine learning problems and applications. However, DNNs often require a large amount of labeled data to train well-generalized models and as more classical methods, DNNs rely on the assumption that training data and test data are drawn from the same underlying distribution. In some cases, collecting large volumes of labeled training data is expensive or even prohibitive. Transfer learning (Pan and Yang, 2010) addresses this challenge of data scarcity by utilizing previously-labeled data from one or more source tasks. The hope is that this source domain is related to the target domain and thus transferring knowledge from the source can improve the performance within the target domain. This powerful paradigm has been studied under various settings (Weiss et al., 2016) and has been proven effective in a wide range of applications (Long et al., 2015; Moon and Carbonell, 2017; Zamir et al., 2018).

However, the success of transfer learning is not always guaranteed. If the source and target domains are not sufficiently similar, transferring from such weakly related source may hinder the performance in the target, a phenomenon known as negative transfer. The notion of negative transfer has been well recognized within the transfer learning community (Pan and Yang, 2010; Weiss et al., 2016). An early paper (Rosenstein et al., 2005) has conducted empirical study on a simple binary classification problem to demonstrate the existence of negative transfer. Some more recent work (Cao et al., 2018a; Duan et al., 2012; Ge et al., 2014) has also observed similar negative impact while performing transfer learning on more complex tasks under different settings.

Despite these empirical observations, little research work has been published to analyze or predict negative transfer, and the following questions still remain open: First, while the notion being quite intuitive, it is not clear how negative transfer should be defined exactly. For example, how should we measure it at test time? What type of baseline should we compare with? Second, it is also unknown what factors cause negative transfer, and how to exploit them to determine that negative transfer may occur. Although the divergence between the source and target domain is certainly crucial, we do not know how large it must be for negative transfer to occur, nor if it is the only factor. Third and most importantly, given limited or no labeled target data, how to detect and/or avoid negative transfer.

In this chapter, we take a step towards addressing these questions. We first derive a formal definition of negative transfer that is general and tractable in practice. Here tractable means we can explicitly measure its effect given the testing data. This definition further reveals three underlying factors of negative transfer that give us insights on when it could occur. Motivated by these theoretical observations, we develop a novel and highly generic technique based on adversarial networks to combat negative transfer. In our approach, a discriminator estimating both marginal and joint distributions is used as a gate to filter potentially harmful source data by reducing the bias between source and target risks, which corresponds to the idea of importance reweighting (Cortes et al., 2010; Yu and Szepesvári, 2012). Our experiments involving eight transfer learning methods and four benchmark datasets reveal the three factors of negative transfer. In addition, we apply our method to six state-of-the-art deep methods and compare their performance, demonstrating that our approach substantially improves the performance of all base methods under potential negative transfer conditions by largely avoiding negative transfer.

3.2 Rethink Negative Transfer

Notation. We will use $P_S(X, Y)$ and $P_T(X, Y)$, respectively, to denote the the joint distribution in the source and the target domain, where X is the input random variable and Y the output. Following the convention, we assume having access to labeled source set $\mathcal{S} = \{(x_s^i, y_s^i)\}_{i=1}^{n_s}$ sampled from the source joint $P_S(X, Y)$, a labeled target set $\mathcal{T}_l = \{(x_l^j, y_l^j)\}_{j=1}^{n_l}$ drawn from the target joint $P_T(X, Y)$, and an unlabeled target set $\mathcal{T}_u = \{x_u^k\}_{k=1}^{n_u}$ from the target marginal $P_T(X)$. For convenience, we define $\mathcal{T} = (\mathcal{T}_l, \mathcal{T}_u)$. Notice that we focus on domain adaptation setup in this chapter, and assume the label spaces are the same between the source and the target domain. This allows us to study negative transfer in details and we leave more complex transfer learning settings for future work.

Transfer Learning. Under the notation, transfer learning aims at designing an algorithm A, which takes both the source and target domain data $\mathcal{S}, \mathcal{T}$ as input, and outputs a better hypothesis (model) $h = A(\mathcal{S}, \mathcal{T})$, compared to only using the target-domain data. For model comparison, we will adapt the standard expected risk, which is defined as

$$R_{P_T}(h) \coloneqq \mathbb{E}_{x,y \sim P_T} \left[\ell(h(x), y) \right], \tag{3.1}$$

with ℓ being the specific task loss. To make the setting meaningful, it is often assumed that $n_s \gg n_l$.

Negative Transfer. The notion of negative transfer lacks a rigorous definition. A widely accepted description of negative transfer (Pan and Yang, 2010; Weiss et al., 2016) is stated as "*transferring knowledge from the source can have a negative impact on the target learner*". While intuitive, this description conceals many critical factors underlying negative transfer, among which we stress the following three points:

1. *Negative transfer should be defined w.r.t. the transfer learning algorithm.* Specifically, the informal description above does not specify what the negative impact is compared with. For example, it will be misleading to only compare with the best possible algorithm only using the target data, i.e., defining negative transfer as

$$R_{P_T}(A(\mathcal{S}, \mathcal{T})) > \min_{A'} R_{P_T}(A'(\emptyset, \mathcal{T})), \tag{3.2}$$

because the increase in risk may not come from using the source-domain data, but the difference in algorithms. Therefore, to study negative transfer, one should focus on a specific algorithm at a time and compare its performance with and without the source-domain data. Hence, we define the *negative transfer condition* (NTC) for any algorithm A as

$$R_{P_T}(A(\mathcal{S}, \mathcal{T})) > R_{P_T}(A(\varnothing, \mathcal{T})). \tag{3.3}$$

For convenience, we also define the *negative transfer gap* (NTG) as a quantifiable measure of negative transfer:

$$\text{NTG}(A, S, T) = R_{P_T}(A(\mathcal{S}, \mathcal{T})) - R_{P_T}(A(\varnothing, \mathcal{T})), \tag{3.4}$$

and we say that negative transfer occurs if the negative transfer gap is positive and vice versa.

2. *Divergence between the joint distributions is the root to negative transfer*. As negative transfer is algorithm specific, it is natural to ask the question that whether there exists a transfer learning algorithm that can always improve the expected risk compared to its target-domain only baseline. It turns out this depends on the divergence between $P_S(X, Y)$ and $P_T(X, Y)$ (Gong et al., 2016). As an extreme example, assume $P_S(X) = P_T(X)$ and $P_S(Y \mid x)$ is uniform for any x. In the case, there is no meaningful knowledge in $P_S(X, Y)$ at all. Hence, exploiting $\mathcal{S} \sim P_S(X, Y)$ will almost surely harm the estimation of $P_T(Y \mid X)$, unless $P_T(Y \mid X)$ is uniform.

   In practice, we usually deal with the case where there exists some "systematic similarity" between $P_S(X, Y)$ and $P_T(X, Y)$. Then, an ideal transfer learning algorithm would figure out and take advantage of the similar part, leading to improved performance. However, if an algorithm fails to discard the divergent part and instead relies on it, one can expect negative transfer to happen. Thus, regardless of the algorithm choice, the distribution shift is the actual root to negative transfer.

3. *Negative transfer largely depends on the size of the labeled target data*. While the previous discussion focuses on the distribution level, an overlooked factor of negative transfer is the size of the labeled target data, which has a mixed effect.

   On one hand, for the same algorithm and distribution divergence, NTC depends on how well the algorithm can do using target data alone, i.e. the RHS of Eq.(3.3). In zero-shot transfer learning[1] (Ganin and Lempitsky, 2015; Pei et al., 2018) where there is no labeled target data ($n_l = 0$), only using unlabeled target data would result in a weak random model and thus

[1] It is often referred to as unsupervised domain adaptation in the literature.

NTC is unlikely to be satisfied. When labeled target data is available (Moon and Carbonell, 2017; Rosenstein et al., 2005; Tzeng et al., 2015), a better target-only baseline can be obtained using semi-supervised learning methods and so negative transfer is *relatively* more likely to occur. At the other end of the spectrum, if there is an abundance of labeled target data, then transferring from an even slightly different source domain could hurt the generalization. Thus, this shows that negative transfer is *relative*.

On the other hand, the amount of labeled target data has a direct effect on the feasibility and reliability of discovering shared regularity between the joint distributions. As discussed above, the key component of a transfer learning algorithm is to discover the similarity between the source joint $P_S(X,Y)$ and the target joint $P_T(X,Y)$. When labeled target data is not available ($n_l = 0$), one has to resort to the similarity between the marginals $P_S(X)$ and $P_T(X)$, which though has a theoretical limitation on generalization bound (Ben-David et al., 2007). In contrast, if one has a considerable number of samples $(x_l, y_l) \sim P_T(X,Y)$ and $(x_s, y_s) \sim P_S(X,Y)$, the problem would be manageable. Therefore, an ideal transfer learning algorithm may be able to utilize labeled target data to mitigate the negative impact of unrelated source information.

While the discussion above are based on the underlying setting of domain adaptation, we believe some ideas are also shared across the other settings such as covariate shift and concept shift. For example, the analysis on the divergence of joint distributions is generic and is applicable for those settings as well. Other ideas are also worth exploring in future research. With these points in mind, we next turn to the problem of how to avoid negative transfer in a systematic way.

3.3 Proposed Method

As discussed in Section 3.2, the key to improving transfer is to discover and exploit shared underlying structures between $P_S(X,Y)$ and $P_T(X,Y)$. In practice, there are many possible regularities one may take advantage of. To motivate our proposed method, we first review an important line of work and show how the observation in Section 3.2 helps us to identify the limitation.

3.3.1 Domain Adversarial Network

As a notable example, a recent line of work (Ganin and Lempitsky, 2015; Long et al., 2015; Tzeng et al., 2017) has successfully utilized a domain-invariant feature space assumption to achieve knowledge transfer. Specifically, it is assumed that there exists a feature space that is both shared by both source and target domains and discriminative enough for predicting the output. By learning a feature extractor F that can map both the source and target input to the same feature space, classifier learned on the source data can transfer to the target domain.

To find such a feature extractor, a representative solution is the Domain Adversarial Neural Network (DANN) (Ganin et al., 2016), which exploits a generative adversarial network (GAN) framework to train the feature extractor F such that the feature distributions $P(F(X_S))$ and $P(F(X_T))$ cannot be distinguished by the discriminator D. Based on the shared feature space, a simple classifier C is trained on both source and target data. Formally, the objective can be written as:

$$\operatorname*{argmin}_{F,C} \operatorname*{argmax}_{D} \mathcal{L}_{\text{CLF}}(F, C) - \mu \mathcal{L}_{\text{ADV}}(F, D), \tag{3.5}$$

$$\begin{aligned} \mathcal{L}_{\text{CLF}}(F, C) =& \mathbb{E}_{x_l, y_l \sim \mathcal{T}_L} \left[\ell_{\text{CLF}}(C(F(x_l)), y_l)\right] \\ &+ \mathbb{E}_{x_s, y_s \sim \mathcal{S}} \left[\ell_{\text{CLF}}(C(F(x_s)), y_s)\right], \end{aligned} \tag{3.6}$$

$$\begin{aligned} \mathcal{L}_{\text{ADV}}(F, D) =& \mathbb{E}_{x_u \sim P_T(X)} \left[\log D(F(x_u))\right] \\ &+ \mathbb{E}_{x_s \sim P_S(X)} \left[\log(1 - D(F(x_s)))\right]. \end{aligned} \tag{3.7}$$

Intuitively, $\mathcal{L}_{\text{CLF}}$ is the supervised classification loss on both the target and source labeled data, $\mathcal{L}_{\text{ADV}}$ is the standard GAN loss treating $F(x_u)$ and $F(x_s)$ as the true and fake features respectively, and μ is a hyper-parameter balancing the two terms. For more details and theoretical analysis, we refer readers to the original work (Ganin and Lempitsky, 2015).

Now, notice that the DANN objective implicitly makes the following assumption: For any $x_s \in \mathcal{X}_s$, there exists a $x_t \in \mathcal{X}_t$ such that

$$P_S(Y|x_s) = P_T(Y|x_t) = P(Y|F(x_s)) = P(Y|F(x_t)).$$

In other words, it is assumed that every single source sample can provide meaningful knowledge for transfer learning. However, as we have discussed in Section 3.2, some source samples may not be able to provide any knowledge at all. Consider the case where there is a source input $x_s \in \mathcal{X}_s$ such that $P_S(Y \mid x_s) \neq P_T(Y \mid x_t)$ for any x_t. Since $P(F(X_s)) = P(F(X_t))$ as a result of the GAN objective, there exists a $x' \in \mathcal{X}_t$ such that $F(x') = F(x_s)$ and hence $P(Y \mid F(x')) = P(Y \mid F(x_s))$. Then, if $P(Y \mid F(x_s))$ is trained on the source data to match $P_S(Y \mid x_s)$, it follows

$$P(Y|F(x')) = P(Y|F(x_s)) = P(Y|x_s) \neq P(Y|x').$$

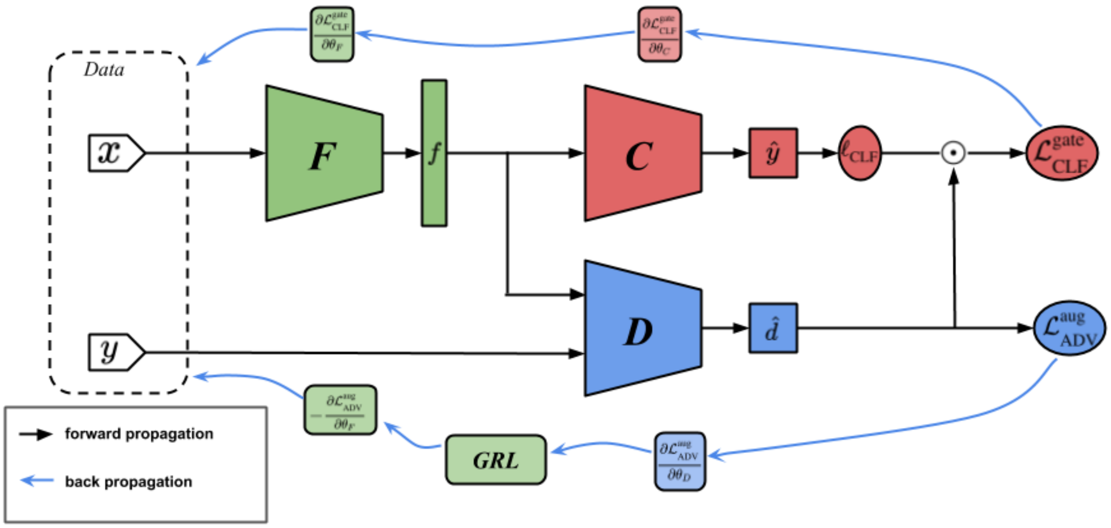

$\frac{\partial \mathcal{L}_{\text{CLF}}^{\text{gate}}}{\partial \theta_F}$
$\frac{\partial \mathcal{L}_{\text{CLF}}^{\text{gate}}}{\partial \theta_C}$
Data
x
y
F
f
C
$\hat{y}$
ℓ_{CLF}
$\odot$
$\mathcal{L}_{\text{CLF}}^{\text{gate}}$
D
$\hat{d}$
$\mathcal{L}_{\text{ADV}}^{\text{aug}}$
$-\frac{\partial \mathcal{L}_{\text{ADV}}^{\text{aug}}}{\partial \theta_F}$
GRL
$\frac{\partial \mathcal{L}_{\text{ADV}}^{\text{aug}}}{\partial \theta_D}$
forward propagation
back propagation

Figure 3.1: The architecture of proposed discriminator gate, where f is the extracted feature layer, $\hat{y}$ and ℓ_{CLF} are predicted class label and its loss, $\hat{d}$ is the predicted domain label, $\mathcal{L}_{\text{CLF}}^{\text{gate}}$ is the classification loss, $\mathcal{L}_{\text{ADV}}^{\text{aug}}$ is the adversarial learning loss; GRL stands for Gradient Reversal Layer and $\odot$ is the Hadamard product.

As a result, relying on such "unrelated" source samples can hurt the performance, leading to negative transfer. Motivated by this limitation, we next present a simple yet effective method to deal with harmful source samples in a systematic way.

3.3.2 Discriminator Gate

The limitation of DANN comes from the unnecessary assumption that all source samples are equally useful. To eliminate the weakness, a natural idea is to reweight each source sample in some proper manner. To derive an appropriate weight, notice that the standard supervised learning objective can be rewritten as

$$\begin{aligned}\mathcal{L}_{\text{SUP}} &= \mathbb{E}_{x,y\sim P_T(X,Y)}\left[\ell_{\text{CLF}}(C(F(x)),y)\right] \\ &= \mathbb{E}_{x,y\sim P_S(X,Y)}\left[\frac{P_T(x,y)}{P_S(x,y)}\ell_{\text{CLF}}(C(F(x)),y)\right]\end{aligned} \tag{3.8}$$

where the density ratio $\frac{P_T(x,y)}{P_S(x,y)}$ naturally acts as an importance weight (Cortes et al., 2010; Yu and Szepesvári, 2012) for the source data. Hence, the problem reduces to the classic problem of density ratio estimation.

Here, we exploit a GAN discriminator to perform the density ratio estimation (Uehara et al., 2016). Specifically, the discriminator takes both x and the paired y as input, and tries to classify whether the pair is from the source domain (fake) or the target domain (true). At any point, the optimal discriminator is given by $D(x,y) = \frac{P_T(x,y)}{P_T(x,y)+P_S(x,y)}$, which implies

$$\frac{P_T(x,y)}{P_S(x,y)} = \frac{D(x,y)}{1-D(x,y)}.$$

In our implementation, to save model parameters, we reuse the feature extractor to obtain the features of x and instantiate $D(x,y)$ as $D(F(x),y)$. With the weight ratio, we modify the classification objective (3.6) in DANN as

$$\begin{aligned}\mathcal{L}_{\text{CLF}}^{\text{gate}}(C,F) &= \mathbb{E}_{x_l,y_l\sim\mathcal{T}_L}\left[\ell_{\text{CLF}}(C(F(x_l)),y_l)\right] \\ &+\lambda\mathbb{E}_{x_s,y_s\sim\mathcal{S}}\left[\omega(x_s,y_s)\ell_{\text{CLF}}(C(F(x_s)),y_s)\right], \\ \omega(x_s,y_s) &= \text{SG}\left(\frac{D(x_s,y_s)}{1-D(x_s,y_s)}\right)\end{aligned} \tag{3.9}$$

where $\text{SG}(\cdot)$ denotes stop gradient and λ is another hyper-parameter introduce to scale the density ratio. As the density ratio acts like a gating function, we will refer to mechanism as discriminator gate.

On the other hand, we also augment the adversarial learning objective (3.7) by incorporating terms for matching the joint distributions:

$$\begin{aligned}\mathcal{L}_{\text{ADV}}^{\text{aug}}(F,D) = &\mathbb{E}_{x_u\sim P_T(X)}\left[\log D(F(x_u),\texttt{nil})\right] \\ &+\mathbb{E}_{x_s\sim P_S(X)}\left[\log(1-D(F(x_s),\texttt{nil}))\right] \\ &+\mathbb{E}_{x_l,y_l\sim\mathcal{T}_L}\left[\log D(F(x_l),y_l)\right] \\ &+\mathbb{E}_{x_s,y_s\sim\mathcal{S}}\left[\log(1-D(F(x_s),y_s))\right],\end{aligned} \tag{3.10}$$

where `nil` denotes a dummy label which does not provide any label information and it is included to enable the discriminator D being used as both a marginal discriminator and a joint discriminator. As a benefit, the joint discriminator can utilize unlabeled target data since labeled data could be scarce. Similarly, under this objective, the feature network F will receive gradient from both the marginal discriminator and the joint discriminator. Theoretically speaking, the joint matching objective subsumes the the marginal matching objective, as matched joint distribution implied matched marginals. However, in practice, the labeled target data $\mathcal{T}_L$ is usually limited, making the joint matching objective itself insufficient. This particular design choice echos our discussion about how the size of labeled target data can influence our algorithm design in Section 3.2.

Combining the gated classification objective (3.9) and the augmented adversarial learning objective (3.10), we arrive at our proposed approach to transfer learning

$$\underset{F,C}{\operatorname{argmin}}\,\underset{D}{\operatorname{argmax}}\,\mathcal{L}_{\text{CLF}}^{\text{gate}}(F,C) - \mu\mathcal{L}_{\text{ADV}}^{\text{aug}}(F,D). \tag{3.11}$$

The overall architecture is illustrated in Figure 3.1. Finally, although the presentation of the proposed method is based on DANN, our method is highly general and can be applied directly to other adversarial transfer learning methods. In fact, we can even extend non-adversarial methods to achieve similar goals. In our experiments, we adapt six deep methods (Cao et al., 2018b; Ganin and Lempitsky, 2015; Long et al., 2015; Sankaranarayanan et al., 2018; Sun and Saenko, 2016; Tzeng et al., 2017) of three different categories to demonstrate the effectiveness of our method.

3.4 Experiments

We conduct extensive experiments on four benchmark datasets to (1) analyze negative transfer and its three underlying aspects, and (2) evaluate our proposed discriminator gate on six state-of-the-art methods.

3.4.1 Datasets

We use four standard datasets with different levels of difficulties: (1) small domain shift: Digits dataset, (2) moderate domain shift: Office-31 dataset, and (3) large domain shift: Office-Home and VisDA datasets.

Digits contains three standard digit classification datasets: MNIST, USPS, SVHN. Each dataset contains large amount of images belonging to 10 classes (0-9). This dataset is relatively easy due to its simple data distribution and therefore we only consider a harder case: **SVHN→MNIST**. Specifically, SVHN (Netzer et al., 2011) contains 73K images cropped from house numbers in Google Street View images while MNIST (LeCun et al., 1998) consists of 70K handwritten digits captured under constrained conditions.

Office-31 (Saenko et al., 2010) is the most widely used dataset for visual transfer learning. It contains 4,652 images of 31 categories from three domains: Amazon(**A**) which contains images from amazon.com, Webcam(**W**) and DSLR(**D**) which consist of images taken by web camera and SLR camera. We evaluate all methods across three tasks: **W**→**D**, **A**→**D**, and **D**→**A**. We select these three settings because the other three possible cases yield similar results.

Office-Home (Venkateswara et al., 2017) is a more challenging dataset that consists of about 15,500 images of 65 categories that crawled through several search engines and online image directories. In particular, it contains four domains: Artistic images(**Ar**), Clip Art(**Cl**), Product images(**Pr**) and Real-World images(**Rw**). We want to test on more interesting and practical transfer learning tasks involving adaptation from synthetic to real-world and thus we consider three transfer tasks: **Ar**→**Rw**, **Cl**→**Rw**, and **Pr**→**Rw**. In addition, we choose to use the first 25 categories in alphabetic order to make our results more comparable to previous studies (Cao et al., 2018b).

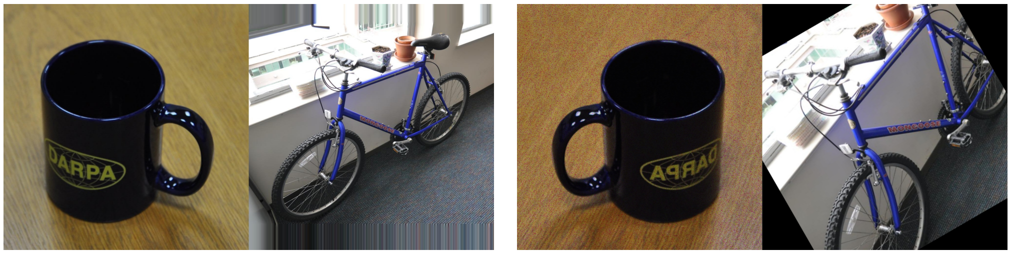
VisDA (Peng et al., 2017) is another challenging synthetic to real dataset. We use the training set as the synthetic source and the testing set as the real-world target (**Synthetic**→**Real**). Specifically, the training set contains 152K synthetic images generated by rendering 3D models and the testing set contains 72K real images from crops of Youtube Bounding Box dataset (Real et al., 2017), both contain 12 categories.

3.4.2 Experimental Setup

To better study negative transfer effect and evaluate our approach, we need to control the three factors discussed in Section 3.2, namely *algorithm factor*, *divergence factor* and *target factor*. In our experiments, we adopt the following mechanism to control each of them.

Divergence factor: Since existing benchmark datasets usually contain domains that are similar to each other, we need to alter their distributions to better observe negative transfer effect. In our experiments, we introduce two perturbation rates ϵ_x and ϵ_y to respectively control the marginal divergence and the conditional divergence between two domains. Specifically, for each source domain data we independently draw a Bernoulli variable of probability ϵ_x, and if it returns one, we add a series of random noises to the input image such as random rotation, random salt&pepper noise, random flipping, etc (examples shown Figure 3.2). According to studies in (Azulay and Weiss, 2018; Szegedy et al., 2014), such perturbation is enough to cause misclassification for neural networks and therefore is sufficient for our purpose. In addition, we draw a second independent Bernoulli variable of probability ϵ_y and assign a randomly picked label if it returns one.

Target factor: Similar to previous works, we use all labeled source data for training. For the target data, we first split 50% as training set and the rest 50% for testing. In addition, we use all



(a) Original (b) Perturbed

Figure 3.2: Example images before & after perturbation

of target training data as unlabeled target data and use $L_{\%}$ percent of them as labeled target data. A systematic study of source data can be found in (Wang and Carbonell, 2018).

Algorithm factor: To provide a more comprehensive study of negative transfer, we evaluate the performance of eight transfer learning methods of five categories: **TCA** (Pan et al., 2011), **KMM** (Huang et al., 2007), **DAN** (Long et al., 2015), **DCORAL** (Sun and Saenko, 2016), **DANN** a.k.a RevGrad (Ganin and Lempitsky, 2015), **ADDA** (Tzeng et al., 2015), **PADA** (Cao et al., 2018b), **GTA** (Sankaranarayanan et al., 2018). Specifically, (1) TCA is a conventional method based on MMD-regularized PCA, (2) KMM is a conventional sample reweighting method, (3) DAN and DCORAL are non-adversarial deep methods which use a distribution measurement as an extra loss, (4) DANN, ADDA and PADA use adversarial learning and directly train a discriminator, (5) GTA is a GAN based method that includes a generator to generate actual images in additional to the discriminator. We mainly follow the default settings and training procedures for model selection as explained in their respective papers. However, for fair comparison, we use the same feature extractor and classifier architecture for all deep methods. In particular, we use a modified LeNet as detailed in (Sankaranarayanan et al., 2018) for the Digits dataset. For other datasets, we fine-tune from the ResNet-50 (He et al., 2016) pretrained on ImageNet with an added 256-dimension bottleneck layer between the *res5c* and *fc* layers. To compare the performance of our proposed approach, we adapt a gated version for each of the six deep methods (e.g **DANN$_{\textbf{gate}}$** is the gated DANN). Specifically, we extend DANN, ADDA and PADA straightforwardly as described in Section 4.2. For GTA, we extend the discriminator to take in class labels and output domain label predictions as gates. For DAN and DCORAL, we add an extra discriminator network to be used as gates but the general network is not trained adversarially. For hyper-parameters, we set $\lambda = 1$ and μ progressively increased from 0 to 1 in all our experiments. For each transfer task, we compare the average classification accuracy over

	W→D					A→D					
	ϵ =0.0	ϵ =0.3	ϵ =0.7	ϵ =0.9	Avg	ϵ =0.0	ϵ =0.3	ϵ =0.7	ϵ =0.9	Avg	$L_{\%}$
DANN	99.1±0.8	83.2±1.4	47.2±2.7	32.2±3.5	65.4	76.2±1.5	40.9±1.1	21.3±2.7	12.9±3.7	37.8	
NTG$_1$	-96.5	-80.3	-44.1	-28.3	-62.3	-73.7	-37.3	-17.2	-9.7	-34.5	
DANN$_{\text{gate}}$	98.9±0.6	83.3±2.1	48.4±2.5	32.1±3.1	65.7	76.0±1.2	41.0±1.6	21.5±3.1	13.2±2.4	37.9	0%
NTG$_2$	-96.3	-80.4	-45.3	-28.2	-62.6	-73.5	-37.4	-17.4	-10.0	-34.6	
Δ	↓0.2	↑0.1	↑1.2	↓0.1	↑0.3	↓0.2	↑0.1	↑0.2	↑0.3	↑0.1	
DANN	99.5±0.4	86.8±2.8	73.1±3.3	48.8±4.3	77.0	78.6±2.7	54.8±3.1	49.6±2.1	32.3±2.6	53.8	
NTG$_1$	-48.7	-37.8	-23.6	1.6	-27.1	-28.4	-4.4	1.2	18.4	-3.3	
DANN$_{\text{gate}}$	99.2±0.3	85.4±2.6	79.4±2.9	50.4±3.2	78.6	85.1±1.7	60.2±2.1	58.3±2.0	49.1±2.5	63.2	10%
NTG$_2$	-48.4	-36.4	-29.9	0.0	-28.7	-34.9	-9.8	-7.5	1.6	-12.7	
Δ	↓0.3	↓1.4	↑6.3	↑1.6	↑1.6	↑6.5	↑5.4	↑8.7	↑16.8	↑9.4	
DANN	99.6±0.2	89.7±1.6	78.4±2.5	70.5±4.3	84.6	80.2±2.0	73.3±2.2	70.2±3.3	51.3±4.3	68.8	
NTG$_1$	-18.5	-10.3	1.8	8.2	-4.7	-1.5	6.5	8.9	28.4	10.6	
DANN$_{\text{gate}}$	100.0±0.1	90.4±1.8	82.0±1.8	79.9±3.8	88.1	89.0±1.5	82.6±1.0	81.3±2.1	80.6±1.8	83.4	30%
NTG$_2$	-18.9	-11.0	-1.8	-1.2	-8.2	-10.3	-2.8	-2.2	-0.9	-4.1	
Δ	↑0.4	↑0.7	↑3.6	↑9.4	↑2.6	↑8.8	↑9.3	↑11.1	↑29.3	↑14.6	
DANN	100.0±0.0	92.2±1.7	85.8±2.3	78.2±4.8	89.1	84.5±1.9	77.6±3.8	70.6±4.9	65.4±6.3	74.5	
NTG$_1$	-11.7	-3.2	3.8	10.4	-0.2	4.6	12.1	18.8	23.2	14.7	
DANN$_{\text{gate}}$	100.0±0.0	93.3±1.7	91.2±1.5	89.5±3.4	92.5	93.2±1.3	91.4±1.2	90.2±2.0	89.8±1.9	91.2	50%
NTG$_2$	-11.7	-4.3	-1.6	-0.9	-4.6	-4.1	-1.7	-0.8	-1.2	-2.0	
Δ	→0.0	↑1.1	↑5.4	↑11.3	↑4.5	↑8.7	↑13.8	↑19.6	↑24.4	↑16.7	

Table 3.1: Classification accuracy (%) of DANN and DANN$_{\text{gate}}$ on tasks W→D and A→D. Perturbation rates are set equal, i.e. $\epsilon = \epsilon_x = \epsilon_y$. NTG_1 and NTG_2 are negative transfer gaps for DANN and DANN$_{\text{gate}}$. Δ is the performance gain of DANN$_{\text{gate}}$ compared to DANN.

five random repeats. To test whether negative transfer occurs, we measure the negative transfer gap (NTG) as the gap between the accuracy of target-only baseline and that of the original method. For instance, for DANN, the target-only baseline is DANN$_T$ which treats labeled target data as “source" data and uses unlabeled data as usual. A positive NTG indicates the occurrence of negative transfer and vice versa.

3.4.3 Study of Negative Transfer

To reveal the three dependent factors, we study the effect of negative transfer under different methods with varying perturbation rates (ϵ_x, ϵ_y) and target labeled data $(L_{\%})$.

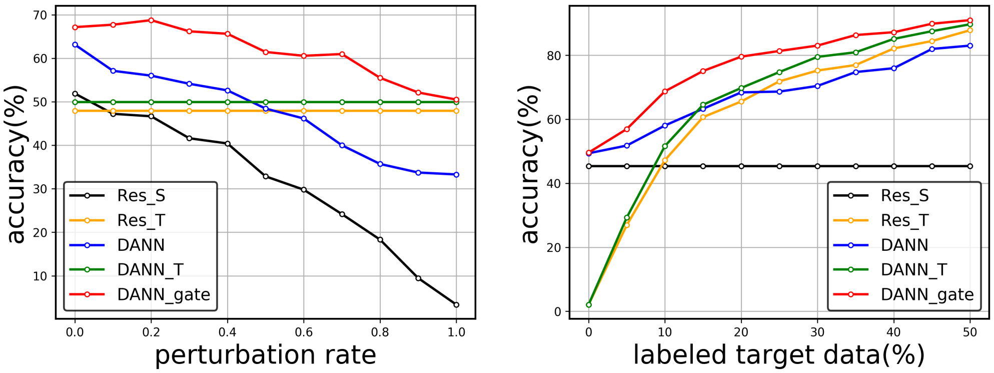
Divergence factor. The performance of DANN under different settings of ϵ and $L_{\%}$ on two tasks of Office-31 are shown in Table 3.1. We observe an increasing negative transfer gap as we increase the perturbation rate in all cases. In some cases such as $L_{\%} = 10\%$, we can even observe a change in the sign of NTG. For a more fine-grained study, we investigate a wider spectrum of distribution divergence by gradually increasing ϵ from 0.0 to 1.0 in Figure 3.3(a). Although DANN is better than DANN$_T$ when ϵ is small, its performance degrades quickly as ϵ increases and drops below DANN$_T$, indicating the occurrence of negative transfer. On the other hand, by fixing $\epsilon_y = 0$ and using two domains **W** and **D** that are known to be particularly similar, we study negative transfer under the assumption of covariate shift in Table 3.3, and observe that negative transfer does *not* occur even with high ϵ_x and descent $L_\%$. These experimental results confirms that the distribution divergence is an important factor of negative transfer.

Target factor. Fixing a specific ϵ, we observe that the negative transfer gap increases as $L_\%$ increases in Table 3.1. In the extreme case of unsupervised adaptation ($L_\% = 0\%$), NTG stays negative even if two domains are far apart ($\epsilon = 0.9$). In Figure 3.3(b), we fix $\epsilon = 0.2$ and plot the performance curve as $L_\%$ increases. We can see that while both DANN and DANN$_T$ perform better with more labeled target data, DANN is affected by the divergence factor and outperformed by DANN$_T$ when $L_\%$ becomes larger. This observation shows that negative transfer is relative and it depends on target labeled data.

Algorithm factor. In Table 3.2, we compare the results of all methods under a more practically interesting scenario of moderately different distributions and limited amount of labeled target data. We observe that some methods are more vulnerable to negative transfer then the other even using the same training data. For conventional methods, instance-reweighting method KMM achieves smaller NTG compared to feature selection method TCA, possibly because KMM can assign small weights to source instances with dissimilar input features. For deep methods, we find GTA to be the most robust method against negative transfer since it takes both label information and random noises as inputs to the generator network. More interestingly, we observe that methods based on distribution measurement such as MMD (e.g. DAN) achieve smaller NTG than methods based on adversarial networks (e.g. DANN), even though the later tends to perform better when distributions are similar. This is consistent with findings in previous works (Cao et al., 2018a) and one possible explanation is that adversarial network's better capability of matching source and target domains leads to more severe negative transfer. Similarly, ADDA has better matching power by using two separate feature extractors, but it results in larger NTG compared to DANN.

3.4.4 Evaluation of Discriminator Gate

We compare our gated models with their respective state-of-the-art methods on the benchmarks in Table 3.2. Even using limited amount of labeled target data, our proposed method consistently



(a) $L_{\%}$ fixed at 20%

(b) ϵ fixed at 0.2

Figure 3.3: Incremental performance on task Pr→Rw. Res_S and Res_T are ResNet-50 baselines trained using only source data and only target data. Perturbation rates are set equal, i.e. $\epsilon = \epsilon_x = \epsilon_y$.

improves the performance for all deep methods on all tasks. More importantly, our method can largely eliminate the negative impact of less related source data and avoid negative transfer (e.g. $\text{DANN}_{\text{gate}}$ achieves negative average NTG while DANN gets positive NTG). Specifically, our method achieves larger accuracy gains on harder tasks such as synthetic to real-world tasks in Office-Home and VisDA. This is mainly because source domains in these tasks tend to contain more unrelated samples. This finding is also consistent with results in Table 3.1 and Figure 3.3(a) where we can observe larger performance gains as perturbation rates increase. In the extreme case where the source domain is degenerate ($\epsilon = 1.0$ in Figure 3.3(a)), the gated model achieves comparable results to those of DANN_T. On the other hand, the results of DANN and $\text{DANN}_{\text{gate}}$ are similar when source domain is closely related to the target ($\epsilon = 0.0$ on task W→D in Table 3.1). This indicates that the discriminator gate can control the trade-off between maximal transfer and alleviating negative impact.

Ablation Study. We report the results of ablation study in Table 3.4 and analyze the effects of several components in our method subject to different settings of transfer tasks. First, both $\text{DANN}_{\text{gate-only}}$ and $\text{DANN}_{\text{label-only}}$ perform better than DANN but worse than $\text{DANN}_{\text{gate}}$, showing that the discriminator gate and estimating joint distributions can both improve performance but their combination yields full performance benefit. Second, $\text{DANN}_{\text{joint}}$ obtains higher accuracy results than $\text{DANN}_{\text{marginal}}$ and $\text{DANN}_{\text{none}}$ since matching joint distributions is the key to avoid negative transfer when both marginal and conditional distributions shift. However, while $\text{DANN}_{\text{joint}}$ achieves comparable results as $\text{DANN}_{\text{gate}}$ when $L_{\%} = 30\%$, it performs

	Digits	Office-31			Office-Home			VisDA	
Method	SVHN→MNIST	W→D	A→D	D→A	Ar→Rw	Cl→Rw	Pr→Rw	Synthetic→Real	Avg
TCA	58.7(18.2)	54.2(-4.2)	11.4(20.5)	13.1(18.4)	-	-	-	-	34.4(13.2)
KMM	70.9(6.0)	58.7(-8.5)	18.5(13.4)	17.7(13.8)	-	-	-	-	41.5(6.2)
DAN	78.5(-4.4)	76.3(-19.5)	55.0(-1.3)	39.2(4.9)	43.2(3.8)	30.2(5.8)	47.2(4.0)	28.4(7.2)	49.8(0.1)
DAN_{gate}	82.2(-8.1)	78.7(-21.9)	60.4(-6.7)	43.9(0.2)	46.8(0.2)	38.0(-2.0)	50.4(0.8)	36.2(-0.6)	54.6(-4.7)
Δ_{DAN}	↑3.7	↑2.4	↑5.4	↑4.7	↑3.6	↑7.8	↑3.2	↑7.8	↑4.8
DCORAL	75.2(-1.2)	75.7(-18.9)	53.8(-0.4)	37.4(5.0)	44.0(3.7)	32.4(4.1)	48.0(2.2)	30.5(5.7)	49.6(0.0)
$\text{DCORAL}_{\text{gate}}$	81.0(-7.0)	78.2(-21.4)	59.0(-5.6)	43.2(-0.8)	48.5(-0.8)	40.0(-3.5)	51.6(-1.4)	35.8(0.4)	54.7(-5.1)
Δ_{DCORAL}	↑5.8	↑2.5	↑5.2	↑5.8	↑4.5	↑7.6	↑3.6	↑5.3	↑5.1
DANN	68.3(7.7)	75.0(-19.2)	51.0(2.3)	38.2(5.6)	42.8(4.2)	28.5(7.7)	42.0(10.0)	29.9(6.0)	47.0(3.0)
$\text{DANN}_{\text{gate}}$	78.1(-2.1)	80.2(-24.4)	61.8(-8.5)	48.3(-4.5)	51.2(-4.2)	43.8(-7.6)	55.2(-3.2)	40.5(-4.6)	57.4(-7.4)
Δ_{DANN}	↑9.8	↑5.2	↑10.8	↑10.1	↑9.4	↑14.7	↑13.2	↑10.6	↑10.4
ADDA	63.2(12.2)	74.5(-18.1)	49.9(2.2)	38.3(5.1)	41.4(6.0)	25.2(13.5)	43.2(7.2)	28.0(7.3)	45.5(4.4)
$\text{ADDA}_{\text{gate}}$	79.4(-4.0)	82.9(-26.5)	64.2(-12.1)	47.7(-4.3)	52.2(-4.8)	48.0(-9.3)	58.2(-7.8)	43.0(-7.7)	59.5(-9.6)
Δ_{ADDA}	↑16.2	↑8.4	↑14.3	↑9.4	↑10.8	↑22.8	↑15.0	↑15.0	↑14.0
PADA	69.7(6.5)	75.5(-19.0)	50.2(1.9)	38.7(5.1)	43.2(3.8)	30.1(5.5)	43.4(6.6)	32.2(5.5)	47.9(2.0)
$\text{PADA}_{\text{gate}}$	81.8(-5.6)	81.6(-25.1)	62.1(-10.0)	44.8(-1.0)	52.8(-5.8)	45.2(-9.6)	54.5(-4.5)	41.4(-5.7)	58.0(-8.1)
Δ_{PADA}	↑12.1	↑5.9	↑11.9	↑6.1	↑9.6	↑15.1	↑11.1	↑11.2	↑10.1
GTA	81.2(-6.8)	78.9(-20.5)	58.4(-7.2)	42.2(2.8)	48.2(1.0)	33.1(5.1)	50.2(-0.1)	31.2(4.2)	52.9(-2.7)
GTA_{gate}	83.3(-8.9)	85.8(-27.4)	66.7(-15.5)	48.5(-3.5)	55.0(-5.8)	44.9(-6.7)	58.0(-7.7)	43.8(-8.4)	60.8(-10.6)
Δ_{GTA}	↑2.1	↑6.9	↑8.3	↑6.3	↑6.8	↑11.8	↑7.8	↑12.6	↑7.9
ΔAvg	↑8.3	↑5.2	↑8.1	↑7.1	↑7.5	↑13.3	↑8.9	↑10.4	

Table 3.2: Classification accuracy (%) of state-of-the-art methods on four benchmark datasets with negative transfer gap shown in brackets. Perturbation rates are fixed at $\epsilon_x = \epsilon_y = 0.7$. Target labeled ratio is set at $L_\% = 10\%$ and we further enforce each task to use at most 3 labeled target samples per class.

worse than $\text{DANN}_{\text{gate}}$ when $L_\% = 10\%$. This shows that utilizing unlabeled target data to match marginal distributions can be beneficial when labeled target data is scarce. Lastly, it is inspiring to see $\text{DANN}_{\text{gate}}$ outperforms $\text{DANN}_{\text{oracle}}$ when perturbation rate is high. This is because less unperturbed source data are used for $\text{DANN}_{\text{oracle}}$ but $\text{DANN}_{\text{gate}}$ can utilize perturbed source data that contain related information. This further shows the effectiveness of our approach.

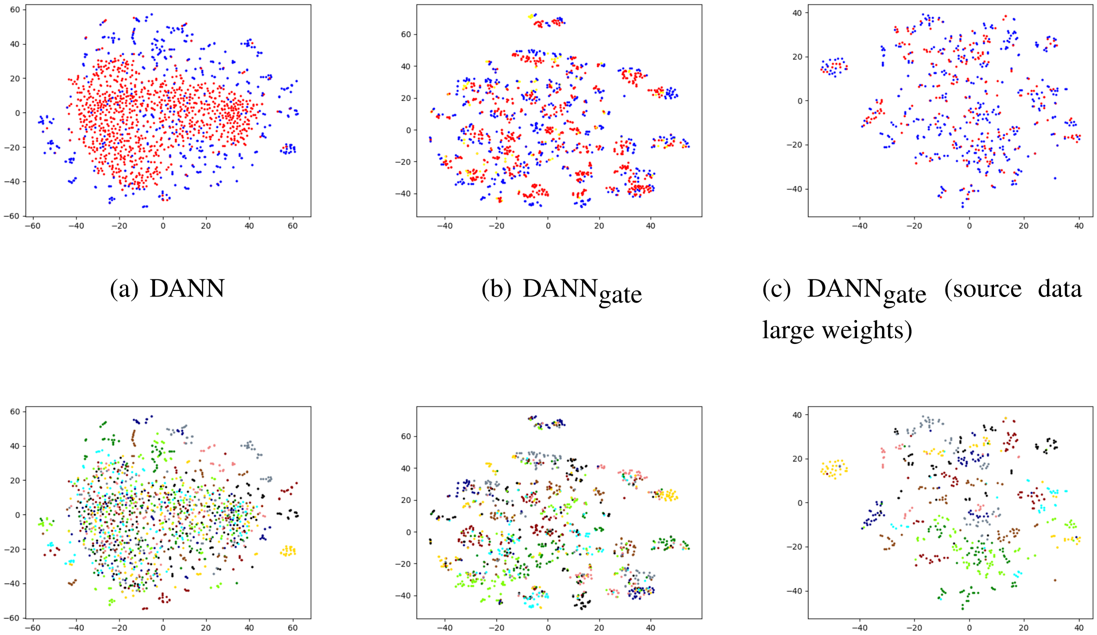
Feature Visualization. We visualize the t-SNE embeddings (Donahue et al., 2014) of the bottleneck representations in Figure 3.4. The first column shows that, when perturbation rate is high, DANN cannot align the two domains well and it fails to discriminate both source and target classes as different classes are mixed together. The second column illustrates the discriminator gate can improve the alignment by assigning less weights to unrelated source data. For instance, we can see some source data from different classes mixed in the yellow cluster at the center right

Method	ϵ_x=0.7 $L_{\%}$=10%	ϵ_x=1.0 $L_{\%}$=30%
DAN	81.2(-29.3)	85.8(-6.2)
DANN	83.0(-30.8)	86.1(-6.5)
GTA	85.5(-33.5)	88.1(-8.0)

Table 3.3: Classification accuracy (%) under the Covariate Shift assumption on task W→D. ϵ_y is fixed at 0. Negative transfer gap is shown in brackets.

