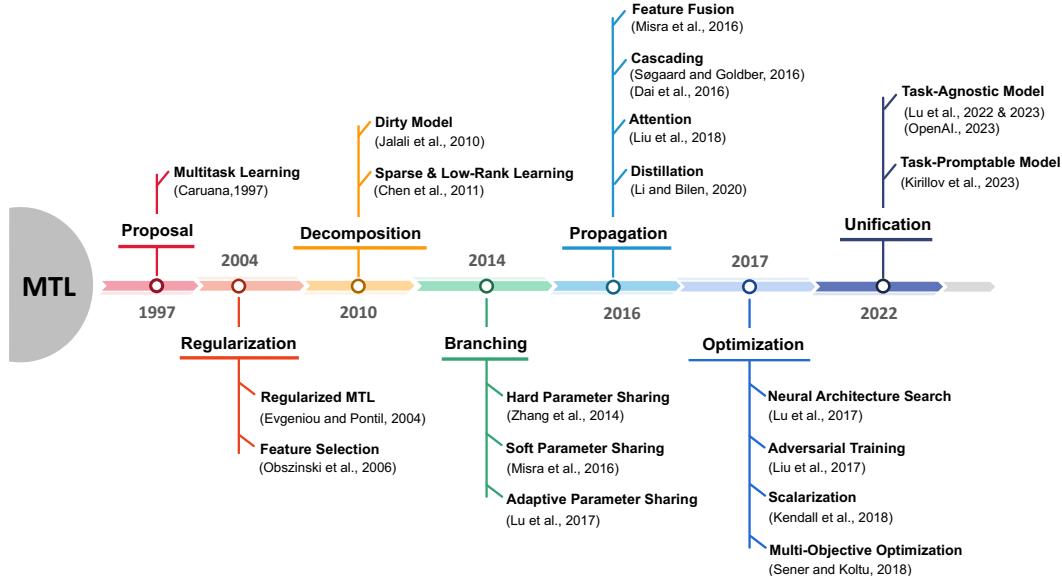


## Unleashing the Power of Multi-Task Learning: A Comprehensive Survey Spanning Traditional, Deep, and Pretrained Foundation Model Eras

Jun Yu<sup>♦♡, †, ‡</sup>, Yutong Dai<sup>◊</sup>, Xiaokang Liu<sup>♡▲</sup>, Jin Huang<sup>◆</sup>, Yishan Shen<sup>♡</sup>, Ke Zhang<sup>◆</sup>, Rong Zhou<sup>◆</sup>, Eashan Adhikarla<sup>◆</sup>, Wenxuan Ye<sup>◆</sup>, Yixin Liu<sup>◆</sup>, Zhao ming Kong<sup>◆</sup>, Kai Zhang<sup>◆</sup>, Yilong Yin<sup>◆</sup>, Vinod Namboodiri<sup>◆△</sup>, Brian D. Davison<sup>◆</sup>, Jason H. Moore<sup>▽</sup>, Yong Chen<sup>♡, ‡</sup>

<sup>♦</sup> Department of Computer Science and Engineering, Lehigh University, USA  
<sup>♡</sup> Department of Biostatistics, Epidemiology and Informatics, University of Pennsylvania, USA  
<sup>◊</sup> Department of Industrial and Systems Engineering, Lehigh University, USA  
<sup>▲</sup> Department of Statistics, University of Missouri, USA  
<sup>◆</sup> School of Software, Shandong University, China  
<sup>◆</sup> Department of Computer Science, University of Hong Kong, China  
<sup>◆</sup> Department of Computer Science and Engineering, South China University of Technology, China  
<sup>△</sup> Department of Community and Population Health, Lehigh University, USA  
<sup>▽</sup> Department of Computational Biomedicine, Cedars-Sinai Medical Center, USA



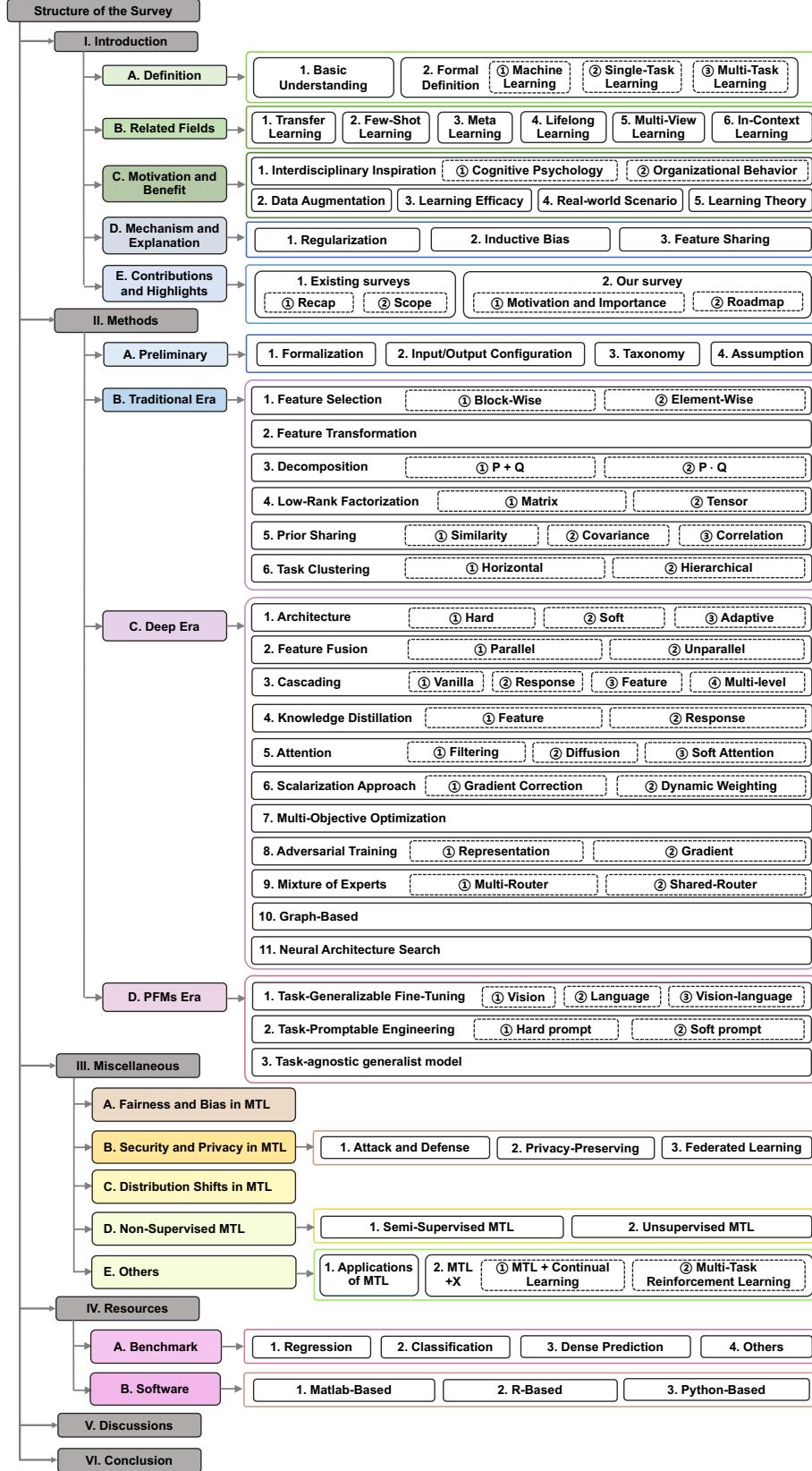
**Figure 1.** Significant landmarks in the evolution of Multi-Task Learning (MTL) highlighted over time.

<sup>†</sup>This work includes efforts as a visiting student at Upenn.

<sup>‡</sup>Corresponding to juy220@lehigh.edu or ychen123@pennmedicine.upenn.edu.

**ABSTRACT.** Multi-Task Learning (MTL) is a learning paradigm that effectively leverages both task-specific and shared information to address multiple related tasks simultaneously. In contrast to Single-Task Learning (STL), MTL offers a suite of benefits that enhance both the training process and the inference efficiency. MTL's key advantages encompass streamlined model architecture, performance enhancement, and cross-domain generalizability. Over the past twenty years, MTL has become widely recognized as a flexible and effective approach in various fields, including computer vision, natural language processing, recommendation systems, disease prognosis and diagnosis, and robotics. This survey provides a comprehensive overview of the evolution of MTL, encompassing the technical aspects of cutting-edge methods from traditional approaches to deep learning and the latest trend of pretrained foundation models. Our survey methodically categorizes MTL techniques into five key areas: regularization, relationship learning, feature propagation, optimization, and pre-training. This categorization not only chronologically outlines the development of MTL but also dives into various specialized strategies within each category. Furthermore, the survey reveals how the MTL evolves from handling a fixed set of tasks to embracing a more flexible approach free from task or modality constraints. It explores the concepts of task-promptable and -agnostic training, along with the capacity for zero-shot learning, which unleashes the untapped potential of this historically coveted learning paradigm. Overall, we hope this survey provides the research community with a comprehensive overview of the advancements in MTL from its inception in 1997 to the present in 2023. We address present challenges and look ahead to future possibilities, shedding light on the opportunities and potential avenues for MTL research in a broad manner. This project is publicly available at <https://github.com/junfish/Awesome-Multitask-Learning>.

**Keywords:** Deep Learning, Generative Pretrained Transformers, Multi-Objective Optimization, Multi-Task Learning, Pretrained Foundation Models, Prompt Learning

**Figure 2.** The structure of this survey.

**Table 1.** Alphabetically sorted index table of acronyms.

| Abbreviation | Expanded Form                     | Abbreviation | Expanded Form                       |
|--------------|-----------------------------------|--------------|-------------------------------------|
| AD           | Alzheimer’s Disease               | AGM          | Accelerated Gradient Method         |
| APM          | Accelerated Proximal Method       | CE           | Cross-Entropy                       |
| CNN          | Convolutional Neural Network      | CT           | Computed Tomography                 |
| CV           | Computer Vision                   | DA           | Domain Adaptation                   |
| DNN          | Deep Neural Network               | FCN          | Fully Convolutional Network         |
| FNN          | Feedforward Neural Network        | FSL          | Few Shot Learning                   |
| GAN          | Generative Adversarial Network    | GCN          | Graph Convolutional Network         |
| GNN          | Graph Neural Network              | GP           | Gaussian Process                    |
| GPT          | Generative Pretrained Transformer | GPU          | Graphics Processing Unit            |
| GRL          | Gradient Reversal Layer           | I/O          | Input/Output                        |
| KD           | Knowledge Distillation            | LLM          | Large Language Model                |
| LSTM         | Long Short-Term Memory            | MAP          | Maximum A Posteriori                |
| MCI          | Mild Cognitive Impairment         | MDP          | Markov Decision Process             |
| MIM          | Masked Image Modeling             | MIML         | Multi-Instance Multi-Label learning |
| MIMO         | Multi-Input Multi-Output          | MISO         | Multi-Input Single-Output           |
| ML           | Machine Learning                  | MLM          | Masked Language Modeling            |
| MLP          | Multi-Layer Perceptron            | MoE          | Mixture-of-Experts                  |
| MOO          | Multi-Objective Optimization      | MRI          | Magnetic Resonance Imaging          |
| MSE          | Mean Squared Error                | MTL          | Multi-Task Learning                 |
| MTRL         | Multi-Task Reinforcement Learning | MVL          | Multi-View Learning                 |
| NAS          | Neural Architecture Search        | NLI          | Natural Language Inference          |
| NLP          | Natural Language Processing       | OCR          | Optical Character Recognition       |
| OOD          | Out-Of-Distribution               | PET          | Positron Emission Tomography        |
| PFM          | Pretrained Foundation Model       | PSD          | Positive Semi-Definite              |
| RL           | Reinforcement Learning            | RNN          | Recurrent Neural Network            |
| seq2seq      | sequence to sequence              | SIMO         | Single-Input Multi-Output           |
| SNP          | Single Nucleotide Polymorphism    | SGD          | Stochastic Gradient Descent         |
| SSL          | Self-Supervised Learning          | SOTA         | State-Of-The-Art                    |
| STL          | Single-Task Learning              | SVD          | Singular Value Decomposition        |
| SVM          | Support Vector Machine            | TL           | Transfer Learning                   |
| TPU          | Tensor Processing Unit            | VLM          | Vision-Language Model               |
| VQA          | Visual Question Answering         | ZSL          | Zero-Shot Learning                  |

This table excludes abbreviations pertaining to datasets, institutions, and newly proposed methods.