	Setting (ϵ,$L_{\%}$)				
Method	0.7, 30%	0.7, 10%	0.3, 30%	0.3, 10%	Avg
DANN	70.4	49.4	72.5	54.3	61.7
DANN$_T$	79.5	50.7	80.3	50.1	65.2
DANN$_{\text{oracle}}$	81.6	58.5	**89.1**	**85.4**	**78.7**
DANN$_{\text{gate-only}}$	76.3	53.8	78.0	55.7	66.0
DANN$_{\text{label-only}}$	74.4	52.5	77.5	55.0	64.9
DANN$_{\text{joint}}$	82.3	57.6	83.1	59.4	70.6
DANN$_{\text{marginal}}$	80.6	56.5	81.5	58.6	69.3
DANN$_{\text{none}}$	79.6	52.4	79.7	57.5	67.3
DANN$_{\text{gate}}$	**82.5**	**58.7**	82.7	60.7	71.2

Table 3.4: Ablation Study on task A→D. DANN$_{\text{gate-only}}$ applies only the discriminator gate while DANN$_{\text{label-only}}$ only uses label information without the gate. DANN$_{\text{joint}}$ is a variant of DANN$_{\text{gate}}$ where the feature network only matches the joint distribution (last two lines of Eq.3.10), DANN$_{\text{marginal}}$ only matches the marginal distribution, and DANN$_{\text{none}}$ matches none of them. DANN$_{\text{oracle}}$ excludes perturbed source data via human oracle.



(a) DANN

(b) $\text{DANN}_{\text{gate}}$

(c) $\text{DANN}_{\text{gate}}$ (source data with large weights)

(d) DANN

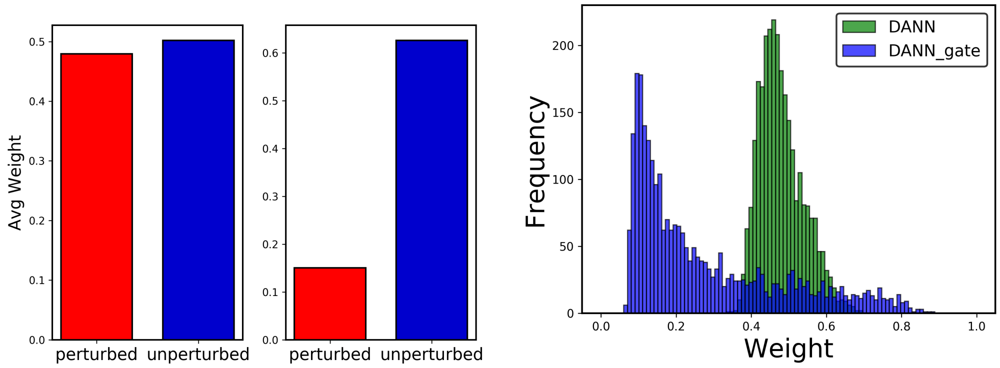
(e) $\text{DANN}_{\text{gate}}$

(f) $\text{DANN}_{\text{gate}}$ (source data with large weights)

Figure 3.4: Visualization on A→W, with $\epsilon = 0.7$, $L_{\%} = 30\%$. The t-SNE visualization. First row shows domain info with red for source samples (yellow for weights > 0.4) and blue for target samples. Second row shows corresponding class info.

but they get assigned smaller weights. The third column shows the embeddings after we remove source data with small discriminator weights (< 0.4). We can observe that target data are much better clustered compared to that of DANN. These in-depth results demonstrate the efficacy of discriminator gate method.

Statistics of Instance Weights. We illustrate the discriminator output ($\frac{P_T(x_s^i, y_s^i)}{P_T(x_s^i, y_s^i) + P_S(x_s^i, y_s^i)}$) for each source data in Figure 3.5(b). We can observe that DANN fails to discriminate unrelated source data as all weights concentrate around 0.5 in the middle. On the other hand, $\text{DANN}_{\text{gate}}$ assigns smaller weights to a large portion of source data (since perturbation rate is high) and thus filters out unrelated information. Figure 3.5(a) further shows that DANN assign similar average weights for perturbed and unperturbed source data while $\text{DANN}_{\text{gate}}$ outputs much smaller values for perturbed data but higher ones for unperturbed data.



(a) Left: DANN Right:DANN$_{\text{gate}}$

(b) Source Sample Weights

Figure 3.5: Left shows the histogram of discriminator weights for source samples. Right shows average weights for perturbed and unperturbed samples.

3.5 Negative Transfer Definition

In this section we further discuss how the negative transfer condition (NTC) in Eq.(3.3) is derived.

The intuitive definition given earlier in Section 3.2 does not lead to a rigorous definition. There are two key questions that are not clear: (1) Should negative transfer be defined to be algorithm-specific? (2) What is the negative impact being compared with?

First, if negative transfer is completely algorithm-agnostic, then its definition would be independent to which transfer learning algorithm is being used. Mathematically, this may yield the following:

$$\min_{A} R_{P_T}(A(\mathcal{S},\mathcal{T})) > \min_{A'} R_{P_T}(A'(\varnothing,\mathcal{T})). \tag{3.12}$$

However, it is easy to see that this condition is never satisfied. To show this, given source data $\mathcal{S}$ and target data $\mathcal{T}$, consider an algorithm A_1 that minimizes the expected risk on the RHS:

$$A_1 \in \operatorname*{argmin}_{A'} R_{P_T}(A'(\varnothing,\mathcal{T})).$$

Then we can always construct a new algorithm A_1' such that $A_1'(\mathcal{S},\mathcal{T}) = A_1(\varnothing,\mathcal{T})$, i.e. A_1' always ignores the source data. As a result, we must have:

$$\begin{aligned} \min_{A} R_{P_T}(A(\mathcal{S},\mathcal{T})) &\le R_{P_T}(A_1'(\mathcal{S},\mathcal{T})) \\ &= R_{P_T}(A_1(\varnothing,\mathcal{T})) \\ &= \min_{A'} R_{P_T}(A'(\varnothing,\mathcal{T})) \end{aligned} \tag{3.13}$$

Therefore, the condition defined in Eq.(3.12) is never true and we conclude that negative transfer must be algorithm-specific. This answers the first question.

Given the answer, the condition in Eq.(3.12) could be modified to consider only a specific transfer algorithm A, i.e.,

$$R_{P_T}(A(\mathcal{S},\mathcal{T})) > \min_{A'} R_{P_T}(A'(\varnothing,\mathcal{T})). \tag{3.14}$$

However, there are still two problems with this definitions:

(a) This condition cannot be measured in practice since we cannot evaluate the RHS even at test time;

(b) An algorithm that does not utilize any source at all still satisfies the condition, which is counterintuitive. For instance, consider a degenerated algorithm A_2 such that $A_2(\mathcal{S},\mathcal{T}) = A_2(\varnothing,\mathcal{T})$ and $R_{P_T}(A_2(\varnothing,\mathcal{T})) > \min_{A'} R_{P_T}(A'(\mathcal{S},\mathcal{T}))$. This algorithm does not perform any meaningful transfer from the source, but negative transfer occurs in this case according to Eq. (3.14) since:

$$R_{P_T}(A_2(\mathcal{S},\mathcal{T})) = R_{P_T}(A_2(\varnothing,\mathcal{T})) > \min_{A'} R_{P_T}(A'(\varnothing,\mathcal{T})).$$

Therefore, it is misleading to only compare with the best possible algorithm and we propose the following definition:

Definition 1 (Negative Transfer). *Given a source dataset $\mathcal{S}$, a target dataset $\mathcal{T}$ and a transfer learning algorithm A, the negative transfer condition (NTC) is defined as:*

$$R_{P_T}(A(\mathcal{S},\mathcal{T})) > R_{P_T}(A(\varnothing,\mathcal{T})) \geq \min_{A'} R_{P_T}(A'(\varnothing,\mathcal{T})), \tag{3.15}$$

which is exactly Eq.(3.3) since the “$\geq$" constraint on the right hand side is true for any A. This definition of NTC resolves the two questions mentioned above. Furthermore, it is consistent with the intuitive definition and is also tractable at test time.

3.6 Related Work

Transfer learning (Pan and Yang, 2010; Yang et al., 2013) uses knowledge learned in the source domain to assist training in the target domain. Early methods exploit conventional statistical techniques such as instance weighting (Huang et al., 2007) and feature mapping (Pan et al., 2011; Uguroglu and Carbonell, 2011). Compared to these earlier approaches, deep transfer networks achieve better results in discovering domain invariant factors (Yosinski et al., 2014). Some deep methods (Long et al., 2015; Sun and Saenko, 2016) transfer via distribution (mis)match measurements such as Maximum Mean Discrepancy (MMD) (Huang et al., 2007). More recent work(Cao et al., 2018a; Ganin et al., 2016; Sankaranarayanan et al., 2018; Tzeng et al., 2015) exploit generative adversarial networks (GANs) (Goodfellow et al., 2014) and add a subnetwork as a domain discriminator. These methods achieve state-of-the-art on computer vision tasks (Sankaranarayanan et al., 2018) and some natural language processing tasks (Moon and Carbonell, 2017). However, none of these techniques are specifically designed to tackle the problem of negative transfer.

Negative transfer Early work that noted negative transfer (Rosenstein et al., 2005) was targeted at simple classifiers such as hierarchical Naive Bayes. Later, similar negative effects have also been observed in various settings including multi-source transfer learning (Duan et al., 2012), imbalanced distributions (Ge et al., 2014) and partial transfer learning (Cao et al., 2018a). While the importance of detecting and avoiding negative transfer has raised increasing attention (Weiss et al., 2016), the literature lacks in-depth analysis.

3.7 Summary

In this chapter, we analyze the problem of negative transfer in domain adaptation and propose a novel discriminator gate technique to avoid it. We show that negative transfer directly relates to specific algorithms, domain divergence and target data. Experiments demonstrate these factors and the efficacy of our method. Our method consistently improves the performance of base methods and largely avoids negative transfer. However, we focus on feature matching methods in this chapter and our results may not generalize to other transfer learning methods, which we are looking forward to study in a future work. Understanding negative transfer in more complex transfer tasks and settings is an important direction, which we study in the next chapter.

Chapter 4

Negative Interference in Multilingual Unsupervised Representation Learning

In the previous chapter, we have analyzed negative transfer in domain adaptation with one single source and target. This chapter explores a more complex setting of transferring among multiple tasks where each task is a language. Modern multilingual models are trained on concatenated text from multiple languages in hopes of conferring benefits to each (positive transfer), with the most pronounced benefits accruing to low-resource languages. However, recent work has shown that this approach can degrade performance on high-resource languages, a form of negative transfer among multiple tasks known as *negative interference*. In this chapter, we present the first systematic study of negative interference. We show that, contrary to previous belief, negative interference also impacts low-resource languages. While parameters are maximally shared to learn language-universal structures, we demonstrate that language-specific parameters do exist in multilingual models and they are a potential cause of negative interference. Motivated by these observations, we also present a meta-learning algorithm that obtains better cross-lingual transferability and alleviates negative interference, by adding language-specific layers as meta-parameters and training them in a manner that explicitly improves shared layers' generalization on all languages. Overall, our results show that negative interference is more common than previously known, suggesting new directions for improving multilingual representations.

4.1 Introduction

Advances in pretraining language models (Devlin et al., 2018; Liu et al., 2019; Yang et al., 2019) as general-purpose representations have pushed the state of the art on a variety of natural language tasks. However, not all languages enjoy large public datasets for pretraining and/or downstream tasks. Multilingual language models such as mBERT (Devlin et al., 2018) and XLM (Lample and Conneau, 2019) have been proven effective for cross-lingual transfer learning by pretraining a single shared Transformer model (Vaswani et al., 2017) jointly on multiple languages. The goals of multilingual modeling are not limited to improving language modeling in low-resource languages (Lample and Conneau, 2019), but also include zero-shot cross-lingual transfer on downstream tasks—it has been shown that multilingual models can generalize to target languages even when labeled training data is only available in the source language (typically English) on a wide range of tasks (Hu et al., 2020; Pires et al., 2019; Wu and Dredze, 2019).

However, multilingual models are not equally beneficial for all languages. Conneau et al. (2020) demonstrated that including more languages in a single model can improve performance for low-resource languages but hurt performance for high-resource languages, while Neubig and Hu (2018) found more source language pairs can hurt transfer into low-resource languages. Similarly, recent work (Aharoni et al., 2019; Arivazhagan et al., 2019; Johnson et al., 2017; Tan et al., 2019) in multilingual neural machine translation (NMT) also observed performance degradation on high-resource language pairs. In multi-task learning (Ruder, 2017), this phenomenon is known as *negative interference* or *negative transfer* (Wang et al., 2019b), where training multiple tasks jointly hinders the performance on individual tasks.

Despite these empirical observations, little prior work analyzed or showed how to mitigate negative interference in multilingual language models. Particularly, it is natural to ask: (1) Can negative interference occur for low-resource languages also? (2) What factors play an important role in causing it? (3) Can we mitigate negative interference to improve the model's cross-lingual transferability?

In this chapter, we take a step towards addressing these questions. We pretrain a set of monolingual and bilingual models and evaluate them on a range of downstream tasks to analyze negative interference. We seek to individually characterize the underlying factors of negative interference through a set of ablation studies and glean insights on its causes. Specifically, we examine if training corpus size and language similarity affect negative interference, and also measure gradient and parameter similarities between languages.

Our results show that negative interference can occur in both high-resource and low-resource languages. In particular, we observe that neither subsampling the training corpus nor adding typologically similar languages substantially impacts negative interference. On the other hand, we show that gradient conflicts and language-specific parameters do exist in multilingual models, suggesting that languages are fighting for model capacity, which potentially causes negative interference. We further test whether explicitly assigning language-specific modules to each language can alleviate negative interference, and find that the resulting model performs better within each individual language but worse on zero-shot cross-lingual tasks.

Motivated by these observations, we further propose to meta-learn these language-specific parameters to explicitly improve generalization of shared parameters on all languages. Empirically, our method improves not only within-language performance on monolingual tasks but also cross-lingual transferability on zero-shot transfer benchmarks. To the best of our knowledge, this is the first effort to systematically study and remedy negative interference in multilingual language models.

4.2 Motivation

Multilingual transfer learning aims at utilizing knowledge transfer across languages to boost performance on low-resource languages. State-of-the-art multilingual language models are trained on multiple languages jointly to enable cross-lingual transfer through parameter sharing. However, languages are heterogeneous, with different vocabularies, morphosyntactic rules, and different pragmatics across cultures. It is therefore natural to ask, *is knowledge transfer beneficial for all languages in a multilingual model?* To analyze the effect of knowledge transfer from other languages on a specific language *lg*, we can compare multilingual models with the monolingual model trained on *lg*. For example, in Figure 4.1, we compare the performance on a named entity recognition (NER) task of monolingually-trained models vs. bilingual models (trained on *lg* and English, details shown later) vs. state-of-the-art XLM(Conneau et al., 2020). We can see that monolingual models outperform multilingual models on four out of six languages (See Section 4.3.3 for details). This shows that language conflicts may induce negative impacts on certain languages, which we refer to as *negative interference*. Here, we investigate the causes of negative interference (Section 4.3.3) and methods to overcome it (Section 4.4).

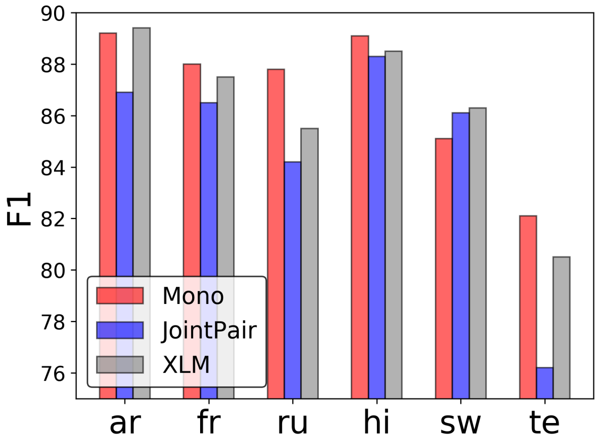


Figure 4.1: Comparing monolingual vs multilingual models on NER. Lower performance of multilingual models is likely an indicator of negative interference.

4.3 Investigating the Sources of Negative Interference in Multilingual Models

4.3.1 Methodology

To study negative interference, we compare multilingual models with monolingual baselines. Without loss of generality, we focus on analyzing bilingual models to minimize confounding factors. For two languages lg_1 and lg_2, we pretrain a single bilingual model and two monolingual models. We then assess their performance on downstream tasks using two different settings. To examine negative interference, we evaluate both monolingual and multilingual models using the **within-language monolingual** setting, such that the pretrained model is finetuned and tested on the same language. For instance, if the monolingual model of lg_1 outperforms the bilingual model on lg_1, we know that lg_2 induces negative impact on lg_1 in the bilingual model. Besides, since multilingual models are trained to enable cross-lingual transfer, we also report their performance on the **zero-shot cross-lingual transfer** setting, where the model is only finetuned on the source language, say lg_1, and tested on the target language lg_2.

We hypothesize that the following factors play important roles in causing negative interference and study each individually:

Training Corpus Size While prior work mostly report negative interference for high-resource languages (Arivazhagan et al., 2019; Conneau et al., 2020), we hypothesize that it can also occur for languages with fewer resources. We study the impact of training data size per language on negative interference. We subsample a high-resource language, say lg_1, to create a "low-resource version". We then retrain the monolingual and bilingual models and compare with the results of their high-source counterparts. Particularly, we test if reducing lg_1's training size also reduces negative interference on lg_2.

Language Similarity Language similarity has been shown important for effective transfer in multilingual models. Wu et al. (2020) shows that bilingual models trained on more similar language pairs result in better zero-shot transfer performance. We thus expect it to play a critical role in negative interference as well. For a specific language lg_1, we pair it with languages that are closely and distantly related. We then compare these bilingual models' performance on lg_1 to investigate if more similar languages cause less severe interference. In addition, we further add a third language lg_3 that is similar to lg_1 and train a trilingual model on lg_1-lg_2-lg_3. We compare the trilingual model with the bilingual model to examine if adding lg_3 can mitigate negative interference on lg_1.

Gradient Conflict Recent work (Yu et al., 2020) shows that gradient conflict between dissimilar tasks, defined as negative cosine similarity between gradients, is predictive of negative interference in multi-task learning. Therefore, we study whether gradient conflicts exist between languages in multilingual models. In particular, we sample one batch for each language in the model and compute the corresponding gradients' cosine similarity for every 10 steps during pretraining.

Parameter Sharing State-of-the-art multilingual models aim to share as many parameters as possible in the hope of learning a language-universal model for all languages (Wu et al., 2020). While prior studies measure the latent embedding similarity between languages, we instead examine model parameters directly. The idea is to test whether model parameters are **language-universal**[1] or **language-specific**. To achieve this, we prune multilingual models for each language using relaxed L_0 norm regularization (Louizos et al., 2017), and compare parameter similarities between languages. Formally, for a model $f(\cdot;\boldsymbol{\theta})$ parameterized by $\boldsymbol{\theta} = \{\theta_i\}_{i=1}^{n}$ where each θ_i represents an individual parameter or a group of parameters, the method introduces a set of binary masks $\mathbf{z}$, drawn from some distribution $q(\mathbf{z}|\boldsymbol{\pi})$ parametrized by $\boldsymbol{\pi}$, and learns a

[1] Ideally, we would like to call a parameter shared among many languages to be universal. Here, we only focus on two languages and we use the same term for consistency.

	en	ar	fr	ru	hi	sw	te
corpus size	44.6	8.7	16.2	13.1	0.5	0.2	0.3
NER	✓	✓	✓	✓	✓	✓	✓
POS	✓	✓	✓	✓	✓		✓
QA	✓	✓		✓		✓	✓
XNLI	✓	✓	✓	✓	✓	✓	

Table 4.1: Language training corpra statitstics and downstream tasks availability. Corpus size measured in millions of sentences.

sparse model $f(\cdot; \boldsymbol{\theta} \odot \mathbf{z})$ by optimizing:

$$\begin{aligned} \min_{\boldsymbol{\pi}} \quad & \mathbb{E}_{q(\mathbf{z}|\boldsymbol{\pi})} \left[\frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(f(x_i; \tilde{\boldsymbol{\theta}}), y_i) + \lambda \|\tilde{\boldsymbol{\theta}}\|_0 \right] \\ \text{s.t.} \quad & \tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} \odot \mathbf{z}, \end{aligned} \tag{4.1}$$

where $\odot$ is the Hadamard (elementwise) product, $\mathcal{L}(\cdot)$ is some task loss and λ is a hyperparameter. We follow the work of (Louizos et al., 2017) and use the Hard Concrete distribution for the binary mask $\mathbf{z}$, such that the above objective is fully differentiable. Then, for each bilingual model, we freeze its pretrained parameter weights and learn binary masks $\mathbf{z}$ for *each* language independently. As a result, we obtain two independent sets of mask parameters $\boldsymbol{\pi}$ which can be used to determine parameter importance. Intuitively, for each parameter group, it is language-universal if both languages consider it important (positive $\boldsymbol{\pi}$ values). On the other hand, if one language assigns a positive value while the other assigns a negative, it shows that the parameter group is language-specific. We compare them across languages and layers to analyze parameter similarity in multilingual models.

4.3.2 Experimental Setup

We focus on standard multilingual masked language modeling (MLM) used in mBERT and XLM. We first pretrain models and then evaluate their performance on four NLP benchmarks.

For pretraining, we mainly follow the setup and implementation of XLM(Lample and Conneau, 2019) [2]. We focus on monolingual and bilingual models for a more controllable comparison, which we refer to as **Mono** and **JointPair** respectively. In particular, we always include

[2] We only focus on pretraining on monolingual corpus while XLM uses resources beyond that. For our study purpose, we utilize its settings on monolingual data only.

English (En) in bilingual models to compare on zero-shot transfer settings with prior work. Besides, we consider three high-resource languages {Arabic (Ar), French (Fr), Russian (Ru)} and three low-resource languages {Hindi (Hi), Swahili (Sw), Telugu (Te)} (see Table 4.1 for their statistics). We chose these six languages based their data availability in downstream tasks. We use Wikipedia as training data with statistics shown in Table 4.1. For each model, we use BPE (Sennrich et al., 2016) to learn 32k subword vocabulary shared between languages. For multilingual models, we sample language proportionally to $P_i = (\frac{L_i}{\sum_j L_j})^{\frac{1}{T}}$, where L_i is the size of the training corpus for i-th language pair and T is the temperature. Each model is a standard Transformer (Vaswani et al., 2017) with 8 layers, 12 heads, 512 embedding size and 2048 hidden dimension for the feedforward layer. Notice that we specifically consider a smaller model capacity to be comparable with existing models with larger capacity but also include much more (over 100) languages. We use the Adam optimizer (Kingma and Ba, 2014) and exploit the same learning rate schedule as Lample and Conneau (2019). We train each model with 4 NVIDIA V100 GPUs with 32GB of memory. Using mixed precision, we fit a batch of 128 for each GPU and the total batch size is 512. Each epoch contains 10k steps and we train for 50 epochs.

For evaluation, we consider four downstream tasks: named entity recognition (NER), part-of-speech tagging (POS), question answering (QA), and natural language inference (NLI). Notice that XNLI only has training data in available in English so we only evaluate zero-shot cross-lingual performance on it. Following (Hu et al., 2020), we finetune the model for 10 epochs for NER and POS, 2 epochs for QA and 200 epochs for XNLI. For NER, POS and QA, we search the following hyperparameters: batch size {16, 32}; learning rate {2e-5, 3e-5, 5e-5}. We use English dev set for zero-shot cross-lingual setting and the target language dev set for within-language monolingual setting. For XNLI, we search for: batch size {4, 8}; encoder learning rate {1e-6, 5e-6, 2e-5}; classifier learning rate {5e-6, 2e-5, 5e-5}. For models with language-specific components, we test freezing these components or finetuning them together. We discover that finetuning the whole network always yields better results. For all experiments, we save checkpoints after each epoch.

NER We use the WikiAnn (Pan et al., 2017) dataset, which is a sequence labelling task built automatically from Wikipedia. A linear layer with softmax classifier is added on top of pretrained models to predict the label for each word based on its first subword. We report the F1 score.

POS Similar to NER, POS is also a sequence labelling task but with a focus on synthetic knowledge. In particular, we use the Universal Dependencies treebanks (Nivre et al., 2018). Task-specific layers are the same and we report F1, as in NER.

QA We choose to use the TyDiQA-GoldP dataset (Clark et al., 2020) that covers typologically

Model	NER (F1)							POS (F1)					
	ar	fr	ru	hi	sw	te	avg	ar	fr	ru	hi	te	avg
Within-language Monolingual													
Mono	89.2	88.0	87.8	89.1	85.1	82.1	86.9	92.7	76.2	96.7	97.0	94.5	91.4
JointPair	86.9	86.5	84.2	88.3	86.1	76.2	84.7	89.2	75.8	93.2	95.2	88.7	88.4
+ ffn	88.2	88.4	86.6	88.9	85.4	81.2	86.5	92.4	76.1	95.6	96.1	92.4	90.5
+ attn	87.3	86.8	84.1	88.5	84.9	77.4	84.8	91.8	75.4	94.4	95.3	90.9	89.6
+ adpt	87.8	86.8	84.5	87.7	86.3	77.0	85.0	91.7	75.6	94.0	95.2	91.5	89.6
+ share adpt	86.8	86.7	84.3	88.6	86.1	76.0	84.8	89.3	76.4	93.5	95.2	88.2	88.5
+ meta adpt	88.9	88.3	85.1	88.4	86.5	79.5	86.1	92.4	75.9	95.1	95.8	92.2	90.3
XLM	89.4	87.5	85.5	88.5	86.3	80.5	86.3	94.5	72.9	96.6	97.1	92.2	90.7
Zero-shot Cross-lingual													
JointPair	38.1	77.5	57.5	61.4	64.8	45.2	57.4	58.5	44.2	80.1	58.9	72.8	62.9
+ ffn	8.9	35.2	5.8	10.5	9.7	12.5	13.8	5.4	8.1	4.5	3.3	7.7	5.8
+ attn	15.4	39.4	10.2	9.9	13.4	11.6	16.7	6.2	4.5	7.5	4.8	6.9	6.0
+ adpt	37.2	75.5	59.2	61.0	64.4	44.7	57.0	57.0	43.5	81.6	58.2	73.5	62.8
+ share adpt	38.5	77.8	58.4	62.0	65.4	44.5	57.8	58.7	43.8	82.5	59.7	71.8	63.3
+ meta adpt	44.4	78.5	62.4	66.0	67.3	50.1	61.5	63.5	44.6	84.9	62.7	78.5	66.8
XLM	44.8	78.3	63.6	65.8	68.4	49.3	61.7	62.8	42.4	86.3	65.7	76.9	66.8

Table 4.2: NER and POS results. We observe negative interference when monolingual models outperform multilingual models. Besides, adding language-specific layers (e.g. ffn) mitigates interference but sacrifices transferability.

diverse languages. Similar to popular QA datasets such as SQuAD (Rajpurkar et al., 2018), this is a span prediction task where task-specific linear classifiers are used to predict start/end positions of the answer. Standard metrics of F1 and Exact Match (EM) are reported.

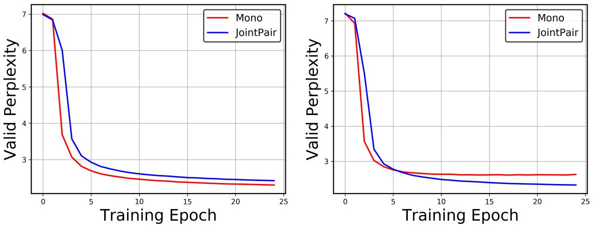
NLI XNLI (Conneau et al., 2018b) is probably the most popular cross-lingual benchmark. Notice that the original dataset only contains training data for English. Consequently, we only evaluate this task on the zero-shot transfer setting while we consider both settings for the rest of other tasks.

Model	ar	ru	sw	te	avg
Within-language Monolingual					
Mono	74.2/62.5	63.1/49.2	52.5/37.4	58.2/41.0	62.0/47.5
JointPair	71.3/58.1	58.2/43.1	52.8/39.0	52.2/36.4	58.6/44.2
+ ffn	73.4/61.2	61.2/45.8	51.4/34.3	57.5/40.5	60.9/45.5
+ attn	72.8/61.0	60.8/45.4	51.2/34.0	52.8/36.8	59.4/44.3
+ adpt	71.5/58.7	59.4/44.8	52.1/38.7	55.5/38.9	59.6/45.3
+ share adpt	71.0/57.8	58.5/43.2	52.8/39.0	53.9/37.2	59.1/44.3
+ meta adpt	73.0/61.4	61.8/46.7	54.5/40.0	56.2/39.5	61.4/36.4
XLM	74.3/63.2	62.5/48.7	58.7/40.4	55.4/38.3	62.7/47.7
Zero-shot Cross-lingual					
JointPair	54.1/39.5	43.2/27.5	41.5/22.2	21.5/14.7	40.1/26.0
+ ffn	2.2/1.5	0.0/0.0	4.4/3.7	0.0/0.0	1.7/1.3
+ attn	3.7/2.0	2.1/1.2	0.7/1.0	0.0/0.0	1.6/1.1
+ adpt	53.4/39.1	44.7/27.9	41.2/21.8	20.4/13.8	39.9/25.7
+ share adpt	54.3/39.6	44.8/27.8	42.2/22.9	22.7/15.6	41.0/26.5
+ meta adpt	57.5/40.8	45.8/28.8	43.0/24.2	23.1/17.7	42.4/27.9
XLM	59.4/41.2	47.3/29.8	42.3/22.0	16.3/7.2	41.3/25.1

Table 4.3: TyDiQA-GoldP results (F1/EM).

4.3.3 Results and Analysis

In Table 4.2 and 4.3, we report our results on NER, POS and QA together with XLM-100, which is trained on 100 languages and contains 827M parameters. In particular, we observe that monolingual models outperform bilingual models for all languages except Swahili on all three tasks. In fact, monolingual models even perform better than XLM on four out of six languages including *hi* and *te*, despite that XLM is much larger in model sizes and trained with much more resources. This shows that negative interference *can* occur on low-resource languages as well. While the negative impact is expected to be more prominent on high-resource languages, we demonstrate that it may occur for languages with resources fewer than commonly believed. The existence of negative interference confirms that state-of-the-art multilingual models cannot



((a)) hi ((b)) sw

Figure 4.2: Validation perplexity during pretraining.

generalize equally well on all languages, and there is still a gap compared to monolingual models on certain languages.

We next turn to dissect negative interference by studying the four factors described in Section 4.3.1.

Training Corpus Size By comparing the validation perplexity on Swahili and Telugu in Figure 4.2, we find that while both monolingual models outperform bilingual models in the first few epochs, the Swahili model's perplexity starts to *increase* and is eventually surpassed by the bilingual model in later epochs. This matches the intuition that monolingual models may overfit when training data size is small. To verify this, we subsample French and Russian to 100k sentences to create a "low-resource version" of them (denoted as fr$_{l}$/ru$_{l}$). As shown in Table 4.4, while the performance for both models drop compared to their "high-resource" counterparts, bilingual models indeed outperform monolingual models for fr$_{l}$/ru$_{l}$, in contrast for fr/ru. This suggests that multilingual models can stimulate positive transfer for low-resource languages when monolingual models overfit. On the other hand, when we compare bilingual models on English, models trained using different sizes of fr/ru data obtain similar performance, indicating that the training size of the source language has little impact on negative interference on the target language (English in this case). While more training data usually implies larger vocabulary and more diverse linguistic phenomena, negative interference seems to arise from more fundamental conflicts contained in even small training corpus.

Language Similarity As illustrated by Table 4.4, the in-language performance on English drops as the paired language becomes more distantly related (French vs Russian). This verifies that transferring from more distant languages results in more severe negative interference.

It is therefore natural to ask if adding more similar languages can mitigate negative interference, especially for low-resource languages. We then train two trilingual models, adding Marathi to English-Hindi, and Kannada to English-Telugu. Compared to their bilingual counter-

Model	NER (F1)				POS (F1)				QA (F1/EM)	
	fr	fr$_l$	ru	ru$_l$	fr	fr$_l$	ru	ru$_l$	ru	ru$_l$
Within-language Performance on fr/ru										
Mono	88.0	81.7	87.8	82.4	76.2	68.5	96.7	88.7	63.1/49.2	47.2/29.5
JointPair	86.5	83.2	84.2	82.7	75.8	71.4	93.2	89.5	58.2/43.1	49.5/30.4
Within-language Performance on en										
JointPair	78.6	78.4	75.8	75.9	94.5	94.5	92.7	92.3	61.7/49.8	62.1/50.2

Table 4.4: Evaluating effects of training corpus sizes on negative interference.

parts (Table 4.5), trilingual models obtain similar within-language performance, which indicates that adding similar languages *cannot* mitigate negative interference among existing languages in the model. However, they do improve zero-shot cross-lingual performance. One possible explanation is that even similar languages can fight for language-specific capacity but they may nevertheless benefit the generalization of the shared knowledge. Notice that prior work (Lin et al., 2019b) has found language similarity to be crucial for better transfer, we emphasize that our results are conditioned on unsupervised representation learning in multilingual models and should be verified for broader multilingual learning settings in future.

Gradient Conflict In Figure 4.3, we plot the gradient cosine similarity between Arabic-English and French-English in their corresponding bilingual models over the first 25 epochs. We also plot the similarity within English, measured using two independently sampled batches[3]. Specifically, gradients between two different languages are indeed less similar than those within the same language. The gap is more evident in the early few epochs, where we observe negative gradient similarities for Ar-En and Fr-En while those for En-En are positive. In addition, gradients in Ar-En are less similar than those in Fr-En, indicating that distant language pair can cause more severe gradient conflicts. These results confirm that gradient conflict exists in multilingual models and is correlated to per language performance, suggesting it may introduce optimization challenge that results in negative interference.

Parameter Sharing The existence of gradient conflicts may imply that languages are fighting for capacity. Thus, we next study how language-universal these multilingual parameters are. Figure 4.4(a) shows the cosine similarity of mask parameters $\boldsymbol{\pi}$ across different layers. We observe that within-language similarity (En-En) is near perfect, which validates the pruning method's robustness. The trend shows that model parameters are better shared in the bottom layers than

[3] Notice that we use gradient accumulation to sample an effectively larger batch of 4096 sentences to calculate the gradient similarity.

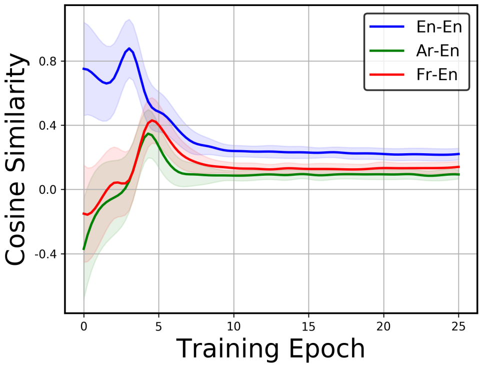


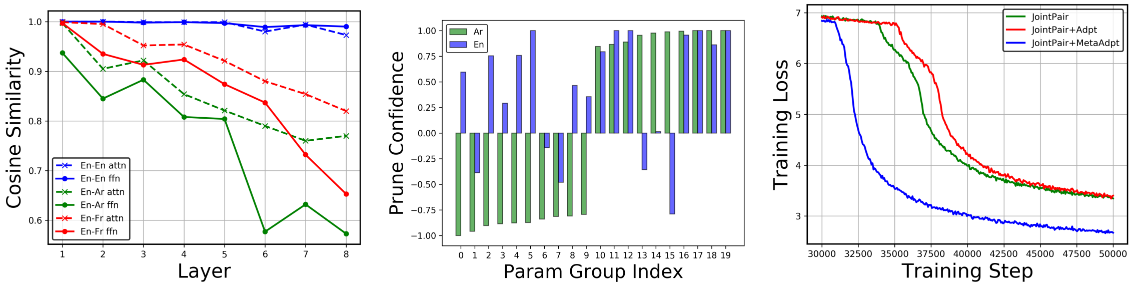
Figure 4.3: Gradients similarity throughout training. “En-En” refers to gradients of two English batches within the Ar-En model, while “Ar-En” and “Fr-En” refer to gradients of two batches, one from each language, within Ar-En and Fr-En models respectively.

Model	NER (F1)		POS (F1)	
	hi	te	hi	te
Within-language Monolingual				
JointPair	88.3	76.2	95.2	88.7
JointTri	87.8	76.4	95.3	88.7
Zero-shot Cross-lingual				
JointPair	61.4	45.2	58.9	72.8
JointTri	63.5	47.6	59.5	74.4

Table 4.5: Comparing trilingual models with bilingual models. This shows the effect of adding a third similar language to bilingual models.

the upper ones. Besides, it also demonstrates that parameters in multi-head attention layers obtain higher similarities than those in feedforward layers, suggesting that the attention mechanism might be more language-universal. We additionally inspect π parameters with the highest absolute values and plot those values for Ar (Figure 4.4(b)), together with their En counterparts. A more negative value indicates that the parameter is more likely to be pruned for that language and vice versa. Interestingly, while many parameters with positive values (on the right) are language-universal as both languages assign very positive values, parameters with negative values (on the left) are mostly language-specific for Ar as En assigns positive values. We observe similar patterns for other languages as well. These results demonstrate that language-specific parameters do exist in multilingual models.

Having language-specific capacity in shared parameters is sub-optimal. It is less transferable and thus can hinder cross-lingual performance. Moreover, it may also take over capacity



((a)) ((b)) ((c))

Figure 4.4: **Left:** Parameter similarity across layers. **Middle:** Normalized pruning variables of highest absolute values for Ar in Ar-En model. 10 parameter groups with most negative values are shown on the left and 10 with most positive values are shown on the right. **Right:** Average MLM training loss after the warm-up stage.

budgets for other languages and degrade their within-language performance, i.e., causing negative interference. A natural next question is whether explicitly adding language-specific capacity into multilingual models can alleviate negative interference. We thus train variants of bilingual models that contain language-specific components for each language. Particularly, we consider adding language-specific feedforward layers, attention layers, and residual adapter layers (Houlsby et al., 2019; Rebuffi et al., 2017), denoted as ffn, attn and adpt respectively. For each type of component, we create two separate copies in each Transformer layer, one designated for each language, while the rest of the network remains unchanged. As shown in Table 4.2 and 4.3, adding language-specific capacity does mitigate negative interference and improve monolingual performance. We also find that language-specific feedforward layers obtain larger performance gains compared to attention layers, consistent with our prior analysis. However, these gains come at a cost of cross-lingual transferability, such that their zero-shot performance drops tremendously. Our results suggest a tension between addressing interference versus improving transferability. In the next section, we investigate how to address negative interference in a manner that can improve performance on *both* within-language tasks and cross-lingual benchmarks.

Algorithm 1 Training XLM with Meta Language-specific Layers

```
Input: Training data
Output: The converged model $\{\boldsymbol{\theta}^*, \boldsymbol{\phi}^*\}$
Initialize model parameters $\{\boldsymbol{\theta}^{(0)}, \boldsymbol{\phi}^{(0)}\}$
while not converged do
    Sample language $i$
    Update language-specific parameters as:
        $\phi_i^{(t+1)} \leftarrow \text{GradientUpdate}(\phi_i^{(t)}, \nabla_{\phi_i^{(t)}} \frac{1}{L} \sum_{j=1}^{L} \mathcal{L}_{\text{val}}^{j}(\theta_i^{(t)} - \beta \nabla_{\boldsymbol{\theta}^{(t)}} \mathcal{L}_{\text{train}}^{i}(\boldsymbol{\theta}^{(t)}, \phi_i^{(t)}), \phi_j^{(t)}))$
    Update shared parameters as:
        $\boldsymbol{\theta}^{(t+1)} \leftarrow \text{GradientUpdate}(\boldsymbol{\theta}^{(t)}, \nabla_{\boldsymbol{\theta}^{(t)}} \mathcal{L}_{\text{train}}(\boldsymbol{\theta}^{(t)}, \boldsymbol{\phi}^{(t+1)}))$
end while=0
```

4.4 Mitigating Negative Interference via Meta Learning

4.4.1 Proposed Method

In the previous section, we demonstrated that while explicitly adding language-specific components can alleviate negative interference, it can also hinder cross-lingual transferability. We notice that a critical shortcoming of language-specific capacity is that they are **agnostic** of the rest of other languages, since by design they are trained on the designated language only. They are thus more likely to overfit and can induce optimization challenges for shared capacity as well. Inspired by recent work in meta learning (Flennerhag et al., 2019) that utilizes meta parameters to improve gradient geometry of the base network, we propose a novel meta-learning formulation of multilingual models that exploits language-specific parameters to improve the generalization of shared parameters.

For a model with some predefined language-specific parameters $\boldsymbol{\phi} = \{\phi_i\}_{i=1}^{L}$, where ϕ_i is designated for the i-th language, and shared parameters $\boldsymbol{\theta}$, our solution is to treat $\boldsymbol{\phi}$ as meta parameters and $\boldsymbol{\theta}$ as base parameters. Ideally, we want $\boldsymbol{\phi}$ to store non-transferable language-specific knowledge to resolve conflicts and improve generalization of $\boldsymbol{\theta}$ in all languages (a.k.a. mitigate negative interference and improve cross-lingual transferability). Therefore, we train $\boldsymbol{\phi}$ based on the following principle: *if $\boldsymbol{\theta}$ follows the gradients on training data for a given $\boldsymbol{\phi}$, the resulting $\boldsymbol{\theta}$ should obtain a good validation performance on all languages*. This implies a bilevel optimization problem (Colson et al., 2007) formally written as:

$$\begin{aligned} &\min_{\boldsymbol{\phi}} \quad \frac{1}{L}\sum_{i=1}^{L}\mathcal{L}_{\text{val}}^{i}(\boldsymbol{\theta}^{*},\phi_i) \\ &\text{s.t.} \quad \boldsymbol{\theta}^{*}=\arg\min_{\boldsymbol{\theta}}\frac{1}{L}\sum_{i=1}^{L}\mathcal{L}_{\text{train}}^{i}(\boldsymbol{\theta},\phi_i), \end{aligned} \tag{4.2}$$

where $\mathcal{L}_{\text{val}}^{i}$ and $\mathcal{L}_{\text{train}}^{i}$ denote the training and the validation MLM loss for the i-th language. Since directly solving this problem can be prohibitive due to the expensive inner optimization, we approximate $\boldsymbol{\theta}^{*}$ by adapting the current $\boldsymbol{\theta}^{(t)}$ using a single gradient step, similar to techniques used in prior meta-learning methods (Finn et al., 2017). This results in a two-phase iterative training process shown in Algorithm 1.