## 1. INTRODUCTION

In the introduction, we hope to answer the following five research questions (RQs) before we overview the methodologies of Multi-task Learning (MTL):

- **RQ1:** What is the concept and definition of MTL? (See § 1.1)
- **RQ2:** How does MTL distinguish itself from other learning paradigms? (See § 1.2)
- **RQ3:** What motivates the use of MTL in learning scenarios? (See § 1.3)
- **RQ4:** What underlying principles does the efficacy of MTL rest on? (See § 1.4)
- **RQ5:** In what ways does our survey differentiate from previous studies? (See § 1.5)

In § 1.1, we progressively introduce Multi-Task Learning (MTL), starting with a broad sense and culminating in a formal definition. Subsequently, § 1.2 explores the position of MTL within the Machine Learning (ML) landscape, drawing comparisons with related paradigms such as Transfer Learning (TL), Few-Shot Learning (FSL), lifelong learning, Multi-View Learning (MVL), to name a few. § 1.3 delves into the motivations for employing MTL, offering insights from both explicit and subtle angles, while also addressing how MTL benefits the involved tasks. In § 1.4, we delve deeper into the fundamental mechanisms and theories underpinning MTL, specifically: 1) regularization, 2) inductive bias, and 3) feature sharing, providing an understanding of its underlying principles. Finally, § 1.5 reviews existing surveys on MTL, underscoring the unique contributions of our survey and laying out a structured roadmap for the remainder of this work. The structure of our survey is depicted in Fig. 2. Before delving into this survey, readers can quickly refer to Table 1 for a list of acronyms not related to datasets, institutions, and newly proposed methods, while an overview of mathematical notations is provided in Table 3 and Table 6.

### 1.1. Definition.

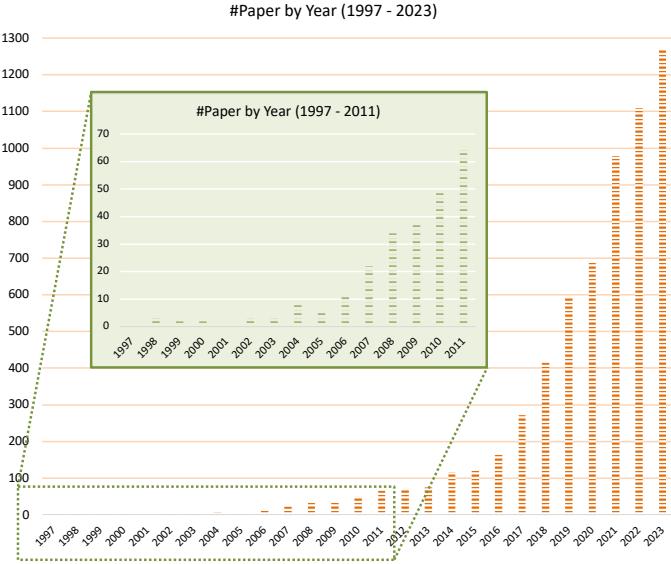
The increasing popularity of MTL over the past few decades is evident in **Fig. 3**, which displays the trend in the number of papers associated with “*allintitle: ‘multitask learning’ OR ‘multi-task learning’*” as a keyword search, according to data from Google Scholar.<sup>1</sup>

As the name suggests, MTL is a subfield of ML where multiple tasks are jointly learned. In this manner, we hope to leverage useful information across these related tasks and break from the tradition of performing different tasks in isolation. In Single-Task Learning (STL), data specific to the task at

hand is the only source to couch a learner. However, MTL can conveniently transfer extra knowledge learned from other tasks. The essence of MTL is to exploit consensual and complementary information among tasks by combining data resources and sharing knowledge. This sheds light on a better learning paradigm that can reduce memory burden and data consumption, and improve training speed and testing performance. For instance, learning the monocular depth estimation (scaling the distance to the camera) (Eigen et al., 2014) and semantic segmentation (assigning a class label to every pixel value) (K.-S. Fu & Mui, 1981) simultaneously in images is beneficial since both tasks need to perceive meaningful objects. MTL has become increasingly ubiquitous as experimental and theoretical analyses continue to validate its promising results. For example, using Face ID to unlock an iPhone is a typical but imperceptible MTL application that involves simultaneously locating the user’s face and identifying the user. In general, multitasking occurs when we attempt to handle two or more objectives during the optimization stage in practice.

Consequently, MTL exists everywhere in ML, even when performing STL with regularization. This can be understood as having one target task and an additional artificial task of human preference, such as learning a constrained model via  $\ell_2$  regularizer or a parsimonious model via  $\ell_1$  regularizer. These hypothesis preferences can serve as an inductive bias to enhance an inductive learner (Caruna, 1993). In the early exploration of MTL (R. Caruana, 1997), the extra information that the involved tasks provide is regarded as a domain-specific inductive bias for the other tasks. Since collecting training signals from other tasks is more practical than acquiring inductive bias from model design or human expertise, we can thus empower any ML models via this MTL paradigm.

**1.1.1. Formal Definition.** To comprehensively understand MTL, we provide a formal definition of MTL. Suppose we have a sample dataset  $\mathbf{X}$  drawn from the feature space  $\mathcal{X}$ , and its respective ground-truth label set  $\mathbf{Y}$  drawn from the label space  $\mathcal{Y}$ . We can define *experience*  $\mathcal{E} \subseteq \{\mathbf{X}, \mathbf{Y}\}$ ,



**Figure 3.** The total number of published papers (*y*-axis) has surged for the MTL topic from 1997 to 2023 (*x*-axis).

<sup>1</sup><https://scholar.google.com>

*domain*  $\mathcal{D} = (\mathcal{X}, P(\mathbf{X}))$ , and *task*  $\mathcal{T} = (\mathcal{Y}, f)$ , where  $P(\mathbf{X})$  is the distribution of  $\mathbf{X}$  and  $f$  maps a data sample  $\mathbf{x} \in \mathbf{X}$  to a prediction  $\tilde{\mathbf{y}} \in \mathbf{Y}$ . These predictive values consist of the predictive label set  $\tilde{\mathbf{Y}} = \{\tilde{\mathbf{y}} | \tilde{\mathbf{y}} = f(\mathbf{x}), \mathbf{x} \in \mathbf{X}\}$ . Following the ML settings, we should define a *measurement*  $\mathcal{P} = (\mathbf{Y}, \tilde{\mathbf{Y}}, \mathcal{L})$ , where  $\mathcal{L}$  is a function to measure the distance between any pairs of  $(\mathbf{y}, \tilde{\mathbf{y}})$ . More basic notations please refer to Table 3. Based on the definitions of four basic elements (*experience*, *domain*, *task*, and *measurement*) above, we first restate the general definition of machine learning by Mitchell (1997) to a more exact form as follows.

**Definition 1** (Machine Learning, Mitchell (1997)). *A computer program is said to learn from experience  $\mathcal{E}$  with respect to a set of tasks  $\{\mathcal{T}^{(t)}\}_{t=1}^T$  and performance measurement  $\mathcal{P}$ , if its performance at tasks  $\{\mathcal{T}^{(t)}\}_{t=1}^T$ , as measured by  $\mathcal{P}$ , improves with experience  $\mathcal{E}$ .*

The definition above inherently considers both single-task and multi-task scenarios during the ML process but deviates from a meticulous definition to characterize MTL that includes recent developments. Now, let us first define STL to induce the formal definition of MTL.

**Definition 2** (Single-Task Learning). *A type of machine learning specified by  $\mathcal{E}, \{\mathcal{T}^{(t)}\}_{t=1}^T$  and  $\mathcal{P}$ , where  $\{\mathcal{T}^{(t)}\}_{t=1}^T$  contains only one task (i.e.  $T = 1$ ) on a specific domain  $\mathcal{D}$ .*

As recent developments in MTL focus more on heterogeneous tasks (e.g., regression + classification) than homogeneous ones, each task should be represented by its own *experience*  $\mathcal{E}$  on its corresponding *domain*  $\mathcal{D}$ . Due to this diversity, we always employ distinct *measurement*  $\mathcal{P}$  to evaluate the learning performance of each task. We accordingly define the MTL as follows.

**Definition 3** (Multi-Task Learning). *A super set of STL specified by  $\bigcup_{t=1}^T \mathcal{E}^{(t)}, \{\mathcal{T}^{(t)}\}_{t=1}^T$  and  $\{\mathcal{P}^{(t)}\}_{t=1}^T$ , where experience  $\mathcal{E}^{(t)} \subseteq \{\mathbf{X}^{(t)}, \mathbf{Y}^{(t)}\}$  is with respect to task  $\mathcal{T}^{(t)}$  on its corresponding domain  $\mathcal{D}^{(t)}$ . Accordingly, MTL is a computer program to learn from the experience set  $\bigcup_{t=1}^T \mathcal{E}^{(t)}$  with respect to the task set  $\{\mathcal{T}^{(t)}\}_{t=1}^T$  and the corresponding performance measurement set  $\{\mathcal{P}^{(t)}\}_{t=1}^T$ , if its total performance at any task  $\mathcal{T}^{(t)}$ , as measured by its corresponding  $\mathcal{P}^{(t)}$ ,  $t = 1, \dots, T$ , improves with experience set  $\bigcup_{t=1}^T \mathcal{E}^{(t)}$ .*

We note that the formal MTL definition above has no conflict with the homogeneous or heterogeneous MTL.

## 1.2. Related Fields.

Having established a formal definition of MTL grounded in fundamental ML elements, a thorough understanding can be achieved by analytically comparing it with related domains. These include Transfer Learning (TL), Meta-Learning, and In-Context Learning (ICL), among others. This comparison not only clarifies the distinct characteristics of MTL but also situates it within the broader context of these interconnected fields.

*Transfer Learning (TL).* TL (Pan & Yang, 2009) is a prevalent learning paradigm that solves the problem of lacking labeled data when applying ML to real-world data (Pan & Yang, 2009; Zhuang et al., 2020). Specifically, TL improves the performance of a target model on target domains by transferring the knowledge in different but related source domains to the target domains. Such properties make TL well-appreciated in real-world applications, such as healthcare (Kao et al., 2021; Pérez-García et al., 2021; Song et al., 2021) and recommender systems (Bonab et al., 2021; W. Liu et al., 2021; Y. Zhang et al., 2021a). According to the availability of labels in the source and target domains, TL is categorized into three types, i.e., transductive TL (aka *Domain Adaptation (DA)*), Patel et al. (2015) and Redko et al. (2019)), inductive TL, and unsupervised TL (Pan &

Yang, 2009; Zhuang et al., 2020).

*Few-Shot Learning (FSL).* FSL (Fei-Fei et al., 2006; Fink, 2004; Y. Wang et al., 2020) is a specific application case of TL. It aims at obtaining a model for the target task under a certain scenario where limited labeled samples from the target domain are available (Y. Wang et al., 2020). FSL is well-acknowledged in tackling different real-world problems such as identifying atypical ailments (X. Jia et al., 2020; Quellec et al., 2020), visual navigation (Al-Halah et al., 2022; Luo et al., 2021), and cold-start item recommendation (H. Sun et al., 2021; Y. Zhang et al., 2021b)).

*Meta-Learning.* Meta-Learning (Hospedales et al., 2021) is an implementation approach to achieve TL. The main concept is to obtain a meta-learner (a model) that can have satisfying performance for an unseen target domain (Hospedales et al., 2021). Such meta-learner first extracts the meta-knowledge, i.e., the universally applicable principles, across source domains. With meta-knowledge, the meta-learner can be easily generalized to the target domain by leveraging the target samples. Meta-learning has been successfully applied in various problems such as hyper-parameter optimization (Bohdal et al., 2021; Raghu et al., 2021), algorithm selection for data mining (Simchowitz et al., 2021), and neural architecture search (NAS) (Ding et al., 2022; H. Lee et al., 2021).

Though TL paradigms, including FSL and meta-learning, involve multi-domain data, their ultimate goal is to obtain a model with satisfied performance or can be easily generalized to one target task. In other words, TL leverages the knowledge in different tasks to assist the model in learning a single task, which intersects with MTL according to our definition in Definition 3. Thus, TL can bring merits to MTL, such as capturing the relations among tasks and extracting shared knowledge among involved tasks. Notably, the transfer of knowledge from pretrained foundation models (PFMs) proves beneficial for a myriad of downstream tasks in recent advancements (Bommasani et al., 2021; C. Zhou et al., 2023).

*Lifelong Learning.* Lifelong Learning (Parisi et al., 2019), aka Continual Learning, Sequential Learning, or Incremental Learning, studies the problem of learning from an infinite stream of data (De Lange et al., 2021). The goal is to gradually extend the acquired knowledge and use it for future data, mitigating the occurrence of catastrophic forgetting or interference (McClelland et al., 1995). With only a small portion of the input data from one or few tasks available at once, lifelong learning particularly tends to preserve the knowledge learned from the previous input when learning on new data, i.e., addressing the stability-plasticity dilemma (Grossberg, 2012). There are extensive applications of lifelong learning in solving tasks in ever-evolving systems, such as recommendations (Z. Chen et al., 2021; Yao et al., 2021) and anomaly detection (Doshi & Yilmaz, 2022; H. Peng et al., 2021). Lifelong learning differs from MTL in the sense that its training object is a dynamic data stream, while MTL studies data from multiple tasks available at the beginning of the learning process.

*Multi-View Learning (MVL).* MVL (Y. Li et al., 2018; C. Xu et al., 2013; J. Zhao et al., 2017) studies the problem of jointly learning from multi-view data samples, whose goal is to optimize the generalization performance for the jointly learning model (Y. Li et al., 2018). In real-world applications, the multi-view data indicates objects being described by multi-modal measurements, such as image+text, audio+video, and audio+articulation. Multi-Instance Multi-Label learning (MIML) (Z.-H. Zhou et al., 2012) is a specific subtype of MVL, where an example is described by multiple instances and associated with multiple class labels. Due to the vast existence of multi-view

data in realistic, MVL has attracted much attention in both research and industry, and the respective solutions play essential roles in cross-media retrieval (P.-Y. Huang et al., 2020; Zhen et al., 2019), video analysis (Y. Wang, Dong, et al., 2022; Zellers et al., 2021), recommender system (Chai et al., 2022; W. Wei et al., 2022), etc. MVL, including MIIML, can be considered a specialized form of MTL, where the input contains data from multiple domains that are handled as distinct tasks, but the output is still in one label space.

*In-Context Learning (ICL).* ICL (Q. Dong et al., 2022) has aroused interest as a novel learning paradigm for natural language processing (NLP) within Large Language Models (LLMs). ICL relies on templates in natural language that can demonstrate different tasks, such as solving mathematical reasoning problems (J. Wei, Wang, et al., 2022) and learning natural language inference (NLI) (H. Liu et al., 2021). LLMs can then make predictions by taking this demonstration and its corresponding query pair as input. While both ICL and MTL involve leveraging shared knowledge or context to enhance task generalizability, ICL is specifically tailored to the target task within a narrower scope in real-world applications. However, recent large PFMs, like GPT-4 (OpenAI, 2023), are inherently task-agnostic, accommodating various tasks owing to the diversity of demonstration templates encountered during their large-scale training stage.

### 1.3. Motivation and Benefit.

MTL can be motivated from the following five perspectives with different benefits: cognitive/social psychology, data augmentation, learning efficacy, real-world scenarios, and learning theory.

- Psychologically, humans are inherent with flexible adaptability to new problems and settings, as the human learning process can transfer knowledge from one experience to another (Council et al., 2000). Therefore, MTL is inspired by simulating this process to empower a model with the potentiality of multitasking. Coincidentally, another example of this knowledge transfer happens among organizations (Argote et al., 2000). It is proved that organizations with more effective knowledge transfer are more productive and likely to survive than those with less. These prior successes of transfers or mutualizations in other areas encourage the joint learning of tasks in ML (R. Caruana, 1997).
- In the pre-big data era, real-world problems were usually represented by small but high-dimensional datasets (# samples < # features). This data bottleneck forces early methods to learn a sparse-structured model, which always leads to a parsimonious solution to a problem with insufficient data. However, the MTL emerged to aggregate labeled data from different domains or tasks to enlarge the training dataset against overfitting.
- The pursuit of efficiency and effectiveness is also one of the motivations. MTL can aggregate data from different sources together, and the joint training process of multiple tasks can save both computation and storage resources. In addition, the potential of performance enhancement makes it popular in research communities. In brief, universal representations for any tasks can be learned from multi-source data, and benefit all tasks in terms of both the learning cost and performance.
- Motivated by the majority of real-world problems naturally being multimodal or multitasking, MTL is proposed to remedy the suboptimal achieved by STL that only models parts of the whole problem separately. For example, predicting the progression of Alzheimer’s Disease (AD) biomarkers for Mild Cognitive Impairment (MCI) risk and clinical diagnosis is simultaneously based on multimodal data such as computed tomography (CT), Magnetic Resonance Imaging (MRI), and Positron Emission Tomography (PET) (H. Chen et

al., 2022; Jie et al., 2015; Kwak et al., 2018). Autonomous driving, another example, also involves multiple subtasks to calculate the final prediction (Chowdhuri et al., 2019; Z. Yang et al., 2018), including the recognition of surrounding objects, adjustments to the fastest route according to the traffic conditions, the balance between efficiency and safety, etc.

- From the perspective of learning theory, bias-free learning is proved to be impossible (Mitchell, 1980), so we can motivate the MTL by using the extra training signals for related tasks. Generally, MTL is one of the ways to achieve inductive transfer via multitasking assistance, which improves both learning speed and generalization. Specifically, during the process of the combined training of multiple tasks, some tasks can be provided inductive bias from other related tasks, and these stronger inductive biases (compared with universal regularizers, e.g.,  $\ell_2$ ) enable the knowledge transfer and yield more generalization abilities on a fixed training dataset. In other words, task-related biases make a learner prefer hypotheses that can explain more than one task and prevent specific task from overfitting.

#### 1.4. Mechanism and Explanation.

In this section, we explore three key mechanisms – regularization, inductive bias, and feature sharing – shedding light on how MTL operates to achieve enhanced performance across multiple tasks.

*Regularization.* In MTL, the total loss function is a combination of multiple loss terms with respect to each task. The related tasks play a role as regularizers, enhancing the generalizability across them. The hypothesis space of an MTL model is confined to a more limited scope as it tackles multiple tasks simultaneously. Consequently, this constraint on the hypothesis space reduces model complexity, mitigating the risk of overfitting.

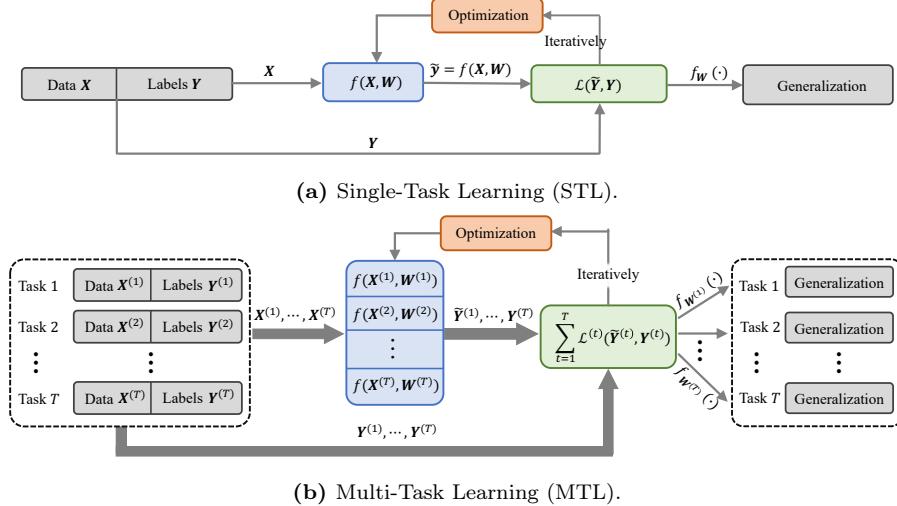
*Inductive Bias.* The training signals from co-training tasks act as mutual inductive biases due to their shared domain information. These biases facilitate cross-task knowledge transfer during training, guiding the model to favor task-related concepts rather than the tasks themselves. Consequently, this broadens the model’s horizons beyond a singular task, enhancing its generalization capabilities for unseen out-of-distribution (OOD) data.

*Feature Sharing.* MTL can enable feature sharing across related tasks. One approach involves selecting overlapping features and maximizing their utility across all tasks. This is referred to as “eavesdropping” (Ruder, 2017), considering that some features may be unavailable for specific tasks but can be substituted by those learned from related tasks. Another way is to concatenate all the features extracted by different tasks together; these features can be holistically used across tasks via linear combination or nonlinear transformation.

Overall, MTL can be an efficient and effective way to boost the performance of the ML model on multiple tasks by regularization, inductive transfer, and feature sharing.