To be specific, at each training step t on the i-th language during pretraining, we first adapt a gradient step on $\boldsymbol{\theta}$ to obtain a new $\boldsymbol{\theta}'$ and update ϕ_i based on the $\boldsymbol{\theta}'$'s validation MLM loss:

$$\begin{aligned} \phi_i^{(t+1)} &= \phi_i^{(t)}-\alpha\nabla_{\phi_i^{(t)}}\frac{1}{L}\sum_{j=1}^{L}\mathcal{L}_{\text{val}}^{j}(\boldsymbol{\theta}',\phi_j^{(t)}) \\ \boldsymbol{\theta}' &= \boldsymbol{\theta}^{(t)}-\beta\nabla_{\boldsymbol{\theta}^{(t)}}\mathcal{L}_{\text{train}}^{i}(\boldsymbol{\theta}^{(t)},\phi_i^{(t)}), \end{aligned} \tag{4.3}$$

where α and β are learning rates. Notice that $\boldsymbol{\theta}'$ is a function of $\phi_i^{(t)}$ and thus this optimization requires computing the gradient of gradient. Particularly, by applying chain rule to the gradient of $\phi_i^{(t)}$, we can observe that it contains a higher-order term:

$$\left[\nabla^2_{\phi_i^{(t)},\boldsymbol{\theta}^{(t)}}\mathcal{L}_{\text{train}}^{i}(\boldsymbol{\theta}^{(t)},\phi_i^{(t)})\right]\cdot\left[\nabla_{\boldsymbol{\theta}'}\frac{1}{L}\sum_{j=1}^{L}\mathcal{L}_{\text{val}}^{j}(\boldsymbol{\theta}',\phi_j^{(t)})\right] \tag{4.4}$$

This is important, since it shows that ϕ_i can obtain information from other languages through higher-order gradients. In other words, language-specific parameters are **not** agnostic of other languages anymore without violating the language-specific requirement. This is because, in Eq. 4.3, while $\nabla_{\boldsymbol{\theta}^{(t)}}$ is based on the i-th language only, the validation loss is computed for all languages. Finally, in the second phase, we update $\boldsymbol{\theta}$ based on the new $\boldsymbol{\phi}^{(t+1)}$:

$$\boldsymbol{\theta}^{(t+1)}=\boldsymbol{\theta}^{(t)}-\beta\nabla_{\boldsymbol{\theta}^{(t)}}\mathcal{L}_{\text{train}}(\boldsymbol{\theta}^{(t)},\boldsymbol{\phi}^{(t+1)}) \tag{4.5}$$

4.4.2 Evaluation

While our method is generic, we evaluate it applied on bilingual models with adapter networks. Adapters have been effectively utilized in multilingual models (Bapna et al., 2019), and we choose them for practical consideration of limiting per-language capacity. Unlike prior works that finetune adapters for adaptation, we train them jointly with shared parameters during pre-training. We follow Houlsby et al. (2019) and insert language-specific adapters after attention and feedforward layers. We leave a more thorough investigation of how to better pick language-specific structures for future work. For downstream task evaluation, we finetune all layers. Notice that computing the gradient of gradient in Eq. 4.3 doubles the memory requirement. In practice, we utilize the finite difference approximation as follows.

Let z_i be the output of the i-th layer of dimension d. The residual adapter network (Bapna et al., 2019; Houlsby et al., 2019; Rebuffi et al., 2017) is a bottleneck layer that first projects z_i to an inner layer with dimension b:

$$h_i = g(W_i^z z_i) \tag{4.6}$$

where $W_i^z \in \mathbb{R}^{d\times b}$ and g is some activation function such as $relu$. It is then projected back to the original input dimension d with a residual connection:

$$o_i = W_i^h h_i + z_i \tag{4.7}$$

where $W_i^h \in \mathbb{R}^{b\times d}$. In our experiments, we fix $b = \frac{1}{4}d$. For a bilingual model of lg_1 and lg_2, we inject two langauge-specific adapters after each attention and feedforward layer, one for each language. For example, if the input text is in lg_1, the network will be routed to adapters designated for lg_1. The rest of the network and training protocol remain unchanged.

The injected adapter layers mimic the warp layers interleaved between base network layers in Flennerhag et al. (2019). Warp layers are meta parameters that aim to improve the performance of the base network. They precondition base network gradients to obtain better gradient geometry. In our experiments, we treat language-specific adapters as meta parameters to improve generalization of the shared network. The adapters are updated according to Eq 4.3, which doubles the memory requirement. In particular, the high-order term in Eq 4.4 requires computing the gradient of gradient. In practice, we approximate this term using the finite difference approximation as:

$$\frac{\nabla_{\phi_i^{(t)}} \mathcal{L}^i_{\text{train}}(\boldsymbol{\theta}^+, \phi_i^{(t)}) - \nabla_{\phi_i^{(t)}} \mathcal{L}^i_{\text{train}}(\boldsymbol{\theta}^-, \phi_i^{(t)})}{2\epsilon} \tag{4.8}$$

where $\boldsymbol{\theta}^{\pm} = \boldsymbol{\theta}^{(t)} \pm \epsilon \nabla_{\boldsymbol{\theta}'} \frac{1}{L} \sum_{j=1}^{L} \mathcal{L}^j_{\text{val}}(\boldsymbol{\theta}', \phi_j^{(t)})$ and ϵ is a small scalar. We use the same value for learning rates α and β in Eq 4.3, to be consistent with standard learning rate schedule used in XLM (Lample and Conneau, 2019).

By evaluating their performance on the zero-shot transfer settings (Table 4.2, 4.3 and 4.6), we observe that our method, denoted as meta adpt, consistently improves the performance over

Model	ar	fr	ru	hi	sw	avg
JointPair	67.1	73.5	69.2	61.5	62.3	66.7
+ ffn	42.5	51.4	40.7	36.2	34.8	41.1
+ attn	48.5	50.7	41.2	33.3	35.1	41.8
+ adpt	67.8	73.7	69.5	62.2	59.7	66.6
+ share adpt	67.9	73.4	70.0	61.8	62.2	67.1
+ meta adpt	68.5	74.8	70.2	64.5	61.5	67.9
XLM	68.2	75.2	72.3	65.4	58.1	67.8

Table 4.6: XNLI results (Accuracy).

JointPair baselines, while ordinary adapters (adpt) perform worse than JointPair. This shows that, the proposed method can effectively utilize the added language-specific adapters to improve generalization of shared parameters across languages. At the same time, our method also mitigates negative interference and outperforms JointPair on within-language performance, closing the gap with monolingual models. In particular, it performs better than ordinary adapters in both settings. We hypothesize that this is because it alleviates language conflicts during training and thus converges more robustly. For example, we plot training loss in the early stage in Figure 4.4(c), which shows that ordinary adapters converge slower than JointPair due to overfitting of language-specific adapters while meta adapters converge much faster. For ablation studies, we also report results for JointPair trained with adapters shared between two languages, denoted as share adpt. Unlike language-specific adapters that can hinder transferability, shared adapters improve both within-language and cross-lingual performance with the extra capacity. However, meta adapters still obtain better performance. These results show that mitigating negative interference can improve multilingual representations.

4.5 Related Work

Unsupervised multilingual language models such as mBERT (Devlin et al., 2018) and XLM (Conneau et al., 2020; Lample and Conneau, 2019) work surprisingly well on many NLP tasks without parallel training signals (Pires et al., 2019; Wu and Dredze, 2019). A line of follow-up work (Artetxe et al., 2019b; Karthikeyan et al., 2020; Wu et al., 2020) study what contributes to the cross-lingual ability of these models. They show that vocabulary overlap is not required for multilingual models, and suggest that abstractions shared across languages emerge automatically during pretraining. Another line of research investigate how to further improve these shared knowledge, such as applying post-hoc alignment (Cao et al., 2020; Wang et al., 2020d) and utilizing better calibrated training signal (Huang et al., 2019a; Mulcaire et al., 2019). While prior work emphasize how to share to improve transferability, we study multilingual models from a different perspective of how to unshare to resolve language conflicts.

Our work is also related to transfer learning (Pan and Yang, 2010) and multi-task learning (Ruder, 2017). In particular, prior work have observed (Rosenstein et al., 2005) and studied (Wang et al., 2019b) negative transfer, such that transferring knowledge from source tasks can degrade the performance in the target task. Others show it is important to remedy negative transfer in multi-source settings (Ge et al., 2014; Wang and Carbonell, 2018). In this chapter, we study negative transfer in multilingual models, where languages contain heavily unbalanced training data and exhibit complex inter-task relatedness.

In addition, our work is related to methods that measure the similarity between cross-lingual representations. For example, existing methods utilize statistical metrics to examine cross-lingual embeddings such as singular vector canonical correlation analysis (Kudugunta et al., 2019; Raghu et al., 2017), eigenvector similarity (Søgaard et al., 2018), and centered kernel alignment (Kornblith et al., 2019; Wu et al., 2020). While these methods focus on testing latent representations, we directly compare similarity of neural network structures through network pruning. Finally, our work is related to meta learning, which sets a meta task to learn model initialization for fast adaptation (Finn et al., 2017; Flennerhag et al., 2019; Gu et al., 2018), data selection (Wang et al., 2020b), and hyperparameters (Baydin et al., 2018). In our case, the meta task is to mitigate negative interference.

4.6 Summary

In this chapter, we present the first systematic study of negative interference in multilingual models and shed light on its causes. We further propose a method and show it can improve cross-lingual transferability by mitigating negative interference. While prior efforts focus on improving sharing and cross-lingual alignment, we provide new insights and a different perspective on unsharing and resolving language conflicts. This concludes the first part of this thesis on understanding negative transfer. In the next part, we turn to improve model generalization by addressing task conflicts and mitigating negative transfer.

Part II

Enhancing Transfer Generalization by Addressing Task Conflicts

Chapter 5

Representation Alignment

The previous part of this thesis dissects negative transfer and demonstrates task conflict to be the root of it. As a result, this part tries to resolve task conflict by explicitly aligning tasks through representation and gradient, with the goal of improving the model's generalization. In this chapter, we first align representation learned for each individual language thereby leading to improved cross-lingual transferability. Learning multilingual representations of text has proven a successful method for many cross-lingual transfer learning tasks. There are two main paradigms for learning such representations: (1) alignment, which maps different independently trained monolingual representations into a shared space, and (2) joint training, which directly learns unified multilingual representations using monolingual and cross-lingual objectives jointly. We first conduct direct comparisons of representations learned using both of these methods across diverse cross-lingual tasks. Our empirical results reveal a set of pros and cons for both methods, and show that the relative performance of alignment versus joint training is task-dependent. Stemming from this analysis, we propose a simple and novel framework that combines these two previously mutually-exclusive approaches. Extensive experiments demonstrate that our proposed framework alleviates limitations of both approaches, and outperforms existing methods on the MUSE bilingual lexicon induction (BLI) benchmark. We further show that this framework can generalize to contextualized representations such as Multilingual BERT, and produces strong results on the CoNLL cross-lingual NER benchmark.

5.1 Introduction

Continuous word representations (Bojanowski et al., 2017; Mikolov et al., 2013a; Pennington et al., 2014) have become ubiquitous across a wide range of NLP tasks. In particular, methods for *cross-lingual word embeddings* (CLWE) have proven a powerful tool for cross-lingual transfer for downstream tasks, such as text classification (Klementiev et al., 2012), dependency parsing (Ahmad et al., 2019), named entity recognition (NER) (Chen et al., 2019; Xie et al., 2018), natural language inference (Conneau et al., 2018b), language modeling (Adams et al., 2017), and machine translation (MT) (Artetxe et al., 2018b; Lample et al., 2018a,b; Zou et al., 2013). The goal of these CLWE methods is to learn embeddings in a *shared vector space* for two or more languages. There are two main paradigms for learning CLWE: *cross-lingual alignment* and *joint training*.

The most successful approach has been the cross-lingual embedding alignment method (Mikolov et al., 2013b), which relies on the assumption that monolingually-trained continuous word embedding spaces share similar structures across different languages. The underlying idea is to first independently train embeddings in different languages using monolingual corpora alone, and then learn a mapping to align them to a shared vector space. Such a mapping can be trained in a supervised fashion using parallel resources such as bilingual lexicons (Jawanpuria et al., 2019; Joulin et al., 2018; Smith et al., 2017; Xing et al., 2015), or even in an unsupervised ("supervision" refers to that provided by a parallel corpus or bilingual dictionaries). manner based on distribution matching (Artetxe et al., 2018a; Conneau et al., 2018a; Zhang et al., 2017b; Zhou et al., 2019). Recently, it has been shown that alignment methods can also be effectively applied to contextualized word representations (Aldarmaki and Diab, 2019; Schuster et al., 2019).

Another successful line of research for CLWE considers joint training methods, which optimize a monolingual objective predicting the context of a word in a monolingual corpus along with either a hard or soft cross-lingual constraint. Similar to alignment methods, some early works rely on bilingual dictionaries (Ammar et al., 2016; Duong et al., 2016) or parallel corpora (Gouws et al., 2015; Luong et al., 2015) for direct supervision. More recently, a seemingly naive *unsupervised* joint training approach has received growing attention due to its simplicity and effectiveness. In particular, Lample et al. (2018b) reports that simply training embeddings on concatenated monolingual corpora of two related languages using a shared vocabulary without any cross-lingual resources is able to produce higher accuracy than the more sophisticated alignment methods on unsupervised MT tasks. Besides, for contextualized representations, unsupervised multilingual language model pretraining using a shared vocabulary has produced state-of-the-art results on multiple benchmarks.

Despite a large amount of research on both alignment and joint training, previous work has neither performed a systematic comparison between the two, analyzed their pros and cons, nor elucidated when we may prefer one method over the other. Particularly, it's natural to ask: (1)

Does the phenomenon reported in Lample et al. (2018b) extend to other cross-lingual tasks? (2) Can we employ alignment methods to further improve unsupervised joint training? (3) If so, how would such a framework compare to supervised joint training methods that exploit equivalent resources, i.e., bilingual dictionaries? (4) And lastly, can this framework generalize to contextualized representations?

In this chapter, we attempt to address these questions. Specifically, we first evaluate and compare alignment versus joint training methods across three diverse tasks: BLI, cross-lingual NER, and unsupervised MT. We seek to characterize the conditions under which one approach outperforms the other, and glean insight on the reasons behind these differences. Based on our analysis, we further propose a simple, novel, and highly generic framework that uses unsupervised joint training as initialization and alignment as a refinement to combine both paradigms. Our experiments demonstrate that our framework improves over both alignment and joint training baselines, and outperforms existing methods on the MUSE BLI benchmark. Moreover, we show that our framework can generalize to contextualized representations such as Multilingual BERT, producing state-of-the-art results on the CoNLL cross-lingual NER benchmark. To the best of our knowledge, this is the first framework that combines previously mutually-exclusive alignment and joint training methods.

5.2 Background: Cross-lingual Representations

Notation. We assume we have two different languages $\{L_1, L_2\}$ and access to their corresponding training corpora. We use $V_{L_i} = \{w^j_{L_i}\}^{n_{L_i}}_{j=1}$ to denote the vocabulary set of the ith language where each $w^j_{L_i}$ represents a unique token, such as a word or subword. The goal is to learn a set of embeddings $E = \{\boldsymbol{x}^j\}^m_{j=1}$, with $\boldsymbol{x}^j \in \mathbb{R}^d$, in a *shared* vector space, where each token $w^j_{L_i}$ is mapped to a vector in E. Ideally, these vectorial representations should have similar values for tokens with similar meanings or syntactic properties, so they can better facilitate cross-lingual transfer.

5.2.1 Alignment Methods

Given the notation, alignment methods consist of the following steps:
Step 1: Train an embedding set $E_0 = E_{L_1} \cup E_{L_2}$, where each subset $E_{L_i} = \{x^j_{L_i}\}^{n_{L_i}}_{j=1}$ is trained independently using the ith language corpus and contains an embedding $x^j_{L_i}$ for each token $w^j_{L_i}$.
Step 2: Obtain a seed dictionary $D = \{(w^i_{L_1}, w^j_{L_2})\}^K_{k=1}$, either provided or learnt unsupervised.

Step 3: Learn a projection matrix $W \in \mathbb{R}^{d \times d}$ based on D, resulting in a final embedding set $E_A = (W \cdot E_{L_1}) \cup E_{L_2}$ in a shared vector space.

To find the optimal projection matrix W, Mikolov et al. (2013b) proposed to solve the following optimization problem:

$$\min_{W \in \mathbb{R}^{d \times d}} \|WX_{L_1} - X_{L_2}\|_F \tag{5.1}$$

where X_{L_1} and X_{L_2} are matrices of size $d \times K$ containing embeddings of the words in D. Xing et al. (2015) later showed further improvement could be achieved by restricting W to an orthogonal matrix, which turns the Eq.(5.1) into the Procrustes problem with the following closed form solution:

$$W^* = UV^T, \tag{5.2}$$

$$\text{with } U\Sigma V^T = \text{SVD}(X_{L_2} X_{L_1}^T) \tag{5.3}$$

where W^* denotes the optimal solution and SVD$(\cdot)$ stands for the singular value decomposition.

As surveyed in Section 5.5, different methods (Artetxe et al., 2018a; Conneau et al., 2018a; Joulin et al., 2018; Smith et al., 2017) differ in the way how they obtain the dictionary D and how they solve for W in step 3. However, most of them still involve solving the Eq.(5.2) as a crucial step.

5.2.2 Joint Training Methods

Joint training methods in general have the following objective:

$$\mathcal{L}_J = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{R}(L_1, L_2) \tag{5.4}$$

where $\mathcal{L}_1$ and $\mathcal{L}_2$ are monolingual objectives and $\mathcal{R}(L_1, L_2)$ is a cross-lingual regularization term. For example, Klementiev et al. (2012) use language modeling objectives for $\mathcal{L}_1$ and $\mathcal{L}_2$. The term $\mathcal{R}(L_1, L_2)$ encourages alignment of representations of words that are translations. Training an embedding set $E_J = E_{L_1} \cup E_{L_2}$ is usually done by directly optimizing $\mathcal{L}_J$.

While supervised joint training requires access to parallel resources, recent studies (Artetxe and Schwenk, 2019; Devlin et al., 2018; Lample and Conneau, 2019; Lample et al., 2018b) have suggested that unsupervised joint training without such resources is also effective. Specifically, they show that the cross-lingual regularization term $\mathcal{R}(L_1, L_2)$ does not require direct cross-lingual supervision to achieve highly competitive results. This is because the shared words between $\mathcal{L}_1$ and $\mathcal{L}_2$ can serve implicitly as anchors by sharing their embeddings to ensure that representations of different languages lie in a shared space. Using our notation, the unsupervised joint training approach takes the following steps:

Step 1: Construct a joint vocabulary $V_J = V_{L_1} \cup V_{L_2}$ that is *shared* across two languages.

Step 2: Concatenate the two training corpora and learn an embedding set E_J corresponding to V_J.

The joint vocabulary is composed of three disjoint sets: V_J^1, V_J^2, V_J^s, where $V_J^s = V_{L_1} \cap V_{L_2}$ is the shared vocabulary set and V_J^i is the set of tokens that appear in the ith language only. Note that a key difference of existing supervised joint training methods is that embeddings corresponding to V_J^s are not shared between E_{L_1} and E_{L_2}, meaning that they are disjoint, as in alignment methods.

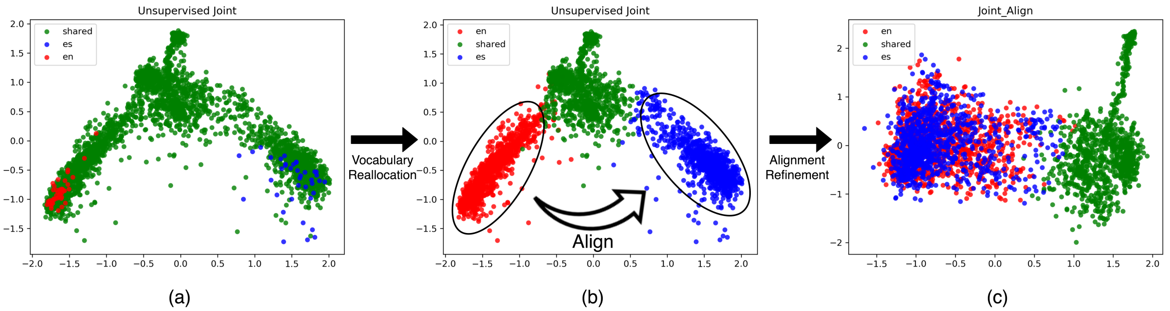
5.2.3 Discussion

While alignment methods have had great success, there are still some critical downsides, among which we stress the following points:

1. While recent studies in unsupervised joint training have suggested the potential benefits of word sharing, alignment methods rely on two disjoint sets of embeddings. Along with some possible loss of information due to no sharing, one consequence is that finetuning the aligned embeddings on downstream tasks may be sub-optimal due to the lack of cross-lingual constraints at the finetuning stage, whereas shared words can fulfill this role in jointly trained models.
2. A key assumption of alignment methods is the isomorphism of monolingual embedding spaces. However, some recent papers have challenged this assumption, showing that it does not hold for many language pairs (Patra et al., 2019; Søgaard et al., 2018). Also notably, Ormazabal et al. (2019) suggests that this limitation results from the fact that the two sets of monolingual embeddings are independently trained.

On the other hand, the *unsupervised* joint training method is much simpler and doesn't share these disadvantages with the alignment methods, but there are also some key limitations:

1. It assumes that all shared words across two languages serve implicitly as anchors and thus need not be aligned to other words. Nonetheless, this assumption is not always true, leading to misalignment. For example, the English word "the" will most likely also appear in the training corpus of Spanish, but preferably it should be paired with Spanish words such as "el" and "la" instead of itself. We refer to this problem as oversharing.
2. It does not utilize any explicit form of seed dictionary as in alignment methods, resulting in potentially less accurate alignments, especially for words that are not shared.



(a)

(b)

(c)

Figure 5.1: PCA visualization of English and Spanish embeddings learned by unsupervised joint training as in Lample et al. (2018b). As shown by plots (a) and (b), most words are shared in the initial embedding space but not well-aligned, hence the oversharing problem. Plots (b) and (c) shows that the vocabulary reallocation step effectively mitigates oversharing while the alignment refinement step further improves the poorly aligned embeddings by projecting them into a close neighborhood.

Lastly, while the *supervised* joint training approach does not have the same issues of unsupervised joint training, it shares limitation 1 of the alignment methods.

We empirically compare both joint training and alignment approaches in Section 5.4 and shed light on some of these pros and cons for both paradigms (See Section 5.4.3).

5.3 Proposed Framework

Motivated by the pros and cons of both paradigms, we propose a unified framework that first uses unsupervised joint training as a coarse initialization and then applies alignment methods for refinement, as demonstrated in Figure 9.1. Specifically, we first build a single set of embeddings with a shared vocabulary through unsupervised joint training, so as to alleviate the limitations of alignment methods. Next, we use a vocabulary reallocation technique to mitigate oversharing, before finally resorting back to alignment methods to further improve the embeddings' quality. Lastly, we show that this framework can generalize to contextualized representations.

5.3.1 Unifying Alignment with Joint Training

Our proposed framework mainly involves three components and we discuss each of them as follows.

Joint Initialization. We use unsupervised joint training (Lample et al., 2018b) to train the initial CLWE. As described in Section 5.2.2, we first obtain a joint vocabulary V_J and train its corresponding set of embeddings E_J on the concatenated corpora of two languages. This allows us to obtain a single set of embeddings that maximizes sharing across two languages. To train embeddings, we used fastText[1] (Bojanowski et al., 2017) in all our experiments for both word and subword tokens.

Vocabulary Reallocation. As discussed in Section 5.2.3, a key issue of unsupervised joint training is oversharing, which prohibits further refinement as shown in Figure 9.1. To alleviate this drawback, we attempt to "unshare" some of the overshared words, so their embeddings can be better aligned in the next step. Particularly, we perform a vocabulary reallocation step such that words appearing mostly exclusively in the ith language are reallocated from the shared vocabulary V_J^s to V_J^i, whereas words that appear similarly frequent in both languages stay still in V_J^s. Formally, for each token w in the shared vocabulary V_J^s, we use the ratio of counts within each language to determine whether it belongs to the shared vocabulary:

$$r = \frac{T_{L_2}}{T_{L_1}} \cdot \frac{C_{L_1}(w)}{C_{L_2}(w)}, \tag{5.5}$$

where $C_{L_i}(w)$ is the count of w in the training corpus of the ith language and $T_{L_i} = \sum_w C_{L_i}(w)$ is the total number of tokens. The token w is allocated to the shared vocabulary if

$$\frac{1-\gamma}{\gamma} \leq r \leq \frac{\gamma}{1-\gamma}, \tag{5.6}$$

where γ is a hyper-parameter. Otherwise, we put w into either V_J^1 or V_J^2, where it appears mostly frequent. The above process generates three new disjoint vocabulary sets $V_J^{1'}, V_J^{2'}, V_J^{s'}$ and their corresponding embeddings $E_J^{1'}, E_J^{2'}, E_J^{s'}$ that are used thereafter. Note that, $V_J^{'} = V_J$ and $E_J^{'} = E_J$.

Alignment Refinement. The unsupervised joint training method does not explicitly utilize any dictionary or form of alignment. Thus, the resulting embedding set is coarse and ill-aligned in the shared vector space, as demonstrated in Figure 9.1. As a final refinement step, we utilize any off-the-shelf alignment method to refine alignments across the non-sharing embedding sets, i.e. mapping $E_J^{1'}$ to $E_J^{2'}$ and leaving $E_J^{s'}$ untouched. This step could be conducted by either the supervised or unsupervised alignment method and we compare both in our experiments.

[1] https://github.com/facebookresearch/fastText

5.3.2 Extension to Contextualized Representations

As our framework is highly generic and applicable to any alignment and unsupervised joint training methods, it can naturally generalize to contextualized word representations by aligning the fixed outputs of a multilingual encoder such as multilingual BERT (M-BERT) (Devlin et al., 2018). While our vocab reallocation technique is no longer necessary as contextualized representations are dependent on context and thus dynamic, we can still apply alignment refinement on extracted contextualized features for further improvement. For instance, as proposed by Aldarmaki and Diab (2019), one method to perform alignment on contextualized representations is to first use word alignment pairs extracted from parallel corpora as a dictionary, learn an alignment matrix W based on it, and apply W back to the extracted representations. To obtain W, we can solve Eq.(5.1) as described in Section 5.2.1, where the embedding matrices X_{L_1} and X_{L_2} now contain contextualized representations of aligned word pairs. Note that this method is applicable to fixed representations but not finetuning.

5.4 Experiments

We evaluate the proposed approach and compare with alignment and joint training methods on three NLP benchmarks. This evaluation aims to: (1) systematically compare alignment vs. joint training paradigms and reveal their pros and cons discussed in Section 5.2.3, (2) show that the proposed framework can effectively alleviate limitations of both alignment and joint training, and (3) demonstrate the effectiveness of the proposed framework in both non-contextualized and contextualized settings.

5.4.1 Evaluation Tasks

Bilingual Lexicon Induction (BLI) This task has been the *de facto* evaluation task for CLWE methods. It considers the problem of retrieving the target language translations of source language words. We use bilingual dictionaries complied by Conneau et al. (2018a) and test on six diverse language pairs, including Chinese and Russian, which use a different writing script than English. Each test set consists of 1500 queries and we report *precision at 1* scores (P@1), following standard evaluation practices (Conneau et al., 2018a; Glavas et al., 2019).

Name Entity Recognition (NER) We also evaluate our proposed framework on cross-lingual NER, a sequence labeling task, where we assign a label to each token in a sequence. We evaluate both non-contextualized and contextualized word representations on the CoNLL 2002 and 2003 benchmarks (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003), which contain 4 European languages. To measure the quality of CLWE, we perform zero-shot cross-lingual classification, where we train a model on English and directly apply it to each of the other 3 languages.

Unsupervised Machine Translation (UMT) Lastly, we test our approach using the unsupervised MT task, on which the initialization of CLWE plays a crucial role (Lample et al., 2018b). Note that our purpose here is to directly compare with similar studies in Lample et al. (2018b), and thus we follow their settings and consider two language pairs, English-French and English-German, and evaluate on the widely used WMT'14 en-fr and WMT'16 en-de benchmarks.

5.4.2 Experimental Setup

For the BLI task, we compare our framework to recent state-of-the-art methods. We obtain numbers from the corresponding papers or Zhou et al. (2019), and use the official tools for MUSE (Conneau et al., 2018a), GeoMM (Jawanpuria et al., 2019) and RCSLS (Joulin et al., 2018) to obtain missing results. We consider the method of Duong et al. (2016) for supervised joint training based on bilingual dictionaries, which is comparable to supervised alignment methods in terms of resources used. For unsupervised joint training, we train uncased joint fastText word vectors of dimension 300 on concatenated Wikipedia corpora of each language pair with default parameters. The hyper-parameter γ is selected from {0.7, 0.8, 0.9, 0.95} on validation sets. For the alignment refinement step in our proposed framework, we use **RCSLS** and **GeoMM** to compare with supervised methods, and **MUSE** for unsupervised methods. In addition, we include an additional baseline of joint training, denoted as **Joint - Replace**, which is identical to unsupervised joint training except that it utilizes a seed dictionary to randomly replace words with their translations in the training corpus. Following standard practices, we consider the top 200k most frequent words and use the cross-domain similarity local scaling (CSLS) (Conneau et al., 2018a) as the retrieval criteria. Note that a concurrent work (Artetxe et al., 2019a) proposed a new retrieval method based on MT systems and produced state-of-the-art results. Although their method is applicable to our framework, it has a high computational cost.

For the NER task: (1) For non-contextualized representations, we train embeddings the same way as in the BLI task and use a vanilla Bi-LSTM-CRF model (Lample et al., 2016). For all alignment steps, we apply the supervised Procrustes method using dictionaries from the MUSE library for simplicity. (2) For contextualized representations, we use M-BERT, an unsupervised

	en-es	es-en	en-fr	fr-en	en-de	de-en	en-it	it-en	en-ru	ru-en	en-zh	zh-en	avg
Alignment Methods													
(1) MUSE	81.7	83.3	82.3	82.1	74.0	72.0	77.7	78.2	44.0	59.1	32.5	31.4	66.5
(2) VECMAP	82.3	84.7	82.3	83.6	75.1	74.3	-	-	49.2	65.6	0.0	0.0	-
(3) DeMa-BWE	82.8	84.9	83.1	83.5	77.2	74.4	-	-	49.2	65.7	42.5	37.9	-
(4) Procrustes	81.4	82.9	81.1	82.4	73.5	72.4	77.5	77.9	51.7	63.7	42.7	36.7	68.7
(5) GeoMM	81.4	85.5	82.1	84.1	74.7	76.7	77.9	80.9	51.3	67.6	49.1	45.3	71.4
(6) RCSLS	84.1	86.3	83.3	84.1	79.1	76.3	78.5	79.8	57.9	67.2	45.9	46.4	72.4
(7) RCSLS + IN	83.9	-	**83.9**	-	78.1	-	79.1	-	57.9	-	48.6	-	-
Joint Traing Methods													
(8) Unsupervised Joint	33.4	36.6	42.2	47.4	39.5	41.4	36.8	38.8	4.0	3.5	17.9	10.2	29.3
(9) Supervised Joint	79.7	79.8	78.1	76.7	67.5	68.9	74.4	74.1	41.8	51.8	46.7	43.3	65.2
(10) Joint - Replace	48.2	47.7	49.4	52.1	46.5	46.9	43.8	45.8	20.3	36.6	32.7	34.1	42.0
Joint Align Framework													
(11) Joint_Align (w/o AR)	55.9	62.8	61.8	67.0	49.1	54.6	50.2	51.4	8.7	8.2	19.4	18.2	42.3
(12) Joint_Align + MUSE	81.4	84.2	82.8	83.6	74.2	72.2	77.5	81.5	45.0	58.3	36.1	35.3	67.7
(13) Joint_Align + RCSLS (w/o VR)	34.2	37.0	41.2	46.8	34.0	35.6	35.3	35.1	7.7	5.2	20.2	15.7	29.0
(14) Joint_Align + GeoMM	82.6	85.7	82.5	84.2	75.5	77.2	78.2	81.4	52.4	67.7	50.4	46.5	72.0
(15) Joint_Align + RCSLS	**84.7**	**87.9**	83.5	**85.6**	**79.6**	**78.0**	**80.6**	**84.0**	**59.8**	**67.8**	**54.3**	**48.7**	**74.5**

Table 5.1: **Precision@1 for the BLI task on the MUSE dataset**. Within each category, unsupervised methods are listed at the top while supervised methods are at the bottom. The best result for unsupervised methods is underlined while **bold** signifies the overall best. “IN” refers to iterative normalization proposed in Zhang et al. (2019), “AR” refers to alignment refinement and “VR” refers to vocabulary reallocation.

joint training model, as our base model and apply our proposed framework on it by first aligning its extracted features and then feeding them to a task-specific model (M-BERT Feature + Align). Specifically, we use the sum of the last 4 M-BERT layers’ outputs as the extracted features. To obtain the alignment matrices, one for each layer, we use 30k parallel sentences from the Europarl corpus for each language pair and follow the procedure of Section 5.3.2. We feed the extracted features as inputs to a task-specific model with 2 Bi-LSTM layers and a CRF layer:

- **Alignment** As described in Section 5.3.2, we apply word alignment methods, such as fastalign (Dyer et al., 2013), on parallel data to extract word-aligned pairs for learning the alignment matrix. As M-BERT is based on subword tokens, we use the average of the representations of all subword tokens that correspond to a word as the representation for that word. For instance, assume that an English word “Resumption” is aligned to a German word “Wiederaufnahme”, and they are tokenized by M-BERT as “Res”, “##sumption”, and “Wie”, “##dera”, “##uf”, “##nahme”, respectively. Then the representation for “Resumption” is the average of the representations of subword tokens “Res” and “##sumption”, and the same goes for “Wiederaufnahme”.

	en-es	es-en	en-fr	fr-en	en-de	de-en	en-it	it-en	en-ru	ru-en	en-zh	zh-en	avg
Unsupervised													
(1) MUSE	77.1	82.5	76.4	78.0	67.4	67.8	72.5	77.5	42.7	50.2	28.7	29.1	62.5
(2) Unsupervised Joint	3.7	10.2	5.1	10.7	8.5	10.5	7.8	8.1	0.4	2.7	2.5	6.4	6.4
(3) Joint_Align + MUSE	77.5	83.0	77.0	79.5	66.7	68.0	70.9	78.0	43.5	55.1	32.3	32.7	63.7
Supervised													
(4) RCSLS	78.0	83.9	76.0	78.6	68.2	68.4	71.8	78.2	50.7	56.9	51.0	41.7	67.0
(5) Supervised Joint	76.8	80.8	73.4	76.1	60.1	61.7	69.7	76.2	41.0	51.8	**52.3**	43.3	63.6
(6) Joint_Align + RCSLS	**82.1**	**84.6**	**78.1**	**80.4**	**68.4**	**70.4**	**73.7**	**79.0**	**59.0**	**66.8**	51.4	**45.7**	**70.0**

Table 5.2: **Precision@1 for the BLI task on the MUSE dataset with test pairs of same surface form removed**. The best result for unsupervised methods is underlined while **bold** signifies the overall best.

- **Hyperparameters** For the task-specific NER model, we use a 2-layer Bi-LSTM with a hidden size of 768 followed by a CRF layer. We apply a dropout rate of 0.5 on the input and the output of the Bi-LSTM, and use Adam with default parameters and a learning rate of 0.0001 for optimization. We train the model for 40 epochs with a batch size of 10, and evaluate the model per 150 steps. For prediction, we feed the outputs of the Bi-LSTM that correspond to the first subword tokens of each word to the CRF model. This is identical to finetuning BERT on the NER task, except that in our case the outputs that correspond to the first subword token are fed into a CRF, rather than a linear layer as done in BERT.

We compare our framework to both finetuning (M-BERT Finetune), which has been studied by previous papers, and feature extraction (M-BERT Feature). Lastly, we also compare against XLM, a supervised joint training model.

For the UMT task, we use the exact same data, architecture and parameters released by Lample et al. (2018b). We simply use different embeddings trained with the same data as inputs to the model.

5.4.3 Results and Analysis

Alignment vs. Joint Training

We compare alignment methods with joint training on all three downstream tasks. As shown in Table 5.1 and Table 5.3, we find alignment methods significantly outperform the joint training approach by a large margin in all language pairs for both BLI and NER. However, the unsupervised joint training method is superior than its alignment counterpart on the unsupervised MT task as demonstrated in 5.2(c). While these results demonstrate that their relative performance is task-dependent, we conduct further analysis to reveal three limitations as discussed in Sec 5.2.3.

First, their poor performance on BLI and NER tasks shows that unsupervised joint training fails to generate high-quality alignments due to the lack of a fine-grained seed dictionary as discussed in its limitation 2. To evaluate accuracy on words that are not shared, we further remove test pairs of the same surface form (e.g. (hate, hate) as a test pair for en-de) of the BLI task and report their results in Table 5.2. We find unsupervised joint training (row 2) to achieve extremely low scores which shows that emebddings of non-shared parts are poorly aligned, consistent with the PCA visualization shown in Figure 9.1.

Moreover, we delve into the relative performance of the two paradigms on the MT task by plotting their test BLEU scores of the first 20 epochs in Figure 5.2(a) and 5.2(b). We observe that the alignment method actually obtains *higher* BLEU scores in the first few epochs, but gets surpassed by joint training in later epochs. This shows the importance of parameter sharing as discussed in limitation 1 of alignment methods: shared words can be used as a cross-lingual constraint for unsupervised joint training during fine-tuning but this constraint cannot easily be used in alignment methods. The lack of sharing is also a limitation for the supervised joint training method, which performs poorly on the MT task even with supervision as shown in Figure 5.2(c).

Lastly, we demonstrate that oversharing can be sub-optimal for unsupervised joint training as discussed in its limitation 2. Specifically, we conduct ablation studies for our framework in Table 5.1. Applying alignment refinement on unsupervised joint training without any vocabulary reallocation does not improve its performance (row 13). On the other hand, simple vocabulary reallocation alone boosts the performance by quite a margin (row 11). This shows some words are shared erroneously across languages in unsupervised joint training, thereby hindering its performance. In addition, while utilizing a seed dictionary (row 10) improves the performance of unsupervised joint training, it still suffers from the oversharing problem and performs worse compared to supervised joint training (row 9).

Evaluation of Proposed Framework

As shown in Table 5.1, Table 5.3, and Figure 5.2, our proposed framework substantially improves over the alignment and joint training baselines on all three tasks. In particular, it outperforms existing methods on all language pairs for the BLI task (using the CSLS as retrieval metric) and achieves state-of-the-art results on 2 out of 3 language pairs for the NER task. Besides, we show that it alleviates limitations of alignment and joint training methods shown in the previous

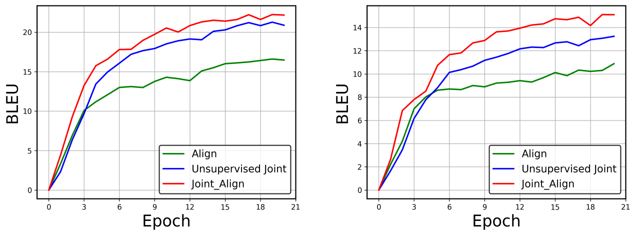
	es	nl	de	avg
Non-contextualized				
Unsupervised Joint	50.28	42.77	21.49	38.18
Supervised Joint	63.16	63.60	36.24	54.33
Joint - Replace	65.28	68.44	51.59	61.77
Align	69.00	71.33	52.17	64.17
Joint_Align	70.46	72.10	56.47	66.34
Xie et al.‡	71.67	70.90	57.43	66.67
Chen et al.‡	73.50	72.40	56.00	67.30
Contextualized				
XLM Finetune *	63.18	-	67.55	-
M-BERT Finetune	73.59	77.36	69.74	73.56
M-BERT Finetune	74.96	77.57	69.56	74.03
M-BERT Finetune	75.00	77.50	68.60	73.70
M-BERT Finetune + Adv	74.30	77.60	**71.90**	74.60
M-BERT Feature	74.23	78.65	67.63	73.50
M-BERT Feature + Align	**75.77**	**79.03**	70.54	**75.11**

Table 5.3: **F1 score for the cross-lingual NER task.** "Adv" refers to adversarial training. ‡ denotes results that are not directly comparable due to different resources and architectures used. * denotes supervised XLM model trained with MLM and TLM objectives. Its Dutch (nl) result is blank because the model is not pretrained on it. **Bold** signifies state-of-the-art results. We report the average of 5 runs.

section.

First, the proposed framework largely improves the poor alignment of unsupervised joint training, especially for non-sharing parts. As shown in Table 5.1, the proposed Joint_Align framework achieves comparable results to prior methods in the unsupervised case (row 12) and it outperforms previous state-of-the-art methods in the supervised setting (row 15). Specifically, our proposed framework can generate well-aligned embeddings after alignment refinement is applied to the initially ill-aligned embeddings, as demonstrated in Figure 9.1. This is further verified by results in Table 5.2, where our proposed framework largely improves accuracy on words not shared between two languages over the unsupervised joint training baseline (row 3 and 6 vs row 2).

Besides, our ablation study in Table 5.1 further shows the effectiveness of the proposed vocabulary reallocation technique, which alleviates the issue of oversharing. Particularly, we observe no improvement compared to unsupervised joint training baseline (row 8) when an alignment refinement step is used without vocabulary reallocation (row 13), while a vocabulary reallocation step alone significantly improves the performance (row 11). This is consistent with



(a) fr-en

(b) en-de

	en-fr	fr-en	en-de	de-en
Unsupervised Joint†	25.14	24.18	17.16	21.00
Align†	22.00	21.30	-	-
Unsupervised Joint	23.68	23.00	15.04	19.21
Align	20.01	19.09	12.86	17.05
Joint_Align	**25.74**	**25.24**	**17.20**	**22.18**
Supervised Joint‡	17.54	16.90	11.02	13.88

(c)

Figure 5.2: (a)(b): **Results on MT of Align, Joint and our framework for the first 20 training epochs.** Results after 20 epochs have similar patterns. (c): **BLEU scores for the MT task.** Results evaluated on the WMT'14 English-French and WMT'16 German-English. All training settings are the same for each language pair except the embedding initialization. Note that we are not trying to outperform state-of-the-art methods (Song et al., 2019) but rather to observe improvements afforded by embedding initialization. †Results reported by (Lample et al., 2018b). Our results are obtained using the official code released by the author. ‡Duong et al. (2016) is a supervised method that we include for analysis purpose only and is not directly comparable to other results in this table.

Figure 9.1 and shows that oversharing is a bottleneck for applying alignment methods to joint training. It also suggests detecting what to share is crucial to achieve better cross-lingual transfer.

Lastly, while supervised joint training shares the limitation 1 of alignment methods and performs poorly when finetuned, our proposed framework exploits the same idea of vocabulary sharing used in unsupervised joint training. In the MT tasks, our framework obtains a maximum gain of 2.97 BLEU over baselines we ran and consistently performs better than results reported in (Lample et al., 2018b). In addition, Figure 5.2 shows that Joint_Align not only converges faster in earlier training epochs but also consistently outperforms the two baselines thereafter. These empirical findings demonstrate the effectiveness of our proposed methods in the non-contextualized case.

Contextualized Word Representations

As can be seen in Table 5.3, when using our framework (M-BERT Feature + Align), we achieve state-of-the-art results on cross-lingual NER on 2 out of 3 languages and the overall average. This shows that our framework can effectively generalize to contextualized representations. Specifically, our framework improves over both the M-BERT feature extraction and finetuning baselines on all three language pairs. However, when compared to non-contextualized results, the gain of using alignment refinement on top of unsupervised joint training is much smaller. This suggests that, as the contextualized unsupervised joint training model performs very well already even without any supervision, it is harder to achieve large improvements. While alignment refinement relies on word alignment, a process that is noisy itself, a better alignment approach may be warranted. Lastly, the reason why a supervised joint training model, XLM, performs worse than its unsupervised counterpart, M-BERT, is likely that XLM uses an uncased vocabulary, where casing information is important for NER tasks.

5.5 Related Work

Word embeddings (Mikolov et al., 2013a; Ruder et al., 2019) are a key ingredient to achieving success in monolingual NLP tasks. However, directly using word embeddings independently trained for each language may cause negative transfer (Wang et al., 2019b) in cross-lingual transfer tasks. In order to capture the cross-lingual mapping, a rich body of existing work relying on cross-lingual supervision, including bilingual dictionaries (Artetxe et al., 2016; Duong et al., 2016; Faruqui and Dyer, 2014; Gouws and Søgaard, 2015; Joulin et al., 2018; Mikolov et al., 2013a; Xing et al., 2015), sentence-aligned corpora (Gouws et al., 2015; Hermann and Blunsom, 2014; Kočiský et al., 2014) and document-aligned corpora (Søgaard et al., 2015; Vulić and Moens, 2016).

Besides, unsupervised alignment methods aim to eliminate the requirement for cross-lingual supervision. Early work of Cao et al. (2016) matches the mean and the standard deviation of two embedding spaces after alignment. Barone (2016); Conneau et al. (2018a); Zhang et al. (2017a,b) adapted a generative adversarial network (GAN) (Goodfellow et al., 2014) to make the distributions of two word embedding spaces indistinguishable. Follow-up works improve upon GAN-based training for better stability and robustness by introducing Sinkhorn distance (Xu et al., 2018), by stochastic self-training (Artetxe et al., 2018a), or by introducing latent variables (Dou et al., 2018).