#### 1.5. Contributions and Highlights.

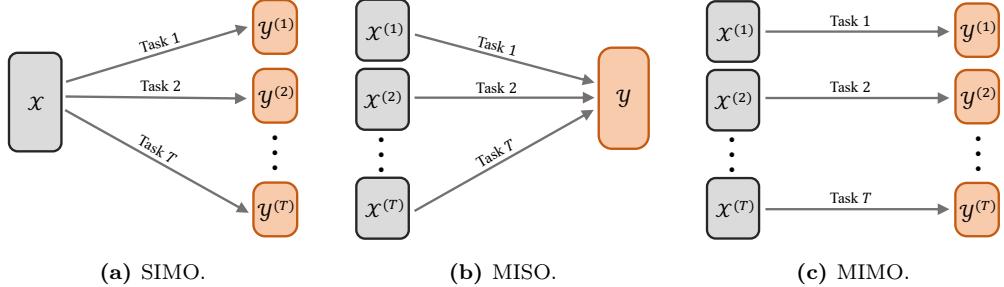
*Existing Surveys.* Ruder (2017) is a pioneering survey in MTL, offering a broad overview of MTL and focusing on advances in deep neural networks from 2015 to 2017. Thung and Wee (2018) reviews MTL methods from a taxonomy perspective of input-output variants, mainly concentrating on traditional MTL prior to 2016. These two reviews can be complementary materials to each other. Vafaeikia et al. (2020) is an incomplete survey that briefly reviews recent deep MTL approaches, particularly focusing on the selection of auxiliary tasks for enhanced learning performance. Crawshaw (2020) presents the well-established and advanced MTL methods before 2020 from the perspective of applications. Vandenhende et al. (2021) provides a comprehensive review of deep MTL in dense prediction tasks, which generate pixel-level predictions such as in semantic segmentation and monocular depth estimation. Y. Zhang and Yang (2021) first give a comprehensive overview



**Figure 4.** The comparison of general framework between STL and MTL. (a) In STL, the learning function  $f$  is trained on a single dataset  $(\mathbf{X}, \mathbf{Y})$ , where  $\mathbf{X}$  represents the input data and  $\mathbf{y}$  represents the corresponding labels. The function  $f$  is parametrized by  $\mathbf{W}$ , and is trained to minimize a predefined loss function  $\mathcal{L}(\tilde{\mathbf{Y}}, \mathbf{Y})$ , where  $\tilde{\mathbf{Y}}$  is the prediction value. Once  $f$  is trained, it can be used to generalize to unseen data. (b) In MTL, the learning pipeline is similar to STL, but instead of training on a single dataset, multiple datasets are combined for different tasks. The multiple tasks are learned jointly by optimizing multiple loss functions  $\mathcal{L}^{(1)}(\tilde{\mathbf{Y}}^{(1)}, \mathbf{Y}^{(1)}), \dots, \mathcal{L}^{(T)}(\tilde{\mathbf{Y}}^{(T)}, \mathbf{Y}^{(T)})$  simultaneously. It should be noted that although multiple tasks are learned jointly, the generalization of each task can still be performed independently.

of MTL models from the taxonomy of feature-based and parameter-based approaches, but with limited inclusion of deep learning methods. Notably, all these surveys overlook the development of MTL in the last three or four years, named the era of large PFM (Bommasani et al., 2021; C. Zhou et al., 2023), exemplified by the GPT-series models (Brown et al., 2020; OpenAI, 2023; Radford et al., 2018, 2019).

*Roadmap.* This survey adopts a well-organized structure, distinguishing it from its predecessors, to demonstrate the evolutionary journey of MTL from traditional methods to deep learning and the innovative paradigm shift introduced by PFM, as shown in Fig. 1. In § 2.1, we provide a comprehensive summary of traditional MTL techniques, including feature selection, feature transformation, decomposition, low-rank factorization, priori sharing, and task clustering. Moving forward, § 2.2 is devoted to exploring the critical dimensions of deep MTL methodologies, encompassing feature fusion, cascading, knowledge distillation, cross-task attention, scalarization, multi-objective optimization (MOO), adversarial training, Mixture-of-Experts (MoE), graph-based methods, and NAS. The recent advancements in PFM are introduced in § 2.3, categorized based on task-generalizable fine-tuning, task promptable engineering, as well as task-agnostic unification. Additionally, we provide a concise overview of the miscellaneous aspects of MTL in § 3. § 4 provides valuable resources and tools to enhance the engagement of researchers and practitioners with MTL. Our discussions and future directions are presented in § 5, followed by our conclusion in § 6. The goal of this review is threefold: 1) to provide a comprehensive understanding of MTL for newcomers; 2) to function as a toolbox or handbook for engineering practitioners; and 3) to inspire experts by providing insights into the future directions and potentials of MTL.



**Figure 5.** The classification of MTL problems into three different input/output configurations: (a) single-input multi-output (MISO), (b) multi-input single-output (MISO), and (c) multi-input multi-output (MISO).

**Table 2.** Summary of MTL methods discussed in § 2.

| MTL Strategy          | Assumption                   | I/O  |      |      | Data Modality |       |      | Task Type |            |                |                  |
|-----------------------|------------------------------|------|------|------|---------------|-------|------|-----------|------------|----------------|------------------|
|                       |                              | SIMO | MISO | MIMO | Table         | Image | Text | Graph     | Regression | Classification | Dense Prediction |
| Regularization        | Feature Selection            | 1    | ✓    |      |               |       |      |           | ✓          |                | ✗                |
|                       | Decomposition                | 1    | ✓    |      |               |       |      |           | ✓          | ✓              |                  |
|                       | Low-Rank Factorization       | 1    | ✓    |      | ✓             | ✓     |      |           | ✓          | ✓              |                  |
| Relationship Learning | Priori Sharing               | 1    | ✓    | ✓    | ✓             | ✓     | ✓    |           | ✓          | ✓              |                  |
|                       | Task Clustering/Grouping     | 1    | ✓    |      | ✓             | ✓     |      |           | ✓          |                |                  |
|                       | Group-Based Learning         | 1    | ✓    | ✗    | ✗             | ✗     | ✓    |           | ✓          | ✓              |                  |
|                       | Mixture-of-Experts           | 1    | ✓    |      | ✓             | ✓     | ✓    |           | ✓          | ✓              |                  |
| Feature Propagation   | Feature Fusion               | 2    | ✓    | ✗    | ✗             | ✗     | ✓    | ✓         | ✓          | ✓              | ✓                |
|                       | Cascading                    | 2    | ✓    | ✓    | ✗             | ✗     | ✓    | ✓         | ✓          | ✓              | ✓                |
|                       | Knowledge Distillation       | 2    | ✓    | ✗    | ✗             | ✗     | ✓    | ✓         | ✓          | ✓              | ✓                |
| Optimization          | Cross-Task Attention         | 2    | ✓    | ✗    | ✗             | ✗     | ✓    | ✓         | ✓          | ✓              | ✓                |
|                       | Scalarization                | 3    | ✓    |      | ✓             | ✓     | ✓    | ✓         | ✓          | ✓              | ✓                |
|                       | Multi-Objective Optimization | 3    | ✓    |      | ✓             | ✓     | ✓    | ✓         | ✓          | ✓              | ✓                |
|                       | Adversarial Training         | 3    | ✓    | ✓    |               | ✓     | ✓    | ✓         | ✓          | ✓              | ✓                |
| Pre-training          | Neural Architecture Search   | 1    | ✓    |      |               |       |      |           | ✓          | ✓              | ✓                |
|                       | Downstream Fine-tuning       | 1    | ✓    | ✗    | ✗             | ✓     | ✓    | ✓         | ✓          | ✓              | ✓                |
|                       | Task Prompting               | 1    | ✓    | ✓    | ✓             | ✓     | ✓    | ✓         | ✓          | ✓              | ✓                |
|                       | Multi-Modal Unification      | 1    | ✓    | ✓    | ✓             | ✓     | ✓    | ✓         | ✓          | ✓              | ✓                |

✓ indicates common practice in the research community. ✗ indicates not applicable due to technical constraints.

## 2. MTL MODELS

*Formalization.* In machine learning, no matter the problem (discriminative, generative, adversarial, etc.), we hope to learn a predictive model by minimizing the regularized empirical loss as follows.

$$(2.1) \quad \min_{\mathbf{W}} \mathcal{L}(f_{\mathbf{W}}(\mathbf{X}), \mathbf{Y}) + \lambda \Omega(\mathbf{W}),$$

where  $(\mathbf{X}, \mathbf{Y})$  is data pairs sampled from a single task, and  $\mathbf{W}$  includes weights of learning model  $f(\cdot)$ . In general,  $\mathcal{L}$  measures the distance between the predictions and ground-truth, and  $\Omega$  adds constraints to the learning model, e.g., sparsity. The trade-off parameter  $\lambda$  controls the balance between the loss and penalty. Fig. 4a shows the detailed framework of STL. In comparison, as shown in Fig. 4b, the optimization in MTL is conducted on the multiple loss functions to achieve joint learning, and each task can maintain a specific loss function. Accordingly, MTL considers the problem in the following way.

$$(2.2) \quad \min_{\{\mathbf{W}^{(t)}\}_{t=1}^T} \sum_{t=1}^T \mathcal{L}^{(t)}(f_{\mathbf{W}^{(t)}}(\mathbf{X}^{(t)}), \mathbf{Y}^{(t)}) + \lambda \Omega(\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(T)}),$$

where  $T$  denotes the number of tasks, and  $f(\cdot)$  is the MTL model to be learned. In MTL,  $f(\cdot)$  always encodes both task-specific and -shared representations, and  $\Omega(\cdot)$  builds task relatedness and reciprocity; both contribute to the effectiveness and efficiency of MTL.

*I/O Configurations.* To accommodate data in Eq. (2.2), it is necessary to consider various input/output (I/O) configurations that may impose constraints on the MTL modeling process. For instance, tasks such as semantic segmentation and depth estimation can utilize the same input images, and the applications are always developed using datasets where each image is attached with dense prediction labels for both segmentation and depth. On the other hand, when dealing with a digital

recognition problem involving multiple domains (e.g., handwritten digits and license plate digits), different inputs are mapped to the same output space. We refer the former as a single-input multi-output (SIMO) configuration and the latter as a multi-input single-output (MISO) configuration. In MTL, the most prevalent scenarios reside in multi-input multi-output (MIMO) configuration where each task maintains its own set of samples and the labels are omnivorous, e.g., autonomous driving that involves pedestrian detection and traffic sign recognition. Let us denote the data input space and its corresponding label space for the  $t$ -th task ( $t = 1, \dots, T$ ) by  $\mathcal{X}^{(t)}$  and  $\mathcal{Y}^{(t)}$ , respectively. We classify the MTL problems into three cases: SIMO, MISO, and MIMO. Fig. 5 shows the illustration of these three configurations. It is worth noting that the I/O configurations do not significantly impact the taxonomy of methods in MTL. As indicated in Table 2, there are numerous shared practices of applying different methods to these I/O configurations, as well as various data modalities and task types.