While alignment methods utilize embeddings trained independently on different languages, joint training methods train word embeddings at the same time. Klementiev et al. (2012) train a bilingual dictionary-based regularization term jointly with monolingual language model objectives while Kočiský et al. (2014) defines the cross-lingual regularization with the parallel corpus. Another branch of methods (Ammar et al., 2016; Duong et al., 2016; Gouws and Søgaard, 2015; Xiao and Guo, 2014) build a pseudo-bilingual corpus by randomly replacing words in monolingual corpus with their translations and use monolingual word embedding algorithms to induce bilingual representations. The unsupervised joint method by Lample and Conneau (2019) simply exploit words that share the same surface form as bilingual "supervision" and directly train a shared set of embedding with joint vocabulary. Recently, unsupervised joint training of contextualized word embeddings through the form of multilingual language model pretraining using shared subword vocabularies has produced state-of-the-art results on various benchmarks (Artetxe and Schwenk, 2019; Devlin et al., 2018; Lample and Conneau, 2019; Pires et al., 2019; Wu and Dredze, 2019).

A concurrent work by Ormazabal et al. (2019) also compares alignment and joint method in the bilingual lexicon induction task. Different from their setup which only tests on supervised settings, we conduct analysis across various tasks and experiment with both supervised and unsupervised conditions. While (Ormazabal et al., 2019) suggests that the combination of the alignment and joint model could potentially advance the state-of-art of both worlds, we propose such a framework and empirically verify its effectiveness on various tasks and settings.

5.6 Summary

In this chapter, we systematically compare the alignment and joint training methods for CLWE. We point out that the nature of each category of methods leads to certain strengths and limitations. The empirical experiments on extensive benchmark datasets and various NLP tasks verified our analysis. To further improve the state-of-art of CLWE, we propose a simple hybrid framework that combines the strength from both worlds and achieves significantly better performance in the BLI, MT and NER tasks. Our work opens a promising new direction that combines two previously exclusive lines of research. The method proposed in this chapter is post-hoc and improves cross-lingual generalization; in the next chapter, we study how to encourage better alignment during the optimization process.

Chapter 6

Gradient Alignment

This chapter presents aligning gradients among tasks. Unlike previous chapter which align tasks after the training procedure, the method proposed in this chapter align gradients adaptively during the optimization process. Massively multilingual models subsuming tens or even hundreds of languages pose great challenges to multi-task optimization. While it is a common practice to apply a language-agnostic procedure optimizing a joint multilingual task objective, how to properly characterize and take advantage of its underlying problem structure for improving optimization efficiency remains under-explored. In this chapter, we attempt to peek into the black-box of multilingual optimization through the lens of loss function geometry. We find that gradient similarity measured along the optimization trajectory is an important signal, which correlates well with not only language proximity but also the overall model performance. Thus, we propose two simple algorithms to encourage more geometrically aligned parameter updates for close tasks, namely Gradient Vaccine and Meta Smoothing. Empirically, our method obtains significant model performance gains on multilingual machine translation and XTREME benchmark tasks for multilingual language models. Our work reveals the importance of properly measuring and utilizing language proximity in multilingual optimization, and has broader implications for multi-task learning beyond multilingual modeling.

6.1 Introduction

Modern multilingual methods, such as multilingual language models (Conneau et al., 2020; Devlin et al., 2018; Lample and Conneau, 2019) and multilingual neural machine translation (NMT) (Aharoni et al., 2019; Arivazhagan et al., 2019; Firat et al., 2016; Johnson et al., 2017), have been showing success in processing tens or hundreds of languages simultaneously in a single large model. These models are appealing for two reasons: (1) Efficiency: training and deploying a single multilingual model requires much less resources than maintaining one model for each language considered, (2) Positive cross-lingual transfer: by transferring knowledge from high-resource languages (HRL), multilingual models are able to improve performance on low-resource languages (LRL) on a wide variety of tasks (Hu et al., 2020; Pires et al., 2019; Siddhant et al., 2020; Wu and Dredze, 2019).

Despite their efficacy, how to properly analyze or improve the optimization procedure of multilingual models remains under-explored. In particular, multilingual models are *multi-task learning (MTL)* (Ruder, 2017) in nature but existing literature often train them in a monolithic manner, naively using a single language-agnostic objective on the concatenated corpus of many languages. While this approach ignores task relatedness and might induce *negative interference* (Wang et al., 2020c), its optimization process also remains a black-box, muffling the interaction among different languages during training and the cross-lingual transferring mechanism.

In this chapter, we attempt to open the multilingual optimization black-box via the analysis of loss geometry. Specifically, we aim to answer the following questions: (1) Do typologically similar languages enjoy more similar loss geometries in the optimization process of multilingual models? (2) If so, in the joint training procedure, do more similar gradient trajectories imply less interference between tasks, hence leading to better model quality? (3) Lastly, can we deliberately encourage more geometrically aligned parameter updates to improve multi-task optimization, especially in real-world massively multilingual models that contain heavily noisy and unbalanced training data?

Towards this end, we perform a comprehensive study on massively multilingual neural machine translation tasks, where each language pair is considered as a separate task. We first study the correlation between language and loss geometry similarities, characterized by gradient similarity along the optimization trajectory. We investigate how they evolve throughout the whole training process, and glean insights on how they correlate with cross-lingual transfer and joint performance. In particular, our experiments reveal that gradient similarities across tasks correlate strongly with both language proximities and model performance, and thus we observe that typologically close languages share similar gradients that would further lead to well-aligned multilingual structure (Wu et al., 2020) and successful cross-lingual transfer. Based on these findings, we identify a major limitation of a popular multi-task learning method (Yu et al., 2020) applied in multilingual models and propose a *preemptive* method, **Gradient Vaccine**, that leverages task relatedness to set gradient similarity objectives and adaptively align task gradients to achieve such objectives. Empirically, our approach obtains significant performance gain over the

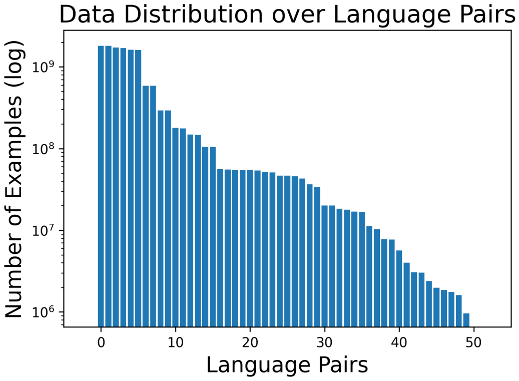


Figure 6.1: Per language pair data distribution of the dataset used to train our multilingual model. The yaxis depicts the number of training examples available per language pair on a logarithmic scale.

standard monolithic optimization strategy and popular multi-task baselines on large-scale multilingual NMT models and multilingual language models. To the best of our knowledge, we are the first to systematically study and improve loss geometries in multilingual optimization at scale.

6.2 Investigating Multi-task Optimization in Massively Multilingual Models

While prior work have studied the effect of data (Arivazhagan et al., 2019; Wang et al., 2020b), architecture (Blackwood et al., 2018; Escolano et al., 2020; Sachan and Neubig, 2018; Vázquez et al., 2019) and scale (Huang et al., 2019c; Lepikhin et al., 2020) on multilingual models, their optimization dynamics are not well understood. We hereby perform a series of control experiments on massively multilingual NMT models to investigate how gradients interact in multilingual settings and what are their impacts on model performance, as existing work hypothesizes that gradient conflicts, defined as negative cosine similarity between gradients, can be detrimental for multi-task learning (Yu et al., 2020) and cause negative transfer (Wang et al., 2019b).

6.2.1 Experimental Setup

For training multilingual machine translation models, we mainly follow the setup in Arivazhagan et al. (2019). In particular, we jointly train multiple translation language pairs in a single sequence-to-sequence (seq2seq) model (Sutskever et al., 2014). We use the Transformer-Big

Language	Id	Language Family	Data Size	Language	Id	Language Family	Data Size
French	fr	Western European	High	Finnish	fi	Uralic	High
Spanish	es	Western European	High	Hindi	hi	Indo-Iranian	Medium
German	de	Western European	High	Marathi	mr	Indo-Iranian	Ex-Low
Polish	pl	West Slavic	High	Gujarati	gu	Indo-Iranian	Low
Czech	cs	West Slavic	High	Nepali	ne	Indo-Iranian	Ex-Low
Macedonian	mk	South Slavic	Low	Kazakh	kk	Turkic	Low
Bulgarian	bg	South Slavic	Medium	Kyrgyz	ky	Turkic	Ex-Low
Ukrainian	uk	East Slavic	Medium	Swahili	sw	Benue-Congo	Low
Belarusian	be	East Slavic	Low	Zulu	zu	Benue-Congo	Ex-Low
Russian	ru	East Slavic	High	Xhosa	xh	Benue-Congo	Ex-Low
Latvian	lv	Baltic	Medium	Indonesian	id	Malayo-Polynesian	High
Lithuanian	lt	Baltic	Medium	Malay	ms	Malayo-Polynesian	Medium
Estonian	et	Uralic	Medium				

Table 6.1: Details of all languages considered in our dataset. Notice that since German (Germanic) is particularly similar to French and Spanish (Romance), we consider a larger language branch for them named "Western European". "Ex-Low" indicates extremely low-resource languages in our dataset. We use BCP-47 language codes as labels (Phillips and Davis, 2006).

(Vaswani et al., 2017) architecture containing 375M parameters described in (Chen et al., 2018a), where all parameters are shared across language pairs. We use an effective batch sizes of 500k tokens, and utilize data parallelism to train all models over 64 TPUv3 chips. Sentences are encoded using a shared source-target Sentence Piece Model (Kudo and Richardson, 2018) with 64k tokens, and a <2xx> token is prepended to the source sentence to indicate the target language (Johnson et al., 2017).

To train each model, we use a single Adam optimizer (Kingma and Ba, 2014) with default decay hyper-parameters. We warm up linearly for 30K steps to a learning rate of 1e-3, which is then decayed with the inverse square root of the number of training steps after warm-up. At each training step, we sample from all language pairs according to a temperature based sampling strategy as in prior work (Arivazhagan et al., 2019; Lample and Conneau, 2019). That is, at each training step, we sample each sentence from all language pairs to train proportionally to $P_i = (\frac{L_i}{\sum_j L_j})^{\frac{1}{T}}$, where L_i is the size of the training corpus for language pair i and T is the temperature. We set T=5 for most of our experiments.

[1] Western European includes Romance and Germanic.

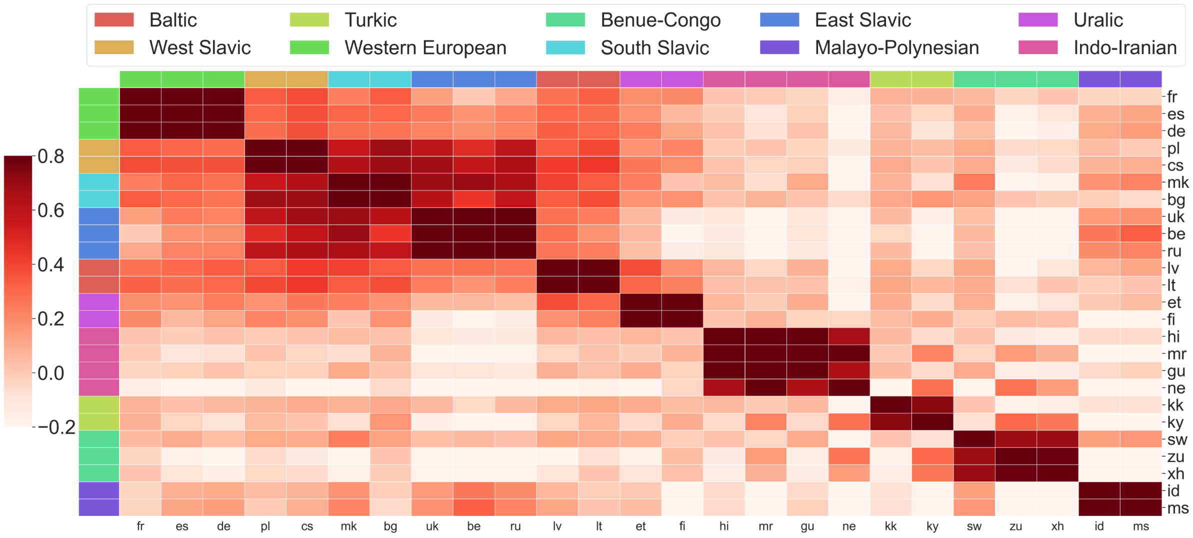
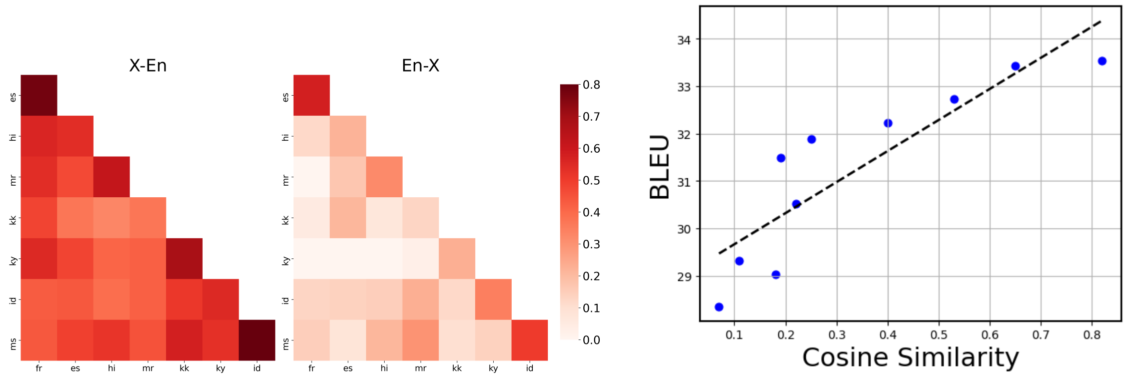


Figure 6.2: Cosine similarities of encoder gradients between *xx-en* language pairs averaged across all training steps. Darker cell indicates pair-wise gradients are more similar. Best viewed in color.[1]

To study real-world multi-task optimization on a massive scale, we use an in-house training corpus (Arivazhagan et al., 2019) generated by crawling and extracting parallel sentences from the web (Uszkoreit et al., 2010), which contains more than 25 billion sentence pairs for 102 languages to and from English. We select 25 languages (50 language pairs pivoted on English), containing over 8 billion sentence pairs, from 10 diverse language families and 4 different levels of data sizes. We then train two models on two directions separately, namely *Any→En* and *En→Any*. Furthermore, to minimize the confounding factors of inconsistent sentence semantics across language pairs, we create a multi-way aligned evaluation set of 3k sentences for all languages[2]. Then, for each checkpoint at an interval of 1000 training steps, we measure pair-wise cosine similarities of the model's gradients on this dataset between all language pairs. We examine gradient similarities at various granularities, from specific layers to the entire model.

We pick languages that belong to different language families (typologically diverse) and with various levels of training data sizes. Specifically, we consider the following languages and their details are listed in 6.1: French (fr), Spanish (es), German (de), Polish (pl), Czech (cs), Macedonian (mk), Bulgarian (bg), Ukrainian (uk), Belarusian (be), Russian (ru), Latvian (lv), Lithuanian (lt), Estonian (et), Finnish (fi), Hindi (hi), Marathi (mr), Gujarati (gu), Nepali (ne), Kazakh (kk),

[2] In other words, 3k semantically identical sentences are given in 25 languages.



(a)

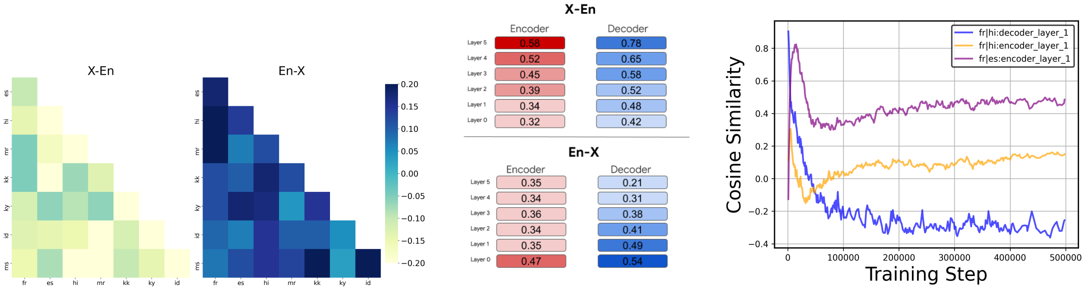
(b)

Figure 6.3: Comparing gradient similarity versus model performance. **(a):** Similarity of model gradients between *xx-en* (left) and *en-xx* (right) language pairs in a single *Any*→*Any* model. **(b):** BLEU scores on *en-fr* of a set of trilingual models versus their gradient similarities. Each model is trained on *en-fr* and another *en-xx* language pair.

Kyrgyz (ky), Swahili (sw), Zulu (zu), Xhosa (xh), Indonesian (id), Malay (ms).

Our corpus has languages belonging to a wide variety of scripts and linguistic families. The selected 25 languages belong to 10 different language families (e.g. Turkic versus Uralic) or branches within language family (e.g. East Slavic versus West Slavic), as indicated in Table 6.1. Families are groups of languages believed to share a common ancestor, and therefore tend to have similar vocabulary and grammatical constructs. We therefore utilize membership of language family to define language proximity.

In addition, our language pairs have different levels of training data, ranging from 10^5 to 10^9 sentence pairs. This is shown in Figure 6.1. We therefore have four levels of data sizes (number of languages in parenthesis): High (7), Medium (8), Low (5), and Extremely Low (5). In particular, we consider tasks with more than 10^8 to be high-resource, $10^7 - 10^8$ to be medium-resource, and rest to be low-resource (with those below 5 million sentence pairs to be extremely low-resource). Therefore, our dataset is both heavily unbalanced and noisy, as it is crawled from the web, and thus introduces optimization challenges from a multi-task training perspective. These characteristics of our dataset make the problem that we study as realistic as possible.



X-En

	Encoder	Decoder
Layer 5	0.58	0.78
Layer 4	0.52	0.65
Layer 3	0.45	0.58
Layer 2	0.39	0.52
Layer 1	0.34	0.48
Layer 0	0.32	0.42

En-X

	Encoder	Decoder
Layer 5	0.35	0.21
Layer 4	0.34	0.31
Layer 3	0.36	0.38
Layer 2	0.34	0.41
Layer 1	0.35	0.49
Layer 0	0.47	0.54

(a) (b) (c)

Figure 6.4: Evaluating gradient similarity across model architecture and training steps. **(a):** Difference between gradient similarities in the encoder and decoder. Positive value (darker) indicates the encoder has more similar gradient similarities. **(b):** Gradient similarities across layers. **(c):** Gradient similarities of different components and tasks across training steps.

6.2.2 Observations

We make the following three main observations. Our findings are consistent across different model architectures and settings.

1. **Gradient similarities reflect language proximities.** We first examine if close tasks enjoy similar loss geometries and vice versa. Here, we use language proximity (defined according to their memberships in a linguistic language family) to control task similarity, and utilize gradient similarity to measure loss geometry. In Figure 6.2, we use a symmetric heatmap to visualize pair-wise gradient similarities, averaged across all checkpoints at different training steps. Specifically, we observe strong clustering by membership closeness in the linguistic family, along the diagonal of the gradient similarity matrix. In addition, all European languages form a large cluster in the upper-left corner, with an even smaller fine-grained cluster of Slavic languages inside. Furthermore, we also observe similarities for Western European languages gradually decrease in West Slavic→South Slavic→East Slavic, illustrating the gradual continuum of language proximity.

2. **Gradient similarities correlate positively with model quality.** As gradient similarities correlate well with task proximities, it is natural to ask whether higher gradient similarities lead to better multi-task performance. In Figure 6.3(a), we train a joint model of all language pairs in both *En*→*Any* and *Any*→*En* directions, and compare gradient similarities between these two. While prior work has shown that *En*→*Any* is harder and less amenable for positive

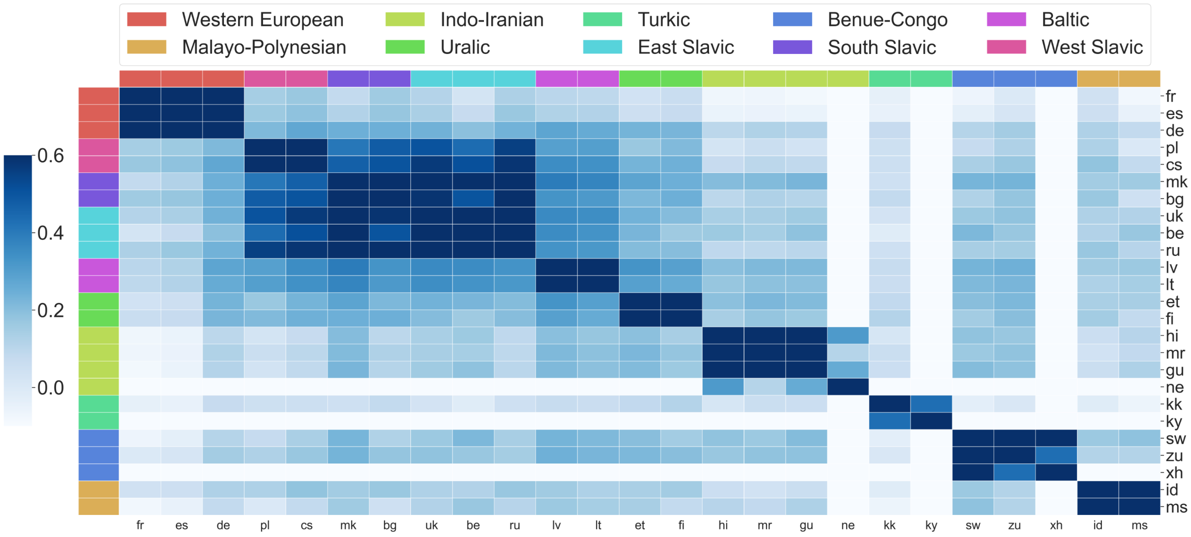


Figure 6.5: Cosine similarities of decoder gradients between *en-xx* language pairs averaged across all training steps. Darker cell indicates pair-wise gradients are more similar.

transfer (Arivazhagan et al., 2019), we find that gradients of tasks in *En*→*Any* are indeed less similar than those in *Any*→*En*. On the other hand, while larger batch sizes often improve model quality, we observe that models trained with smaller batches have less similar loss geometries. These all indicate that gradient interference poses a great challenge to the learning procedure.

To further verify this, we pair En→Fr with different language pairs (e.g. En→Es or En→Hi), and train a set of models with exactly two language pairs[3]. We then evaluate their performance on the En→Fr test set, and compare their BLEU scores versus gradient similarities between paired two tasks. As shown in Figure 6.3(b), gradient similarities correlate positively with model performance, again demonstrating that dissimilar gradients introduce interference and undermine model quality.

3. **Gradient similarities evolve across layers and training steps.** While the previous discussion focuses on the gradient similarity of the whole model averaged over all checkpoints, we now study it across different layers and training steps. Figure 6.4(c) shows the evolution of the gradient similarities throughout the training. Interestingly, we observe diverse patterns for different gradient subsets. For instance, gradients between En→Fr and En→Hi gradually become less similar (from positive to negative) in layer 1 of the decoder but more similar (from

[3] To remove confounding factors, we fix the same sampling strategy for all these models.

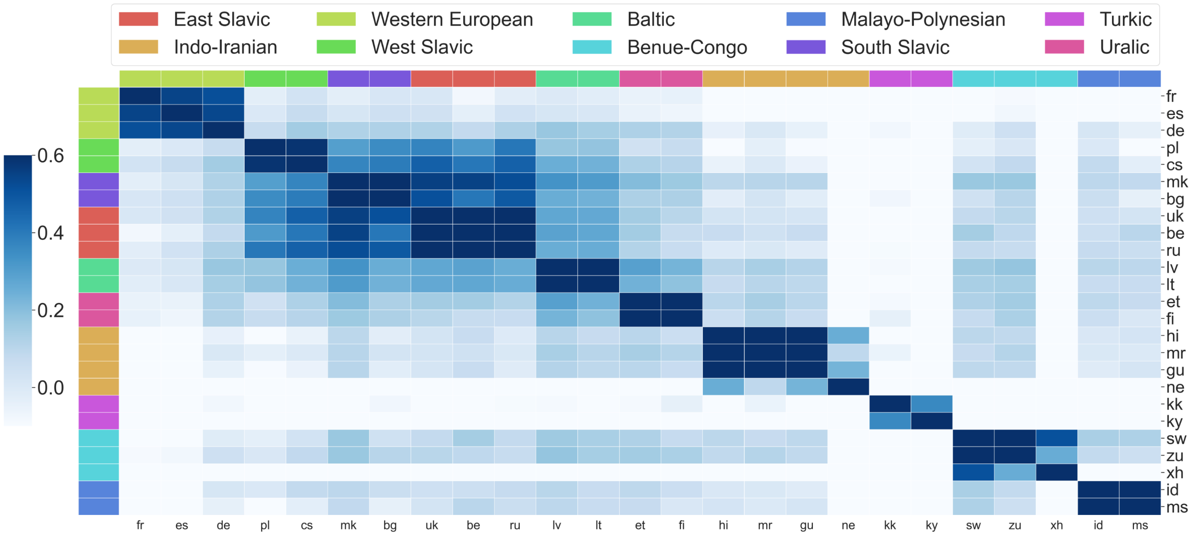


Figure 6.6: Cosine similarities of decoder gradients between *en-xx* language pairs averaged across all training steps. Darker cell indicates pair-wise gradients are more similar. Model trained with smaller batch sizes.

negative to positive) in the encoder of the same layer. On the other hand, gradient similarities between En→Fr and En→Es are always higher than those between En→Fr and En→Hi in the same layer, consistent with prior observation that gradients reflect language similarities.

In addition, we evaluate the difference between gradient similarities in the multilingual encoder and decoder in Figure 6.4(a). We find that the gradients are more similar in the decoder (positive values) for the *Any→En* direction but less similar (negative values) for the *En→Any* direction. This is in line with our intuition that gradients should be more consistent when the decoder only needs to handle one single language. Moreover, we visualize how gradient similarities evolve across layers in Figure 6.4(b). We notice that similarity between gradients increase/decrease as we move up from bottom to top layers for the *Any→En*/*En→Any* direction, and hypothesize that this is due to the difference in label space (English-only tokens versus tokens from many languages). These results demonstrate that the dynamics of gradients evolve over model layers and training time.

Furthermore, in Figure 6.2 we show visualization on models trained using *Any→En* language pairs. Here, we also examine models trained in the other direction, *En→Any*. As shown in Figure 6.5, we have similar observations made in Section 6.2 such that gradient similarities cluster strongly by language proximities. However, the *en-xx* model has smaller scales in cosine

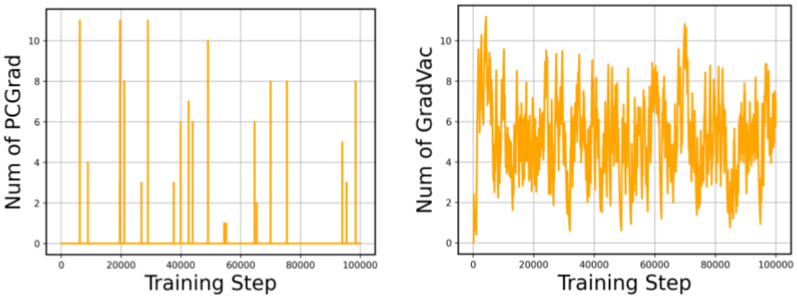


Figure 6.7: Counts of active PCGrad (left) and GradVac (right) during the training process.

similarities and more negative values. For example, Nepali shares mostly conflicting gradients with other languages, except for those belonging to the same language family. This is in line with our above discussion that gradient interference may be a source of optimization challenge, such that the *en-xx* model is harder to train than the *xx-en* model. Moreover, while our previous models are trained using a large batch size for better performance (as observed in prior work (Arivazhagan et al., 2019)), we also evaluate gradients in a model trained with smaller batches (125k tokens) in Figure 6.6. Compared to model trained with larger batch sizes, this model enjoy similar patterns but with smaller gradient cosine similarity values, indicating that gradients are less similar. This presents an additional potential explanation of why larger batch sizes can be more effective for training large models: they may better reflect the correct loss geometries such that gradients are less conflicting in nature. For our case, this means larger batches better reflect language proximities hence gradients of better quality. Finally, these results also reveal that gradient similarities are mostly dependent on task relatedness, as even sentence pairs with identical semantic meanings can have negative cosine similarities due to language differences.

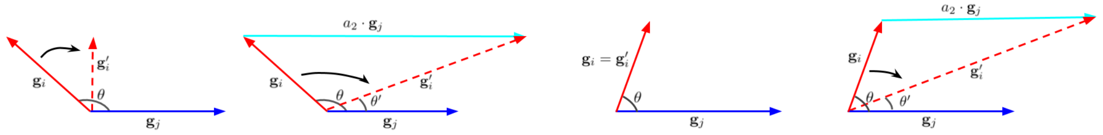
Our analysis highlights the important role of loss geometries in multilingual models. With these points in mind, we next turn to the problem of how to improve multi-task optimization in multilingual models in a systematic way.

6.3 Proposed Method

Following our observations that inter-task loss geometries correlate well with language similarities and model quality, a natural question to ask next is how we can take advantage of such gradient dynamics and design optimization procedures superior to the standard monolithic practice. Since we train large-scale models on real-world dataset consisting of billions of words, of which tasks are highly unbalanced and exhibit complex interactions, we propose an effective ap-

Algorithm 2 GradVac Update Rule

```
**Require:** EMA decay $\beta$, Model Components $\mathcal{M} = \{\boldsymbol{\theta}_k\}$, Tasks for GradVac $\mathcal{G} = \{\mathcal{T}_i\}$
Initialize model parameters
Initialize EMA variables $\hat{\phi}_{ijk}^{(0)} = 0, \forall i, j, k$
Initialize time step $t = 0$
**while** not converged **do**
    Sample minibatch of tasks $\mathcal{B} = \{\mathcal{T}_i\}$
    **for** $\boldsymbol{\theta}_k \in \mathcal{M}$ **do**
        Compute gradients $\mathbf{g}_{ik} \leftarrow \nabla_{\boldsymbol{\theta}_k} \mathcal{L}_{\mathcal{T}_i}, \forall \mathcal{T}_i \in \mathcal{B}$
        Set $\mathbf{g}'_{ik} \leftarrow \mathbf{g}_{ik}$
        **for** $\mathcal{T}_i \in \mathcal{G} \cap \mathcal{B}$ **do**
            **for** $\mathcal{T}_j \in \mathcal{B} \setminus \mathcal{T}_i$ in random order **do**
                Compute $\phi_{ijk}^{(t)} \leftarrow \frac{\mathbf{g}'_{ik} \cdot \mathbf{g}_{jk}}{\|\mathbf{g}'_{ik}\| \|\mathbf{g}_{jk}\|}$
                **if** $\phi_{ijk}^{(t)} < \hat{\phi}_{ijk}^{(t)}$ **then**
                    Set $\mathbf{g}'_{ik} = \mathbf{g}'_{ik} + \frac{\|\mathbf{g}'_{ik}\|(\hat{\phi}_{ijk}^{(t)}\sqrt{1-(\phi_{ijk}^{(t)})^2} - \phi_{ijk}^{(t)}\sqrt{1-(\hat{\phi}_{ijk}^{(t)})^2})}{\|\mathbf{g}_{jk}\|\sqrt{1-(\hat{\phi}_{ijk}^{(t)})^2}} \cdot \mathbf{g}_{jk}$
                **end if**
                Update $\hat{\phi}_{ijk}^{(t+1)} = (1-\beta)\hat{\phi}_{ijk}^{(t)} + \beta\phi_{ijk}^{(t)}$
            **end for**
        **end for**
        Update $\boldsymbol{\theta}_k$ with gradient $\sum \mathbf{g}'_{ik}$
    **end for**
    Update $t \leftarrow t + 1$
**end while**=0
```



(a)

(b)

Figure 6.8: Comparing PCGrad (left) with GradVac (right) in two cases. **(a):** For negative similarity, both methods are effective but GradVac can utilize adaptive objectives between different tasks. **(b):** For positive similarity, only GradVac is active while PCGrad stays “idle".

proach that not only exploits inter-task structures but also is applicable to unbalanced tasks and noisy data. To motivate our method, we first review a state-of-the-art multi-task learning method and show how the observation in Section 6.2 helps us to identify its limitation.

6.3.1 Gradient Surgery

An existing line of work (Chen et al., 2018b; Sener and Koltun, 2018; Yu et al., 2020) has successfully utilized gradient-based techniques to improve multi-task models. Notably, Yu et al. (2020) hypothesizes that negative cosine similarities between gradients are detrimental for multi-task optimization and proposes a method to directly *project conflicting gradients (PCGrad)*, also known as the Gradient Surgery. As illustrated in the left side of Figure 6.8(a), the idea is to first detect gradient conflicts and then perform a "surgery" to deconflict them if needed. Specifically, for gradients $\mathbf{g}_i$ and $\mathbf{g}_j$ of the i-th and j-th task respectively at a specific training step, PCGrad (1) computes their cosine similarity to determine if they are conflicting, and (2) if the value is negative, projects $\mathbf{g}_i$ onto the normal plane of $\mathbf{g}_j$ as:

$$\mathbf{g}_i' = \mathbf{g}_i - \frac{\mathbf{g}_i \cdot \mathbf{g}_j}{\|\mathbf{g}_j\|^2}\mathbf{g}_j. \tag{6.1}$$

The altered gradient $\mathbf{g}_i'$ replaces the original $\mathbf{g}_i$ and this whole process is repeated across all tasks in a random order. For more details and theoretical analysis, we refer readers to the original work.

Now, we can also interpret PCGrad from a different perspective: notice that the gradient cosine similarity will always be zero after the projection, effectively setting a target lower bound. In other words, PCGrad aims to align gradients to match a certain gradient similarity level, and implicitly makes the assumption that *any two tasks must have the same gradient similarity objective of zero*. However, as we shown in Section 6.2, different language proximities would result in diverse gradient similarities. In fact, many language pairs in our model share positive cosine similarities such that the pre-condition for PCGrad would never be satisfied. This is shown in the left of Figure 6.8(b), where PCGrad is not effective for positive gradient similarities and thus it is very sparse during training in the left of Figure 6.7. Motivated by this limitation, we next present our proposed method.

6.3.2 Gradient Vaccine

The limitation of PCGrad comes from the unnecessary assumption that all tasks must enjoy similar gradient interactions, ignoring complex inter-task relationships. To relax this assumption, a natural idea is to set adaptive gradient similarity objectives in some proper manner. An example is shown in the right of Figure 6.8(b), where two tasks have a positive gradient similarity of $\cos(\theta) = \phi_{ij}$. While PCGrad ignores such non-negative cases, the current value of ϕ_{ij} may still be detrimentally low for more similar tasks such as French versus Spanish. Thus, suppose we have some similarity goal of $\cos(\theta') = \phi_{ij}^T > \phi_{ij}$ (e.g. the "normal" cosine similarity between these two tasks), we alter both the magnitude and direction of $\mathbf{g}_i$ such that the resulting gradients match such gradient similarity objective. In particular, we replace g_i with a vector that satisfies such condition in the vector space spanned by $\mathbf{g}_i$ and $\mathbf{g}_j$, i.e. $a_1 \cdot \mathbf{g}_i + a_2 \cdot \mathbf{g}_j$. Since there are infinite numbers of valid combinations of a_1 and a_2, for simplicity, we fix $a_1 = 1$ and by applying Law of Sines in the plane of $\mathbf{g}_i$ and $\mathbf{g}_j$, we solve for the value of a_2 and derive the new gradient for the i-th task as:

$$\mathbf{g}_i' = \mathbf{g}_i + \frac{\|\mathbf{g}_i\|(\phi_{ij}^T\sqrt{1-\phi_{ij}^2} - \phi_{ij}\sqrt{1-(\phi_{ij}^T)^2})}{\|\mathbf{g}_j\|\sqrt{1-(\phi_{ij}^T)^2}} \cdot \mathbf{g}_j. \tag{6.2}$$

This formulation allows us to use arbitrary gradient similarity objective ϕ_{ij}^T in $[-1, 1]$. The remaining question is how to set such objective properly. In the above analysis, we have seen that gradient interactions change drastically across tasks, layers and training steps. To incorporate these three factors, we exploit an exponential moving average (EMA) variable for tasks i, j and parameter group k (e.g. the k-th layer) as:

$$\hat{\phi}_{ijk}^{(t)} = (1-\beta)\hat{\phi}_{ijk}^{(t-1)} + \beta\phi_{ijk}^{(t)}, \tag{6.3}$$

where $\phi_{ijk}^{(t)}$ is the computed gradient similarity at training step t, β is a hyper-parameter, and $\hat{\phi}_{ijk}^{(0)} = 0$. The full method is outlined in Algorithm 2. Notice that gradient surgery is a special case of our proposed method such that $\phi_{ij}^T = 0$. As shown in the right of Figure 6.8(a) and 6.8(b), our method alters gradients more *preemptively* under both positive and negative cases, taking more proactive measurements in updating the gradients (Figure 6.7). We therefore refer to it as **Gradient Vaccine (GradVac)**.

6.4 Experiments

We compare multi-task optimization methods with the monolithic approach in multilingual settings, and examine the effectiveness of our proposed method on multilingual NMT and multilingual language models.

	en-fr	en-cs	en-hi	en-tr	avg
GradVac w. HRL_only	39.07	21.51	14.92	19.63	23.78
GradVac w. LRL_only	39.27	21.67	14.88	19.73	23.89
GradVac w. all_task	38.85	21.47	14.48	19.75	23.64

Table 6.2: Comparing which tasks to be included for GradVac. Parameter granularity fixed at all_layer while β=1e-2.

	en-fr	en-cs	en-hi	en-tr	avg
GradVac w. whole_model	38.76	21.32	14.22	18.89	23.30
GradVac w. enc_dec	39.05	21.73	14.54	19.33	23.66
GradVac w. all_layer	39.27	21.67	14.88	19.73	23.89
GradVac w. all_matrix	38.95	21.56	14.57	19.01	23.52

Table 6.3: Comparing parameter granularity for GradVac. GradVac tasks fixed at LRL_only while β=1e-2.

	en-fr	en-cs	en-hi	en-tr	avg
GradVac w. β=1e-1	38.72	20.74	14.52	19.25	23.31
GradVac w. β=1e-2	39.27	21.67	14.88	19.73	23.89
GradVac w. β=1e-3	38.85	20.96	14.85	19.68	23.59

Table 6.4: Comparing EMA decay rate β for GradVac. Parameter granularity fixed at all_layer and GradVac tasks fixed at LRL_only.

6.4.1 General Setup

We choose three popular scalable gradient-based multi-task optimization methods as our baselnes: **GradNorm** (Chen et al., 2018b), **MGDA** (Sener and Koltun, 2018), and **PCGrad** (Yu et al., 2020). For fair comparison, language-specifc gradients are computed for samples in each batch. The sampling temperature is also fixed at T=5 unless otherwise stated. For the baselines, we mainly follow the default settings and training procedures for hype-parameter selection as explained in their respective papers. For our method, to study how sensitive GradVac is to the distribution of tasks, we additionally examine a variant that allows us to control which languages are considered for GradVac. Specifically, we search the following hyper-parameters on small-scale WMT dataset and transfer to our large-scale dataset: tasks considered for GradVac {HRL_only, LRL_only, all_task}, parameter granularity {whole_model, enc_dec, all_layer, all_matrix}, EMA decay rate β {1e-1, 1e-2, 1e-3}.

First, we examine the effect of what tasks to include for GradVac, i.e. $\mathcal{G}$ in Algorithm 2. We consider three options: (1) HRL_only: only perform GradVac on high-resource languages, (2) LRL_only: only perform GradVac on low-resource languages, (3) all_task: perform GradVac on all languages. Results are shown in Table 6.2. We find that only conducting GradVac on a subset of languages obtain better performance while it is the best to conduct GradVac on low-resource language only. This is probably because the effective batch sizes of low-resource languages are usually smaller due to the sampling strategy.

Next, we compare the effect of parameter granularity on model quality. This corresponds to setting different model components for GradVac ($\mathcal{M}$ in Algorithm 2). We consider four possibilities, from coarse to fine-grained: (1) whole_model: only perform GradVac once on the entire model, (2) enc_dec: perform separately for encoder and decoder, (3) all_layer: perform individually for each layer in encoder and decoder, (4) all_matrix: perform for each parameter matrix in the model. As shown in Table 6.3, we find that choosing proper parameter granularity is important, as neither too coarse nor too fine-grained perform the best.

Finally, we study how sensitive our method is on the hyper-parameter β, i.e. the EMA decay rate. Results in Table 6.4 illustrate that setting an effective "window" of 100 training steps work best for our problem setups. This is expected, as setting a larger β value corresponds to conduct GradVac more aggressively, and vice versa. In general, wWe find {LRL_only, all_layer, 1e-2} to work generally well and use these in the following experiments.

6.4.2 Results and Analysis

WMT Machine Translation. We first conduct comprehensive analysis of our method and other baselines on a small-scale WMT task. We consider two high-resource languages (WMT14 en-fr, WMT19 en-cs) and two low-resource languages (WMT14 en-hi, WMT18 en-tr), and train two models for both to and from English. Results are shown in Table 6.5.

First, we observe that while the naive multilingual baseline outperforms bilingual models on low-resource languages, it performs worse on high-resource languages due to negative interference (Wang et al., 2020c) and constrained capacity (Arivazhagan et al., 2019). Existing baselines fail to address this problem properly, as they obtain marginal or even no improvement (row 3, 4 and 5). In particular, we look closer at the optimization process for methods that utilize gradient signals to reweight tasks, i.e. GradNorm and MGDA, and find that their computed weights are less meaningful and noisy. For example, MGDA assigns larger weight for en-fr in the en-xx model, that results in worse performance on other languages. This is mainly because these methods are designed under the assumption that all tasks have balanced data. Our results show that simply reweighting task weights without considering the loss geometry has limited efficacy.

By contrast, our method significantly outperforms all baselines. Compared to the naive joint training approach, the proposed method improves over not only the average BLEU score but also the individual performance on all tasks. We notice that the performance gain on *En→Any* is larger compared to *Any→En*. This is in line with our prior observation that gradients are less similar and more conflicting in *En→Any* directions.

We next conduct extensive ablation studies for deeper analysis: (1) GradVac applied to all layers vs. whole model (row 8 vs. 9): the all_layer variant outperforms whole_model, showing that setting fine-grained parameter objectives is important. (2) Constant objective vs. EMA (row 7 vs. 9): we also examine a variant of GradVac optimized using a constant gradient objective for all tasks (e.g. $\phi_{ij}^T = 0.5, \forall i, j$) and observe performance drop compared to using EMA variables. This highlights the importance of setting task-aware objectives through task relatedness. (3) GradVac vs. PCGrad (row 8-9 vs. 5-6): the two GradVac variants outperform their PCGrad counterparts, validating the effectiveness of setting preemptive gradient similarity objectives.

Massively Multilingual Machine Translation. We then scale up our experiments and transfer the best setting found on WMT to the same massive dataset used in Section 6.2. We visualize model performance in Figure 6.9 and average BLEU scores are shown in Table 6.6. We additionally compare with models trained with uniform language pairs sampling strategy (T=1) and find that our method outperforms both multilingual models. Most notably, while uniform sampling favor high-resource language pairs more than low-resource ones, GradVac is able to improve both consistently across all tasks. We observe larger performance gain on high-resource languages, illustrating that addressing gradient conflicts can mitigate negative interference on these head language pairs. On the other hand, our model still performs worse on resourceful languages compared to bilingual baselines, most likely limited by model capacity.

XTREME Benchmark. We additionally apply our method to multilingual language models and evaluate on the XTREME benchmark (Hu et al., 2020). We choose tasks where training data are available for all languages, and finetune a pretrained multilingual BERT model (mBERT) (Devlin et al., 2018) on these languages jointly. While other work mostly focus on zero-shot cross-lingual transfer (finetune on English training data and then evaluate on the target language

	En→Any					*Any→En*				
	en-fr	en-cs	en-hi	en-tr	avg	fr-en	cs-en	hi-en	tr-en	avg
	Monolithic Training									
(1) Bilingual Model	41.80	24.76	5.77	9.77	20.53	36.38	29.17	8.68	13.87	22.03
(2) Multilingual Model	37.24	20.22	13.69	18.77	22.48	34.29	27.66	18.48	22.01	25.61
	Multi-task Training									
(3) GradNorm	37.02	18.78	11.57	15.44	20.70	34.58	27.85	18.03	22.37	25.71
(4) MGDA	38.22	17.54	12.02	13.69	20.37	35.05	26.87	18.28	22.41	25.65
(5) PCGrad	37.72	20.88	13.77	18.23	22.65	34.37	27.82	18.78	22.20	25.79
(6) PCGrad w. all_layer	38.01	21.04	13.95	18.46	22.87	34.57	27.84	18.84	22.48	25.93
	Our Approach									
(7) GradVac w. fixed_obj	38.41	21.12	13.75	18.68	22.99	34.55	27.97	18.72	22.14	25.85
(8) GradVac w. whole_model	38.76	21.32	14.22	18.89	23.30	34.84	28.01	18.85	22.24	25.99
(9) GradVac w. all_layer	**39.27**	**21.67**	**14.88**	**19.73**	**23.89**	**35.28**	**28.42**	**19.07**	**22.58**	**26.34**

Table 6.5: BLEU scores on the WMT dataset. The best result for multilingual model is **bolded** while underline signifies the overall best.

test data), we use a different setup of multi-task learning such that we finetune multiple languages jointly and evaluate on all languages. Notice that our goal is not to compare with state-of-the-art results on this benchmark but rather to examine the effectiveness of our proposed method on pre-trained multilingual language models. We therefore only consider tasks that contain training data for all languages: named entity recognition (NER) and part-of-speech tagging (POS).

The NER task is from the WikiAnn (Pan et al., 2017) dataset, which is built automatically from Wikipedia. A linear layer with softmax classifier is added on top of pretrained models to predict the label for each word based on its first subword. We report the F1 score. Similar to NER, POS is also a sequence labeling task but with a focus on synthetic knowledge. In particular, the dataset we used is from the Universal Dependencies treebanks (Nivre et al., 2018). Task-specific layers are the same as in NER and we report F1. We select 12 languages for each task randomly.

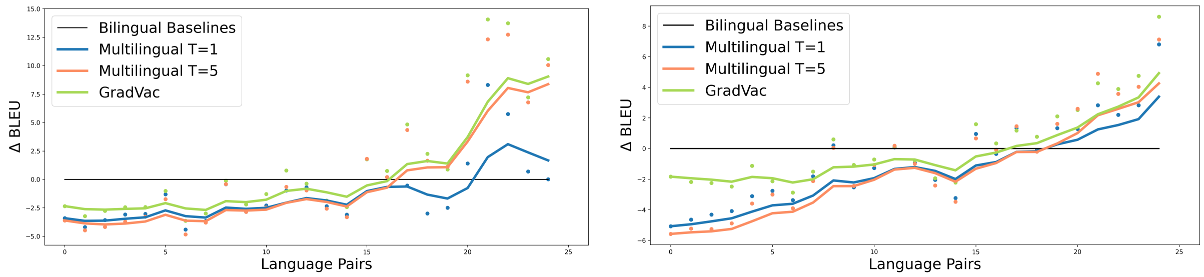
We use the multilingual BERT (Devlin et al., 2018) as our base model, which is a Transformer model pretrained on the Wikipedias of 104 languages using masked language modelling (MLM). It contains 12 layers and 178M parameters. Following (Hu et al., 2020), we finetune the model for 10 epochs for NER and POS, and search the following hyperparameters: batch size {16, 32}; learning rate {2e-5, 3e-5, 5e-5}.

As shown in Table 6.7 and 6.8, our method consistently outperforms naive joint finetuning and other multi-task baselines. This demonstrates the practicality of our approach for general

Any→En	High	Med	Low	All
T=1	28.56	28.51	19.57	24.95
T=5	28.16	28.42	24.32	26.71
GradVac	**28.99**	**28.94**	**24.58**	**27.21**

En→Any	High	Med	Low	All
T=1	22.62	21.53	12.41	18.18
T=5	22.04	21.43	13.07	18.25
GradVac	**24.20**	**21.83**	**13.30**	**19.08**

Table 6.6: Average BLEU scores of 25 language pairs on our massively multilingual dataset.



(a) X-En

(b) En-X

Figure 6.9: Comparing multilingual models with bilingual baselines on our dataset. Language pairs are listed in the order of training data sizes (high-resource languages on the left).

	de	en	es	hi	jv	kk	mr	my	sw	te	tl	yo	avg
mBERT	83.2	77.9	87.5	82.2	77.6	87.6	82.0	**75.8**	87.7	78.9	83.8	90.7	82.9
+ GradNorm	83.5	77.4	87.2	82.7	78.4	**87.9**	81.2	73.4	85.2	78.7	83.6	91.5	82.6
+ MGDA	82.1	74.2	85.6	81.5	77.8	87.8	81.9	74.3	86.5	78.2	87.5	91.7	82.4
+ PCGrad	83.7	78.6	88.2	81.8	79.6	87.6	81.8	74.2	85.9	78.5	85.6	92.2	83.1
+ GradVac	**83.9**	**79.4**	**88.2**	**81.8**	**80.5**	87.4	**82.1**	73.9	**87.8**	**79.3**	**87.8**	**93.0**	**83.8**

Table 6.7: F1 on the NER tasks of the XTREME benchmark.

multilingual tasks.

	ar	bg	de	en	es	fr	hi	hu	mr	ta	te	vi	avg
mBERT	84.2	94.7	92.7	91.0	93.8	93.3	88.0	91.9	83.3	80.3	90.4	79.2	88.6
+ GradNorm	83.5	94.7	92.3	91.0	93.6	93.2	88.2	91.4	83.0	**80.5**	90.6	79.1	88.4
+ MGDA	**84.4**	94.5	92.3	90.4	93.5	92.7	88.1	92.3	83.4	**80.5**	90.2	78.7	88.4
+ PCGrad	83.7	94.8	92.6	91.5	94.2	92.8	**88.5**	91.7	**83.7**	**80.5**	90.8	79.4	88.7
+ GradVac	84.1	**95.0**	**93.6**	**91.7**	**94.4**	**93.9**	**88.5**	**92.4**	83.5	79.8	**90.9**	**79.5**	**88.9**

Table 6.8: F1 on the POS tasks of the XTREME benchmark.

6.5 Related Work

Multilingual models train multiple languages jointly (Aharoni et al., 2019; Arivazhagan et al., 2019; Conneau et al., 2020; Devlin et al., 2018; Firat et al., 2016; Johnson et al., 2017; Lample and Conneau, 2019). Follow-up work study the cross-lingual ability of these models and what contributes to it (Artetxe et al., 2019b; Karthikeyan et al., 2020; Kudugunta et al., 2019; Pires et al., 2019; Wu and Dredze, 2019; Wu et al., 2020), the limitation of such training paradigm (Arivazhagan et al., 2019; Wang et al., 2020c), and how to further improve it by utilizing post-hoc alignment (Cao et al., 2020; Wang et al., 2020d), data balancing (Jean et al., 2019; Wang et al., 2020b), or calibrated training signal (Huang et al., 2019a; Mulcaire et al., 2019). In contrast to these studies, we directly investigate language interactions across training progress using loss geometry and propose a language-aware method to improve the optimization procedure.

On the other hand, multilingual models can be treated as multi-task learning methods (Ruder, 2017; Zamir et al., 2018). Prior work have studied the optimization challenges of multi-task training (Hessel et al., 2019; Schaul et al., 2019), while others suggest to improve training quality through learning task relatedness (Zhang and Yeung, 2012), routing task-specifc paths (Rosenbaum et al., 2019; Rusu et al., 2016), altering gradients directly (Chen et al., 2018b; Du et al., 2018; Kendall et al., 2018; Yu et al., 2020), or searching pareto solutions (Lin et al., 2019a; Sener and Koltun, 2018). However, while these methods are often evaluated on balanced task distributions, multilingual datasets are often unbalanced and noisy. As prior work have shown training with unbalanced tasks can be prone to negative interference (Ge et al., 2014; Wang and Carbonell, 2018), we study how to mitigate it in large models trained with highly unbalanced and massive-scale datasets.

6.6 Summary

In this chapter, we systematically study loss geometry through the lens of gradient similarity for multilingual modeling, and propose a novel approach named GradVac for improvement based on our findings. Leveraging the linguistic proximity structure of multilingual tasks, we validate the assumption that more similar loss geometries improve multi-task optimization while gradient conflicts can hurt model performance, and demonstrate the effectiveness of more geometrically consistent updates aligned with task closeness. We analyze the behavior of the proposed approach on massive multilingual tasks with superior performance. This concludes the second part of this thesis on task alignment, which improves model generalization. In the final part of this thesis, we focus on transfer sample efficiency such that we aim to utilize less human supervision in target tasks.

Part III

Improving Transfer Efficiency with Less Supervision

Chapter 7

Efficient Meta-Lifelong Learning

In the previous part, we have shown that alleviating negative transfer can improve the model's transfer performance, enhancing the model's generalization. In this final part, we study whether mitigating data discrepancy can also improve sample-efficiency of the transfer learning algorithm, such to transfer knowledge efficiently using less alignment data and/or labeled data in the target domain. We first investigate this in the setup of lifelong learning. State-of-the-art lifelong language learning methods store past examples in episodic memory and replay them at both training and inference time. However, as we show later in our experiments, there are three significant impediments: (1) needing unrealistically large memory module to achieve good performance, (2) suffering from negative transfer, (3) requiring multiple local adaptation steps for each test example that significantly slows down the inference speed. We identify three common principles of lifelong learning methods and propose an efficient meta-lifelong framework that combines them in a synergistic fashion. To achieve sample efficiency, our method trains the model in a manner that it learns a better initialization for local adaptation. Extensive experiments on text classification and question answering benchmarks demonstrate the effectiveness of our framework by achieving state-of-the-art performance using merely 1% memory size and narrowing the gap with multi-task learning. We further show that our method alleviates both catastrophic forgetting and negative transfer at the same time.

7.1 Introduction

Humans learn throughout their lifetime, quickly adapting to new environments and acquiring new skills by leveraging past experiences, while retaining old skills and continuously accumulating knowledge. However, state-of-the-art machine learning models rely on the data distribution being stationary and struggle in learning diverse tasks in such a *lifelong learning* setting (Parisi et al., 2019) (see section 7.2 for a formal definition). In particular, they fail to either effectively reuse previously acquired knowledge to help learn new tasks, or they forget prior skills when learning new ones - these two phenomena are known as *negative transfer* (Wang et al., 2019b) and *catastrophic forgetting* (McCloskey and Cohen, 1989), respectively. These downsides limit applications of existing models to real-world environments that dynamically evolve.

Due to its potential practical applications, there is a surge of research interest in the lifelong learning, especially in the vision domain (Chaudhry et al., 2019; Kirkpatrick et al., 2017; Lopez-Paz and Ranzato, 2017; Rusu et al., 2016; Sprechmann et al., 2018; Yoon et al., 2018; Zenke et al., 2017). However, its application to language learning has been relatively less studied. While progress in large-scale unsupervised pretraining (Devlin et al., 2018; Liu et al., 2019; Radford et al., 2019; Raffel et al., 2019; Yang et al., 2019) has recently driven significant advances in the field of natural language processing (NLP), these models require large amounts of in-domain training data and are prone to catastrophic forgetting when trained on new tasks (Yogatama et al., 2019), hindering their deployment in industry or other realistic setups where new tasks/domains continuously emerge.

One successful approach to achieving lifelong learning has been augmenting the learning model with an episodic memory module (Sprechmann et al., 2018). The underlying idea is to first store previously seen training examples in memory, and later use them to perform experience replay (Rolnick et al., 2019) or to derive optimization constraints (Chaudhry et al., 2019; Lopez-Paz and Ranzato, 2017) while training on new tasks. Recently, d'Autume et al. (2019) propose to use such a memory module for sparse experience replay and local adaptation in the language domain, achieving state-of-the-art results for lifelong learning on text classification and question answering tasks. Despite its success, the method has three critical downsides, which we demonstrate later in our experiments:

- It requires an unrealistically large memory module, i.e. storing **all** training examples, in order to achieve optimal performance.
- While the model can mitigate catastrophic forgetting, its local adaptation step is prone to negative transfer such that it performs worse on the most recent task than the naive baseline without any lifelong learning regularization.
- Its inference speed is extremely slow due to a non-trivial amount of local adaptation steps required for **each** test example.

In this chapter, we address these limitations and tackle the problem of *efficient* lifelong language learning. That is, we focus on storing limited training examples in memory. Our contributions are three-fold: First, we identify three common principles underlying lifelong learning methods. We seek to characterize them in language learning and glean insights on overlooked downsides of the existing method. Second, stemming from this analysis, we propose a meta-lifelong framework that unifies these three principles. Our approach is a direct extension of d'Autume et al. (2019) and it explicitly meta-learns the model as a better initialization for local adaptation. Finally, we conduct extensive experiments to demonstrate that our proposed approach can use the identified three principles to achieve efficient lifelong language learning. We find that our framework outperforms prior methods while using *100 times* less memory storage. Moreover, we demonstrate that our method can effectively alleviate catastrophic forgetting and negative transfer, closing the performance gap with the multi-task learning upper bound. It can also potentially obtain *22 times* faster inference speed.

7.2 Background: Principles of Lifelong Language Learning

Following prior work (d'Autume et al., 2019), we consider the lifelong learning setting where a model needs to learn multiple tasks in a sequential order via a stream of training examples without a task descriptor, i.e. the model does not know which task an example comes from during both training and testing. This setup is ubiquitous in practice, as environments consistently evolve without sending an explicit signal.

Formally, during training, the model makes a single pass over the training example stream consisting of N tasks in an ordered sequence, $\mathcal{D}^{train} = \{\mathcal{D}_1^{train}, \cdots, \mathcal{D}_N^{train}\}$, where $\mathcal{D}_t^{train} = \{(\boldsymbol{x}_t^i, y_t^i)\}_{i=1}^{n_t}$ is drawn from the task-specific distribution $P_t(\mathcal{X}, \mathcal{Y})$ of the t-th task. Overall, the goal is to learn a predictor $f_\theta : \mathcal{X} \to \mathcal{Y}$ such as a neural network, parameterized by $\theta \in \mathbb{R}^P$, to minimize the average expected risk of all N tasks:

$$R(f_\theta) \coloneqq \frac{1}{N} \sum_{t=1}^{N} \mathbb{E}_{\boldsymbol{x}, y \sim P_t} \left[\ell(f_\theta(\boldsymbol{x}), y) \right], \tag{7.1}$$

with ℓ being the specific task loss. Notice that while the average risk is most commonly evaluated after the model has seen all tasks, we can also evaluate a specific task at different stages to demonstrate the model's training behavior, and evaluate its robustness against catastrophic forgetting and negative transfer.

While different methods have been developed to optimize Eq.(7.1), we abstract away from their specific assumptions and instead focus on identifying common principles, among which we stress the following three points that are most relevant to language learning:

Generic Representation. Stemming from transfer learning (Ganin and Lempitsky, 2015; Weiss et al., 2016), a key idea of transferring knowledge across diverse tasks is to learn a generic representation (such as a neural network encoder) that is able to encode useful information for all tasks. For instance, regularization based lifelong learning methods (Chaudhry et al., 2019; Kirkpatrick et al., 2017; Schwarz et al., 2018; Zenke et al., 2017) add an extra constraint to prevent the model parameter θ from drastically deviating when training on new tasks, thereby learning a generic model for old tasks as well. In the language domain, as language models have proven success to generate highly generic representation for many language understanding tasks (Raffel et al., 2019; Yogatama et al., 2019), both d'Autume et al. (2019) and Sun et al. (2020) propose utilizing a pretrained language model (Devlin et al., 2018; Radford et al., 2019) to initialize parameters, and further training the model on $\mathcal{D}^{train}$.

Experience Rehearsal. Motivated by the complementary learning systems (CLS) theory (McClelland et al., 1995) that humans rely on episodic memory to store past experiences and conduct experience rehearsal, we can also retrain lifelong learning models on previously seen tasks to reduce forgetting. While prior methods use memory to define optimization constraints (Chaudhry et al., 2019; Lopez-Paz and Ranzato, 2017; Sodhani et al., 2020), recent work use either stored examples (Sprechmann et al., 2018) or generated synthetic data (Shin et al., 2017; Sun et al., 2020) to perform experience replay. Further, d'Autume et al. (2019) shows that a sparse 1$\%$ rate of replaying to learning new examples is sufficient for lifelong language learning.

Task-specific Finetuning. In multi-task learning, injecting task-specific parameters and finetuning on individual tasks have proven effective for different language understanding tasks (Houlsby et al., 2019) or even diverse languages (Bapna et al., 2019). Prior work (Rusu et al., 2016; Yoon et al., 2018) exploit this idea to expand model parameters for new tasks in lifelong learning setting. However, all these methods require a task descriptor in order to know when to add new parameters. When no such signal exists, local adaptation (Sprechmann et al., 2018) uses K stored nearest neighbors of each test example to perform extra finetuning at inference time. Recent work (d'Autume et al., 2019; Khandelwal et al., 2020) demonstrate that the sentence embeddings produced by pretrained models can be used to effectively measure query similarity and that local adaptation can improve performance on text classification, question answering and language modelling.

7.3 Proposed Framework

With these principles in mind, we next turn to the problem of how to achieve efficient lifelong learning. To motivate our proposed framework, we first review the state-of-the-art method, improved MbPA (d'Autume et al., 2019), and show how these principles help us to identify the limitation.

7.3.1 Model-based Parameter Adaptation

As a notable example, a recent line of work (d'Autume et al., 2019; Khandelwal et al., 2020; Sprechmann et al., 2018) have successfully utilized an episodic memory module as a crucial building block for general linguistic reasoning. Specfically, the improved Model-based Parameter Adaptation (MbPA++) (d'Autume et al., 2019) consists of three main components: (i) a predictor network f_θ, (ii) a key network g_ϕ, and (iii) a memory module $\mathcal{M}$. The end goal is to train f_θ to generalize well across all tasks as in Eq.(7.1).

To learn a generic representation, MbPA++ utilizes any state-of-the-art text encoder, such as BERT, to initialize both predictor network f_θ and key network g_ϕ. At each time step, the model receives a training example $(\boldsymbol{x}_t^i, y_t^i) \in \mathcal{D}^{train}$ and updates parameter θ by optimizing the task loss:

$$\mathcal{L}_{\text{TASK}}(\theta; \boldsymbol{x}_t^i, y_t^i) = \ell(f_\theta(\boldsymbol{x}_t^i), y_t^i), \tag{7.2}$$

To determine if the training example should be added to the memory module $\mathcal{M}$, a Bernoulli random variable is drawn with pre-set probability, which is used to control the memory size.

For experience rehearsal, a subset $\mathcal{S}$ of $\mathcal{M}$ is randomly selected, based on a set ratio of replay examples to learning new examples (i.e. revisit n_{re} examples for every n_{tr} training examples). To avoid catastrophic forgetting, the model then updates the following replay loss to adapt θ towards seen tasks:

$$\mathcal{L}_{\text{REP}}(\theta; \mathcal{S}) = \frac{1}{n_{re}} \sum_{\boldsymbol{x}, y \in \mathcal{S}} \ell(f_\theta(\boldsymbol{x}), y), \tag{7.3}$$

At inference time, the key network g_ϕ, which is fixed during training, is used to encode example inputs as keys to obtain the K nearest neighbour context $\mathcal{N}_{\boldsymbol{x}_i}$ of the i-th testing example $\boldsymbol{x}_i$. L local adaptation gradient updates are then performed to achieve task-specific finetuning for the following objective:

$$\mathcal{L}_{\text{LA}}(\tilde{\theta}_i; \theta, \mathcal{N}_{\boldsymbol{x}_i}) = \frac{1}{K} \sum_{\boldsymbol{x}, y \in \mathcal{N}_{\boldsymbol{x}_i}} \ell(f_{\tilde{\theta}_i}(\boldsymbol{x}), y) \\ + \lambda_l \|\tilde{\theta}_i - \theta\|_2^2 \tag{7.4}$$

where λ_l is a hyperparameter. The predictor network $f_{\tilde{\theta}_i}$ is then used to output the final prediction for the i-th testing example.

Despite its effectiveness, the performance gain of MbPA++ comes at a cost of large memory storage and slow inference speed. The root of this inefficiency is the non-synergistic nature of the method - the three principles are performed independently without close interaction. In particular: (i) the generic representation learned is not optimized for local adaptation and thus more steps are required for robust performance, (ii) the memory module is selected randomly and lacks a systematic selection method to effectively reduce its size, (iii) local adaptation only utilizes a few neighbours for each testing example so it is prone to overfit and negative transfer when memory size is small.

7.3.2 Synergistic Meta-lifelong Framework

We notice that there is a discrepancy between training and testing in MbPA++. Specifically, the generic representation is trained on the task loss in Eq.(7.2) directly while it makes prediction *after* the local adaptation at test time. Therefore, the model always overfits to the latest task it has seen, and it never learns how to incorporate experience rehearsal efficiently. According to the CLS theory (McClelland et al., 1995), however, human learning systems are complementary in nature - we learn structured knowledge in a manner that allows us to adapt to episodic information fast. Thus, to resolve the training-testing discrepancy of MbPA++, we change the training goal of generic representation from *how to perform better on the current task* to *how to adapt to episodic memory efficiently*.

In particular, we propose an extension of MbPA++ that exploits a meta learning paradigm to interleave the three key principles: (i) to resolve the training-testing gap, our framework learns a generic representation that is tailored for local adaptation, (ii) to enable robust local adaptation, the memory module uses a diversity-based selection criteria to reduce memory size, (iii) to accommodate small memory, the framework utilizes a coarse local adaptation to alleviate negative transfer. The full framework is outlined in Algorithm 3 and below we detail how each principle is instantiated in a systematic way.

Algorithm 3 Meta-MbPA

```
**Procedure Train**
**Input:** training data $\mathcal{D}^{train}$
**Output:** parameters $\theta$, memory $\mathcal{M}$
Initialize $\theta$ with some pretrained model
**for** $(\boldsymbol{x}_t^i, y_t^i) \in \mathcal{D}^{train}$ **do**
    **[Generic Representation]** Perform a gradient update on $\theta$ to minimize Eq.(7.5)
    **if** training step mod $n_{tr}$ = 0 **then**
        Sample $n_{re}$ examples from $\mathcal{M}$
        **[Experience Rehearsal]** Perform a gradient update on $\theta$ to minimize Eq.(7.6)
    **end if**
    Compute $p(\boldsymbol{x}_t^i)$ according to Eq.(7.7)
    **if** Bernoulli($p(\boldsymbol{x}_t^i)$) = 1 **then**
        Update memory $\mathcal{M} \leftarrow \mathcal{M} \cup (\boldsymbol{x}_t^i, y_t^i)$
    **end if**
**end for**
**Procedure Test**
**Input:** test examples $\boldsymbol{x}$
**Output:** predictions $\hat{\boldsymbol{y}}$
**for** $l = 1, ..., L$ **do**
    Sample K examples from $\mathcal{M}$
    **[Task-specific Finetuning]** Perform a gradient update on $\theta$ to minimize Eq.(7.4)
**end for**
Output prediction $\hat{\boldsymbol{y}}_i = f_\theta(\boldsymbol{x}_i)$ =0
```

Generic Representation. We incorporate local adaptation into training generic representation. In particular, we exploit the idea of meta learning by formulating local adaptation as the base task and representation learning as the meta task. That is, the generic representation is trained such that it should perform well *after* the local adaptation (a.k.a. learning to adapt). Thus, for each training example $(\boldsymbol{x}_t^i, y_t^i) \in \mathcal{D}^{train}$, we formulate the task loss in Eq.(7.2) into a meta-task loss as:

$$
\begin{aligned}
&\mathcal{L}_{\text{TASK}}^{\text{meta}}(\theta; \boldsymbol{x}_t^i, y_t^i) = \ell(f_{\tilde{\theta}_{\boldsymbol{x}_t^i}}(\boldsymbol{x}_t^i), y_t^i) \\
\text{s.t.} \quad &\tilde{\theta}_{\boldsymbol{x}_t^i} = \theta - \alpha \nabla_\theta \mathcal{L}_{\text{LA}}(\theta; \mathcal{N}_{\boldsymbol{x}_t^i})
\end{aligned}
\tag{7.5}
$$

where α is the current learning rate. Notice the differentiation requires computing the gradient of gradient, which can be implemented by modern automatic differentiation frameworks. Intuitively, we first approximate local adaptation using gradient step(s), and then optimize the adapted network.

Experience Rehearsal. With similar rationale to the meta-task loss, we reformulate the memory replay loss in Eq.(7.3) into a meta-replay loss:

$$\begin{aligned} \mathcal{L}_{\text{REP}}^{\text{meta}}(\theta; \mathcal{S}) &= \frac{1}{n_{re}} \sum_{\boldsymbol{x}, y \in \mathcal{S}} \ell(f_{\tilde{\theta}_{\boldsymbol{x}}}(\boldsymbol{x}), y) \\ \text{s.t.} \quad \tilde{\theta}_{\boldsymbol{x}} &= \theta - \alpha \nabla_{\theta} \mathcal{L}_{\text{LA}}(\theta; \mathcal{N}_{\boldsymbol{x}}) \end{aligned} \tag{7.6}$$

with the objective to stimulate efficient local adaptation for all tasks.

We use the same replay ratio as in MbPA++ to keep the meta replay sparse. In addition, we propose a diversity-based selection criterion to determine if a training example $(\boldsymbol{x}_t^i, y_t^i) \in \mathcal{D}^{train}$ should be added to the memory module. Here, we exploit the key network g_ϕ to estimate diversity via the minimum distance of $\boldsymbol{x}_t^i$ to existing memory as:

$$\log(p(\boldsymbol{x}_t^i)) \propto -\frac{\min\limits_{\boldsymbol{x}, y \in \mathcal{M}} \|g_\phi(\boldsymbol{x}_t^i) - g_\phi(\boldsymbol{x})\|_2^2}{\beta}, \tag{7.7}$$

where $p(\boldsymbol{x}_t^i)$ is the probability of the example being selected and β is a scaling parameter. The intuition is to select examples that are less similar to existing memory thereby covering diverse part of data distribution. As shown later, the proposed method outperforms the uncertainty-based selection rule (Ramalho and Garnelo, 2019), which picks examples based on certainty level of the predictor network f_θ. This is because local adaptation is prone to negative transfer when the memory $\mathcal{M}$ misrepresents the true data distribution.

Task-specific Finetuning. With small memory, local adaptation for each testing example is prone to negative transfer. This is because less related memory samples are more likely to be included in $\mathcal{N}_{\boldsymbol{x}_i}$ and the model can easily overfit. Thus, we consider local adaptation with more coarse granularity. For example, we can cluster testing examples and conduct local adaptation for each cluster independently. In our experiments, we find that it is sufficient to take this to the extreme such that we consider all test examples as a single cluster. Consequently, we consider the whole memory as neighbours and we randomly sample from it to be comparable with the original local adaptation formulation (i.e. same batch sizes and gradient steps). As shown in the next section, it has two benefits: (1) it is more robust to negative transfer, (2) it is faster when we evaluate testing examples as a group.

Order	Enc-Dec	Online EWC	A-GEM†	Replay	MbPA++†	MbPA++ (Our Impl.)	Meta-MbPA (1%)	MTL	MTL (1%)	LAMOL$^{0.2}_{TASK}$‡
					Text Classification					
i.	35.5	43.8	70.7	63.4	70.8	75.3	**77.9**	-	-	76.7
ii.	44.8	49.8	65.9	73.0	70.9	74.6	**76.7**	-	-	77.2
iii.	42.4	59.5	67.5	65.8	70.2	75.6	**77.3**	-	-	76.1
iv.	28.6	52.0	63.6	74.0	70.7	75.5	**77.6**	-	-	76.1
Average	37.8	51.3	66.9	69.1	70.6	75.3	**77.3**	78.9	50.4	76.5
					Question Answering					
i.	60.9	58.0	56.1	62.3	62.0	63.3	**64.8**	-	-	-
ii.	57.3	57.2	58.4	61.3	62.4	63.5	**65.3**	-	-	-
iii.	47.0	49.5	52.4	58.3	61.4	61.6	**64.4**	-	-	-
iv.	61.0	58.7	57.9	62.9	62.4	62.4	**65.0**	-	-	-
Average	56.6	55.9	56.2	61.2	62.1	62.7	**64.9**	68.6	44.1	-

Table 7.1: **Accuracy and F_1 scores for text classification and question answering, respectively.** Methods that use the defined lifelong learning setup in Section 7.2 are listed on the left. Where applicable, all methods use $r_{\mathcal{M}} = 100\%$ memory size unless denoted otherwise. The best result for lifelong learning methods is made **bold**. † Results obtained from (d'Autume et al., 2019). ‡ LAMOL (Sun et al., 2020) is not directly comparable due to their different problem setup where task descriptors are available.

7.4 Experiments

7.4.1 Evaluation Dataset

To evaluate the proposed framework, we conduct experiments on text classification and question answering tasks. Following prior work, we consider each dataset as a separate task and the model needs to sequentially learn several tasks of the same category (e.g. all text classification tasks). As pointed out in (McCann et al., 2018), many NLP tasks can be formulated as question answering and thus our setup is general.

Text classification We use five datasets from (Zhang et al., 2015) spanning four text classification tasks: (1) news classification (AGNews), (2) sentiment analysis (Yelp, Amazon), (3) Wikipedia article classification (DBPedia) and (4) questions and answers categorization (Yahoo). To compare our framework with (d'Autume et al., 2019), we follow the same data processing procedure as described by them to produce balanced datasets. In total, we have 33 classes,

	$r_{\mathcal{M}} = 1\%$		$r_{\mathcal{M}} = 10\%$	
Model / Task	class.	QA	class.	QA
MbPA++	73.1	61.9	73.5	62.6
Meta-MbPA	77.3	64.9	78.0	65.5
MTL	50.4	44.1	70.5	56.2

Table 7.2: **Performance of models using different sizes of memory.**

	Replay	MbPA++	Meta-MbPA
Text Classification			
Random	69.2	73.1	76.8
Diversity	69.1	73.0	77.3
Uncertainty	65.4	41.2	62.7
Forgettable	62.7	50.5	61.8
Question Answering			
Random	61.2	61.9	63.8
Diversity	61.5	62.2	64.9
Uncertainty	56.1	50.4	54.2
Forgettable	59.7	52.1	57.5

Table 7.3: **Performance of models using different memory selection criteria.** "Uncertainty" utilizes model's confidence level (Ramalho and Garnelo, 2019). "Forgettable" picks examples according to forgetting events (Toneva et al., 2019). We tune hyperparameters that result in $r_{\mathcal{M}} = 1\%$ memory size for all methods.

$575,000$ training examples and $38,000$ test examples from all datasets.

Question Answering Following (d'Autume et al., 2019), we use three question answering datasets: SQuAD v1.1(Rajpurkar et al., 2016), TriviaQA (Joshi et al., 2017) and QuAC (Choi et al., 2018). TriviaQA has two sections, Web and Wikipedia, which we consider as separate datasets. We process the datasets to follow the same setup as (d'Autume et al., 2019). Our processed datasets includes $60,000$-$90,000$ training and $7,000$-$10,000$ validation examples per task.

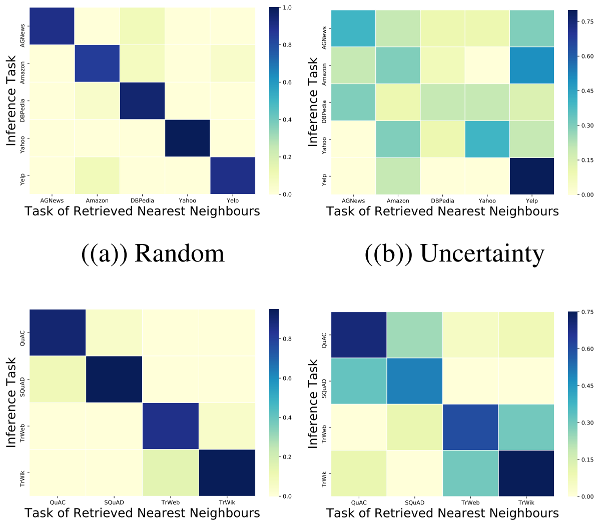
7.4.2 Experimental Setup

We consider the prominent baselines corresponding to each one of the three principles as introduced in Section 7.2. We first consider a standard encoder-decoder model (*Enc-Dec*) which does not utilize any lifelong learning regularization. In the spirit of learning generic representation using parameter regularization, we compare our framework with *Online EWC* (Schwarz et al., 2018) and *A-GEM* (Chaudhry et al., 2019). For experience rehearsal, we implement *Replay*, a model that uses stored examples for sparse experience replay only. Finally, we compare with the state-of-the-art *MbPA++* (d'Autume et al., 2019) which combines experience rehearsal with task-specific finetuning.

Implementation Details We utilize the pre-trained BERT$_{\text{BASE}}$ (Wolf et al., 2019) for initializing the encoder network. BERT$_{\text{BASE}}$ has 12 Transformer layers, 12 self-attention heads, and 768 hidden dimensions (110M parameters). Similar to (d'Autume et al., 2019), we use a separate pre-trained BERT$_{\text{BASE}}$ for key network and freeze it to prevent from drifting while training on a non-stationary data distribution. For text classification, we use encoded representation of the special beginning-of-document symbol `[CLS]` as our key. For question answering, we use the question part of the input to get the encoded representation. For both tasks, we store the input example as its associated memory value. Further, we use Faiss (Johnson et al., 2019) for efficient nearest neighbor search in the memory, based upon the key network.

We mainly set hyper-parameters as mentioned in (d'Autume et al., 2019). We use Adam (Kingma and Ba, 2014) as our optimizer, set dropout (Srivastava et al., 2014) to 0.1 and the base learning rate to $3e^{-5}$. For text classification, we use a training batch of size 32 and set the maximum total input sequence length after tokenization to 128. For question answering, we use a training batch of size 8, set the maximum total input sequence length after tokenization to 384 and to deal with longer documents we set document stride to 128. We also set the maximum question length to 64.

For Online EWC (Schwarz et al., 2018), we set the regularization strength $\lambda = 5000$ and forgetting coefficient $\gamma = 0.95$. For all models with memory module (Replay, MbPA++, Meta-MbPA), we replay 100 examples for every $10,000$ new examples, i.e., $n_{tr} = 10,000$ and $n_{re} = 100$. As mentioned in (d'Autume et al., 2019), for MbPA++, we set the number of neighbors $K = 32$, the number of local adaptation steps $L = 30$ and $\lambda_l = 0.001$. We tune the local adaptation learning rate α for MbPA++ in our re-implementation (MbPA++ Our Impl.) and report the improved numbers as well as their reported numbers in Table 7.1, 7.7, and 7.8. For text classification, we set $\alpha = 5e^{-5}$ and for question answering we set $\alpha = 1e^{-5}$.



((a)) Random ((b)) Uncertainty

((c)) Diversity ((d)) Forgettable

Figure 7.1: **Proportions of source of neighbours used in local adaptation for each task when different memory selection rule is used, e.g. 10% of neighbours retrieved for Yelp belong to Amazon.** Numbers in each row sum to 1. Classification figures are at the top while QA at the bottom (Task ordering i.)

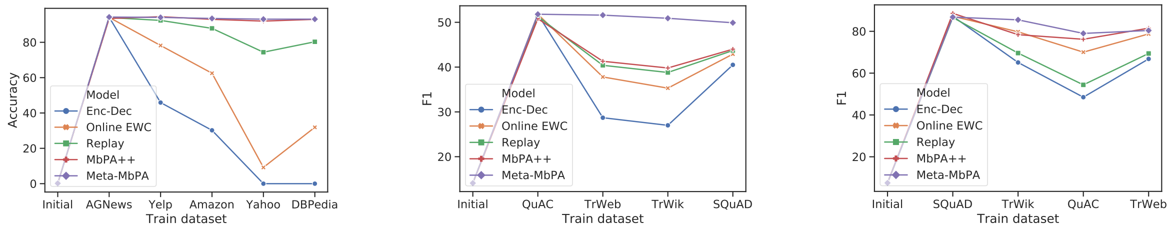
For our framework, Meta-MbPA, unless stated otherwise, we set the number of neighbors $K = 32$ and control the memory size through a write rate $r_{\mathcal{M}} = 1\%$. We use $L = 30$ local adaptation steps and perform local adaptation for whole testing set. That is, we randomly draw $K = 32$ examples from the memory and perform a local adaptation step. Through this, the computational cost is equivalent to MbPA++ but we only need to perform the whole process once while MbPA++ requires conducting local adaptation independently for each testing example. We set $\alpha = 1e^{-5}$ (in Eq. (7.5), (7.6)), $\beta = 10$ (in Eq. (7.7)) and $\lambda_l = 0.001$ (in Eq. (7.4)). All of the experiments are performed using PyTorch (Paszke et al., 2017), which allows for automatic differentiation through the gradient update as required for optimizing the meta-task loss Eq. (7.5) and meta-replay loss Eq. (7.6).

7.4.3 Results

We use four different orderings of task sequences as in (d'Autume et al., 2019) and evaluate the model at the end of all tasks. Following prior work, we report the macro-averaged accuracy for classification and F_1 score for question answering. Table 7.1 provides a summary of our main results. Notice that results on the right are not comparable due to different setups.

	Uncertainty		Forgettable	
Model / Task	class.	QA	class.	QA
Meta-MbPA	62.7	54.2	61.8	57.5
w/o LA	65.8	55.8	67.9	59.2
MbPA++	41.2	50.4	50.5	52.1
w/o LA	65.4	56.1	68.4	59.2

Table 7.4: **Performance of models using the uncertainty-based memory selection methods (correspond to Table 7.3).** “LA” refers to local adaptation.



((a)) Text Classification-AGNews

((b)) Question Answering-QuAC

((c)) Question Answering-SQuAD

Figure 7.2: **Catastrophic Forgetting of the first dataset as training progresses.**

We first compare our framework (Meta-MbPA) with all baselines. Even using only 1% of total training examples as memory, the proposed framework still outperforms existing methods on both text classification and question answering. Specifically, while regularization-based methods (A-GEM and Online EWC) perform better than the standard Enc-Dec model, their performance vary depending on the task ordering and thus are not robust. On the other hand, methods that involve local adaptation (MbPA++ and ours) perform consistently better for all orderings. In particular, our framework improves over MbPA++ while using 100 times less memory, demonstrating the effectiveness of the proposed approach.

We then compare lifelong learning methods to the multitask model MTL, which serves as an upper bound of achievable performance. As shown in Table 7.1, there is still a non-trivial gap between MbPA++ and MTL, albeit MbPA++ stores all training examples as memory. Our framework narrows the gap while using smaller memory.

	Enc-Dec	Replay	MbPA++	Meta-MbPA
class.	82.1	81.8	78.6	82.1
QA	72.6	72.7	70.7	72.1

Table 7.5: **Average performance on the last task across all four task orderings.**

7.4.4 Analysis

Memory Capacity. In Table 7.1, MbPA++ uses 100% memory while our framework only uses 1% memory. To test memory efficiency, we present results for models using equivalent memory resources in Table 7.2. The results demonstrate that the performance of MbPA++ degrades significantly as the memory size decreases. Consequently, the performance gap between MbPA++ and Meta-MbPA enlarges when they both use equal amount of stored examples, compared to results in Table 7.1. It is then natural to ask if using memory alone is sufficient to obtain good performance. We thus compare with MTL trained on subsampled training data, which is equivalent to only performing local adaptation without training the generic representation. Notice that this variant of MTL is *not* an upper bound as it uses less resources. Our method significantly outperforms it, showing that the meta generic representation in our method is also crucial to achieve good performance. These results validate that the proposed framework can utilize the memory module more effectively than existing methods.

We then study the source of improvement of our method. In particular, we show that the prior method is prone to negative transfer. To see this, we first conduct a case study of memory selection rule.

Memory Selection Rule. We consider two popular paradigms in active learning (Donmez et al., 2007), namely the diversity-based method that picks the most representative examples and the uncertainty-based method that picks the most unsure examples. In particular, we compare four selection criteria belonging to these two categories: random selection, our proposed diversity-based method in Eq.(7.7), and two uncertainty-based methods (Ramalho and Garnelo, 2019; Toneva et al., 2019). Notice that random selection is considered as a diversity-based method since it picks examples that represent the true data distribution. As shown in Table 7.3, we observe that the choice of memory selection criteria clearly impacts performance. While the proposed diversity method slightly outperforms random selection, the two uncertain-based methods perform *worse* than the random baseline, consistent with similar findings reported in d'Autume et al. (2019).

We seek an explanation for this phenomenon and visualize the heat maps in Figure 7.1 to show which tasks each testing example's retrieved neighbours come from during the local adaptation phase. Ideally, the model should always use neighbours from the same task and the heat map should be diagonal. We observe that, compared to diversity-based methods, more examples from other tasks are used as nearest neighbours when models use uncertainty-based methods. This is because the selected uncertain examples are usually less representative in the true distribution and could be outliers. Thus, the resulting memory does not have good coverage of the data distribution and no similar examples exist for certain test examples. Consequently, less related examples from other tasks are used for the local adaptation, which causes negative transfer. This is verified in Table 7.4, where models without local adaptation outperform their locally adapted counterparts. More importantly, Meta-MbPA obtains much smaller performance gaps, indicating that it is more robust to negative transfer. We further verify this in the following section.

Trade-off between Catastrophic Forgetting and Negative Transfer. We first verify the models' robustness to catastrophic forgetting. As shown in Table 7.7 and 7.8, the standard Enc-Dec model performs poorly on previously trained tasks, indicating the occurrence of catastrophic forgetting. While all baselines can alleviate the forgetting to some degree, our framework achieves the best performance on previously learned tasks. We also evaluate the model's performance on the first task as it continues to learn more tasks. Figure 7.2 illustrates how each model retains its previously acquired knowledge as it learns new knowledge. We observe that our framework is consistently better than the baselines at mitigating forgetting.

In addition, as prior work have shown transferring from diversely related sources can hurt performance in the target (Ge et al., 2014; Wang and Carbonell, 2018), we study if transferring from multiple tasks learned in the past can induce negative transfer, which is often overlooked in existing studies on lifelong learning. Table 7.5 shows the averaged results on the last task in each task ordering. Surprisingly, compared to the Enc-Dec baseline, MbPA++ actually performs *worse* on the last task despite its improved macro-averaged performance (Table 7.1). This suggests that while it is robust to catastrophic forgetting, MbPA++ fails to utilize prior knowledge to benefit later tasks and thus is prone to negative transfer. Apart from some practical bottlenecks such as limited model capacity, local adaptation is a critical factor of negative transfer as Replay outperforms MbPA++ in Table 7.5. Intuitively, this shows that since Replay already performs well on the last task, further using local adaption can overfit and hurt the performance. On the other hand, the proposed method is trained to learn a more robust initialization for adaptation and uses a coarse adaptation that is less prone to negative transfer. Therefore, it outperforms MbPA++ and closes the gap with Enc-Dec on the last task, consistent with results in Table 7.4.

	$r_{\mathcal{M}} = 1\%$		$r_{\mathcal{M}} = 50\%$	
Model / Task	class.	QA	class.	QA
Meta-MbPA	77.3	64.9	78.2	66.1
w/o Meta	73.1	58.5	74.0	59.6
w/o MS	76.8	63.8	78.1	66.1
w/o LA	75.9	62.0	75.8	62.1

Table 7.6: **Ablation Study on different memory size.** "Meta" refers to the proposed meta optimization in Eq.(7.5) and (7.6)."MS" denotes memory selection based on Eq.(7.7). "LA" refers to local adaptation.

All of these experiments illustrate that there is a trade-off between catastrophic forgetting and negative transfer, such that more adaptations are desired for earlier tasks while less is better for later tasks. While prior studies focus on catastrophic forgetting only, we are the first to show the importance of balancing the trade-off to avoid both negative effects.

Ablation Study. We report the results of ablation study in Table 8.7 and analyze the effects of the three components in our framework subject to different memory sizes. First, we observe that the model without the meta learning optimization performs the worst, which shows the importance of learning a generic representation tailored for local adaptation. More importantly, Meta-MbPA achieves worse performance without any local adaptation step. This demonstrates that learning the generic representation alone is not sufficient enough, and that the meta learning mechanism and local adaptation are complementary, which mimic the complementary human learning systems in the CLS theory. Finally, while the diversity-based memory selection rule contributes to the performance gain when we use a small memory module, it becomes less effective as the memory size increases. This is expected since the memory distribution can well represent the true data distribution with a larger capacity, and thus it demonstrates that the proposed methods mostly contribute to robustly reducing the memory sizes for better efficiency. Overall, these results validate the effectiveness of each component and highlight the importance of complementary lifelong learning systems. To the best of our knowledge, this is the first work to formulate the slow learning of structured knowledge as meta task and the fast learning from episodic memory as base task.

Inference Speed. The ordinary local adaptation requires customized gradient updates for each testing example and thus it is notoriously slow. Using 1 Nvidia Tesla V100 GPU and 128

GB of RAM, it takes 66.6 hours and 89.3 hours to evaluate on text classification and question answering, respectively. On the other hand, we use coarse local adaptation in our method which uses the same updates for all testing examples. Consequently, it takes 2.9 hours and 4.2 hours for our method to finish the evaluation process, achieving a maximum 22 times speedup. Notice that in a pure online learning setup, our method will obtain similar inference speed as MbPA++. In addition, we hypothesize that using a different granularity (e.g. clustering testing examples) is beneficial for tasks that are more conflicting in nature, as it can balance the trade-off between overfitting to nearest neighbours of small memory and performing more sample-specific adaptation for each test example. We leave this exploration for future work.