*Taxonomy.* MTL has seen significant advancement prior to the deep learning era (Ando et al., 2005; Bakker & Heskes, 2003; R. Caruana, 1997; Caruna, 1993; Obozinski et al., 2006; J. Zhang, 2006). Initially, there was a strong focus on weight/parameter regularization, including sparse learning for cross-task feature selection, low-rank learning to uncover underlying factors, and decomposition methods to capture informative components. These approaches, while innovative in integrating intuitive variations from existing methods (e.g., the  $\ell_{2,1}$  regularizer derived from the classic  $\ell_1$  regularizer), still face limitations in practical applications due to the idealistic assumptions and a lack of consideration for task relationships. The emergence of methods like task clustering, prior sharing, graph-based learning, and MoE marked a shift towards more effective task relationship modeling. With the transition to the deep learning era, the abundance of features learned from architectures like convolutional neural networks (CNNs) (Fukushima, 1980; LeCun et al., 1998), recurrent neural networks (RNNs) (Hochreiter & Schmidhuber, 1997; Werbos, 1988) and Transformers (Dosovitskiy et al., 2020; Vaswani et al., 2017) spurred the exploration of feature propagation methods, such as feature fusion, cascading, knowledge distillation, and cross-task attention, all crucial for leveraging multi-source features. Alternatively, optimization-based methods, including scalarization, MOO, adversarial training and NAS, focused on gradients to harmonize optimization directions across tasks. These methods, while not restricted by I/O configurations, are constrained on the pre-defined number of tasks and the use of heterogeneous architectures. Pre-training techniques, which leverages TL, marks a significant advancement towards unified and versatile multitasking, breaking limitations related to data modalities, dimensions, task numbers, model architectures, etc. The only cost is the large computation resources to train a really large model that can accommodate multi-task distributions. The MTL models are accordingly organized into five categories: regularization, relationship learning, feature propagation, optimization, and pre-training. Each contains a series of topics arranged chronologically in § 2.1, § 2.2, and § 2.3. All these topics are derived from three assumptions (but have been widely confirmed through empirical evidences) below:

**Assumption 1** (Parameter Relatedness). *Under the same hypothesis space, models learned to perform related tasks can exhibit similarities.*

**Assumption 2** (Feature Richness). *Given the same level of experience, expanding the number of tasks to be learned can enhance the richness of features.*

**Assumption 3** (Optimization Consistency). *Learning multiple related tasks jointly in a single model can ensure consistency in optimization directions for each task.*

We acknowledge that the presented taxonomy is not exhaustive, and certain methods may be classified differently when viewed from a different perspective. For example, Task Tree (TAT) (Han

**Table 3.** Summary of basic notations used in this paper.

| Notation   | Description  |
|--|--|
| $n, N \in \mathbb{R}$                                      | Scalars are denoted by plain lowercase or uppercase letters.   |
| #object  | The number of object, e.g. #task denoting the number of task.  |
| $\mathbf{x}$ or $\tilde{\mathbf{x}} \in \mathbb{R}^N$      | A vector $\mathbf{x}$ with $N$ entries, denoted by bold lowercase letters.   |
| $\mathbf{X} \in \mathbb{R}^{M \times N}$                   | A matrix $\mathbf{X}$ with size $M \times N$ , denoted by bold uppercase letters.  |
| $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ | A tensor $\mathcal{X}$ with size $\mathbb{R}^{I_1 \times \dots \times I_N}$ , denoted by bold calligraphic letters.  |
| $\{\star^{(i)}\}_{i=1}^N$                                  | A set contains $\star^{(1)}, \dots, \star^{(N)}$ , where $\star$ could be anything, e.g., scalar, vector, data pair, learner, etc.   |
| $x_n \in \mathbb{R}$                                       | The $n$ -th entry for vector $\mathbf{x} \in \mathbb{R}^N, n \in \{1, 2, \dots, N\}$ .   |
| $x_{m,n}$ or $[\mathbf{X}]_{m,n} \in \mathbb{R}$           | The $(m, n)$ -th entry of matrix $\mathbf{X} \in \mathbb{R}^{M \times N}, m \in \{1, 2, \dots, M\}, n \in \{1, 2, \dots, N\}$ .  |
| $\mathbf{X} \odot \mathbf{Y} \in \mathbb{R}^{M \times N}$  | Element-wise product of $\mathbf{X} \in \mathbb{R}^{M \times N}$ and $\mathbf{Y} \in \mathbb{R}^{M \times N}$ , which means the $(m, n)$ -th entry of $\mathbf{X} \odot \mathbf{Y}$ is $x_{m,n} y_{m,n}$ . |
| $\mathbf{x}^n \in \mathbb{R}^M$                            | The $n$ -th column vector of matrix $\mathbf{X} \in \mathbb{R}^{M \times N}, n \in \{1, 2, \dots, N\}$ .   |
| $\mathbf{x}_m \in \mathbb{R}^N$                            | The $m$ -th row vector of matrix $\mathbf{X} \in \mathbb{R}^{M \times N}, m \in \{1, 2, \dots, M\}$ .  |
| $\mathbf{I}_{N \times N} \in \mathbb{R}^{N \times N}$      | The identity matrix of size $N \times N$ , which has ones on the diagonal and zeros elsewhere.   |
| $\text{tr}(\mathbf{X}) \in \mathbb{R}$                     | The trace of a matrix $\mathbf{X} \in \mathbb{R}^{N \times N}$ , defined as the sum of its $N$ components on the diagonal.   |
| $\text{col}(\mathbf{X}) \subseteq \mathbb{R}^M$            | The column space of a matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$ , which consists of all linear combinations of its column vectors.   |
| $\text{rank}(\mathbf{X}) \in \mathbb{R}$                   | The rank of matrix $\mathbf{X}$ , defined as the maximum number of linearly independent column (or row) vectors of $\mathbf{X}$ .  |
| $\text{vec}(\mathbf{X}) \in \mathbb{R}^{MN}$               | The vectorization of the matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$ in the row-by-row stacking way.   |
| $D^* \in \mathbb{R}^{N \times M}$                          | The pseudoinverse of a matrix $D \in \mathbb{R}^{M \times N}$ .  |
| $\mathcal{O}^N \subset \mathbb{R}^{N \times N}$            | The set of $N \times N$ orthogonal matrices.   |
| $\mathbf{X} \in \mathcal{O}^N$                             | The column vectors $\mathbf{x}^1, \dots, \mathbf{x}^N$ of matrix $\mathbf{X}$ are orthogonal.  |
| $\mathcal{S}^N \subset \mathbb{R}^{N \times N}$            | The set of $N \times N$ real symmetric matrices.   |
| $\mathcal{S}_+^N \subset \mathcal{S}^N$                    | The subset of $\mathcal{S}^N$ that contains positive semidefinit matrices.   |
| $\ \mathbf{w}\ _1$   | The $\ell_1$ norm of a vector, calculated as the sum of the absolute vector values.  |
| $\ \mathbf{w}\ _2$   | The $\ell_2$ norm of a vector, calculated as the square root of the sum of the squared vector values.  |
| $\ \mathbf{w}\ _\infty$                                    | The $\ell_\infty$ norm of a vector, calculated as the maximum of the absolute vector values.   |
| $\ \mathbf{W}\ _0$   | The $\ell_0$ norm, i.e., cardinality of a matrix, defined as the number of nonzero components.   |
| $\ \mathbf{W}\ _1$   | The $\ell_1$ norm of a matrix, calculated as the maximum of the $\ell_1$ norm of the column vectors.   |
| $\ \mathbf{W}\ _2$   | The $\ell_2$ norm of a matrix, calculated as its maximum singular value.   |
| $\ \mathbf{W}\ _F$   | The Frobenius norm of a matrix, calculated as the square root of the sum of the squared matrix values.   |
| $\{\sigma_r(\mathbf{W})\}_{r=1}^R$                         | The set of non-increasing ordered singular values of matrix $\mathbf{W}$ .   |
| $\ \mathbf{W}\ _*$   | The trace norm of a matrix, defined as the sum of its singular values, i.e., $\sum_{r=1}^R \sigma_r(\mathbf{W})$ .   |
| $\ \mathbf{W}\ _\infty$                                    | The $\ell_\infty$ norm of a matrix, calculated as the maximum of the $\ell_1$ norm of the row vectors.   |
| $\ \mathbf{W}\ _{p,q}$                                     | The $\ell_{p,q}$ norm of a matrix, defined as the $q$ -norm of the vector whose components are $p$ -norm of $\mathbf{W}$ 's row vectors.   |
| $\ \mathbf{W}\ _{1,1}$                                     | The $\ell_{1,1}$ norm of a matrix, defined as the sum of the absolute matrix components.   |
| $\ \mathbf{W}\ _{1,2}$                                     | The $\ell_{1,2}$ norm of a matrix, calculated as the $\ell_2$ norm of the vector whose components are $\ell_1$ norm of the row vectors.  |
| $\ \mathbf{W}\ _{2,1}$                                     | The $\ell_{2,1}$ norm of a matrix, calculated as the sum of the $\ell_2$ norm of the row vectors.  |

& Zhang, 2015), a clustering MTL method, establishes task hierarchy by decomposing the parameter matrix into different component matrices for each tree layer; we discuss it within the context of clustering MTL. We also acknowledge that some methods that may be of interest to readers may not be included in this survey due to similarities or oversight. We welcome paper recommendations and will update the survey on our project page accordingly.<sup>2</sup>. In Table 2, we summarize their assumptions, common practice, and technical constraints of these topics in terms of I/O configuration, data modality, and task type.

## 2.1. Traditional Era: Provable but Restrictive.

To establish a unified formulation, we start the review of traditional methods by defining a common framework. The notations for subsequent discussions are summarized in Table 3. Building upon this, we initiate our discussion with multiple standard regression models for each task as a paradigm. The weights of these homogeneous models can be arranged into one weight matrix, catalyzing a series of MTL studies through matrix regularization techniques in the traditional era. We denote by  $\{(\mathbf{X}^{(t)}, \mathbf{y}^{(t)})\}_{t=1}^T$  our dataset across  $T$  tasks. For each task indexed by  $t = 1, 2, \dots, T$ , we are given  $N_t$  samples with  $D$  features, i.e.,  $\mathbf{X}^{(t)} \in \mathbb{R}^{N_t \times D}$ , and the corresponding response values  $\mathbf{y}^{(t)} \in \mathbb{R}^{N_t}$ .

The single-task setting of these multiple linear regression problems is as follows.

$$(2.3) \quad \mathbf{y}^{(t)} = \mathbf{X}^{(t)} \mathbf{w}^{(t)} + \epsilon^{(t)}, t = 1, \dots, T,$$

<sup>2</sup><https://github.com/junfish/Awesome-Multitask-Learning>

where  $\mathbf{w}^{(t)} \in \mathbb{R}^D$  for any  $t \in \{1, \dots, T\}$ ,  $\epsilon^{(t)} \sim \mathcal{N}(0, \sigma_t^2 \mathbf{I})$  is the error term independent of  $\mathbf{X}^{(t)}$ , and  $\sigma_t$  is determined by the system state for  $t$ -th task. Each model is separately learned from independent samples  $\{(\mathbf{x}_1^{(t)\top}, y_1^{(t)}), \dots, (\mathbf{x}_{N_t}^{(t)\top}, y_{N_t}^{(t)})\}$ .

A key simplification of the above linear regressions is that all tasks maintain the same feature size  $D$ , thus leading to a natural idea of stacking weight vectors for these tasks:  $\mathbf{W} = [\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(T)}] \in \mathbb{R}^{D \times T}$ , where the matrix-based regularizers come into play. To estimate as  $\mathbf{W}$ , the MTL method minimizes the objective function:

$$(2.4) \quad \min_{\mathbf{W}} \sum_{t=1}^T \frac{1}{n_t} \mathcal{L}^{(t)}(\mathbf{X}^{(t)} \mathbf{w}^t, \mathbf{y}^{(t)}) + \lambda \Omega(\mathbf{W}),$$

where we consider the weights of multiple models, i.e.,  $\mathbf{W}$ , as a union, and denote by  $\mathbf{w}^t$  the  $t$ -th column of  $\mathbf{W}$ . Normally, an identical loss function, e.g., mean squared error (MSE), is always applied to  $\{\mathcal{L}^{(t)}\}_{t=1}^T$ , which originates from the *i.i.d.* assumption of  $\{\epsilon^{(t)}\}_{t=1}^T$ . To capture task relatedness from the Assumption 1 that multiple models are similar to each other,  $\Omega$  is designed to take various regularization forms in traditional MTL.