7.5 Summary

In this chapter, we identify three principles underlying different lifelong language learning methods and show how to unify them in a meta-lifelong framework. Our experiments demonstrate the effectiveness of the proposed framework and we report new state-of-the-art results while using 100 times less memory space. Our analysis also shows that negative transfer is an overlooked factor that could cause sub-optimal performance, and we show that alleviating it can significantly boost efficiency. The method proposed in this chapter reduces the requirement of training examples in lifelong learning, in the last two chapters of this thesis we explore the limit of transfer learning through zero-shot learning, where no training data is utilized in the target task.

Order	Dataset	Enc-Dec	Online	A-GEM[†]	Replay	MbPA++[†]	MbPA++	MbPA++	Meta-MbPA
			EWC				(Our Impl.)	($r_\mathcal{M} = 1\%$)	($r_\mathcal{M} = 1\%$)
i	1	2.0	29.7	42.5	49.2	45.7	59.2	54.2	62.1
	2	4.3	0.1	89.8	50.1	91.6	94.0	91.0	93.7
	3	95.8	97.5	96.0	98.7	96.3	98.5	98.5	99.1
	4	1.3	18.5	56.8	45.2	54.6	57.7	56.7	60.7
	5	74.2	73.2	68.2	74.0	65.6	67.2	66.7	73.8
	Average	35.5	43.8	70.7	63.4	70.8	75.3	73.4	77.9
ii	1	62.2	89.9	80.1	98.7	95.8	98.5	98.0	99.0
	2	0.0	0.1	50.3	54.6	63.1	69.7	61.7	70.2
	3	39.4	40.3	91.3	89.3	92.2	95.0	93.0	92.5
	4	61.3	60.0	57.3	61.5	55.7	55.2	55.2	60.1
	5	61.2	58.5	50.6	61.1	47.7	54.7	52.7	61.5
	Average	44.8	49.8	65.9	73.0	70.9	74.6	72.1	76.7
iii	1	11.4	52.5	41.1	54.8	44.3	59.2	53.7	59.6
	2	2.1	14.9	55.0	31.9	62.7	67.7	60.2	70.2
	3	12.8	40.3	54.6	52.0	54.4	58.2	60.7	63.8
	4	92.5	98.0	93.3	97.4	96.2	98.5	98.0	98.9
	5	93.3	91.8	93.6	93.1	93.4	94.5	92.5	94.1
	Average	42.4	59.5	67.5	65.8	70.2	75.6	73.0	77.3
iv	1	0.0	31.9	90.8	80.3	91.8	94.0	91.0	93.1
	2	8.3	33.3	44.9	59.3	44.9	57.2	54.2	60.8
	3	3.6	22.2	60.2	59.6	55.7	59.7	61.2	61.6
	4	31.8	73.5	65.4	71.9	65.3	68.7	63.7	73.6
	5	99.1	98.9	56.9	99.1	95.8	98.0	98.5	99.1
	Average	28.6	52.0	63.6	74.0	70.7	75.5	73.7	77.6

Table 7.7: **Dataset specific accuracy for text classification tasks for different dataset orders and models.** † Results obtained from (d'Autume et al., 2019). Where applicable, we use $r_\mathcal{M} = 100\%$ unless denoted otherwise.

Order	Dataset	Enc-Dec	Online	A-GEM[†]	Replay	MbPA++[†]	MbPA++	MbPA++	Meta-MbPA
			EWC				(Our Impl.)	($r_{\mathcal{M}} = 1\%$)	($r_{\mathcal{M}} = 1\%$)
i	1	40.5	42.9	36.7	44.1	47.2	44.3	42.6	49.9
	2	60.1	57.4	51.8	60.7	57.7	62.9	60.0	63.1
	3	58.2	53.8	53.4	58.7	58.9	61.2	58.8	61.5
	4	85.0	77.7	82.5	85.5	84.3	84.7	86.8	84.7
	Average	60.9	58.0	56.1	62.3	62.0	63.3	62.0	64.8
ii	1	66.8	78.8	64.2	73.1	72.6	80.4	81.8	80.4
	2	64.2	59.5	62.5	64.2	63.4	65.3	60.7	61.5
	3	31.4	28.6	43.4	41.0	50.5	42.0	41.6	52.1
	4	66.7	61.9	63.5	66.8	63.0	66.1	64.3	67.0
	Average	57.3	57.2	58.4	61.3	62.4	63.5	62.1	65.3
iii	1	41.6	57.2	47.6	58.7	56.0	62.0	59.4	65.7
	2	38.8	51.9	47.0	54.2	56.8	53.4	57.3	59.2
	3	54.4	63.1	57.4	67.7	78.0	81.8	83.9	80.7
	4	53.1	25.5	57.4	52.7	54.9	49.0	46.9	52.1
	Average	47.0	49.5	52.4	58.3	61.4	61.6	61.8	64.4
iv	1	58.1	60.5	54.8	59.4	59.0	58.9	60.8	61.3
	2	39.8	36.3	38.8	45.0	48.7	43.5	39.2	50.4
	3	60.5	60.4	53.4	61.6	58.1	64.2	61.3	63.7
	4	85.6	77.3	84.7	85.6	83.6	82.8	85.3	84.5
	Average	61.0	58.7	57.9	62.9	62.4	62.4	61.6	65.0

Table 7.8: **Dataset specific F_1 scores for question answering tasks for different dataset orders and models.** † Results obtained from (d'Autume et al., 2019). Where applicable, we use $r_{\mathcal{M}} = 100\%$ unless denoted otherwise.

Dataset	Single Model	MTL (1%)	MTL (10%)	MTL (100 %)
Text Classification				
AGNews	93.6	83.1	88.7	94.0
Amazon	61.8	38.6	54.2	63.5
DBPedia	99.2	78.1	91.4	99.3
Yahoo	74.9	15.8	65.6	75.3
Yelp	61.9	36.4	52.8	62.6
Average	78.28	50.4	70.5	78.9
Question Answering				
QuAC	54.0	20.9	30.9	53.5
SQuAD	87.8	60.5	75.2	88.1
Trivia Web	65.8	49.2	62.2	67.7
Trivia Wikipedia	62.9	45.9	56.5	64.9
Average	67.6	44.1	56.2	68.6

Table 7.9: **Single model and Multi-Task Learning (MTL) results for text classification and question answering tasks.** MTL ($X\%$) denotes $X\%$ of the training examples are used per dataset to train MTL models.

First Dataset	Dataset	Enc-Dec	Online EWC	Replay	MbPA++ (Our Impl.)	Meta-MbPA ($r_{\mathcal{M}} = 1\%$)
Text Classification						
AGNews	0 (Initial)	0.2	0.2	0.2	0.2	0.2
	1 (AGNews)	94.2	94.1	94.0	93.5	94.3
	2 (Yelp)	45.9	78.2	92.4	94.5	94.1
	3 (Amazon)	30.2	62.5	87.9	93.0	93.5
	4 (Yahoo)	0.0	9.2	74.4	92.0	93.1
	5 (DBPedia)	0.0	31.9	80.3	93.0	93.1
Yelp	0 (Initial)	0.2	0.2	0.2	0.2	0.2
	1 (Yelp)	62.5	62.0	62.5	57.7	62.5
	2 (Yahoo)	4.3	32.3	58.1	56.7	61.0
	3 (Amazon)	60.4	61.7	60.1	55.7	61.2
	4 (DBPedia)	48.6	61.4	60.3	58.2	61.4
	5 (AGNews)	11.4	52.4	54.8	57.7	59.6
Question Answering						
QuAC	0 (Initial)	14.1	14.1	14.1	14.1	14.1
	1 (QuAC)	51.8	51.8	51.3	50.8	51.8
	2 (TrWeb)	28.7	37.8	40.4	41.3	51.6
	3 (TrWik)	27.0	35.3	38.8	39.8	50.9
	4 (SQuAD)	40.5	42.9	43.7	44.0	49.9
SQuAD	0 (Initial)	7.5	7.5	7.5	7.5	7.5
	1 (SQuAD)	87.2	87.2	86.6	88.6	86.8
	2 (TrWik)	65.1	79.8	69.6	78.4	85.5
	3 (QuAC)	48.5	70.0	54.4	76.2	79.0
	4 (TrWeb)	66.8	78.8	69.4	81.5	80.4

Table 7.10: **Performance of the first dataset as training progresses for text classification and question answering tasks over different dataset orders and models.** Where applicable, we use $r_{\mathcal{M}} = 100\%$ unless denoted otherwise. “0 (Initial)" denotes model before training on any dataset.

Chapter 8

Weakly Supervised Multimodal Transfer Learning

In the previous part of this thesis, we have explored utilizing explicit alignments to enable knowledge transfer across languages. In this chapter, we are interested to investigate a more complex trasfer learning setting of two less similar domains. Specifically, we study how to mitigate negative transfer between two different modalities: vision and language. In addition, we also study whether it is possible to reduce the requirement of human labeled alignment data in the setting of multimodal learning. In particular, Vision-Language Pretraining (VLP) has achieved impressive performance on many multimodal downstream tasks with recent progress in joint modeling of visual and textual representations. However, the requirement for expensive annotations including clean image captions and regional labels limits the scalability of existing approaches, and complicates the pretraining procedure with the introduction of multiple dataset-specific objectives. In this chapter, we relax these constraints and present a minimalist pretraining framework, named **Sim**ple **V**isual **L**anguage **M**odel (**SimVLM**). Unlike prior work, SimVLM reduces the training complexity by exploiting large-scale weak supervision, and is trained end-to-end with a single prefix language modeling objective. Without utilizing extra data or task-specific customization, the resulting model significantly outperforms previous pretraining methods and achieves new state-of-the-art results on a wide range of discriminative and generative vision-language benchmarks, including VQA (+3.74% vqa-score), NLVR2 (+1.17% accuracy), SNLI-VE (+1.37% accuracy) and image captioning tasks (+10.1% average CIDEr score). Furthermore, we demonstrate that SimVLM acquires strong generalization and transfer ability, enabling zero-shot behavior including open-ended visual question answering and cross-modality transfer. These results show it is possible to train a modality-agnostic language model that not only requires less alignment data but also enables zero-shot transfer in downstream tasks.

8.1 Introduction

Self-supervised textual representation learning (Brown et al., 2020; Devlin et al., 2018; Liu et al., 2019; Radford et al., 2018, 2019; Raffel et al., 2019; Yang et al., 2019) based on Transformers (Vaswani et al., 2017) has pushed the state of the art on a wide range of natural language processing (NLP) tasks (Rajpurkar et al., 2016; Sarlin et al., 2020; Wang et al., 2018). One successful approach is to first pretrain the model (e.g. BERT) on large-scale unlabled text corpora using masked language modeling (MLM) objective (Devlin et al., 2018), followed by finetuning on downstream tasks. While this pretraining-finetuning paradigm has been widely adopted, recent work on autoregressive language models (LM) (Brown et al., 2020; Radford et al., 2019) such as GPT-3 has shown strong performance without finetuning by utilizing few-shot prompts (Liu et al., 2021), suggesting the text guided zero-shot generalization is a promising alternative.

Motivated by the success of textual representation pretraining, various efforts have been made to build the multi-modal (visual and textual) counterpart. A line of work (Chen et al., 2020; Li et al., 2019, 2020; Lu et al., 2019; Su et al., 2020; Tan and Bansal, 2019; Zhang et al., 2021) has explored vision-language pretraining (VLP) that learns a joint representation of both modalities to be finetuned on vision-language (VL) benchmarks, such as visual question answering (VQA) (Goyal et al., 2017). In order to capture the alignment between images and text, previous methods have extensively exploited two types of human-labeled datasets from multiple sources, which typically consist of the following steps. Firstly, object detection datasets are used to train a supervised object detector (OD) which allows further extracting region-of-interest (ROI) features from images. Next, datasets with aligned image-text pairs are used for MLM pretraining of a fusion model that usually takes as input the concatenation of the extracted ROI features and the paired text. In addition, due to the limited scale of human annotated data, various task-specific auxiliary losses have been introduced in order to improve performance. These design choices complicate the pretraining protocol of VLP, creating a bottleneck for further quality improvement. What is more, such pretraining-finetuning based approaches usually lack the zero-shot capability, just like their language counterparts. In comparison, another line of work (Jia et al., 2021; Radford et al., 2021; Ramesh et al., 2021) utilizes weakly labeled/aligned data crawled from the web to perform pretraining, achieving good performance and certain zero-shot learning capability on image classification and image-text retrieval. Nonetheless, these methods mainly focus on specific tasks of consideration and thus may not serve as a generic pretraining-finetuning representation for VL benchmarks.

In light of these disadvantages of the existing techniques, we are interested in building a VLP model that: (1) can be seamlessly plugged into the pretraining-finetuning paradigm and achieve competitive performance on standard VL benchmarks; (2) does not require a complicated pretraining protocol as in previous methods; and (3) has the potential towards text guided zero-shot generalization in cross-modal settings. To this end, we propose **SimVLM**, standing for **Sim**ple **V**isual **L**anguage **M**odel, which significantly simplifies VLP by *solely* exploiting language modeling objectives on weakly aligned image-text pairs (Jia et al., 2021). In a nutshell, SimVLM consists of the following components:

- **Objective**. It is trained end-to-end from scratch with a single objective of Prefix Language Modeling (Raffel et al., 2019), which can not only naturally perform text generation as GPT-3, but also process contextual information in a bidirectional manner as BERT does.
- **Architecture**. The framework employs ViT/CoAtNet (Dai et al., 2021; Dosovitskiy et al., 2021) and directly takes raw images as inputs. These models can also fit the large-scale data and are readily compatible with the PrefixLM objective.
- **Data**. These setups relieve the requirement for object detection and allow the model to utilize the large-scale weakly labeled dataset, which has better potential towards zero-shot generalization.

Not only is SimVLM simpler, requiring neither object detection pretraining nor auxiliary losses, but it also obtains better performance than previous work. Empirically, SimVLM consistently outperforms existing VLP models and achieves new state-of-the-art results on 6 VL benchmarks without additional data nor task-specific customization. Besides, it acquires stronger generalization in visual-language understanding that empowers zero-shot image captioning and open-ended VQA. In particular, SimVLM learns unified multimodal representation that enables zero-shot cross-modality transfer, where the model is finetuned on text-only data and directly evaluated on image-and-text test examples without further training. This chapter indicates that it is possible to attain strong vision-language understanding by a simple pretraining framework with weak supervision only, and builds a step towards text guided zero-shot VL generalization. Notice that our proposed framework is not the PrefixLM objective alone, but a recipe that combines objective, architecture and data together to obtain strong cross-modality generalization.

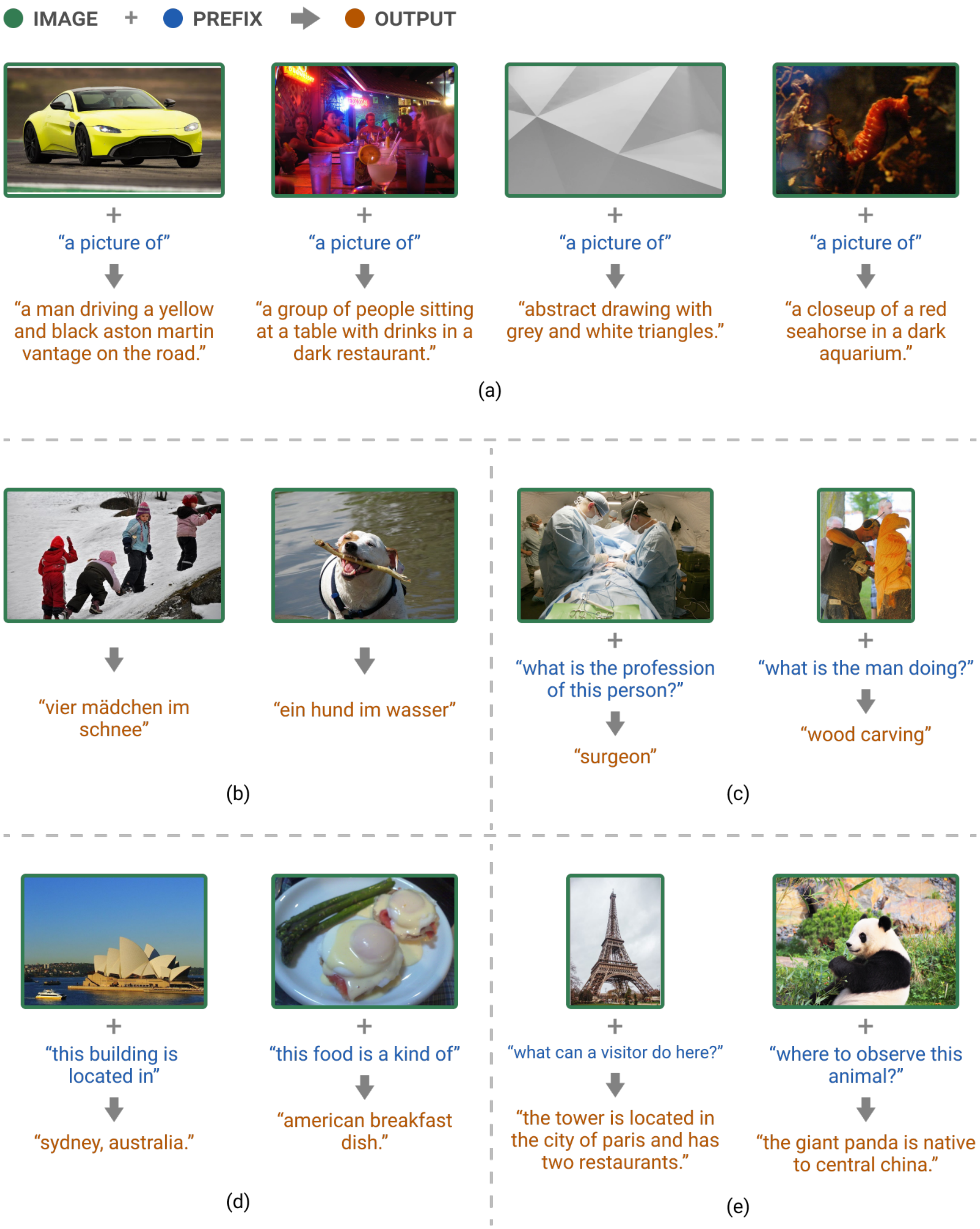


IMAGE
+
PREFIX
OUTPUT
+
"a picture of"
"a man driving a yellow and black aston martin vantage on the road."
+
"a picture of"
"a group of people sitting at a table with drinks in a dark restaurant."
+
"a picture of"
"abstract drawing with grey and white triangles."
+
"a picture of"
"a closeup of a red seahorse in a dark aquarium."
(a)
"vier mädchen im schnee"
"ein hund im wasser"
(b)
+
"what is the profession of this person?"
"surgeon"
+
"what is the man doing?"
"wood carving"
(c)
+
"this building is located in"
"sydney, australia."
+
"this food is a kind of"
"american breakfast dish."
(d)
+
"what can a visitor do here?"
"the tower is located in the city of paris and has two restaurants."
+
"where to observe this animal?"
"the giant panda is native to central china."
(e)

Figure 8.1: Generated examples of SimVLM of various applications: (a) zero-shot image captioning (b) zero-shot cross-modality transfer on German image captioning (c) generative VQA (d) zero-shot visual text completion (e) zero-shot open-ended VQA.

8.2 Related Work

Recent years have seen a rapid progress made in vision-language pretraining (Han et al., 2021; Khan et al., 2021; Uppal et al., 2020). While a variety of approaches have been proposed, a large portion of them require object detection for image region feature regression or tagging as part of the pre-training objectives, for example LXMERT (Tan and Bansal, 2019), VLBERT (Su et al., 2020), VisualBERT (Li et al., 2019), UNITER (Chen et al., 2020), Villa (Gan et al., 2020), Oscar (Li et al., 2020), ERNIE-ViL (Yu et al., 2021), UNIMO (Li et al., 2021), VinVL (Zhang et al., 2021), VIVO (Hu et al., 2021) VL-T5 (Cho et al., 2021) etc. These methods rely on a strong object detection model like Fast(er) R-CNN (Ren et al., 2015), which is often trained on human annotated data sets like Visual Genome (Krishna et al., 2016). Using such labeled training data as a prerequisite increases the cost of building the training pipeline, and makes the approach less scalable. Some recent efforts have also explored VLP without object detection module (Huang et al., 2021; Kim et al., 2021; Xu et al., 2021), but they only use clean pretraining data with small scales and thus their zero-shot capability is limited.

On the other hand, multiple cross-modality loss functions have been proposed as part of the training objectives, for example image-text matching (Lu et al., 2019; Tan and Bansal, 2019; Xu et al., 2021), masked region classification/feature regression (Chen et al., 2020; Tan and Bansal, 2019), object attribute prediction (Xu et al., 2021), contrastive loss (Li et al., 2021, 2020), word-region alignment (Chen et al., 2020) word-patch alignment (Kim et al., 2021). They are often mixed with other objectives including image caption generation and masked language modeling to form compound pre-training losses. This creates the challenge of balancing among different losses and datasets, and thus complicates the optimization procedure.

Our work by contrast, follows a minimalist approach that takes raw image inputs and makes use of only the language modeling loss, without resorting to auxiliary models like faster R-CNN for image region detection. Motivated by recent works (Jia et al., 2021; Radford et al., 2021; Ramesh et al., 2021) that illustrate zero-shot learning in certain image-text tasks, we train our model using large-scale weakly labeled data only. As we will detail in follow-up sections, such a simple setup is sufficiently effective and reaches state-of-the-art results on multiple downstream tasks.

8.3 Proposed Framework

8.3.1 Background

The bidirectional **Masked Language Modeling (MLM)** has been one of the most popular self-supervised training objectives for textual representation learning. As demonstrated by BERT (Devlin et al., 2018), it is based on the idea of denoising autoencoder such that the model is trained to recover the corrupted tokens in a document. Specifically, given a text sequence $\mathbf{x}$, a subset of tokens $\mathbf{x}_m$ are randomly sampled and a corrupted sequence $\mathbf{x}_{\backslash m}$ is constructed by replacing tokens in $\mathbf{x}_m$ with a special `[MASK]` token. The training objective is to reconstruct $\mathbf{x}_m$ from the context $\mathbf{x}_{\backslash m}$ by minimizing the negative log-likelihood:

$$\mathcal{L}_{\text{MLM}}(\theta) = -\mathbb{E}_{\mathbf{x}\sim D}\left[\log P_\theta(\mathbf{x}_m|\mathbf{x}_{\backslash m})\right], \tag{8.1}$$

where θ is the trainable parameters of the model and D is the pretraining data. This approach learns contextualized representations that can be further finetuned for downstream tasks. The MLM-style pretraining has been widely adopted in previous VLP models, whereby the input is an image-text pair and the model needs to predict masked tokens by leveraging image ROI features.

Alternatively, the unidirectional **Language Modeling (LM)** trains the model to directly maximize the likelihood of the sequence $\mathbf{x}$ under the forward autoregressive factorization:

$$\mathcal{L}_{\text{LM}}(\theta) = -\mathbb{E}_{\mathbf{x}\sim D}\left[\log P_\theta(\mathbf{x})\right] = -\mathbb{E}_{\mathbf{x}\sim D}\left[\sum_{t=1}^{T}\log P_\theta(\mathbf{x}_t|\mathbf{x}_{<t})\right]. \tag{8.2}$$

Compared with MLM, the LM pretraining has also been shown to be highly effective for multiple NLP tasks (Radford et al., 2018). More importantly, it facilitates the model with strong generation capability that enables text induced zero-shot generalization without finetuning (Brown et al., 2020). While MLM has become the de facto approach in VLP models reviewed above, the generative LM has been understudied.

8.3.2 Proposed Objective: Prefix Language Modeling

Motivated by the zero-shot capability introduced by pre-training with LM loss, we propose to pretrain vision-language representation using the **Prefix Language Modeling (PrefixLM)**. PrefixLM differs from the standard LM such that it enables bi-directional attention on the prefix

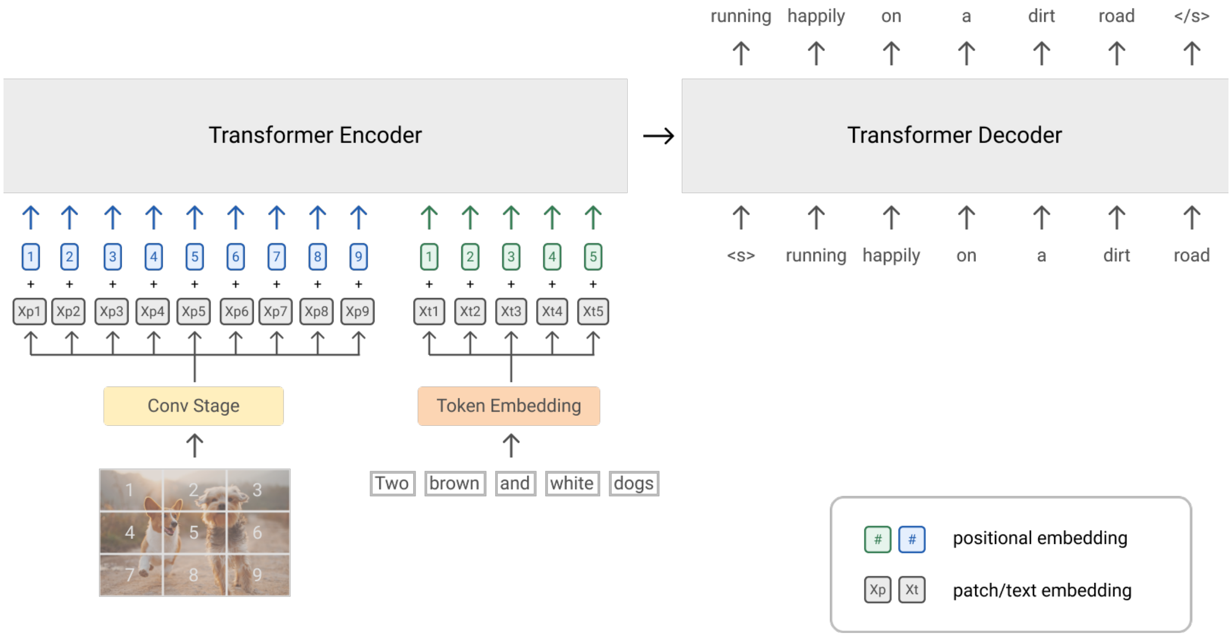


Figure 8.2: Illustration of the SimVLM model. This shows an example of training with PrefixLM of an image-text pair. For text-only corpora, it is straightforward to remove the image patches and utilize textual tokens only.

sequence (e.g. $\mathbf{x}_{<T_p}$ in Eq. (8.3)), and only conducts autoregressive factorization on the remaining tokens (e.g. $\mathbf{x}_{\geq T_p}$ in Eq. (8.3)). During pretraining, a prefix sequence of tokens of (a randomly selected) length T_p is truncated from input sequence and the training objective becomes:

$$\mathcal{L}_{\text{PrefixLM}}(\theta) = -\mathbb{E}_{\mathbf{x}\sim D}\left[\log P_\theta(\mathbf{x}_{\geq T_p}|\mathbf{x}_{<T_p})\right] = -\mathbb{E}_{\mathbf{x}\sim D}\left[\sum_{t=T_p}^{T} P_\theta(\mathbf{x}_t|\mathbf{x}_{[T_p,t]}, \mathbf{x}_{<T_p})\right]. \tag{8.3}$$

Intuitively, images can be considered as prefix for their textual descriptions as they often appear before text in a web document. Therefore, for a given image-text pair, we prepend image feature sequence of length T_i to the text sequence, and enforce the model to sample a prefix of length $T_p \geq T_i$ to calculate LM loss on text data only (an example is shown in Figure 8.2). Compared to prior MLM style VLP methods, our PrefixLM model under the sequence-to-sequence framework not only enjoys the bidirectional contextualized representation as in MLM, but also can perform text generation similar to LM.

8.3.3 Architecture

We adopt Transformer as the backbone of our model due to its success for both language and vision tasks (Devlin et al., 2018; Dosovitskiy et al., 2021). Differently from standard LM, PrefixLM enables bidirectional attention within the prefix sequence, and thus it is applicable for both decoder-only and encoder-decoder sequence-to-sequence language models. In our preliminary experiments, we found that the inductive bias introduced by encoder-decoder model which decouples encoding from generation is conducive to the improvement of downstream task.

An overview of our model architecture is depicted in Figure 8.2. For the visual modality, inspired by ViT (Dosovitskiy et al., 2021) and CoAtNet (Dai et al., 2021), our model receives the raw image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ and maps it into flattened 1D sequence of patches $\mathbf{x}_p \in \mathbb{R}^{T_i \times D}$ as input for the transformer, where D is the fixed hidden size of the transformer layers and $T_i = \frac{HW}{P^2}$ is the length of the image tokens for a given patch size P. Following (Dai et al., 2021), we use a convolution (Conv) stage consist of the first three blocks of ResNet (He et al., 2016) to extract contextualized patches, which we find advantageous over the naive linear projection (equivalent to 1×1 Conv layer) used in ViT, consistent with the observation from (Xiao et al., 2021). For the textual modality, we follow the standard practice to tokenize the input sentence into sub-word tokens (Kudo and Richardson, 2018), and the embeddings are learned for a fixed vocabulary. To retain positional information, we add two trainable 1D positional embeddings for image and text inputs separately, and we additionally add 2D relative attention for the image patches within transformer layers. Notice that we do not add extra modality type embeddings for which we found no improvement in our experiment. We study the effects of various components of the model in Section 8.4.4.

8.3.4 Datasets

Since our approach does not rely on an object detection module and only operates with raw image patch inputs. we pretrain all model parameters from scratch using large-scale noisy image-text data, which has better potential for zero-shot generalization. Specifically, we use the image and alt-text pairs introduced in (Jia et al., 2021), which are crawled from the web with minimal post-processing. On the other hand, our formulation of PrefixLM is modality-agnostic and thus we can additionally include text-only corpora to compensate for noisy text supervision in the alt-text data. As shown later in our experiments, this unified PrefixLM formulation reduces the modality discrepancy and improves the model quality.

Compared to prior VLP methods consisting of two pretraining stages and multiple auxiliary objectives, our model only requires one-pass pretraining using a single language modeling loss in an end-to-end manner, hence the name Simple Visual Language Model (SimVLM).

8.4 Experiments

We conduct systematic experiments on a diversified set of visual-linguistic benchmarks, including visual question answering, image captioning, visual reasoning, visual entailment, and multimodal translation. We not only examine our model as a general-purpose VL representation learning in the pretraining-finetuning paradigm, but also study its zero-shot generalization towards open-ended VL understanding.

8.4.1 Setup

Pretraining

Our models are implemented with the Lingvo framework (Shen et al., 2019). We follow the setup in ViT (Dosovitskiy et al., 2021) to explore 3 variants of SimVLM, namely "Base", "Large", and "Huge". For the Transorformer, each variant follows the same setting as its corresponding ViT variant. For the Conv stage, we use the first three blocks (excluding the Conv stem) of ResNet-101 and ResNet-152 (He et al., 2016) for our Base and Large models respectively, and a larger variant of ResNet-152 with more channels for Huge. During pretraining, we utilize the 224×224 resolution with a fixed patch size of 16×16, resulting in a patch sequence of length 14×14 as visual tokens. For the textual input, we use a vocabulary size of 32,000 and a max sequence length of 256 in both the encoder and decoder. We also share parameters between the embedding and the decoder softmax output layer (Press and Wolf, 2016).

We pretrain on large-scale web datasets for both image-text and text-only inputs. For joint vision and language data, we exploit the training set of ALIGN (Jia et al., 2021), which contains about 1.8B noisy image-text pairs. Notice that we do not use any extra data preprocessing or filtering, except simple random resized cropping. For the text-only copora, we use the Colossal Clean Crawled Corpus (C4) dataset presented in (Raffel et al., 2019) and followed their preprocessing steps. The dataset contains about 800GB of web crawled documents.

All model are pretrained for about 1M steps from scratch. We optimize with the AdamW optimizer (Loshchilov and Hutter, 2017) with $\beta_1 = 0.9, \beta_2 = 0.999$ and weight decay of 0.01.

We warm up the learning rate for the first 2% of updates to a peak value of 5×10^{-4}, and then linearly decay it afterwards. We mix the two pretraining datasets within each batch, which contains 4,096 image-text pairs and 512 text-only documents, sharded across 512 TPU v3 chips (Jouppi et al., 2017).

Finetuning

After pretraining, our model is finetuned on 5 types of downstream tasks with minimal overhead:

Visual question answering: This task requires the model to answer questions about input images, and has been the most widely used VL benchmark. Following prior work, we use the VQA v2 (Goyal et al., 2017) and formulate the task as a classification problem over 3,129 most frequent answers in the training set. The raw image and the corresponding question are used as inputs to the encoder and the decoder respectively, and a task-specific linear classifier is trained to predict answer based on activation corresponding to the last question token from the decoder. We use a resolution of 480×480 for the image and all positional parameters are adapted using linear interpolation.

Visual entailment: The SNLI-VE (Xie et al., 2019a) dataset is adapted from SNLI (Bowman et al., 2015), which is originally designed to predict the relation between a premise sentence and a hypothesis sentence as either entailment, neutral or contradiction, a task known as natural language inference (NLI). For the VL variant, the premise is based on the content of an image rather than textual descriptions. We finetune SimVLM similarly to VQA, such that the image and the sentence are fed to encoder and decoder separately, and the classifier is trained to predict the three relations.

Visual reasoning: The NLVR2 (Suhr et al., 2018) dataset tests the model's ability of jointly reasoning over the language and multiple images by asking whether a textual description is true based on a pair of two images. Following (Zhang et al., 2021), we create two input pairs, each consisting of one image and the textual description, and generate output embeddings for both using the same setup above. The two embeddings are then concatenated for final prediction.

Image captioning: The captioning task requires a model to generate natural language descriptions of input images. We consider two datasets CoCo (Chen et al., 2015) and NoCaps (Agrawal et al., 2019), both finetuned using the CoCo training data. For SimVLM, it is straightforward to first encode the image in the encoder and then generate captions using the decoder. Note that in contrast to prior work that apply task-specific tricks such as CIDEr optimization (Rennie et al., 2017), our model is trained with naive cross-entropy loss only.

	VQA		NLVR2		SNLI-VE		CoCo Caption				NoCaps		Multi30k
	test-dev	test-std	dev	test-P	dev	test	B@4	M	C	S	C	S	En-De
LXMERT	72.42	72.54	74.90	74.50	-	-	-	-	-	-	-	-	-
VL-T5	-	70.30	74.6	73.6	-	-	-	-	116.5	-	-	-	45.5
UNITER	73.82	74.02	79.12	79.98	79.39	79.38	-	-	-	-	-	-	-
OSCAR	73.61	73.82	79.12	80.37	-	-	**41.7**	30.6	140.0	24.5	80.9	11.3	-
Villa	74.69	74.87	79.76	81.47	80.18	80.02	-	-	-	-	-	-	-
SOHO	73.25	73.47	76.37	77.32	85.00	84.95	-	-	-	-	-	-	-
UNIMO	75.06	75.27	-	-	81.11	80.63	39.6	-	127.7	-	-	-	-
VinVL	76.56	76.60	82.67	83.98	-	-	41.0	31.1	140.9	25.2	92.5	13.1	-
SimVLM$_{\text{base}}$	77.87	78.14	81.72	81.77	84.20	84.15	39.0	32.9	134.8	24.0	94.8	13.1	46.6
SimVLM$_{\text{large}}$	79.32	79.56	84.13	84.84	85.68	85.62	40.3	33.4	142.6	24.7	108.5	14.2	47.5
SimVLM$_{\text{huge}}$	**80.03**	**80.34**	**84.53**	**85.15**	**86.21**	**86.32**	40.6	**33.7**	**143.3**	**25.4**	**110.3**	**14.5**	**47.6**

Table 8.1: Single model results for vision-language pretraining methods on popular VL banchmarks. We report vqa-score for VQA, accuracy for NLVR2 and SNLI-VE, BLEU@4 for Multi30k and various metrics for image captioning (B@4: BLEU@4, M: METEOR, C: CIDEr, S: SPICE).

Multimodal translation: The goal of multimodal translation is to translate image descriptions in source language to target language, for which image inputs can be taken advantage of as grounding signal. We train and evaluate on the Multi30k (Elliott et al., 2016) dataset. We utilize the PrefixLM described in previous sections such that the source sentence, together with the image inputs, are fed to the encoder, which will be translated to the target language by the decoder.

8.4.2 Comparison with existing approaches

To examine the quality of vision-language pretraining, we first compare SimVLM on the popular multi-modal tasks described in Sec. 8.4.1 with state-of-the-art (SOTA) VLP methods including LXMERT (Tan and Bansal, 2019), VL-T5 (Cho et al., 2021), UNITER (Chen et al., 2020), OSCAR (Li et al., 2020), Villa (Gan et al., 2020), SOHO (Huang et al., 2021), UNIMO (Li et al., 2021), and VinVL (Zhang et al., 2021).

As can be seen in Table 8.1, SimVLM outperforms all existing models and achieves new SOTA results on all tasks considered, often by a significant margin. This demonstrates our generative pretraining approach is very effective and that simple framework with weak supervision is sufficient to learn high-quality multi-modal representations.

For the discriminative tasks, the SimVLM$_{\text{base}}$ already outperforms all prior methods while using less capacity, and the SimVLM$_{\text{huge}}$ obtains almost 4 points absolute score improvement compared to the previous SOTA (VinVL), pushing the single model performance above 80% on VQA for the first time. In addition, SimVLM also consistently outperforms prior methods on NLVR2 and SNLI-VE, illustrating its capability of processing more complex visual-linguistic reasoning. For the generation tasks including image captioning and image translation, SimVLM also shows large improvements using naive finetuning techniques. Our model outperforms on 3 out of 4 metrics on the public "Karpathy" 5k test split of CoCo captioning as well as the NoCaps benchmark than prior methods trained with more complex reinforcement learning approach of CIDEr optimization (Rennie et al., 2017). Finally, SimVLM is also effective for image translation of Multi30k from English to German. These experiments demonstrate that our model can be seamlessly plugged into the pretraining-finetuning paradigm with superior performance, utilizing minimalist pretraining and finetuning procedures.

8.4.3 Zero-Shot Generalization

A crucial benefit of generative modeling and scaling with weak supervision is the potential of zero-shot generalization. Models (Brown et al., 2020; Jia et al., 2021; Radford et al., 2021) have been shown capable of performing few-shot or zero-shot transfer from pretrained models to downstream datasets, even across language boundaries (Lample and Conneau, 2019). In this section, we examine the generalization of SimVLM and showcase three applications less explored in prior VLP work.

Zero-shot Image Captioning

The pretraining procedure of SimVLM can be interpreted as a noisy image captioning objective on real-world web corpus. Thus, it is natural to ask how well this caption ability generalizes to other datasets in a zero-shot manner. To this end, we take the pretrained SimVLM model and directly decode on image captioning benchmarks without further finetuning on their clean human-labeled data. We follow the same hyperparameter settings of image resolution and beam search in the fully supervised setup. Besides, we also found that using a prefix prompt "`A picture of`" improves the quality of decoded captions, similar to the finding in (Radford et al., 2021).

As shown in Table 8.2, the zero-shot performance of SimVLM is competitive with fully

	Pre.	Sup.	CoCo Caption				NoCaps			
			B@4	M	C	S	In	Near	Out	Overall
BUTD^{a†}		✔	36.3	27.7	120.1	21.4	-	-	-	-
AoANet^{b†}			39.5	29.3	129.3	23.2	-	-	-	-
M2 Transformer^{c†}			39.1	29.2	131.2	22.6	81.2	-	69.4	75.0
SimVLM$_{\text{base}}$	✔		27.1	26.8	96.3	20.1	83.2	84.1	82.5	83.5
SimVLM$_{\text{large}}$			29.3	27.5	101.4	21.5	97.6	96.5	96.3	96.6
SimVLM$_{\text{huge}}$			29.7	27.8	102.3	22.1	101.2	100.4	102.3	101.4
OSCAR†	✔	✔	**41.7**	30.6	140.0	24.5	85.4	84.0	80.3	83.4
VinVL†			41.0	31.1	140.9	25.2	103.7	95.6	83.8	94.3
SimVLM$_{\text{huge}}$			40.6	**33.7**	**143.3**	**25.4**	**113.7**	**110.9**	**115.2**	**112.2**

Table 8.2: Image captioning results on CoCo Karpath-test split and NoCaps validation split (zero-shot and finetuned). "Pre." indicates the model is pretrained and "Sup." means the model is fine-tuned on task-specific supervision. For NoCaps, {In, Near, Out} refer to in-domain, near-domain and out-of-domain respectively. † indicates Cider optimization. Model references: [a]Anderson et al. (2018) [b]Huang et al. (2019b) [c]Cornia et al. (2020).

supervised baselines on CoCo, and it also demonstrates strong generalization on the concept-rich NoCaps benchmark by achieving better scores than pretrained models.

Figure 8.1 (a) illustrates sample captions generated by our model. SimVLM is able to not only capture real-world concepts but also provide a detailed description of the visual input. For example, the decoded samples are able to explain complex scenes with multiple objects (e.g. "people", "table with drinks", "dark restaurant"). Besides, the model also shows understanding of fine-grained abstraction such as specific car brand and model (e.g. "Aston Martin", "Vantage"). SimVLM even performs robustly on challenging images that could be tricky for human, such as abstract or dark pictures. These all illustrate that our model learns a wide range of real-world concepts that generalize well in a zero-shot manner.

Zero-shot cross-modality Transfer

Existing pretraining methods have been shown to be successful in transferring knowledge across heterogeneous data spaces. For example, multilingual language models (Devlin et al., 2018;

Lample and Conneau, 2019) enable zero-shot cross-lingual transfer such that the model is only finetuned using training data from a source language (typically English) and evaluated on the target language without further training. Inspired by this setup, we explore a novel zero-shot cross-modality transfer paradigm of utilizing VLP models, and evaluate how well our model generalizes across modalities. Since text training data are usually cheaper to obtain compared to visual data, we finetune SimVLM on text-only downstream data and then directly evaluate the zero-shot transfer on joint VL tasks.

Specifically, We utilize SNLI-VE and Multi30k to examine the zero-shot transfer performance. For SNLI-VE, we finetune on three text-only NLI datasets such that the premise sentence is used as the encoder's input while the hypothesis is fed to the decoder, and a similar classifier head is trained on the embedding of the last token in the decoder. At inference, the finetuned model is evaluated by taking the premise image as the encoder input and the corresponding hypothesis sentence to the decoder. As shown in Table 8.3, SimVLM performs competitively with fully supervised baselines including UNITER under the zero-shot setting. As a sanity check, we also mask out the image feature and predict using the hypothesis only (denoted as "Image Masked" in Table8.3, which is the average number given by all three versions of SimVLM). This results in performance close to random guess therefore demonstrating the effectiveness of SimVLM's cross-modality transfer capability.

In addition, SimVLM is also capable of domain adaption by transferring from the MNLI dataset to SNLI-VE, whereby data comes not only from a different modality but also another domain. We also find it possible to transfer across different languages and modalities using SimVLM. Specifically, we utilize the German image captioning task from WMT 2016 of Multi30k for evaluation, where our model is finetuned on English-German text-only translation data followed by decoding with image-only input in the encoder. Table 8.3 shows that SimVLM is capable of transferring knowledge across modalities and languages in generative tasks, achieving comparable performance to supervised baselines (decoded examples shown in Figure 8.1 (b)). These results suggest zero-shot cross-modality transfer emerges with the scaling of weakly labeled data.

Open-ended VQA

On the VQA benchmark, the best performing models to date formulate the problem as a discriminative task of multi-label classification over a predefined 3,129 answer candidates, often consisting of short factual terms. In real-world applications, however, it is hard to define a closed

	SNLI-VE				Multi30k	
	SNLI-VE (T)	SNLI	MNLI	Image Masked	B@4	M
Fully Supervised Baseline						
EVE-Image	71.56 / 71.16				-	-
UNITER	78.59 / 78.28				-	-
SOHO	85.00 / 84.95				-	-
LIUM[a]	-				23.8	35.1
GroundedTrans[a]	-				15.8	31.2
Zero-Shot Cross-Modality Transfer						
SimVLM$_{\text{base}}$	71.35 / 71.02	72.65 / 72.24	64.37 / 63.98		15.0	24.8
SimVLM$_{\text{large}}$	72.85 / 72.44	73.62 / 73.23	66.97 / 66.31	34.31 / 34.62	17.7	30.1
SimVLM$_{\text{huge}}$	73.56 / 73.08	74.24 / 73.86	67.45 / 66.97		18.2	32.6

Table 8.3: Zero-shot cross-modality transfer results on SNLI-VE and Multi30k. For SNLI-VE, the zero-shot model is finetuned on three source datasets: text-only SNLI-VE (Xie et al., 2019a), SNLI (Bowman et al., 2015), and MNLI (Williams et al., 2017). Model reference: [a](Specia et al., 2016).

set of candidate answers that covering all possible scenarios, making the true open-ended VQA a challenging setup. Generative models such as SimVLM provide an alternative solution towards this challenge by generating free-form textual answers without being constrained to predefined answers. To this end, we finetune SimVLM using the PrefixLM loss described above where we treat the concatenation of the image and the question as the prefix, and train the model to generate answers.