**2.1.1. Feature Selection.** The high-dimensional scaling (Negahban & Wainwright, 2008) where the number of model weights is much larger than that of the observations/features, i.e.,  $D \gg N$ , arises in many real-world problems, leading it costly and arduous to seek effective predictor variables. Sparse learning with an  $\ell_1$  regularizer that aims to identify a structure characterized by a reduced number of non-zero elements. This parsimonious solution ensures the retention and selection of the most effective and efficient subset of features tailored to the target task (Tibshirani, 1996). In MTL, Assumption 1 underpins the development of all sparse learning models. Under the settings of sparse learning, this assumption posits that *similar sparsity patterns in model parameters suggest the relatedness between tasks*. As a result, sparsity patterns subtly represent task relatedness, underscoring a subset of common features derived from these limited samples. More benefits and efficacy of employing sparsity in MTL have been thoroughly assessed and discussed in Lounici et al. (2009). In this section, our discussion of feature selection in MTL encompasses both the block-wise ( $\ell_{2,1}$ ) and element-wise ( $\ell_{1,1}$ ) approaches. Each approach maintains both shared and task-specific features, optimizing performance across all tasks. In the block-wise approach, tasks can differentiate themselves from others' priorities by attributing distinct weights to the commonly selected features. Conversely, the element-wise approach allows tasks to highlight their distinct preferences on predictors by opting for specific features in addition to the shared ones.

**Block-Wise Sparsity.** Multi-Task Feature Selection (Obozinski et al., 2006) is the first method to address the problem of joint feature selection across a group of related tasks. This method extends the  $\ell_1$  regularization for STL to the  $\ell_{2,1}$  regularization for MTL. The assumption for  $\ell_{2,1}$  regularization scheme is that multiple related tasks have a similar preference for a few common features, which encourages a solution to share the sparsity pattern. Therefore,  $\ell_{2,1}$  imposes a sparse penalty on the  $\ell_2$  norms of the  $T$ -dimensional weight vectors associated with each feature across tasks (i.e., row vectors of the weight matrix  $\mathbf{W}^{D \times T}$ ). This is formulated as follows:

$$(2.5) \quad \min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \mathbf{w}^t - \mathbf{y}^{(t)}\|_2^2 + \lambda \sum_{d=1}^D \|\mathbf{w}_d\|_2,$$

which selects features globally via encouraging several feature-wise weight vectors  $\mathbf{w}_d$  across all tasks to be  $\mathbf{0}$ . The  $\ell_2$  norm imposed on feature-wise weight vectors (i.e.,  $\mathbf{w}_d$ ) before  $\ell_1$  norm here is a magnitude measurement, which could be substituted by any other  $\ell_p$  ( $p \geq 1$ ) norm (Obozinski et al., 2006). This penalty term can be seen as a generalization of  $\ell_1$  regularization when task

number  $T = 1$ . To solve the problem Eq. (2.5), Obozinski et al. (2006) offers a block-coordinate descent optimization method to update the block of weight vector associated with each feature. J. Liu et al. (2012) proposes an accelerated algorithm by reformulating it as two equivalent smooth convex optimization problems.

Multi-Task Lasso (J. Zhang, 2006) extends the efficient  $\ell_{p,1}$  regularizers via imposing  $\ell_\infty$  norm to each feature-wise weight vector  $\mathbf{w}_d$ . Based on the assumption that the number of effective predictor features is much smaller than the total features, Multi-task Lasso learns a sparser structure by

$$(2.6) \quad \min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \mathbf{w}^t - \mathbf{y}^{(t)}\|_2^2 + \lambda \sum_{d=1}^D \|\mathbf{w}_d\|_\infty.$$

The use of  $\ell_\infty$  enforces the procedure to take the maximum value of each feature-wise vector  $\mathbf{w}_d$  across all tasks. This is appropriate if relevant features are not shared by every task, and this situation frequently happens as the number of tasks grows. J. Zhang (2006) proved that this  $\ell_{\infty,1}$  problem can be solved by an efficient convex optimization technique. Furthermore, a full spectrum of  $\ell_{p,1}$  regularization ( $\ell_{1,1}$ , especially) suitable for MTL is investigated and discussed. However, Negahban and Wainwright (2008) proved that the use of  $\ell_{1,\infty}$  can improve learning efficiency only if the overlap of feature entries across tasks is large enough ( $> 2/3$ ), as compared to the situation where each task learns Lasso problem separately.

Temporal Group Lasso (J. Zhou, Yuan, et al., 2011) is an MTL formulation for predicting the disease progression, which considers  $t$  time points of disease progression as related tasks. They first admit the limitation of task independence for the analytical solution  $\mathbf{W} = (\mathbf{X}^\top \mathbf{X} + \lambda_1 \mathbf{I})^{-1} \mathbf{X}^\top \mathbf{Y}$  to the ridge regression problem  $\min_{\mathbf{W}} \|\mathbf{X}\mathbf{W} - \mathbf{Y}\|_F^2 + \lambda_1 \|\mathbf{W}\|_F^2$ , where  $\mathbf{X}$  is identical and  $\mathbf{Y} = [\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(T)}]$  denotes the progression of disease across  $T$  tasks (time points). To capture the temporal smoothness for the adjacent time points, Temporal Group Lasso adds the temporal smoothness term and feature selector term to form the formalization as follows.

$$(2.7) \quad \min_{\mathbf{W}} \frac{1}{2} \|S \odot (\mathbf{X}\mathbf{W} - \mathbf{Y})\|_F^2 + \lambda_1 \sum_{d=1}^D \|\mathbf{w}_d\|_2^2 + \lambda_2 \sum_{t=1}^{T-1} \|\mathbf{w}^t - \mathbf{w}^{t+1}\|_2^2 + \lambda_3 \sum_{d=1}^D \|\mathbf{w}_d\|_2,$$

where  $S \in \mathbb{R}^{N \times T}$  is the indication matrix for the incomplete data, i.e., for any  $n \in \{1, \dots, N\}, t \in \{1, \dots, T\}$ ,  $s_{n,t} = 0$  if the target value of sample  $n$  at the  $t$ -th time point is missing and  $s_{n,t} = 1$  otherwise. It is noted that this problem can be easily solved by accelerated gradient method (AGM) (Y. Nesterov, 2013) using SLEP (J. Liu et al., 2009). However, to avoid the shrinkage of relevant features that would result in sub-optimal performance, J. Zhou, Yuan, et al. (2011) proposed a standard two-stage procedure to relax the  $\ell_1$  regularization.

Adaptive Multi-Task Elastic-Net (X. Chen et al., 2012) aims to address the problem of collinearity existing in the multi-task feature selection method. Inspired by elastic-net (Zou & Hastie, 2005), a natural thought is to add another quadratic penalty  $\sum_{d=1}^D \|\mathbf{w}_d\|_2^2$  to the sparse multi-task constraint  $\sum_{d=1}^D \|\mathbf{w}_d\|_2$ , which forms the corresponding multi-task elastic-net problem as follows.

$$(2.8) \quad \min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \mathbf{w}^t - \mathbf{y}^{(t)}\|_2^2 + \lambda_1 \sum_{d=1}^D \|\mathbf{w}_d\|_2 + \lambda_2 \sum_{d=1}^D \|\mathbf{w}_d\|_2^2,$$

where the traditional  $\ell_{2,1}$  mixed norm shares the same amount of regularization across all features, i.e.,  $\lambda_1$ . As discussed below in the adaptive sparse multi-task lasso (S. Lee et al., 2010), it is promising to learn different regularization weights  $\{\mathbf{w}_d\}_{d=1}^D$  for each feature. However, unlike the application of eQTL detection (S. Lee et al., 2010), where features on single nucleotide polymorphisms (SNPs) make it easier to incorporate prior knowledge for each feature, see Eq. (2.10), the priors scaling the importance of adaptive weights for each feature are always unavailable in many

real-world problems. X. Chen et al. (2012) proposes a three-stage algorithm to estimate the adaptive weights  $\mathbf{w}_d$  via using a data-driven method: (1) estimate the initial regression weights  $\{\hat{\mathbf{w}}_d\}_{d=1}^D$  with uniform weight for each feature, i.e., solve the problem (2.8); (2) construct adaptive scaling weights  $\{\hat{\lambda}_d\}_{d=1}^D, \hat{\lambda}_d = (\|\hat{\mathbf{w}}_d\|_2)^{-\gamma}$  according to the weights estimated in the previous stage, where  $\gamma$  is a fixed constant; (3) compute the final estimated parameters via the multi-task elastic-net with the adaptive scaling weights, i.e.,  $\hat{\mathbf{W}} = \min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \mathbf{w}^t - \mathbf{y}^{(t)}\|_2^2 + \lambda_1 \sum_{d=1}^D \hat{\lambda}_d \|\mathbf{w}_d\|_2 + \lambda_2 \sum_{d=1}^D \|\mathbf{w}_d\|_2^2$ .

Element-Wise Sparsity. Sparse Multi-Task Lasso (S. Lee et al., 2010) allows feature-specific penalty magnitude by incorporating a set of priors with fixed scaling parameters. This method also generalizes the sparse group Lasso penalty (Simon et al., 2013) by suing both the  $\ell_{2,1}$  and  $\ell_{1,1}$  norms to perform joint block-wise and element-wise feature selection. Specifically, sparse multi-task Lasso proposes

$$(2.9) \quad \min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \mathbf{w}^t - \mathbf{y}^{(t)}\|_2^2 + \lambda_1 \sum_{d=1}^D \rho_d \|\mathbf{w}_d\|_2 + \lambda_2 \sum_{d=1}^D \theta_d \|\mathbf{w}_d\|_1,$$

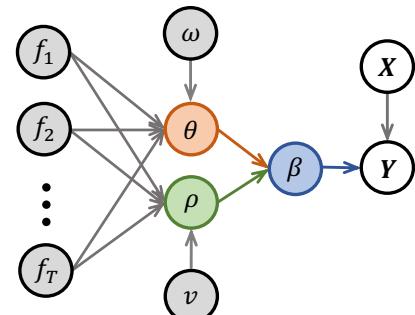
where  $\boldsymbol{\rho} = [\rho_1, \dots, \rho_D]^\top$  and  $\boldsymbol{\theta} = [\theta_1, \dots, \theta_D]^\top$  are the scaling weights for the  $\ell_{2,1}$  and  $\ell_{1,1}$  regularizers, respectively. There exist two advantages of this method: (1) Unlike previous work by Obozinski et al. (2006) and J. Zhang (2006), which considers  $\ell_{p,1}$  ( $p > 1$ ) norm that learns block-wise sparsity well but overlooks element-wise sparsity within each feature group, sparse multi-task Lasso balances the  $\ell_{2,1}$  and  $\ell_{1,1}$  regularizers via  $\lambda_1$  and  $\lambda_2$  to achieve both simultaneously. (2) Unlike Obozinski et al. (2006) and J. Zhang (2006), which treats every feature-wise weight vectors ( $\{\mathbf{w}_d\}_{d=1}^D$ ) equally, i.e.,  $\rho_d = \theta_d = 1, d \in \{1, \dots, D\}$ , the two scaling vectors in S. Lee et al. (2010) can be automatically learned from data. Furthermore, Maurer et al. (2013) uses the  $\ell_1$  regularizer on data preprocessed by a linear mapping function and provides bounds on the generalization error for both MTL and TL settings.