We then compare the generative approach with classification methods in Table 8.4. Firstly, we follow Cho et al. (2021) and evaluate model performance on questions with rare answers in the Karpathy-test split. Here, out-of-domain questions are defined as those with best-scoring answer not included in the 3,129 candidates. Results show that SimVLM outperforms both discriminative and generative baselines on all splits. More importantly, the generative SimVLM significantly improves on the out-of-domain split by over 17 points, demonstrating its strong generalization. However, this setup mainly focuses on rare answers and it remains unclear how well the model generalizes to common unseen answers. We therefore proceed to investigate a more challenging setup where we randomly select 2,085 (about two-thirds of 3,129) in-domain

	Dev	Karpathy-test			Partial Train		
		In-domain	Out-domain	Overall	In-domain	Out-domain	Overall
Discriminative							
UNITER	-	74.4	10.0	70.5	-	-	-
VL-T5	-	70.2	7.1	66.4	-	-	-
VL-BART	-	69.4	7.0	65.7	-	-	-
SimVLM$_{\text{base}}$	73.8	79.0	16.7	75.3	78.4	10.3	70.5
SimVLM$_{\text{large}}$	76.0	80.4	17.3	76.7	79.5	11.0	71.8
SimVLM$_{\text{huge}}$	**76.5**	**81.0**	17.5	**77.2**	**80.2**	11.1	72.2
Generative							
VL-T5	-	71.4	13.1	67.9	-	-	-
VL-BART	-	72.1	13.2	68.6	-	-	-
SimVLM$_{\text{base}}$	73.2	78.3	25.8	75.2	77.1	27.1	71.3
SimVLM$_{\text{large}}$	75.2	79.5	29.6	76.5	78.7	28.4	72.5
SimVLM$_{\text{huge}}$	75.5	79.9	**30.3**	77.0	79.1	**28.8**	**73.0**

Table 8.4: Comparison of discriminative and generative VQA methods. “Dev” refers to standard vqa-score on the VQA validation split. “Karpathy-test” is the setup used in (Cho et al., 2021) for evaluation on the Karpath split with rare answers. “Partial Train” refers to train the model only on partial training data which contain subset of all candidate answers.

answers and partition both train and validation sets into two splits based on whether their best-scoring answers are included in the selected set or not. We then only finetune SimVLM on the in-domain split of the train set and evaluate on the entire validation set. The “Partial Train” column in Table 8.4 shows that the generative SimVLM is also competent in this setup by scoring reasonably well on over 1,000 unseen answers. Overall, we found the generative SimVLM performs competitively with its discriminative counterpart in the standard setup, and works generally better in the out-of-domain case.

Note that we use the exact matching between generated answers and human labels for score calculation in the above experiment, however it is possible that the model generates appropriate answers in different formats or synonyms. Therefore, in addition to the quantitative study above, we show qualitative generation results in Figure 8.1 (c). It can be observed that SimVLM is able to generate answers not included in the 3,129 candidate set (e.g. “surgeon” and “wood carving”),

	CoLA	SST-2	RTE	MRPC	QQP	MNLI	QNLI	WNLI
BERT	**54.6**	**92.5**	62.5	**81.9/87.6**	**90.6/87.4**	**84.2**	**91.0**	48.8
VisualBERT	38.6	89.4	56.6	71.9/82.1	89.4/86.0	81.6	87.0	53.1
UNITER	37.4	89.7	55.6	69.3/80.3	89.2/85.7	80.9	86.0	55.4
VL-BERT	38.7	89.8	55.7	70.6/81.8	89.0/85.4	81.2	86.3	53.1
VilBERT	36.1	90.4	53.7	69.0/79.4	88.6/85.0	79.9	83.8	55.4
LXMERT	39.0	90.2	57.2	69.8/80.4	75.3/75.3	80.4	84.2	46.0
SimVLM$_{\text{base}}$	46.7	90.9	**63.9**	75.2/84.4	90.4/87.2	83.4	88.6	**58.1**

Table 8.5: Text-only task performance on the GLUE benchmark (Dev set). Results for BERT and other VLP methods are obtained from (Iki and Aizawa, 2021). The overall best result is **bolded** while underline signifies the best VLP model.

demonstrating that SimVLM can transfer knowledge from the pretraining corpus to VQA. It is thus natural to ask whether SimVLM can perform zero-shot VQA without finetuning at all. In our experiments, we found that SimVLM is able to "answer" by completing prompting sentences, as shown in Figure 8.1 (d). Nonetheless, we also observed that the model falls short in generating meaningful answers to the real questions. We hypothesize that this is due to the low quality of the pretraining data in which most textual descriptions are short and noisy. To verify our assumption, we continue the pretraining process on the cleaner WIT dataset (Srinivasan et al., 2021) for 50k steps. Examples in Figure 8.1 (e) show that open-ended VQA ability emerges in SimVLM such that it can generate related responses after finetuning on the knowledge-rich wikipedia dataset. Our experiments show promising signs towards zero-shot open-ended VQA and illustrate the potential of generative modeling as future research direction.

8.4.4 Analysis

Single-Modality Tasks

Since SimVLM performs well on joint vision-language benchmarks, it is natural to ask how well the learned representations perform on tasks of single modality. We hope to gain deeper insights into the model behavior by examining its performance on these benchmarks, but it is not our intention to achieve state-of-the-art on single-modality tasks.

Method	Acc@1
SimCLRv2	79.8
DINO	80.1
CLIP	85.4
ALIGN	**85.5**
SimVLM$_{\text{base}}$	80.6
SimVLM$_{\text{large}}$	82.3
SimVLM$_{\text{huge}}$	83.6

Table 8.6: Linear evaluation on ImageNet classification, compared to state-of-the-art representation learning methods.

In Table 8.5, we compare SimVLM with existing VLP models on the GLUE benchmark (Wang et al., 2018), where we mainly follow the text processing procedure in (Raffel et al., 2019) and train our model to classify the fully formatted input without token type embeddings. SimVLM performs better than existing VLP methods and competitively with BERT, indicating that it has good language understanding ability. Additionally, we also compute the top-1 accuracy on ImageNet following the linear evaluation protocol in Table 8.6. Note that our model is not pretrained with a discriminative task such as the contrastive loss, hence we use an average pooling of encoder outputs as image features. Results verify that our model has also learned high-quality image representation. In summary, these experiments show that SimVLM is able to capture modality-specific information effectively, laying a solid foundation for stronger visual-linguistic understanding with weak supervision.

Ablation study

To study the contributions from each model component, we conduct ablation study on SimVLM$_{\text{small}}$ models with an embedding dimension of 512 and 8 layers, with different setups. We make comparisons on VQA and zero-shot image captioning tasks in Table 8.7. First, we compare encoder-decoder models with decoder-only models of comparable model size. We find that although decoder-only model performs competitively on generative tasks, it performs significantly worse on discriminative tasks such as VQA. This suggest the inductive bias of separating bidirectional encoding from unidirectional decoding is beneficial for joint VL representation learning. Next, we study the effectiveness of pretraining objectives by changing the training loss for the text-only

Method	VQA	Zero-Shot Caption
No Pretraining	49.70	-
Decoder-only	65.23	18.0 / 67.9
w/ LM	64.48	17.7 / 63.4
SimVLM$_{\text{small}}$	67.43	18.2 / 68.3
w/o Image2Text	49.23	-
w/o Text2Text	65.25	15.4 / 64.2
w/o conv stage	63.11	17.2 / 62.6
w/ span corruption	66.23	17.4 / 66.2
w/ 2 conv blks	65.57	17.6 / 65.3
w/ 4 conv blks	66.55	17.9 / 67.8

Table 8.7: Ablation study on VQA and image captioning. We compare SimVLM with its decoder-only counterpart and random initialization. “w/ LM” and “w/ span corruption” denote replacing the proposed PrefixLM loss with a different pretraining objective. “Image2Text” and “Text2Text” refer to the noisy image-text data and the text-only data used for pretraining. Finally, we also experiment with various convolution stage (“conv stage”) architecture using either the first 2 blocks (“2 conv blks”) or the first 4 blocks (“4 conv blks”) of ResNet.

data. Results show that the PrefixLM objective outperforms both span corruption (Raffel et al., 2019) and naive LM, illustrating the importance of using a unified objective formulation for both image-text and text-only data. Moreover, we ablate the contribution of the two types of datasets by comparing models trained using either of them or neither. While weakly supervised image-text data are required for bridging the gap between visual and textual representations, text-only corpora also improves the model quality, especially on generative tasks. This is probably because textual signals are extremely noisy in the former and thus the model relies on the text-only data to learn better language understanding. Finally, we study the effect of the convolution stage and find it critical for VL performance. Following (Dai et al., 2021), we experiment with using either the first 2/3/4 ResNet Conv blocks, and empirically observe that the 3 conv block setup works best. This indicates that image and text have different levels of representation granularity and thus utilizing contextualized patches is beneficial.

8.5 Summary

In this chapter, we present a simple yet effective framework of vision-language pretraining. Unlike prior works using object proposal systems and auxiliary losses, our model processes whole image as patches and is trained end-to-end with a single prefix language modeling objective. The proposed model mitigates negative transfer between image and text such that it outperforms prior methods on within-modality tasks. More importantly, on various vision-language benchmarks, the proposed approach not only obtains state-of-the-art performance, but also exhibits intriguing zero-shot behaviors in visual-and-textual understanding. However, we notice that models compared in this chapter are pretrained on different datasets with various scales. Since some datasets are not publicly available due to data sensitivity, it is hard to conduct fair comparison to well examine the algorithmic improvements of these methods, including ours. Similar to many prior work in this domain, our proposed method is a combination of training objective, model architecture and pretraining dataset, rather than algorithmic design alone. Nonetheless, as shown in our ablation studies, different algorithmic design choices can have significant impacts on the model quality, and thus we call for more careful and fair comparison in this aspect for future work. This shows that it is possible to reduce the requirement of human labeled alignment data to enable knowledge transfer across modality boundaries. At the same time, this chapter also builds a step towards zero-shot learning in VLP. In the final chapter, we further push our study of utilizing less human supervision to the extreme, and investigate the setting where no human label is used at all.

Chapter 9

Zero-Label Language Learning

In the previous chapter, we show that it is possible to mitigate data discrepancy across modalities with proper data/model scaling and obtain improved sample efficiency. In this final chapter of the thesis, we further scale the pretrained textual language model and find that scaling deep models can alleviate negative transfer in natural language tasks such that the model is capable of generalizing to a wide range of NLP tasks without finetuning. Specifically, we show that it is possible to achieve zero-label learning in NLP, whereby no human-annotated data is used anywhere during training and models are trained purely on synthetic data. At the core of our framework is a novel approach for better leveraging the powerful pretrained language models. Specifically, inspired by the recent success of few-shot inference on GPT-3 (Brown et al., 2020), we present a training data creation procedure named Unsupervised Data Generation (UDG), which leverages few-shot prompts to synthesize high-quality training data without real human annotations. Our method enables zero-label learning as we train task-specific models solely on the synthetic data, yet we achieve better or comparable results from strong baseline models trained on human-labeled data. Furthermore, when mixed with labeled data, our approach serves as a highly effective data augmentation procedure, achieving new state-of-the-art results on the SuperGLUE benchmark[1]. These results show giant language models can obtain some degree of universal language understanding capability, such that it can enable highly efficient knowledge transfer to downstream tasks.

[1] Notably, our method is also the first to surpass human performance as of Dec 20, 2020.

Model	Setting	SuperGLUE Avg.
Human		89.8
Previous SOTA	Supervised	89.3
T5+UDG		**90.4**
GPT3	Few-Shot	71.8
UDG	Unsupervised	**78.1**

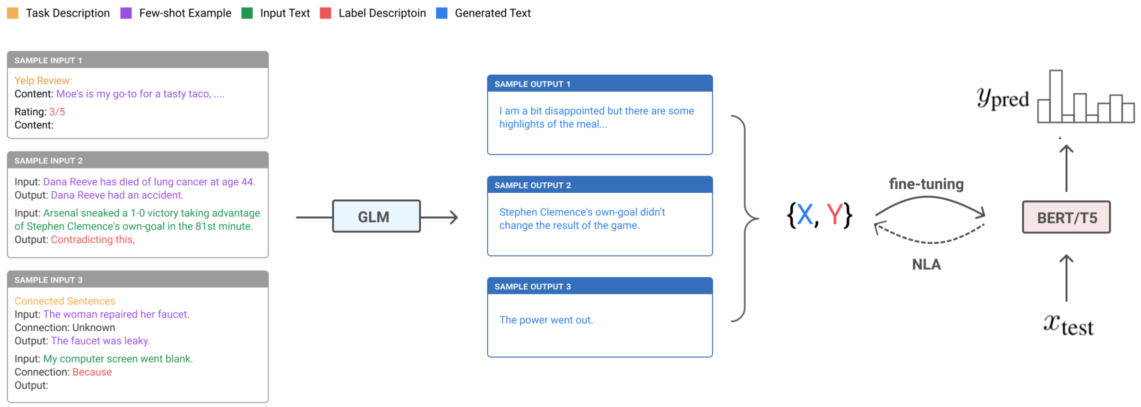
Table 9.1: SuperGLUE summary.

9.1 Introduction

It is well-known that deep learning models are data-hungry. In natural language processing, language model pre-training has become a successful transfer learning approach to effectively reduce the requirement for task-specific labeled data (Brown et al., 2020; Devlin et al., 2018; Liu et al., 2019; Radford et al., 2019; Raffel et al., 2019; Yang et al., 2019). Via training on unsupervised large-scale text corpus, bi-directional language models such as BERT and XLNet are able to learn contextualized text representations that can then be fine-tuned on downstream tasks with small training data sizes, which have pushed the state of the art on a variety of natural language understanding benchmarks.

More recently, gigantic language models (GLM) such as GPT3 (Brown et al., 2020) have been shown to be effective **few-shot learners**. As unsupervised training corpus and model size scaling up, the model is able to generate answers for an unseen NLP task with few-shot inference, based on a manually crafted input prompt consist of a task description and a few examples. Despite no fine-tuning is involved, the language model performs competitively against fine-tuned baselines on a wide range of tasks, whose success suggests a new paradigm of transfer learning in NLP. Yet the gaps between few-shot inference and state-of-the-art fine-tuned methods are still large on many tasks (for example 17.5 below prior state-of-the-art on SuperGLUE as shown in Table 9.1), urging for exploration of applications of giant language models beyond few-shot inference.

Inspired by the few-shot capability of GPT3, we shift our focus towards utilizing GLMs for example creation instead of direct inference, and find that language models are also excellent **few-shot generators**. Similar to the few-shot inference paradigm, we query the model with a prompt with a few examples and a description of the desired label, and the model generates examples aligned with the label while resembling the given samples. Interestingly, we find no



SAMPLE INPUT 1

Yelp Review:
Content: Moe's is my go-to for a tasty taco, ...
Rating: 3/5
Content:

SAMPLE INPUT 2

Input: Dana Reeve has died of lung cancer at age 44.
Output: Dana Reeve had an accident.
Input: Arsenal sneaked a 1-0 victory taking advantage of Stephen Clemence's own-goal in the 81st minute.
Output: Contradicting this,

SAMPLE INPUT 3

Connected Sentences
Input: The woman repaired her faucet.
Connection: Unknown
Output: The faucet was leaky.
Input: My computer screen went blank.
Connection: Because
Output:

SAMPLE OUTPUT 1

I am a bit disappointed but there are some highlights of the meal...

SAMPLE OUTPUT 2

Stephen Clemence's own-goal didn't change the result of the game.

SAMPLE OUTPUT 3

The power went out.

Figure 9.1: Illustration of the framework.

supervision is required for high-quality data creation and thus we only need to use unlabeled examples in our prompts. The dataset created by the model can then used to fine-tune any off-the-shelf model. This approach can therefore be treated as a *zero-label* learning procedure, in which no human label is required throughout the whole process. It differs from the unsupervised learning procedure in that the downstream models still need to be trained with *synthetic data*, however the training example creation requires no human labor.

Following this procedure, we are able to establish a system trained using unsupervised training data only, and thus we refer to it as **Unsupervised Data Generation** (**UDG**). Experiments show that our unsupervised system performs competitively with strong supervised baselines and achieves new state-of-the-art few-shot learning results on text classification and the SuperGLUE language understanding benchmarks. The synthesized data can further be used for data augmentation purpose. When combined with existing labeled data we are able to achieve the first super-human SuperGLUE scores. These results suggest that few-shot training data creation is a promising alternative to few-shot inference with powerful language models.

9.2 Related Work

Data augmentation has traditionally been a popular technique for NLP model quality improvement, especially in low-resource regimes (Wei and Zou, 2019; Yu et al., 2018) While traditionally simple heuristics like token-level modification has been applied to diversify training samples, more recently generative data augmentation has gained popularity due to the progress made in language modeling (Anaby-Tavor et al., 2019; Juuti et al., 2020; Kumar et al., 2021; Lee et al., 2021; Papanikolaou and Pierleoni, 2020). However, they often require labeled examples to finetune generative models and heavy postprocessing for data cleaning. On the other hand, our method generates data in a fully unsupervised manner without finetuning the language model, showcasing a new zero-label learning paradigm.

Perhaps most related to our work is (Schick and Schütze, 2021), in which they also used a large language model (GPT2-XL) to generate training samples, conditioned on task descriptions. However the instruction template they devised is quite restrictive (generating sentence pairs with similar or dissimilar meanings) , only allowing the model to perform semantic similarity tasks. Our proposal by contrast, is much more flexible and capable of generating meaningful examples for a wide range of tasks, following a unified prompting format.

Our approach is also closely related to knowledge retrieval from large language models. These models are known to be good at memorizing facts from training data and capable of performing as open knowledge bases (Carlini et al., 2021; Petroni et al., 2019; Roberts et al., 2020; Wang et al., 2020a). The high quality of training examples created by our approach is to a large part guaranteed by the model's strong knowledge retrieval ability, which reduces the chance of erratic hallucinations irrelevant to the provided labels.

9.3 Method

9.3.1 Background: Few-shot Inference

Given a set of labeled data $\mathcal{L} = \{(x^i, y^i)\}_{i=1}^n$ for a specific downstream task, the most common approach in recent years has been **fine-tuning** that updates the weights of a pre-trained model according to $\mathcal{L}$ (Devlin et al., 2018; Raffel et al., 2019; Yang et al., 2019). While obtaining state-of-the-art performance on a wide range of tasks, fine-tuning requires extra update steps and non-trivial amounts of labeled data in the target task. On the other hand, **few-shot inference** is a more resource-efficient paradigm exhibited in the latest gigantic language models such as GPT3 (Brown et al., 2020; Radford et al., 2019). The idea is to utilize the language model to infer the correct label based on the task description and a few sample input-label pairs. In particular, the input to the model M is a handcrafted ordered prompt consisted of a task description T, a small set of K examples $\mathcal{L}_{\text{few}} = \{(x^i, y^i)\}_{i=1}^K \subseteq \mathcal{L}$, and the query example x_q, and the model is expected to infer the correct label y_q as the most probable next text sequence to the input prompt:

$$y_q = \underset{y}{\operatorname{argmax}}\, P_M(y|[T, \mathcal{L}_{\text{few}}, x_q]). \tag{9.1}$$

Since taking the argmax is intractable, y_q is usually obtained through greedy decoding or beam search. Using much less task-specific data and no gradient update, few-shot inference can obtain performance comparable to fine-tuning methods (e.g. GPT3 performs similarly to fine-tuned BERT on SuperGLUE in Table 9.5). In its extreme format, giant language models can also perform one-shot (K=1) or even zero-shot (K=0) inference.

9.3.2 Unsupervised Data Generation

Despite these interesting findings, few-shot inference using giant language models still underperforms state-of-the-art fine-tuned models on many tasks. In Table 9.5, for instance, T5 largely outperforms GPT3 (89.3 vs 71.8) despite being much smaller in model sizes (11B vs 175B). One potential limitation is that a language model is never explicitly trained to directly conduct inference. Instead, it is trained as a text generator on unsupervised web corpus where inputs (X) and labels (Y) happen to coexist. Consequently, the few-shot inference method finds the proper prompt that 'forces' the model to generate next text sequence X_{next} which happens to be the label Y. However, this could be suboptimal since the labels often emerge prior to the inputs in real-world web documents. For example, in sentiment classification of IMDb movie reviews (Maas et al., 2011), the actual review contexts appear after their corresponding rating scores. Therefore, few-shot inference can force the language model to generate on text distributions that are inconsistent with its training data.

To this end, we propose to utilize language models to perform **few-shot generation**. Instead of generating and predicting the label Y, we let the model to generate the input X instead, decoupling generation from prediction. We aim to formulate the input prompts that are more likely to naturally exist in the training corpus. Specifically, the model is queried to generate x_g corresponding to a pseudo label $\hat{y}_g$ with a prompt consisted of a small set of K *unlabeled* examples $\mathcal{U} = \{x^i\}_{i=1}^K$ and a description of the desired label:

$$x_g \sim P_M(x|[T, \mathcal{U}, \text{Des}(\hat{y}_g)]), \tag{9.2}$$

where Des$(\cdot)$ is a task-specific transformation function that maps a label class to natural language descriptions, as illustrated in Figure 9.1. Different from few-shot inference, our method only requires unsupervised few-shot examples, a *zero-label learning* setting. In addition, we use top-k

		IMDb	Yelp-2	Yelp-5	Amazon-2	Amazon-5	DBpedia	Avg.
XLNet	Supervised	96.80	98.63	72.95	97.89	68.33	99.40	89.00
BERT$_{\text{LARGE}}$		95.49	98.11	70.68	97.37	65.83	99.36	87.81
UDA	Few-Shot	95.80	97.95	67.92	96.50	62.88	98.91	86.66
Few-shot Inf.		90.38	88.79	48.75	92.63	44.21	82.46	74.54
UDG	Unsupervised	95.95	98.22	69.05	97.02	64.54	96.47	86.88
+ NLA		**96.29**	**98.38**	**69.31**	**97.24**	**64.88**	**99.21**	**87.55**

Table 9.2: **Comparison of methods on text classification datasets (Accuracy)**. Results for XLNet are obtained from (Yang et al., 2019) while results for BERT$_{\text{LARGE}}$ and UDA are from (Xie et al., 2019b). The best result for semi-supervised/few-shot setup is **bolded** while underline signifies the overall best.

sampling instead of search-based decoding to sample text from the language model. This allows us to generate a synthetic labeled dataset $\mathcal{L}_{\text{syn}} = \{(x_g^i, \hat{y}_g^i)\}_{i=1}^{n_s}$ with controllable size n_s. We then train task-specific models utilizing this synthetic dataset, either as standalone training data or additional auxiliary data. Unlike existing synthetic data generation systems, our method requires no fine-tuning step of the generative model and uses unsupervised data only, and therefore we refer to it as *Unsupervised* Data Generation to emphasize its resource efficiency. We also hope to emphasize that it is not our intention to leverage the language model to perform generative tasks, but just to take advantage of it to synthesize "labeled" examples for downstream model training.

9.4 Experiments

9.4.1 Unsupervised Text Classification

To examine the effectiveness of the proposed method as well as analyze its properties, we first apply the proposed UDG method on standard text classification tasks.

Experimental Setups. We use six popular text classification benchmark datasets (Maas et al., 2011; Zhang et al., 2015), including IMDb, Yelp-2, Yelp-5, Amazon-2 and Amazon-5 sentiment classification and DBPedia topic classification. We mainly follow the experimental settings in Xie et al. (2019b) and use the corresponding unlabeled data for each task. We apply similar preprocessing steps to clean noisy web texts and truncate the input to 512 subword tokens. For each prompt, we sample $K = 32$ unlabeled examples from the unlabeled data and fit as many

None	0.9→0.8	0.9→0.7	0.9→0.6	0.9→0.5
95.95	96.03	96.08	96.17	96.29

Table 9.3: Comparison of different annealing thresholds on IMBd classification. We observe performance improves as we filter more aggresively.

examples as allowed by the length of the language model's context window. This process is then repeated $n_c = \frac{n_s}{\text{\# Class}}$ times for each label class, where we set $n_c = 10\text{K}$ for sentiment classification tasks and 1000 for topic classification. We then utilize the language model to generate one example for each prompt, resulting in a synthetic labeled dataset of size n_s. We use an in-house language model, which is a variant of Meena (Adiwardana et al., 2020) trained with larger data. We exploit top-k sampling with K=40 and temperature=1.0, and only apply basic post-processing to filter generated examples that are too short/long.

Once we obtain the generated synthetic dataset $\mathcal{L}_{\text{syn}}$, it can be utilized as labeled training data for any task-specific training framework. Here, we choose the state-of-the-art semi-supervised learning framework Unsupervised Data Augmentation (UDA) (Xie et al., 2019b) as the backbone. We use $\text{BERT}_{\text{Large}}$ as our base model and follow the training protocol as described in the UDA paper to tune our hyper-parameters. In our experiment, we find some generated examples are noisy adn thus we additionally implement a *Noisy Label Annealing (NLA)* technique to filter these examples during the training process. Noisiness is a common issue for synthetic data generation. To mitigate this issue, prior work utilize extensive filtering methods to select clean generated examples. While one key benefit of our method being high-quality synthetic data with minimal filtering, we do find some regularization during finetuning to be helpful for better performance, especially on tasks sensitive to noises. In particular, we obverse that the generated examples of the language model may be misaligned with the desired label class. Thus, we introduce a new training technique called Noisy Label Annealing (NLA), which gradually filter out noisy training signals as training progresses. Intuitively, we remove a specific training example if our model disagrees with its label with high confidence. Mathematically, at training step t, a given example $(x_g^i, \hat{y}_g^i)$ is considered noisy and removed, if (1) the model's predicted probability $P(y|x_g^i)$ is higher than a threshold μ_t, and (2) the prediction $\overline{y}^i = \text{argmax}_y\, P(y|x_g^i)$ differs from the synthetic label $\overline{y}^i \neq \hat{y}_g^i$. We set the initial threshold μ_0 to 0.9 and gradually anneal it to $\frac{1}{K}$ where K is the number of classes. Intuitively, the model is less accurate at the early stage of the finetuning process and thus we demand a very high confidence level to filter noises, whereas

	K=0	K=1	K=4	K=32
IMDb Acc.	64.21	91.34	95.86	96.29
Yelp-2 Acc.	67.34	90.27	98.22	98.38
Amz-5 Acc.	47.35	58.79	62.14	64.88

Table 9.4: Ablation of number of examples in each prompt.

we can safely decrease the “bar” as the model gets better trained. We explore different final annealing values in Table 9.3 and find a more aggressive strategy works often better.

For text classifications, we mainly follow the experimental setups in (Xie et al., 2019b). We truncate the input to 512 subwords using BERT’s vocabulary, keeping the last tokens. For the finetuning process, we search the learning rate in {1e-5, 2e-5, 5e-5} and batch size in {32, 64, 128}. We also tune the number of epochs based on the size of generated data, ranging from 5 to 30. As with (Xie et al., 2019b), we also fine-tune the BERT model on in-domain unsupervised data prior to the final training stage. For UDA hyperparameters, we tune the batch size and weight for both unsupervised and generated data, as well as different strategies of Training Signal Annealing (TSA). Notice that TSA is orthogonal to our NLA technique and thus we can apply them at the same time. Experiments are conducted on 32 v3 TPUs.

Results. We compare models of trained using fully supervised, semi-supervised/few-shot and unsupervised settings in Table 9.2. We first compare few-shot inference using our giant language model with fine-tuned methods. Despite requiring no additional training costs, the few-shot inference paradigm performs significantly worse than supervised or even semi-supervised UDA, which utilizes similar amounts of labeled data. The gap is more evident on multi-way classification tasks such as Yelp-5 or DBpedia, where the model is required to predict complex labels beyond simple answers such as ‘True/False’. In contrast, the proposed few-shot generation paradigm obtains strong performance while using less supervision. When combined with NLA, our UDG framework consistently outperforms UDA and few-shot inference on all six tasks, achieving new state-of-the-art few-shot learning results. Besides, without using any label, our method outperforms fully supervised BERT$_{\text{LARGE}}$ on IMDb and Yelp-2 and is also competitive on other tasks. Since both UDA and our method rely on BERT$_{\text{LARGE}}$, we expect using XLNet may further boost our unsupervised performance, which we choose to leave for future work.

Analysis. We first examine the effect of data noisiness on model performance. As is the case for other data augmentation methods, few-shot generation using giant language models can produce

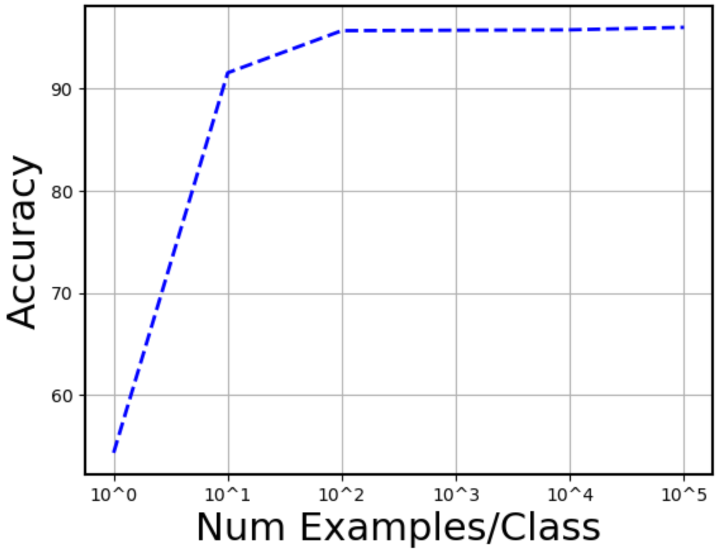


Figure 9.2: Ablation of number of examples generated per label class.

examples that are inaccurate to the desired labels. To reduce the negative impact of these noisy labels, we utilize a simple NLA technique to filter out examples when the task-specific models disagree with the synthetic labels with high confidence levels. As shown in Table 9.2, NLA robustly improves UDG performance on all tasks, especially ones that are sensitive to noise such as DBpedia.

A crucial difference distinguishing our work from existing data generation methods is that we directly query the pretrained language model without any fine-tuning nor supervision. To achieve this, the model needs to not only infer correct knowledge corresponding to the input pseudo label but also generate text with similar styles of the sample unsupervised examples. Thus, we compare the results when the language model uses different amounts of in-context examples in Table 9.4. The model fails to generate high-quality data when no sample is given, indicating the importance of few-shot generation. On the other hand, including more unsupervised examples does improve the quality of synthetic dataset which leads to better performance.

Finally, we evaluate the impact of the synthetic data sizes in Figure 9.2. Despite there is a diminishing return trend, we find the final performance to continuously improve with more generated data, showing that the language model can generate diverse examples. In addition, one key benefit of our method is that we can sample as much data as needed with no additional cost or supervision. This is particularly useful for tasks from low-resource domains with limited unsupervised data available.

		BoolQ	CB	COPA	MultiRC	ReCoRD	RTE	WiC	WSC	Avg.
Human		89.0	95.8/98.9	100.0	81.8/51.9	91.7/91.3	93.6	80.0	100.0	89.8
BERT++[a]	Sup.	79.0	84.8/90.4	73.8	70.0/24.1	72.0/71.3	71.7	69.6	64.4	71.5
RoBERTa[b]		87.1	90.5/95.2	90.6	84.4/52.5	90.6/90.0	88.2	69.9	89.0	84.6
T5[c]		91.2	93.9/96.8	94.8	88.1/63.3	94.1/93.4	92.5	76.9	93.8	89.3
DeBERTa[d]		90.4	94.9/97.2	96.8	**88.2/63.7**	**94.5/94.1**	**93.2**	76.4	95.9	89.9
T5 + UDG		**91.4**	**95.8/97.6**	**98.0**	88.3/63.0	94.2/93.5	93.0	**77.9**	**96.6**	**90.4**
GPT3[e]	Few-Shot	76.4	52.0/75.6	92.0	75.4/30.5	91.1/90.2	69.0	49.4	80.1	71.8
iPET[f]		81.2	79.9/88.8	90.8	74.1/31.7	85.9/85.4	70.8	49.3	88.4	75.4
ADAPET[g]		80.0	82.3/92.0	85.4	76.2/35.7	86.1/85.5	75.0	53.5	85.6	76.0
UDG	Unsup.	81.0	86.2/92.4	80.4	81.1/47.1	82.8/81.8	80.7	67.5	79.5	78.1

Table 9.5: **Comparison of single-model methods on SuperGLUE test scores.** Results obtained from the official SuperGLUE leaderboard[2]. The best result for semi-supervised/few-shot setup is underlined while **bold** signifies the overall best. Model references: [a]Devlin et al. (2018) [b]Liu et al. (2019) [c]Raffel et al. (2019) [d]Devlin et al. (2018) [e]Brown et al. (2020) [f]Schick and Schütze (2020) [g]Tam et al. (2021)

9.4.2 Unsupervised Language Understanding

To evaluate the proposed framework in a more challenging and comprehensive setting, we extend it to perform on complex language understanding tasks.

Experimental Setups. We use the SuperGLUE benchmark (Wang et al., 2019a) for general-purpose language understanding in English, which consists of 8 natural language understanding tasks. Tasks cover textual entailment (CB and RTE), question answering (BoolQ, MultiRC and ReCoRD), common sense reasoning (COPA), word sense disambiguation (WiC), and coreference resolution (WSC). We mainly follow the same generation protocol as described in the previous sections, with some minor changes in prompt templates and data post-processing steps for specific tasks (Please see Section 9.5 for list of all prompts and generated examples). As before, we use K=32 unlabeled examples and generate using the same language model. For each task, we use all original labeled data as unsupervised examples for training data creation.

For the downstream model, we use T5 (Raffel et al., 2019) for fine-tuning on the created data. Different from the released T5 checkpoints that are pretrained on multi-task data, we pretrain our own models on unsupervised Colossal Clean Crawled Corpus (C4) data only and thus the combined framework remains unsupervised. For fair comparison with existing models, we pretrain and then fine-tune a T5-Large model using the created data set. The model is trained using the T5 framework with a batch size of 2,048 during the pretraining stage, each containing 512 max tokens. Following T5, we use Adafactor optimizer with a learning rate of 1e-3 and the same learning rate schedule. The task ratio is also incorporated similarly to T5. The model is trained for 1 million steps on 256 TPU v3 chips, roughly using 10 days. Following Raffel et al. (2019), we use a fine-tuning batch size of 8 with 512 sequence length. The model is further finetuned for 200k steps and we utilize the dev scores to pick the best checkpoint for submission to the test server for final evaluation.

Results. We compare models trained under different settings in Table 9.5. The GPT3 model (Brown et al., 2020) using the few-shot inference method outperform BERT++ with less supervision and no fine-tuning. However, despite containing much more model parameters, it performs worse than other fine-tuned fully supervised models and few-shot methods. On the other hand, our unsupervised framework using few-shot generation outperforms all few-shot learning systems without using any label, and thus it achieves new state-of-the-art results on this benchmark for methods that exploit little-to-no supervision. In particular, our performance gains largely come from natural language entailment tasks (CB and RTE) as well as word sense disambiguation, where GPT3 performs similarly to random guessing. This indicates that language models do contain language knowledge that few-shot inference fails to leverage. On the other hand, our method is extremely simple to use, requiring no further finetuning for data generation or task-specifc tricks, as the case for (Schick and Schütze, 2020). This demonstrates the effectiveness and simplicity of the proposed UDG framework, and also shows that it is possible to obtain competitive performance on the challenging SuperGLUE benchmarks without any human label, a new paradigm that has broader implication for future research in NLP.

9.4.3 UDG as Data Augmentation

In previous sections we only use the created examples as pseudo supervision to explore the limits of transfer learning using language models. Nonetheless, the synthetic data can be also treated as augmented data and combined with existing labeled data.

Experimental Setups. We follow the similar procedure in the previous experiment to generate synthetic data, as the downstream tasks remain the same. However, since we are working with the fully supervised learning setting here, we can optionally include human labels in the input prompts for the language model. For example, instead of 'Sample sentence', we can use

[2] https://super.gluebenchmark.com/leaderboard

’Positive/Negative sentence’ instead. We found including actual labels to be helpful in our experiments.

To finetune the model on both original supervised data and the generated data, we directly mix both dataset and then conduct finetuning once. To achieve this, we first choose to generate samples much larger than the size of the corresponding ground truth data, and then mix the two with a specific ratio (e.g. mix every original example with 2 generated examples). We experiment with different mixing ratios, and choose the best ratio using their dev scores. The finetuning parameters are kept the same as before for all ratios and tasks.

Results. We fine-tune the public T5-XXL checkpoint using both labeled data and generated data, in the form of mixture of data. As shown in Table 9.5, our method combines well with existing labeled data and brings substantial improvements. This is particularly the case for tasks with small data sizes such as COPA and WSC. Moreover, the combined model outperforms not only prior methods but also the human baselines for the first time on this important NLP benchmark, setting a new milestone for natural language understanding with machine learning models.

9.5 Prompts & Generated Examples

Here in this section, we present the prompts we used to generate synthetic data for each task considered.

9.5.1 Text Classification

IMDb

Input Prompt:

```
Sample Movie Review: This is the most saccharine romance I ever sat
through. The perfect film for an idle housewife in kerchief, housedress,
and ostrich-trimmed high-heeled mules to watch in the afternoon,
lying on the couch eating bonbons. In fact, bonbons play a prominent
role in the movie.  The only reason I was able to watch to the end,
is that I finally was able to gaze at Keanu Reeves’ dreamy face in
almost every scene. In most of his films, he moves too fast to get
a good look. The only rapid action in this show is Giancarlo Giannini
waving his hands with Latin emotionality - more Italian than Mexican,
```

really.

Negative Movie Review:

Generated Output:

This remake of the 1966 French film by François Truffaut and Claude Chabrol is just plain awful. The only positive I can say about the film is that the casting was just about perfect. Otherwise, the movie has no redeeming qualities whatsoever. The plot is a mess and is filled with numerous inconsistencies and continuity flaws throughout. It is as if the director just made it up on the spot and made it up bad!

Yelp-2

Input Prompt:

Sample Business Review: I had the most delicious burger ever! I also ordered the truffle fries, which were amazing! The sauce they served with the fries was also very good! Definitely come here!! There was great service and wonderful food!

Positive Business Review:

Generated Output:

We had our wedding at The Grand in October of 2011. The Grand and their staff are absolutely wonderful to work with. We had over 100 people at the wedding and the room was packed for the reception. We had a great time and the service, food, and atmosphere were fantastic. I can not say enough good things about working with and having our wedding at The Grand. We highly recommend The Grand for your wedding or special event.

Yelp-5

Input Prompt:

Yelp Review

Content: I waited FOR 35 MINUTES tonight just for 2 brisket meals to go. They weren't even hot when I arrived home and I live less than 1 minute away. It was still a fine meal, saved by the amazing cornbread!! But, I have no intention of going back any time soon.

Yelp Review
Rating: 3/5
Content:

Generated Output:

I came here to eat after the ridiculous long lines died down a bit. I must say, I don't get the hype about this place and felt a bit disappointed. I suggested coming here with a group of friends. I was pretty excited to eat here and found that nothing impressed me and the price tag could not be justified. Nothing stood out for me that would make me want to come here again. I think the highlight of my meal was the ginseng chicken soup. All other items, I've had better at other buffets. I didn't think their selection was that impressive.

Amazon-2

Input Prompt:

Sample Product Review
Title: Nice to have
Content: My dog loves this bed. I don't like to have my dog sleep on the floor. I know I spoiled my dog. I put a huge pillow on top of this bed to give her the extra comfort. My dog loves sleeping on something soft. Now I have a happy dog that sleeps comfortably every night. Money well spend.

Negative Product Review
Title:

Generated Output:

Not worth it
Content: I am so very disappointed. I bought this for my granddaughter for Christmas. I have a few concerns, but first and foremost, the box that the doll comes in says that it must be assembled by an adult. The instructions are very confusing. My mom and I put it together for her Christmas Eve. Then we realized that the doll has a small hole in the back of the head that is on the end of the seam. I don’t know where the hole came from, but since she is a collectible, I can’t return it.

Amazon-5

Input Prompt:

Product Review
Title: Good Book
Content: Good book, easy to read, with a lot of suggestions about where read the exposure. Also if the book is more than 150 pages, there are a lot of big photos, everyone with a comment; instead of this I would have prefereed more photos with comment also if they would have been smaller. So, I read this book in too little time. My rate for this book is 3.5.

Product Review
Rating: 3/5
Title:

Generated Output:

An ok game
Content: This game is good, but I wouldn’t buy this game unless you are a star wars fan. The game has lots of levels and it takes quite a while to beat. If you don’t know anything about star wars, you would be lost on the story. There are lots of good things and bad things about this game.

DBpedia

Input Prompt:

```
Wikipedia Article
Title: University of Georgia School of Law
Article:  The University of Georgia School of Law (also referred
to as Georgia Law) is a graduate school of the University of Georgia.


Wikipedia Article
Category: Educational Institution
Title:
```

Generated Output:

```
Vanung University
Article: Vanung University (VNU) is a private university in Zhongli
City Taoyuan County Taiwan. It is the first private university in
Taiwan to be accredited by AACSB.
```

9.5.2 SuperGLUE

BoolQ

Input Prompt:

```
Article: Russian Blue -- Russian Blues are plush short-haired, shimmering
pale blue-gray cats with emerald green eyes. Guard hairs are distinctly
silver-tipped giving the cat a silvery sheen or lustrous appearance.
They have been used on a limited basis to create other breeds such
as the Havana Brown or alter existing breeds such as the Nebelung.
They are being used in Italy as a way to make Oriental Shorthairs
healthier and more robust called RUS4OSH in FIFe.
Sample statement based on the article: do all russian blue cats have
green eyes.

Article: Okinawa Churaumi Aquarium -- The Okinawa Churaumi Aquarium
```

```
(Okinawa Churaumi Suizokukan) is located within the Ocean Expo Park
in Okinawa, Japan. It welcomed its 20 millionth visitor on 30 March
2010 and is a member of the Japanese Association of Zoos and Aquariums
(JAZA). It was the largest aquarium in the world until it was surpassed
by the Georgia Aquarium in 2005. The aquarium has the theme of ``Encounter
the Okinawan Sea''.
False statement based on the article:
```

Generated Output:

```
is okinawa churaumi aquarium the largest aquarium in the world.
```

CB

Input Prompt:

```
Input: Jim waited. He waited a long time, and when the young doctor
finally came out, it was almost dark. Jim could nonetheless tell
by his anxious face that something was wrong.
Output: Something was wrong.

Input: She swallowed hard, unsure if she had the nerve to go ahead.
The memory of the pain in Tara’s eyes last night decided her. Did
he really expect her to believe that Tara was only the housekeeper?
Output: Contradicting this,
```

Generated Output:

```
Tara was only a temporary housekeeper.
```

COPA

Input Prompt:

```
Input: My body cast a shadow over the grass.
Output: The sun was rising.

Input: My computer screen went blank.
Connection: Because
Output:
```

Generated Output:

```
The power went out.
```

MultiRC

Input Prompt:

```
Article: On a day it celebrated 50 years of giving free legal advice,
Southern Arizona Legal Aid announced it has received two grants totaling
more than 7 million ...
Question: Who will the Web site benefit?
Answer: Help bridge the "digital divide" that keeps the poor from
using the Internet as a resource

Article: While this process moved along, diplomacy continued its
rounds. Direct pressure on the Taliban had proved unsuccessful ...
Question:
```

Generated Output:

```
Which country did President Bill Clinton visit on March 25, 2000?
Correct Answer: Pakistan
```

ReCoRD

Input Prompt:

```
Article: The harrowing stories of women and children locked up for
so-called ’moral crimes’ in Afghanistan’s notorious female prison
have been revealed after cameras were allowed inside. Mariam has
been in Badam Bagh prison for three months after she shot a man who
just raped her at gunpoint and then turned the weapon on herself
- but she has yet to been charged. Nuria has eight months left to
serve of her sentence for trying to divorce her husband. She gave
birth in prison to her son and they share a cell together. Scroll
down for video Nuria was jailed for trying to divorce her husband.
Her son is one of 62 children living at Badam Bagh prison.
```

Keyword: Badam Bagh prison ### Relevant Fact: The women and children are jailed in Badam Bagh prison.