Adaptive Sparse Multi-Task Lasso (S. Lee et al., 2010) is induced as a super-problem from above. This method adaptively incorporates prior knowledge on SNPs (Brookes, 1999) to learn two scaling vectors  $\boldsymbol{\rho}$  and  $\boldsymbol{\theta}$ , which are defined as the mixtures of features on the  $j$ -th SNP

$$(2.10) \quad \begin{aligned} \rho_d &= \sum_i v_i f_i^d \text{ and } \theta_d = \sum_i \omega_i f_i^d, d = 1, \dots, D, \\ \text{s.t. } \sum_i v_i &= \sum_i \omega_i = 1, \end{aligned}$$

where  $f_i^d$  is the  $i$ -th feature of the  $d$ -th SNP. Here, we clarify that the component  $x_{n_t, d} \in \{0, 1, 2\}$  of  $\mathbf{X}^{(t)} \in \mathbb{R}^{N_t \times D}$  in Eq. (2.9) denotes the number of minor alleles at the  $d$ -th SNP of the  $n_t$ -th sample. S. Lee et al. (2010) uses a directed graphical model as an elegant Bayesian tool to find the maximum a posteriori (MAP) estimate of all the above learnable weights, shown in Fig. 6. Then the conditional probability of weight matrix  $\mathbf{W}$  given  $\boldsymbol{\rho}$  and  $\boldsymbol{\theta}$  is

$$(2.11) \quad P(\mathbf{W} | \boldsymbol{\rho}, \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\rho}, \boldsymbol{\theta})} \prod_{d=1}^D \prod_{t=1}^T \exp(-\theta_d |w_{d,t}|) \times \prod_{d=1}^D \exp(-\rho_d \|\mathbf{w}_d\|_2),$$



**Figure 6.** The Bayesian graph for adaptive sparse multi-task Lasso model.

where the normalization factor  $Z(\boldsymbol{\rho}, \boldsymbol{\theta})$  is upper-bounded by the inference of high dimensional multivariate Laplace distribution (Gómez et al., 1998). Accordingly, S. Lee et al. (2010) proposes

an alternating minimization approach that iteratively optimizes one of  $(v, \omega)$  and  $\mathbf{W}$  by fixing another until convergence.

Convex Fused Sparse Group Lasso (cFSGL) (J. Zhou et al., 2012) considers a formulation that additionally allows the element-wise feature selection compared to the temporal group Lasso (J. Zhou, Yuan, et al., 2011). cFSGL encourages the sparsity for joint feature selection across tasks and specific feature selection within a task. The formulation can be written as

$$(2.12) \quad \min_{\mathbf{W}} \frac{1}{2} \|S \odot (\mathbf{X}\mathbf{W} - \mathbf{Y})\|_F^2 + \lambda_1 \sum_{d=1}^D \|\mathbf{w}_d\|_1 + \lambda_2 \sum_{t=1}^{T-1} \|\mathbf{w}^t - \mathbf{w}^{t+1}\|_1 + \lambda_3 \sum_{d=1}^D \|\mathbf{w}_d\|_2,$$

where  $\sum_{t=1}^{T-1} \|\mathbf{w}^t - \mathbf{w}^{t+1}\|_1$  is the fused Lasso penalty, and the combination of  $\ell_{1,1}$  and  $\ell_{2,1}$  is well-known as the sparse group Lasso penalty (Simon et al., 2013). Thus, this problem with three non-smooth regularization terms can be solved by AGM via computing the decoupled proximal operator.

Multi-Stage Multi-Task Feature Learning (P. Gong, Ye, & Zhang, 2012) represents a pioneering approach to address the sub-optimal solutions observed in prior convex sparse regularization problems. This sub-optimality can be attributed to the challenges in approximating  $\ell_0$  regularization. In response to this limitation, the method introduces a non-convex formulation utilizing capped  $\ell_{1,1}$  regularization for MTL:

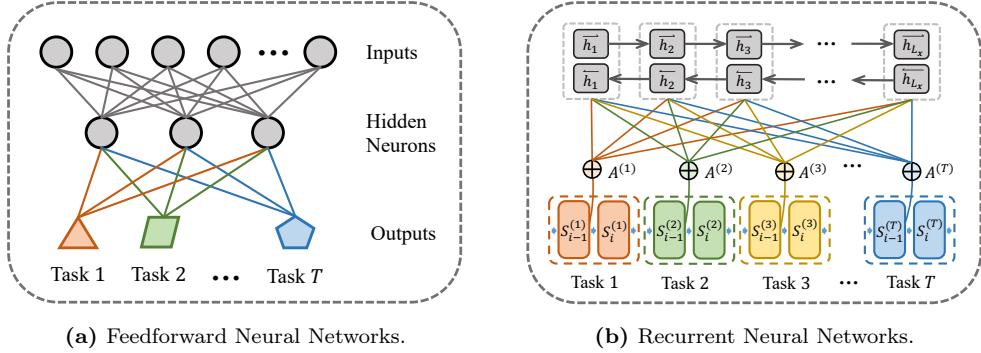
$$(2.13) \quad \min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \mathbf{w}^t - \mathbf{y}^{(t)}\|_2^2 + \lambda \sum_{d=1}^D \min\{\|\mathbf{w}_d\|_1, \tau\},$$

where  $\tau$  is a threshold to tailor the  $\ell_1$  norm of weight vectors, i.e.,  $\{\|\mathbf{w}_d\|\}_{d=1}^D$  corresponding to each feature, and the term  $\sum_{d=1}^D \min\{\|\mathbf{w}_d\|_1, \tau\}$  is a natural generalization of capped  $\ell_1$  norm in T. Zhang (2010, 2013). To solve this non-convex problem (2.13), P. Gong, Ye, and Zhang (2012) proposed an efficient algorithm and investigated the estimation error bound of the resulting estimator.

### Remarks

- (i) Feature selection can highlight task relatedness, especially in scenarios with limited data availability ( $\#\text{feature} > \#\text{data}$ ).
- (ii) The  $\ell_1$  regularization easily facilitates feature selection, offering broad generalizability across various parametric models in MTL.
- (iii) In MTL contexts with plenty of training resources, feature selection might compromise performance; however, it enhances interpretability through the selected features.
- (iv) Given limited data, some feature selection techniques become susceptible to minor data variations, potentially affecting the stability of the learning process.

**2.1.2. Feature Transformation.** Different from the sparse learning methods discussed in §2.1.1 that assumes that directly use the observed features, the feature transformation method tends to combine or transform—not simply select—the raw features into new representations, thus allowing for coarse-grained input data. Sparse learning in MTL builds task relatedness into model  $f(\cdot)$  through sharing similar weight structure across multiple tasks, however, feature learning in MTL makes tasks be related to each other via enforcing a common underlying representation (Argyriou et al., 2006). For example, J. Yu et al. (2019) points out that two tasks of aesthetic quality assessment and emotional recognition in digital image analysis share similar feature representations. Another example from R. Caruana (1997) and Caruna (1993), as shown in Fig. 7a, reveals that different tasks can synchronously learn from the same feature encodings in feedforward neural networks (FNNs).



**Figure 7.** Hard-parameter sharing in FNNs and RNNs. (a) The most early version of hard parameters sharing. The connections between inputs and hidden neurons jointly transform features, which are then utilized for Task 1 to Task  $T$ . (b) A modern-day RNN used for multiple-target language translation, which jointly transforms features from shared sequence-based representations.  $(h_1, \dots, h_{L_x})$  represent the sequence of bidirectional recurrent representations, where  $L_x$  is the number of tokens for the source sentence  $\mathbf{x}$ .  $s_i^{(t)}$  is a recurrent neural network hidden state at time  $i$  for the  $t$ -th task, which is estimated based on the combination of  $(h_1, \dots, h_{L_x})$  weighted by  $A^{(t)}$ .

Multi-Task Feature Learning (Argyriou et al., 2006) linearly combines observations/features via introducing a transformation matrix  $\mathbf{U} \in \mathcal{O}^D$ , which can be extended to nonlinear combinations by using kernel methods. As formulated in the following,

$$(2.14) \quad \min_{\mathbf{U}, \mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|(\mathbf{X}^{(t)} \mathbf{U}) \mathbf{w}^t - \mathbf{y}^{(t)}\|_2^2 + \lambda \left( \sum_{d=1}^D \|\mathbf{w}_d\|_2 \right)^2, \quad \text{s.t. } \mathbf{U} \in \mathcal{O}^D,$$

we need to estimate  $\mathbf{U}$  and  $\mathbf{W}$  from the data. The  $\ell_{2,1}$  norm imposed on  $\mathbf{W}$  ensures that the transformed features, i.e.,  $\mathbf{X}^{(t)} \mathbf{U}$ , with a fixed  $\mathbf{U}$ , would be collectively selected across tasks. To learn the transformed features, Argyriou et al. (2006) fixed  $\mathbf{W}$  to minimize the objective function (2.14) over  $\mathbf{U}$  under the orthogonal constraints. Even with this two-step iterated optimization algorithm to solve for  $\mathbf{W}$  and  $\mathbf{U}$ , solving the problem (2.14) is still a non-convex problem. Accordingly, it is transformed into an equivalent convex problem<sup>3</sup> as follows.

$$(2.15) \quad \begin{aligned} & \min_{\mathbf{V}, \mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \mathbf{w}^t - \mathbf{y}^{(t)}\|_2^2 + \lambda \sum_{t=1}^T \mathbf{w}^{t\top} \mathbf{V}^+ \mathbf{w}^t, \\ & \text{s.t. } \mathbf{V} \in \mathcal{S}_+^D, \text{tr}(\mathbf{V}) \leq 1, \text{col}(\mathbf{W}) \subseteq \text{col}(\mathbf{V}). \end{aligned}$$

D. Dong et al. (2015) first extends the neural machine translation to an MTL framework which shares a bidirectional recurrent representation with forward and backward sequence information, as shown in Fig. 7b. Suppose we have  $T$  different language pairs  $\{(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})\}_{t=1}^T$ , for instance, from English to many other languages like French, Spanish, Dutch, and Portuguese, the probability of generating each translated word at time step  $i$  is

$$(2.16) \quad p(y_i^{(t)} | y_1^{(t)}, \dots, y_{i-1}^{(t)}, \mathbf{x}^{(t)}) = f(y_{i-1}^{(t)}, s_i^{(t)}, c_i^{(t)}), \quad t = 1, \dots, T,$$

where  $f$  is parameterized by a FNN,  $s_i^{(t)}$  is the hidden state of a recurrent neural network at time step  $i$ , and  $c_i^{(t)}$  is a context vector calculated from a sequence of annotations  $(h_1, \dots, h_{L_x})$ , which is mapped from the original sentence  $\mathbf{x}$  by an encoder. More details of bidirectional sequence learning can refer to D. Dong et al. (2015). After that, all annotations  $h_j$  ( $j = 1, \dots, L_x$ ) are collectively

<sup>3</sup>It is also known as convex multi-task feature learning (Argyriou et al., 2006, 2008), which is mentioned in Argyriou et al. (2006) and further discussed in Argyriou et al. (2008) with the learning of non-linear features using kernel methods.

transformed by soft alignment parameters  $A^{(t)}$  ( $t = 1, \dots, T$ ) for each encoder-decoder to achieve cross-task communications.

### Remarks

- (i) Feature transformation can facilitate multiple tasks to share the same underlying representations.
- (ii) The features from different tasks can interact with each other, providing mutual benefits across all tasks.