Article: China’s space agency today distributed fresh new pictures of the Moon’s surface from its lunar lander and rover. The Jade Rabbit rover can be seen trundling across the pockmarked surface of Earth’s satellite in pictures released today by the Chinese Academy of Sciences. The successful soft-landing of the Chang’e 3 lander in December made China the third country to complete a lunar rover mission after the United States and Russia. Chasing the horizon: The sky over the Moon appears inky black in this picture showing the Jade Rabbit rover trundling across the lunar surface taken from the Chang’e 3 lander and released today by Chinese scientists.
Keyword: China ### Relevant Fact:

Generated Output:

Knock off or not, there’s no denying the ambition of the China’s Moon mission, which on December 14 achieved the first lunar soft landing in nearly four decades and aims at a manned mission by 2025.

RTE

Input Prompt:

Input: No Weapons of Mass Destruction Found in Iraq Yet.
Output: Weapons of Mass Destruction Found in Iraq.

Input: CBS newsman Harry Reasoner is returning to his Iowa hometown to get married Saturday.
Output: In other words,

Generated Output:

Harry Reasoner is getting married.

WiC

Input Prompt:

```
Word: meet
Example: The company agrees to meet the cost of any repairs.

Word: skin
Definition 1:
```

Generated Output:

```
the outer covering of the body of a vertebrate animal.
Example: The skin of a snake.
```

WSC

Input Prompt:

```
Noun: Sam ### Pronoun: he ### Sample Sentence: Sam took French classes
from Adam , because he was eager to speak it fluently.

Noun: Ann ### Pronoun: She ###  Sentence where She refers to Ann:
```

Generated Output:

```
It seems that Ann has had a run of bad luck. She never wins at cards.
```

9.6 Summary

In this chapter, we propose a “zero-label” training procedure and show that language models are also few-shot example creators in that they can be used to generate high-quality synthetic data in a fully unsupervised manner. Through this, we demonstrate that universal knowledge transfer across various natural language understanding tasks emerge with the scaling of textual language models, such that it is possible to obtain strong results without any human annotated label. This concludes the final part of the thesis, which shows that mitigating data discrepancy leads to improves sample efficiency in transfer learning.

Chapter 10

Conclusion

This thesis explores the problem of task conflicts in the form of negative transfer in transferable machine learning models. We conduct an in-depth analysis on this issue and glean insights on its causes. Each part is centered around the goal of understanding and mitigating negative transfer. In the first part, we find negative transfer to be model-dependent while task conflict is the root of it. The second and third parts then turn to study methods to address the aforementioned problem, where each chapter is a case study on a specific setup of transfer learning. Overall, we propose novel methods to address negative transfer and show improvements on both model generalization and sample efficiency. In the following sections, we summarize our main contributions and future directions.

10.1 Summary of Contributions

This thesis facilitates transfer learning research on characterizing and mitigating negative transfer, with the goal to improve model performance and efficiency. In the following we summarize the contributions of this thesis.

Defining and understanding negative transfer. While the term negative transfer has existed for a long time in the literature, little prior work has carefully studied it. It is crucial to define and characterize negative transfer in order to further understand its factors and design algorithms to mitigate it. To this end, we conduct analysis under two popular transfer learning settings (domain adaptation and multilingual language modeling) and make the following contributions:

- We present the first formal definition of negative transfer that is general and tractable in practice. This definition further introduces a metric negative transfer gap, that can be used to measure and observe negative transfer for future research.

- The negative transfer definition further reveals three key underlying factors of negative transfer that are verified through empirical observations. Our analysis shows task conflict is the root to negative transfer.
- We propose the first method that utilizes the discriminator as gate for sample estimation and we show this can effectively mitigate negative transfer in domain adaptation.
- We conduct the first systematic analysis of negative transfer in multilingual models and demonstrate gradient conflict and parameter sharing as a key cause of negative transfer. Our results also illustrate negative transfer can occur to low-resource tasks as well under certain setups.
- We propose a new formulation of meta-learning shared parameters in multilingual language models through a bi-level optimization. The method enables knowledge transfer in language-specific parameters through second-order gradients, and is able to mitigate interference among languages, improving both with-in language performance and cross-lingual transfer.

Addressing task conflict to improve model generalization. Following analysis of negative transfer, task conflict has been shown the root of it. Hence, we explore utilizing explicit alignment methods to mitigate negative transfer in multilingual models, where we made the following contributions:

- We present the first thorough comparison of two popular paradigms of cross-lingual embedding training methods, joint training and alignment, and glean insights on their pros and cons.
- We introduce a unified framework that enjoys the benefits of joint training and alignment. The simple framework obtains state-of-the-art results on the word translation benchmark, and could also be extended to contextualized embedding models to further boost their performance.
- We conduct the first study of multilingual optimization process in multilingual machine translation systems and show that gradient similarities closely resemble language proximities.
- We develop the first optimization method designed for multilingual training, namely Gradient Vaccine, that encourages better aligned gradient updates adaptively. This method is scalable and we verify its efficiency in a large-scale machine translation system. It is also generic for other multi-task learning settings and has potential beyond multilingual

settings.

- We also investigate using meta learning to learn how to smooth the output space for multilingual machine translation systems. This method shows improvement over joint training baselines and improves gradient alignments as well.

Achieving efficient transfer learning with less supervision. Transfer learning aims to improve model performance in low-resource settings, and therefore it is our interest to study how to improve better sample efficiency. At a high level, we observe that alleviating data discrepancy and negative transfer enables more efficient transfer such that model requires fewer training examples to obtain strong performance. In some cases, it can also enable few-shot or even zero-shot learning. We made the following contributions in this direction:

- We conduct the first study to compare the trade-off between negative transfer and catastrophic forgetting in the lifelong learning setup, and further show that their tension limits the efficiency of existing methods.
- We propose a synergistic meta-lifelong framework that trains finetuning-aware lifelong learning models. This framework resolves the training-testing discrepancy of existing lifelong learning methods and mitigates negative transfer by jointly optimizing the model for local adaptation for all tasks. Consequently, it largely improves sample efficiency of lifelong learning by outperforming prior methods using 100 times less training data stored in memory.
- We develop a novel vision-language pretraining framework named simple visual language model (SimVLM). The model is significantly simplified compared to prior methods by utilizing a single pretraining objective and does not utilize any human labeled pretraining data.
- We show that SimVLM obtains state-of-the-art results on six vision-language benchmarks, and hence even a simple pretraining method with less supervision can enable effective knowledge transfer.
- We explore new zero-shot capabilities of vision-language model using SimVLM, including zero-shot cross-modality transfer and zero-shot open-ended visual question answering.
- We present a novel unsupervised data generation (UDG) method to generate data from giant language models without human label nor further finetuning. This immediately enables zero-label language learning, where no human label is utilized in the whole training process. Our method obtains strong zero-shot performance on SuperGLUE, outperforming

strong few-shot learning baselines such as GPT3 as well as fully supervised BERT model.

- We show that UDG combines well with labeled data, achieving the first super-human performance on SuperGLUE.

Our contributions cover different aspects of machine learning including methodological improvements as well as data-driven approaches. At a high level, training a (deep) machine learning model for prediction can involve three major components: data, underlying model architecture, and training algorithm. Here, methodological innovations refer to new architecture designs (e.g. transformer (Vaswani et al., 2017)) and better training objectives (e.g. BERT (Devlin et al., 2018)), while the source of improvement of data-driven approaches mainly comes from cleaner and larger datasets (e.g. ALIGN (Jia et al., 2021), T5 (Raffel et al., 2019)).

In this thesis, each part/chapter can focus on tackling one or more aspects of these components. In the first part, we study the cause and characteristics of negative transfer conditioned on fixed model architectures and datasets. Here, through controlled experiments, we demonstrate that negative transfer is algorithm-specific. Following this analysis, in the second part, we design new alignment-based algorithms to mitigate negative transfer and improve model quality. These parts study transferring knowledge to low-resource settings such as low-resource languages and unsupervised target domains, and therefore focus on methodological innovations. On the other hand, we investigate data-driven approaches in the last part of this thesis for transfer learning settings with abundance of data, such as building connections between two modalities (despite each of them contains plenty of training data) and continuously learning new tasks throughout the lifetime. Here, our focus switches to how to better utilize large amount of unsupervised data that are cheap to collect to enable effective transfer to downstream tasks where supervised data are expensive. Consequently, the contributions of this thesis contain both algorithmic and data-driven methods.

While we have different focuses in each part of this thesis, the training components mentioned above are not independent and are often combined to achieve best results. For instance, we also investigate language-specific architectures to assist algorithmic improvements in chapter 4 while SimVLM is a framework that contributes to all three aspects of data, pretraining objective and model architecture. Therefore, algorithm design is important not only for low-resource transfer learning, but also for data-driven approaches. In light of recent work on the potential of model/data scaling (Ghorbani et al., 2021; Kaplan et al., 2020), it is crucial to study how to combine algorithmic improvements with these large-scale models. Some key challenges include training efficiency and stability: for example, the meta-learning methods explored in this thesis require significant computing power and careful hyperparameter tuning, making it hard to adapt to large-scale training. Designing efficient approximation for these algorithms can be a fruitful direction for scaling up transfer learning models. On the other hand, it is becoming increasingly challenging to conduct fair comparison among models to demonstrate the superiority of a specific training aspect such as training objective. This is particularly the case when different research institutes have different levels of computational power. Thus, it is also important to build public benchmarks with standard and accessible evaluation protocol for the research community to understand algorithmic innovations in a better way.

10.2 Future Work

This thesis focuses on understanding and addressing negative transfer for better generalization and sample efficiency. While we have made significant progress, the observations and conclusions made in this thesis are conditioned on specific and often synthetic settings, and therefore there are many interesting directions and unsolved problems remain to be explored, among which we list a few key directions below.

Negative transfer in other settings and algorithm components: In this thesis, we have studied negative transfer in widely-used transfer learning settings such as domain adaptation and multilingual NLP. Many real-world setups such as heterogeneous multi-task learning and the pretraining-finetuning paradigm are yet to be explored. It will be interesting to study whether negative transfer occurs in those settings, and to understand its characteristics. For example, while the pretraining-finetuning paradigm works well in NLP, it is less effective in the vision domain. And thus it is interesting to study whether negative transfer is present in vision pretraining under certain circumstances, and whether it is possible to mitigate it to facilitate better transfer to downstream tasks. In addition, throughout the thesis we mainly focus on negative transfer caused by data discrepancy and conduct analysis conditioned on the specific task settings and algorithms, despite that negative transfer depends on algorithms and problem settings. Therefore, more thoroughly studying negative transfer induced by other factors should be a fruitful future direction. As a concrete next step, a potential effort is to understand the algorithm factor of negative transfer, such as data preprocessing, data sampling, optimizer choices and architecture designs.

More intelligently mitigating negative transfer: In previous chapters, we have explored several methods to encourage better alignments among tasks to mitigate data discrepancy and improve model transferability. However, our methods are built on top of assumptions and human heuristics (e.g. utilizing training corpus sizes to decide vocabulary sharing and sampling strategies). While we have empirically verified their effectiveness in our experimental settings, task-specific adaptation is likely required on challenging real-world problems where new tasks constantly emerge. Therefore, one important direction for future work is to develop more intelligent and automatic methods to address negative transfer, such that they can adaptively fit new transfer environments. Here, we outline a few proposed directions:

- Multilingual Vocabulary Sharing: In Chapter (chapter 5) we have shown representation alignment to work well and mitigating multilingual data discrepancy. We highlight the problem of vocabulary over-sharing and under-sharing, but only rely on heuristic to decide what to share. It is therefore worth exploring how to share multilingual vocabulary automatically. One possibility is to utilize the AutoML approach to develop a search space and search for optimal sharing strategies.
- Meta-level Objective: We proposed several meta learning based approach in this thesis. Typically, a meta level objective is defined over a set of validation data. Nonetheless, the effects of changing meta objectives and validation sets are not clear and require further investigation. For instance, future work should verify whether it is possible to control the behavior of meta learning by adapting these meta-level settings, and if so, explore how to set these objectives automatically and efficiently.
- Gradient Alignment Objective: We have explored gradient alignment methods in this thesis and found setting adaptive gradient similarity objective to be useful. However, we only experimented with a few human-designed methods such as setting objectives based on language family proximity and utilizing exponential moving average variables. These methods introduce additional hyper-parameters and are hard to tune. Therefore, we believe one key missing component of this line of research is to design better alignment objectives of gradients among tasks to address negative transfer more effectively.
- Fine-grained Control of Gradient Alignment: We propose the method of GradVac in chapter (chapter 6). For gradients of two tasks, the method yields a linear combination to improve their alignment. A more generic approach would be to learn a mapping function that map arbitrary input gradients in the gradient space to a single final gradient in the same space. This should be explored in future work.

Improving sample efficiency through zero-shot/few-shot learning: In the final part of this thesis, we have explored zero-shot and few-shot learning in both NLP and multimodal settings, showing new patterns of utilizing pretrained models. In particular, we show that it is possible to obtain strong performance without using any human label. However, the performance of zero-shot learning still does not compete directly with fully supervised methods, and the applicability of existing methods is also limited. Therefore, it is crucial to continually improve transfer learning sample efficiency, and we outline a few potential directions below:

- Scope of few-shot learning: In the last two chapters we have explored various types of few-shot learning in natural language understanding and vision-language understanding tasks, using both the inference model and generation model paradigms. However, these methods of few-shot learning may not generalize to other real-world tasks such as answering scientific questions and generating driving instructions [1]. How to enable few-shot transfer learning to these open-domain tasks is a challenging next step.
- Robustness of few-shot learning: We proposed unsupervised data generation (UDG) using giant language models in this thesis, where we provide few unlabeled examples in the prompt to query the model. However, the effect of these samples is not well understood. In particular, we have observed large variance when changing the template of the prompt as well as using different sampling strategies for the input examples. Therefore, it is important to improve the robustness of this few-shot learning paradigm. One potential solution is meta learning, such that to train an additional component to automatically decide what template/examples to use for unseen tasks.
- Dedicated pretraining procedure: In this thesis, we have focused on improving the transfer efficiency by designing algorithms to better utilize pretrained language models. On the other hand, the zero-shot/few-shot learning capabilities emerge by scaling language models without dedicated pretraining objectives. It is therefore interesting to study whether it is possible to design curated pretraining procedure to improve the zero-shot capability of language models. For example, we can design task template to automatically create pseudo tasks from unsupervised corpus and combine them with unsupervised training objectives in pretraining.
- Multimodal few-shot learning: Finally, in the last part we have explored training language models to bridge the gap between image and text, and the resulting model have shown certain degree of zero-shot generalization. For future work, one direction is to include additional modality options such as acoustic or video representations. More importantly, another direction is to enable few-shot learning capability similar to GPT3, such that the model can generate predictions without task-specific finetuning, based on input task

[1] See more examples at: `https://github.com/google/BIG-bench`.

prompts consist of examples in the form of multiple modalities.

Bibliography

Oliver Adams, Adam Makarucha, Graham Neubig, Steven Bird, and Trevor Cohn. Cross-lingual word embeddings for low-resource language modeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 937–947, Valencia, Spain, April 2017. Association for Computational Linguistics. 5.1

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020. 9.4.1

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. nocaps: novel object captioning at scale. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8948–8957, 2019. 8.4.1

Roee Aharoni, Melvin Johnson, and Orhan Firat. Massively multilingual neural machine translation. *CoRR*, abs/1903.00089, 2019. 2.2.1, 2.2.2, 4.1, 6.1, 6.5

Wasi Uddin Ahmad, Zhisong Zhang, Xuezhe Ma, Eduard Hovy, Kai-Wei Chang, and Nanyun Peng. On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing. In *Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Minneapolis, USA, June 2019. 5.1

Hanan Aldarmaki and Mona Diab. Context-aware cross-lingual mapping. In *Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Minneapolis, USA, June 2019. 5.1, 5.3.2

Waleed Ammar, George Mulcaire, Yulia Tsvetkov, Guillaume Lample, Chris Dyer, and Noah A Smith. Massively multilingual word embeddings. *arXiv preprint arXiv:1602.01925*, 2016. 5.1, 5.5

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. Not enough data? deep learning to the rescue!, 2019. 9.2

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018. 8.2

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*, 2019. 1, 2.2.1, 2.2.2, 4.1, 4.3.1, 6.1, 6.2, 6.2.1, 6.2.1, 2, 6.2.2, 6.4.2, 6.5

Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610, 2019. 5.2.2, 5.5

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, 2016. 2.2.1, 5.5

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of ACL 2018*, pages 789–798, 2018a. 5.1, 5.2.1, 5.5

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. Unsupervised neural machine translation. In *International Conference on Learning Representations*, 2018b. 5.1

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Bilingual lexicon induction through unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5002–5007, 2019a. 5.4.2

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. *arXiv preprint arXiv:1910.11856*, 2019b. 2.2.1, 4.5, 6.5

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018. 3.4.2

Ankur Bapna, Naveen Arivazhagan, and Orhan Firat. Simple, scalable adaptation for neural machine translation. *arXiv preprint arXiv:1909.08478*, 2019. 4.4.2, 7.2

Antonio Valerio Miceli Barone. Towards cross-lingual distributed representations without parallel text trained with adversarial autoencoders. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, pages 121–126, 2016. 5.5

Atilim Gunes Baydin, Robert Cornish, David Martinez Rubio, Mark Schmidt, and Frank Wood. Online learning rate adaptation with hypergradient descent. 2018. 2.2.1, 4.5

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 137–144, 2007. 3

Graeme W. Blackwood, Miguel Ballesteros, and Todd Ward. Multilingual neural machine translation with task-specific attention. *CoRR*, abs/1806.03280, 2018. 6.2

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. ISSN 2307-387X. 5.1, 5.3.1

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. 8.4.1, 8.3

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 2.2.1, 8.1, 8.3.1, 8.4.3, 9, 9.1, 9.1, 9.3.1, 9.5, 9.4.2

Hailong Cao, Tiejun Zhao, Shu ZHANG, and Yao Meng. A distribution-based model to learn bilingual word embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1818–1827, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. 5.5

Steven Cao, Nikita Kitaev, and Dan Klein. Multilingual alignment of contextual word representations. 2020. 2.2.1, 4.5, 6.5

Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Partial transfer learning with selective adversarial networks. 2018a. 2.2.1, 2.2.2, 3.1, 3.4.3, 3.6

Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. 2018b. 3.3.2, 3.4.1, 3.4.2

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models, 2021. 9.2

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997. 2.2.1

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-GEM. In *International Conference on Learning Representations*, 2019. 2.2.1, 7.1, 7.2, 7.4.2

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, et al. The best of both worlds: Combining recent advances in neural machine translation. *arXiv preprint arXiv:1804.09849*, 2018a. 6.2.1

Xilun Chen, Ahmed Hassan Awadallah, Hany Hassan, Wei Wang, and Claire Cardie. Multi-source cross-lingual model transfer: Learning what to share. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3098–3112, Florence, Italy, July 2019. Association for Computational Linguistics. 5.1

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 8.4.1

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020. 8.1, 8.2, 8.4.2

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803. PMLR, 2018b. 6.3.1, 6.4.1, 6.5

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. *arXiv preprint arXiv:2102.02779*, 2021. 8.2, 8.4.2, 8.4.3, 8.4

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. Quac: Question answering in context. In *Proceedings of the 2018 Conference* *on Empirical Methods in Natural Language Processing*, pages 2174–2184, 2018. 7.4.1

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. *arXiv preprint arXiv:2003.05002*, 2020. 4.3.2

Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of operations research*, 153(1):235–256, 2007. 4.4.1

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *International Conference on Learning Representations*, 2018a. 5.1, 5.2.1, 5.4.1, 5.4.2, 5.5

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium, October-November 2018b. Association for Computational Linguistics. 4.3.2, 5.1

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, July 2020. 1, 2.2.1, 2.2.2, 4.1, 4.2, 4.3.1, 4.5, 6.1, 6.5

Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10578–10587, 2020. 8.2

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In *Advances in neural information processing systems*, pages 442–450, 2010. 3.1, 3.3.2

Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *arXiv preprint arXiv:2106.04803*, 2021. 8.1, 8.3.3, 8.4.4

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. Episodic memory in lifelong language learning. In *Advances in Neural Information Processing Systems*, 2019. 7.1, 7.2, 7.2, 7.3, 7.3.1, 7.1, 7.4.1, 7.4.1, 7.4.2, 7.4.3, 7.4.4, 7.7, 7.8

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

2018. 1, 2.2.1, 4.1, 4.5, 5.2.2, 5.3.2, 5.5, 6.1, 6.4.2, 6.5, 7.1, 7.2, 8.1, 8.3.1, 8.3.3, 8.4.3, 9.1, 9.3.1, 9.5, 10.1

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014. 3.4.4

Pinar Donmez, Jaime G Carbonell, and Paul N Bennett. Dual strategy active learning. In *European Conference on Machine Learning*, pages 116–127. Springer, 2007. 7.4.4

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 8.1, 8.3.3, 8.4.1

Zi-Yi Dou, Zhi-Hao Zhou, and Shujian Huang. Unsupervised bilingual lexicon induction via latent variable models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 621–626, 2018. 5.5

Yunshu Du, Wojciech M Czarnecki, Siddhant M Jayakumar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity. *arXiv preprint arXiv:1812.02224*, 2018. 6.5

Lixin Duan, Dong Xu, and Shih-Fu Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1338–1345. IEEE, 2012. 2.2.2, 3.1, 3.6

Long Duong, Hiroshi Kanayama, Tengfei Ma, Steven Bird, and Trevor Cohn. Learning crosslingual word embeddings without bilingual corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1285–1295, 2016. 2.2.1, 5.1, 5.4.2, 5.2, 5.5

Chris Dyer, Victor Chahuneau, and Noah A. Smith. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia, June 2013. Association for Computational Linguistics. 5.4.2

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. Multi30k: Multilingual english-german image descriptions. *arXiv preprint arXiv:1605.00459*, 2016. 8.4.1

Carlos Escolano, Marta R Costa-jussà, José AR Fonollosa, and Mikel Artetxe. Multilingual machine translation: Closing the gap between shared and language-specific encoder-decoders. *arXiv preprint arXiv:2004.06575*, 2020. 6.2

Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. 2.2.1, 5.5

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv preprint arXiv:2101.03961*, 2021. 1

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017. 2.2.1, 4.4.1, 4.5

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. Multi-way, multilingual neural machine translation with a shared attention mechanism. *arXiv preprint arXiv:1601.01073*, 2016. 2.2.1, 6.1, 6.5

Sebastian Flennerhag, Andrei A Rusu, Razvan Pascanu, Hujun Yin, and Raia Hadsell. Meta-learning with warped gradient descent. *arXiv preprint arXiv:1909.00025*, 2019. 2.2.1, 4.4.1, 4.4.2, 4.5

Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. *arXiv preprint arXiv:2006.06195*, 2020. 8.2, 8.4.2

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015. 3, 3.3.1, 3.3.1, 3.3.2, 3.4.2, 7.2

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 2.2.1, 3.3.1, 3.6

Liang Ge, Jing Gao, Hung Ngo, Kang Li, and Aidong Zhang. On handling negative transfer and imbalanced distributions in multiple source transfer learning. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 7(4):254–271, 2014. 2.2.2, 3.1, 3.6, 4.5, 6.5, 7.4.4

Behrooz Ghorbani, Orhan Firat, Markus Freitag, Ankur Bapna, Maxim Krikun, Xavier Garcia, Ciprian Chelba, and Colin Cherry. Scaling laws for neural machine translation. *arXiv preprint arXiv:2109.07740*, 2021. 10.1

Goran Glavas, Robert Litschko, Sebastian Ruder, and Ivan Vulic. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. *arXiv preprint arXiv:1902.00508*, 2019. 5.4.1

Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *International conference on machine learning*, pages 2839–2848, 2016. 2

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2.2.1, 3.6, 5.5

Stephan Gouws and Anders Søgaard. Simple task-specific bilingual word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1386–1390, 2015. 2.2.1, 5.5

Stephan Gouws, Yoshua Bengio, and Greg Corrado. BilBOWA: Fast bilingual distributed representations without word alignments. In *Proceedings of ICML 2015*, pages 748–756, 2015. 2.2.1, 5.1, 5.5

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017. 8.1, 8.4.1

Jiatao Gu, Yong Wang, Yun Chen, Victor OK Li, and Kyunghyun Cho. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, 2018. 2.2.1, 4.5

Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, Zhaohui Yang, Yiman Zhang, and Dacheng Tao. A survey on visual transformer, 2021. 8.2

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3.4.2, 8.3.3, 8.4.1

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of* *the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

Karl Moritz Hermann and Phil Blunsom. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 58–68, 2014. 2.2.1, 5.5

Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado van Hasselt. Multi-task deep reinforcement learning with popart. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3796–3803, 2019. 6.5

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. *arXiv preprint arXiv:1902.00751*, 2019. 4.3.3, 4.4.2, 7.2

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. In *International Conference on Machine Learning (ICML)*, July 2020. 4.1, 4.3.2, 6.1, 6.4.2

Xiaowei Hu, Xi Yin, Kevin Lin, Lijuan Wang, Lei Zhang, Jianfeng Gao, and Zicheng Liu. Vivo: Surpassing human performance in novel object captioning with visual vocabulary pre-training. In *AAAI*, February 2021. 8.2

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. 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)*, pages 2485–2494, Hong Kong, China, November 2019a. Association for Computational Linguistics. 2.2.1, 4.5, 6.5

Jiayuan Huang, Arthur Gretton, Karsten M Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 601–608, 2007. 2.2.1, 3.4.2, 3.6

Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4634–4643, 2019b. 8.2

Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *Advances in neural information process-*

ing systems, pages 103–112, 2019c. 1, 6.2

Zhicheng Huang, Zhaoyang Zeng, Yupan Huang, Bei Liu, Dongmei Fu, and Jianlong Fu. Seeing out of the box: End-to-end pre-training for vision-language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12976–12985, 2021. 8.2, 8.4.2

Taichi Iki and Akiko Aizawa. Effect of vision-and-language extensions on natural language understanding in vision-and-language models. *arXiv preprint arXiv:2104.08066*, 2021. 8.5

Pratik Jawanpuria, Arjun Balgovind, Anoop Kunchukuttan, and Bamdev Mishra. Learning multilingual word embeddings in latent metric space: a geometric approach. *Transactions of the Association for Computational Linguistics*, 7:107–120, 2019. 5.1, 5.4.2

Sébastien Jean, Orhan Firat, and Melvin Johnson. Adaptive scheduling for multi-task learning, 2019. 2.2.1, 6.5

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *arXiv preprint arXiv:2102.05918*, 2021. 8.1, 8.2, 8.3.4, 8.4.1, 8.4.3, 10.1

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019. 7.4.2

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. 2.2.1, 2.2.2, 4.1, 6.1, 6.2.1, 6.5

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, 2017. 7.4.1

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2984, 2018. 2.2.1, 5.1, 5.2.1, 5.4.2, 5.5

Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C. Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma, Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon. In-datacenter performance analysis of a tensor processing unit, 2017. 8.4.1

Mika Juuti, Tommi Gröndahl, Adrian Flanagan, and N. Asokan. A little goes a long way: Improving toxic language classification despite data scarcity. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2991–3009, Online, November 2020. Association for Computational Linguistics. 9.2

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 10.1

K Karthikeyan, Zihan Wang, Stephen Mayhew, and Dan Roth. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*, 2020. 2.2.1, 4.5, 6.5

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018. 6.5

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey, 2021. 8.2

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2020. 7.2, 7.3.1

Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision, 2021. 8.2

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4.3.2, 6.2.1, 7.4.2

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 2.2.1, 7.1, 7.2

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. Inducing Crosslingual Distributed Representations of Words. In *Proceedings of COLING 2012*, 2012. 5.1, 5.2.2, 5.5

Tomáš Kočiskỳ, Karl Moritz Hermann, and Phil Blunsom. Learning bilingual word representations by marginalizing alignments. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 224–229, 2014. 2.2.1, 5.5

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, 2019. 4.5

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. 8.2

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018. 6.2.1, 8.3.3

Sneha Reddy Kudugunta, Ankur Bapna, Isaac Caswell, Naveen Arivazhagan, and Orhan Firat. Investigating multilingual nmt representations at scale. *arXiv preprint arXiv:1909.02197*, 2019. 2.2.1, 4.5, 6.5

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. Data augmentation using pre-trained transformer models, 2021. 9.2

Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019. 1, 2.2.1, 4.1, 4.3.2, 4.4.2, 4.5, 5.2.2, 5.5, 6.1, 6.2.1, 6.5, 8.4.3, 8.4.3

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL*, pages 260–270. The Association for Computational Linguistics, 2016. ISBN 978-1-941643-91-4. 5.4.2

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*, 2018a. 5.1

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, et al. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, 2018b. 5.1, 5.2.2, 5.1, 5.3.1, 5.4.1, 5.4.2, 5.2, 5.4.3

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. 3.4.1

Kenton Lee, Kelvin Guu, Luheng He, Tim Dozat, and Hyung Won Chung. Neural data augmentation via example extrapolation, 2021. 9.2

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020. 1, 6.2

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language, 2019. 8.1, 8.2

Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. UNIMO: Towards unified-modal understanding and generation via cross-modal contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2592–2607, Online, August 2021. Association for Computational Linguistics. 8.2, 8.4.2

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. *ECCV 2020*, 2020. 8.1, 8.2, 8.4.2

Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. In *Advances in Neural Information Processing Systems*, pages 12060–12070, 2019a. 6.5

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, et al. Choosing transfer languages for cross-lingual learning. *arXiv preprint arXiv:1905.12688*, 2019b. 4.3.3

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, 2021. 8.1

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. 1, 2.2.1, 4.1, 7.1, 8.1, 9.1, 9.5

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, 2015. 2.2.1, 3.1, 3.3.1, 3.3.2, 3.4.2, 3.6

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017. 1

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017. 2.2.1, 7.1, 7.2

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 8.4.1

Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017. 4.3.1, 4.3.1

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 8.1, 8.2

Thang Luong, Hieu Pham, and Christopher D. Manning. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, 2015. 5.1

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting* *of the association for computational linguistics: Human language technologies*, pages 142–150, 2011. 9.3.2, 9.4.1

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*, 2018. 7.4.1

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995. 2.2.2, 7.2, 7.3.2

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 2.2.2, 7.1

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *ICLR*, 2013a. 2.2.1, 5.1, 5.5

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*, 2013b. 5.1, 5.2.1

Seungwhan Moon and Jaime Carbonell. Completely heterogeneous transfer learning with attention-what and what not to transfer. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2508–2514, 2017. 3.1, 3, 3.6

Phoebe Mulcaire, Jungo Kasai, and Noah A Smith. Polyglot contextual representations improve crosslingual transfer. 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 3912–3918, 2019. 2.2.1, 4.5, 6.5

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. volume 2011, page 5, 2011. 3.4.1

Graham Neubig and Junjie Hu. Rapid adaptation of neural machine translation to new languages. *arXiv preprint arXiv:1808.04189*, 2018. 4.1

Joakim Nivre, Mitchell Abrams, Željko Agić, Lars Ahrenberg, Lene Antonsen, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Mohammed Attia, et al. Universal dependencies 2.3. 2018. 4.3.2, 6.4.2

Aitor Ormazabal, Mikel Artetxe, Gorka Labaka, Aitor Soroa, and Eneko Agirre. Analyzing the limitations of cross-lingual word embedding mappings. *arXiv preprint arXiv:1906.05407*, 2019. 2, 5.5

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010. 1, 2.1, 2.2.1, 3.1, 3.2, 3.6, 4.5

Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011. 2.2.1, 3.4.2, 3.6

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, 2017. 4.3.2, 6.4.2

Yannis Papanikolaou and Andrea Pierleoni. Dare: Data augmented relation extraction with gpt-2, 2020. 9.2

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019. 2.2.1, 7.1

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 7.4.2

Barun Patra, Joel Ruben Antony Moniz, Sarthak Garg, Matthew R. Gormley, and Graham Neubig. Bilingual lexicon induction with semi-supervision in non-isometric embedding spaces. In *The 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, Florence, Italy, July 2019. 2

Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *AAAI Conference on Artificial Intelligence*, 2018. 3

Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge, 2017. 3.4.1

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014. 5.1

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language models as knowledge bases?, 2019. 9.2

Hieu Pham, Qizhe Xie, Zihang Dai, and Quoc V Le. Meta pseudo labels. *arXiv preprint arXiv:2003.10580*, 2020. 1, 2.2.1

Addison Phillips and Mark Davis. Tags for identifying languages. Technical report, BCP 47, RFC 4646, September, 2006. 6.1

Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy, July 2019. Association for Computational Linguistics. 2.2.1, 4.1, 4.5, 5.5, 6.1, 6.5

Ofir Press and Lior Wolf. Using the output embedding to improve language models. *arXiv preprint arXiv:1608.05859*, 2016. 8.4.1

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. 8.1, 8.3.1

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1, 2.2.1, 7.1, 7.2, 8.1, 9.1, 9.3.1

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 8.1, 8.2, 8.4.3, 8.4.3

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. 1, 2.2.1, 7.1, 7.2, 8.1, 8.4.1, 8.4.4, 8.4.4, 9.1, 9.3.1, 9.5, 9.4.2, 10.1

Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *Advances in Neural Information Processing Systems*, pages 6076–6085, 2017. 4.5

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of EMNLP 2016*, 2016. 7.4.1, 8.1

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018. 4.3.2

Tiago Ramalho and Marta Garnelo. Adaptive posterior learning: few-shot learning with a surprise-based memory module. In *International Conference on Learning Representations*, 2019. 7.3.2, 7.3, 7.4.4

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021. 8.1, 8.2

Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 7464–7473. IEEE, 2017. 3.4.1

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, pages 506–516, 2017. 4.3.3, 4.4.2

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. 8.2

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7008–7024, 2017. 8.4.1, 8.4.2

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online, November 2020. Association for Computational Linguistics. 9.2

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, pages 348–358, 2019. 2.2.1, 7.1

Clemens Rosenbaum, Ignacio Cases, Matthew Riemer, and Tim Klinger. Routing networks and the challenges of modular and compositional computation. *arXiv preprint arXiv:1904.12774*, 2019. 6.5

Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning*, volume 898, pages 1–4, 2005.

1, 2.2.2, 3.1, 3, 3.6, 4.5

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 2.2.2, 4.1, 4.5, 6.1, 6.5

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. A Survey of Cross-lingual Word Embedding Models. *Journal of Artificial Intelligence Research*, 65:569–631, 2019. 5.5

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 2.2.1, 6.5, 7.1, 7.2

Devendra Sachan and Graham Neubig. Parameter sharing methods for multilingual self-attentional translation models. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 261–271, Brussels, Belgium, October 2018. Association for Computational Linguistics. 6.2

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010. 3.4.1

Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. *Computer Vision and Pattern Recognition (CVPR)*, 2018. 2.2.1, 3.3.2, 3.4.2, 3.6

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 8.1

Tom Schaul, Diana Borsa, Joseph Modayil, and Razvan Pascanu. Ray interference: a source of plateaus in deep reinforcement learning. *arXiv preprint arXiv:1904.11455*, 2019. 6.5

Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*, 2020. 9.5, 9.4.2

Timo Schick and Hinrich Schütze. Generating datasets with pretrained language models, 2021. 9.2

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In *Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Minneapolis, USA, June 2019. 5.1

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, pages 4535–4544, 2018. 7.2, 7.4.2

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, pages 527–538, 2018. 6.3.1, 6.4.1, 6.5

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, 2016. 4.3.2

Jonathan Shen, Patrick Nguyen, Yonghui Wu, Zhifeng Chen, et al. Lingvo: a modular and scalable framework for sequence-to-sequence modeling, 2019. 8.4.1

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2990–2999, 2017. 7.2

Aditya Siddhant, Melvin Johnson, Henry Tsai, Naveen Ari, Jason Riesa, Ankur Bapna, Orhan Firat, and Karthik Raman. Evaluating the cross-lingual effectiveness of massively multilingual neural machine translation. In *AAAI*, pages 8854–8861, 2020. 6.1

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 5.1, 5.2.1

Shagun Sodhani, Sarath Chandar, and Yoshua Bengio. Toward training recurrent neural networks for lifelong learning. *Neural computation*, 32(1):1–35, 2020. 7.2

Anders Søgaard, Željko Agić, Héctor Martínez Alonso, Barbara Plank, Bernd Bohnet, and Anders Johannsen. Inverted indexing for cross-lingual nlp. In *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2015)*, 2015. 2.2.1, 5.5

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 778–788, Melbourne, Australia, July 2018. Association for Computational Linguistics. 4.5, 2

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936, 2019. 5.2

Lucia Specia, Stella Frank, Khalil Sima'An, and Desmond Elliott. A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 543–553, 2016. 8.3

Pablo Sprechmann, Siddhant Jayakumar, Jack Rae, Alexander Pritzel, Adria Puigdomenech Badia, Benigno Uria, Oriol Vinyals, Demis Hassabis, Razvan Pascanu, and Charles Blundell. Memory-based parameter adaptation. In *International Conference on Learning Representations*, 2018. 2.2.1, 7.1, 7.2, 7.3.1

Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. *arXiv preprint arXiv:2103.01913*, 2021. 8.4.3

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 7.4.2

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2020. 8.1, 8.2

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. *arXiv preprint arXiv:1811.00491*, 2018. 8.4.1

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. pages 443–450. Springer, 2016. 2.2.1, 3.3.2, 3.4.2, 3.6

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. Lamol: Language modeling for lifelong language learning. In *International Conference on Learning Representations*, 2020. 7.2, 7.1

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. 6.2.1

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *ICLR*, 2014. 3.4.2

Derek Tam, Rakesh R Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. Improving and simplifying pattern exploiting training. *arXiv preprint arXiv:2103.11955*, 2021. 9.5

Hao Tan and Mohit Bansal. LXMERT: Learning cross-modality encoder representations from transformers. 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)*, pages 5100–5111, Hong Kong, China, November 2019. Association for Computational Linguistics. 8.1, 8.2, 8.4.2

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 1

Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. Multilingual neural machine translation with knowledge distillation. *arXiv preprint arXiv:1902.10461*, 2019. 2.2.2, 4.1

Erik F. Tjong Kim Sang. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *CoNLL*, pages 1–4, 2002. 5.4.1

Erik F Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *CoNLL*, pages 142–147, 2003. 5.4.1

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019. 7.3, 7.4.4

Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4068–4076, 2015. 2.2.1, 3, 3.4.2, 3.6

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 4, 2017. 3.3.1, 3.3.2

Masatoshi Uehara, Issei Sato, Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. Generative adversarial nets from a density ratio estimation perspective. *arXiv preprint arXiv:1610.02920*, 2016. 3.3.2

Selen Uguroglu and Jaime Carbonell. Feature selection for transfer learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 430–442. Springer, 2011. 2.2.1, 3.6

Shagun Uppal, Sarthak Bhagat, Devamanyu Hazarika, Navonil Majumdar, Soujanya Poria, Roger Zimmermann, and Amir Zadeh. Multimodal research in vision and language: A review of current and emerging trends, 2020. 8.2

Jakob Uszkoreit, Jay Ponte, Ashok Popat, and Moshe Dubiner. Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1101–1109, 2010. 6.2.1

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 4.1, 4.3.2, 6.2.1, 8.1, 10.1

Raúl Vázquez, Alessandro Raganato, Jörg Tiedemann, and Mathias Creutz. Multilingual NMT with a language-independent attention bridge. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 33–39, Florence, Italy, August 2019. Association for Computational Linguistics. 6.2

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *(IEEE) Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3.4.1

Ivan Vulić and Marie-Francine Moens. Bilingual distributed word representations from document-aligned comparable data. *Journal of Artificial Intelligence Research*, 55:953–994, 2016. 2.2.1, 5.5

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. *arXiv preprint arXiv:1804.07461*, 2018. 8.1, 8.4.4

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Proceedings of NeurIPS 2019*, 2019a. 1.2, 9.4.2

Chenguang Wang, Xiao Liu, and Dawn Song. Language models are open knowledge graphs, 2020a. 9.2

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. Balancing training for multilingual neural machine translation. *arXiv preprint arXiv:2004.06748*, 2020b. 2.2.1, 4.5, 6.2, 6.5

Zirui Wang and Jaime Carbonell. Towards more reliable transfer learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 794–810, 2018. 3.4.2, 4.5, 6.5, 7.4.4

Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11293–11302, 2019b. 4.1, 4.5, 5.5, 6.2, 7.1

Zirui Wang, Zachary C Lipton, and Yulia Tsvetkov. On negative interference in multilingual models: Findings and a meta-learning treatment. In *EMNLP*, 2020c. 2.2.1, 6.1, 6.4.2, 6.5

Zirui Wang, Jiateng Xie, Ruochen Xu, Yiming Yang, Graham Neubig, and Jaime Carbonell. Cross-lingual alignment vs joint training: A comparative study and a simple unified framework. In *International Conference on Learning Representations*, 2020d. 2.2.1, 4.5, 6.5

Jason W. Wei and Kai Zou. EDA: easy data augmentation techniques for boosting performance on text classification tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *EMNLP-IJCNLP*, 2019. 9.2

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3(1):9, 2016. 1, 2.1, 2.2.2, 3.1, 3.2, 3.6, 7.2

Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017. 8.3

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019. 7.4.2

Shijie Wu and Mark Dredze. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. 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)*, pages 833–844, Hong Kong, China, November 2019. Association for Computational Linguistics. 2.2.1, 4.1, 4.5, 5.5, 6.1, 6.5

Shijie Wu, Alexis Conneau, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. Emerging cross-lingual structure in pretrained language models. In *Annual Conference of the Association for Computational Linguistics (ACL)*, July 2020. 2.2.1, 4.3.1, 4.5, 6.1, 6.5

Min Xiao and Yuhong Guo. Distributed word representation learning for cross-lingual dependency parsing. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 119–129, 2014. 5.5

Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better, 2021. 8.3.3

Jiateng Xie, Zhilin Yang, Graham Neubig, Noah A Smith, and Jaime Carbonell. Neural cross-lingual named entity recognition with minimal resources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 369–379, 2018. 5.1

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. Visual entailment: A novel task for fine-grained image understanding. *arXiv preprint arXiv:1901.06706*, 2019a. 8.4.1, 8.3

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019b. 9.2, 9.4.1, 9.4.1

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011, Denver, Colorado, May–June 2015. Association for Computational Linguistics. 2.2.1, 5.1, 5.2.1, 5.5

Haiyang Xu, Ming Yan, Chenliang Li, Bin Bi, Songfang Huang, Wenming Xiao, and Fei Huang. E2E-VLP: End-to-end vision-language pre-training enhanced by visual learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 503–513, Online, August 2021. Association for Computational Linguistics. 8.2

Ruochen Xu, Yiming Yang, Naoki Otani, and Yuexin Wu. Unsupervised cross-lingual transfer of word embedding spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2465–2474, 2018. 5.5

Liu Yang, Steve Hanneke, and Jaime Carbonell. A theory of transfer learning with applications to active learning. *Machine learning*, 90(2):161–189, 2013. 2.2.1, 3.6

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019. 1, 2.2.1, 4.1, 7.1, 8.1, 9.1, 9.3.1, 9.2

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomas Kocisky, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, et al. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*, 2019. 2.2.1, 7.1, 7.2

Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *International Conference on Learning Representations*, 2018. 2.2.1, 7.1, 7.2

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014. 2.2.1, 3.6

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. In *ICLR*, 2018. 9.2

Fei Yu, Jiji Tang, Weichong Yin, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. Ernie-vil: Knowledge enhanced vision-language representations through scene graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3208–3216, 2021. 8.2

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*, 2020. 4.3.1, 6.1, 6.2, 6.3.1, 6.4.1, 6.5

Yao-Liang Yu and Csaba Szepesvári. Analysis of kernel mean matching under covariate shift. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1147–1154, 2012. 3.1, 3.3.2

Amir R Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018. 3.1, 6.5

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3987–3995. JMLR. org, 2017. 2.2.1, 7.1, 7.2

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. Earth mover's distance minimization for unsupervised bilingual lexicon induction. In *Proceedings of EMNLP 2017*, pages 1934–1945, 2017a. 5.5

Mozhi Zhang, Keyulu Xu, Ken-ichi Kawarabayashi, Stefanie Jegelka, and Jordan Boyd-Graber. Are girls neko or sh\= ojo? cross-lingual alignment of non-isomorphic embeddings with iterative normalization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, page 3180–3189, 2019. 5.1

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5579–5588, June 2021. 8.1, 8.2, 8.4.1, 8.4.2

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015. 7.4.1, 9.4.1

Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. Adversarial feature matching for text generation. In *ICML*, 2017b. 5.1, 5.5

Yu Zhang and Dit-Yan Yeung. A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536*, 2012. 6.5

Chunting Zhou, Xuezhe Ma, Di Wang, and Graham Neubig. Density matching for bilingual word embedding. 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 1588–1598, 2019. 5.1, 5.4.2

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. 5.1