**2.1.3. Decomposition.** Task-relatedness can be learned based on assumptions that similar tasks possess the same non-zeros and these tasks can learn richer representations via transformation or low-rank regularizer. The decomposition methods that we will discuss in this section can learn multiple task-relatedness (i.e., sparsity and low-rankness) based on the decomposition of model weights into a summation or product of several different components. These components can capture not only the shared information but also the task-specific information that is beneficial to each task. The flexibility of decomposition techniques reveals more insights into the nature of multitasking. It allows for the exploration of various combinations of regularizers, catering to different types of multitasking, including the incorporation of irrelevant (outlier) tasks. Nevertheless, decomposition methods come with a limitation. The regularization applied to complex components may result in a non-smooth optimization problem with an excessive number of variables. This complexity can pose challenges in efficiently solving the devised decomposition problem. In the MTL setting, the general formalization of decomposition problems can be written as

$$(2.17) \quad \begin{aligned} & \min_{\mathbf{W}} \sum_{t=1}^T \mathcal{L}^{(t)}(f(\mathbf{X}^{(t)}, \mathbf{w}^t), \mathbf{y}^{(t)}) + \lambda_1 \text{reg}_1(\mathbf{P}) + \lambda_2 \text{reg}_2(\mathbf{Q}), \\ & \text{s.t. } \mathbf{W} = \mathbf{P} + \mathbf{Q} \text{ or } \mathbf{W} = \mathbf{P} \cdot \mathbf{Q}, \end{aligned}$$

where the  $\text{reg}_1$  and  $\text{reg}_2$  are regularizers for the learning of different task-relatedness.

Form “ $\mathbf{P} + \mathbf{Q}$ ”. The Dirty Block-Sparse Model (Jalali et al., 2010) is introduced by recognizing that block-sparsity regularizers ( $\ell_{p,1}$ ) are influenced by the degree of feature overlap among tasks. Acknowledging the prevalence of dirty high-dimensional data<sup>4</sup> in many multi-task scenarios, this model adeptly addresses the challenges posed by explicitly permitting the decomposition of the weight matrix into element-wise sparse and block-sparse components:

$$(2.18) \quad \min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)}(\mathbf{s}^t + \mathbf{b}^t) - \mathbf{y}^{(t)}\|_2^2 + \lambda_1 \sum_{d=1}^D \|\mathbf{s}_d\|_1 + \lambda_2 \sum_{d=1}^D \|\mathbf{b}_d\|_\infty, \quad \text{s.t. } \mathbf{W} = \mathbf{S} + \mathbf{B},$$

where the  $\mathbf{s}^t$  and  $\mathbf{b}^t$  are the  $t$ -th columns of  $\mathbf{S}$  and  $\mathbf{B}$ , respectively. The  $\ell_{1,1}$  norm learns an uneven sparse structure (Obozinski et al., 2006; J. Zhang, 2006) while  $\ell_{\infty,1}$  norm guarantees features that admit block-wise sparsity to be learned collectively across tasks (J. Zhang, 2006). Jalali et al. (2010) proves that Eq. (2.18) can match Lasso ( $\ell_1$ ) for no-sharing STL and  $\ell_{\infty,1}$  for fully-sharing MTL, and it strictly outperforms both methods elsewhere, including the dirty setting.

Robust Multi-Task Feature Learning (rMTFL) (P. Gong, Ye, & Zhang, 2012) can capture the task-shared features among relevant tasks and identify outlier tasks simultaneously. Specifically, the weight matrix for all tasks is first decomposed into two components. And then, P. Gong, Ye, and Zhang (2012) impose the well-known  $\ell_{2,1}$  penalty on the first component and the  $\ell_{1,2}$  penalty

---

<sup>4</sup>It refers to data that are not only high-dimensional (containing a large number of features or attributes) but also contain errors, inaccuracies, or misleading information.

on the second component. Formally, the proposed rMTFL can be formulated as

$$(2.19) \quad \min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \mathbf{w}^t - \mathbf{y}^{(t)}\|_2^2 + \lambda_1 \sum_{d=1}^D \|\mathbf{p}_d\|_2 + \lambda_2 \sqrt{\sum_{d=1}^D \|\mathbf{q}_d\|_1^2}, \quad \text{s.t. } \mathbf{W} = \mathbf{P} + \mathbf{Q},$$

where the penalty on the rows of the weight matrices captures the shared information as the same non-zeros are selected across all tasks. At the same time, the penalty on the columns enforces the weights for the outlier tasks to be zeros. In P. Gong, Ye, and Zhang (2012), a theoretical bound is established to quantify the approximation accuracy of the optimization in relation to the true evaluation. Additionally, error bounds between the estimated weights of rMTFL and the underlying true weights are provided. It is important to note that this method is specifically applicable to MTL settings where some of the tasks are considered outliers.

Robust Multi-Task Learning (RMTL) (J. Chen et al., 2011) addresses real-world applications where certain tasks are irrelevant to other aggregated groups in MTL, impacting the learning performance of different tasks. RMTL is designed to capture task relatedness by learning a low-rank structure while identifying outlier tasks. This approach draws inspiration from previous research on group sparsity (S. Lee et al., 2010; Obozinski et al., 2006). It is formulated as a non-smooth convex optimization problem as

$$(2.20) \quad \min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} (\mathbf{p}^t + \mathbf{q}^t) - \mathbf{y}^{(t)}\|_2^2 + \lambda_1 \|\mathbf{P}\|_* + \lambda_2 \sum_{t=1}^T \|\mathbf{q}_t\|_2, \quad \text{s.t. } \mathbf{W} = \mathbf{P} + \mathbf{Q}.$$

Different from feature selection techniques,  $\ell_{2,1}$  norm here is imposed on the columns of the weight matrix. This penalty aims to learn group sparsity of different tasks across all features. It enforces that the weights associated with outlier tasks are constrained to approach zero, thereby diminishing the negative influence of outlier tasks. The low-rank structure encoded in RMTL encapsulates the positive effectiveness, mitigating the impact of outlier tasks. This differs from Hsu et al. (2010), which focus on learning both low-rank and sparse structures and provide a theoretically existing and unique decomposition. RMTL, on the other hand, simultaneously learns both the low-rank and task-wise sparse structures through an accelerated proximal method (APM) (Nemirovski, 1994; Y. Nesterov, 1998). The performance bound of this integrated approach is also proven.

Sparse and Low-Rank Multi-Task Learning (J. Chen et al., 2012) also decomposes the weight matrix into a low-rank component and a sparse component. Unlike J. Chen et al. (2011) that jointly optimizes both structures in the objective function, J. Chen et al. (2012) uses a trace norm constraint to implicitly encourage the low-rank structure. The formulation is

$$(2.21) \quad \min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \mathbf{w}^t - \mathbf{y}^{(t)}\|_2^2 + \lambda \sum_{d=1}^D \|\mathbf{p}_d\|_1, \quad \text{s.t. } \mathbf{W} = \mathbf{P} + \mathbf{Q}, \|\mathbf{Q}\|_* \leq \tau.$$

It is proved to be the tightest convex surrogate function to the non-convex NP-hard problem with a cardinality regularization term ( $\ell_0$  norm) and a low-rank constraint. A general projected gradient scheme (Boyd et al., 2004) is applied to solve this relaxed convex problem (2.21), which can also be accelerated using Nesterov's method (Y. Nesterov, 1998).

Form “ $\mathbf{P} \cdot \mathbf{Q}$ ”. Alternating Structure Optimization (ASO) (Ando et al., 2005) aims to facilitate structural learning from multiple tasks. By introducing an auxiliary variable  $\mathbf{u}^{(t)}$  for each task  $t$  such that  $\mathbf{u}^{(t)} = \mathbf{w}^{(t)} + \Theta^\top \mathbf{v}^{(t)}$ , the problem is formulated as

$$(2.22) \quad \begin{aligned} & \min_{\{\mathbf{W}, \mathbf{V}\}, \Theta} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \mathbf{u}^{(t)} - \mathbf{y}^{(t)}\|_2^2 + \lambda \sum_{d=1}^D \|\mathbf{w}_d\|_2^2, \\ & \text{s.t. } \Theta \Theta^\top = \mathbf{I} \end{aligned}$$

The solution process for problem (2.22) comprises two steps: fixing  $(\Theta, \mathbf{v})$  and then  $\mathbf{u}$ . The first step involves a convex problem, easily addressed by classic optimization methods such as stochastic gradient descent (SGD). The second step can be tackled using singular value decomposition (SVD) along with a series of linear algebra transformations. However, it is important to note that the non-convex ASO algorithm is not guaranteed to converge to a global optimum and may encounter challenges like getting stuck in local optima.

Convex ASO (cASO) (J. Chen et al., 2009) investigates the use of convex relaxations to improve the convergence properties of the algorithm and can converge to a global optimum. Firstly, an improved ASO (iASO) formulation is proposed as an initial non-convex problem

$$(2.23) \quad \begin{aligned} & \sum_{t=1}^T \frac{1}{N_t} \max\{\mathbf{0}, 1 - (\mathbf{X}^{(t)} \mathbf{u}^{(t)}) \cdot \mathbf{y}^{(t)}\} + \lambda_1 \|\mathbf{u}^{(t)} - \Theta^\top \mathbf{v}^{(t)}\|^2 + \lambda_2 \|\mathbf{u}^{(t)}\|^2, \\ & \text{s.t. } \Theta \Theta^\top = \mathbf{I}, \end{aligned}$$

where the intercept is omitted for simplicity. In Eq. (2.23), the constraint terms effectively manage both task relatedness and model complexity. It is noteworthy that the traditional ASO formulation, represented Eq. (2.22), serves as a special case of iASO, irrespective of the loss function choices.

To address the non-convex iASO problem (2.23), based on the observation that  $\mathbf{u}^{(t)} = \Theta^\top \mathbf{v}^{(t)}$  minimizes the constraint terms, the formulation of the constraint term can be restructured as

$$(2.24) \quad \mathbf{G}(\mathbf{U}, \Theta) = \lambda_1 \eta (1 - \eta) \text{tr}(\mathbf{U}^\top (\eta \mathbf{I} + \Theta^\top \Theta)^{-1} \mathbf{U}),$$

where  $\eta = \lambda_2 / \lambda_1 > 0$  and  $\mathbf{U} = [\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(T)}]$ . Thus, the convex ASO formulation can be written as

$$(2.25) \quad \begin{aligned} & \sum_{t=1}^T \frac{1}{N_t} \max\{\mathbf{0}, 1 - (\mathbf{X}^{(t)} \mathbf{u}^{(t)}) \cdot \mathbf{y}^{(t)}\} + \mathbf{G}(\mathbf{U}, \Theta), \\ & \text{s.t. } \Theta \Theta^\top = \mathbf{I}. \end{aligned}$$

The convex optimization procedures contain the alternating steps of the estimation of  $\mathbf{U}$  with the fixed  $\Theta^\top \Theta$  and the estimation of  $\Theta^\top \Theta$  with a fixed  $\mathbf{U}$ . Via the convergence analysis, it is proved that cASO (2.25) can converge to a global optimum (J. Chen et al., 2009).

Multi-level Lasso (Lozano & Swirszcz, 2012) is a method based on the decomposition of the regression coefficients into a product between one component that is common across all tasks and another component that captures the task-specific features. Specifically, Lozano and Swirszcz (2012) suppose that the “global” sparsity would be controlled by a part of the “main effect” variables. Thus, an alternative decomposition is proposed to satisfy the desired property by rewriting  $\mathbf{w}^t$  as

$$(2.26) \quad \mathbf{w}_d^t = \theta_d \gamma_d^{(t)}, d = 1, \dots, D,$$

where  $\theta_d$  indicates the “effect” from the  $d$ -th feature, and  $\gamma_d^{(t)}$  reflects task specificity. Accordingly, the optimization problem can be written as

$$(2.27) \quad \min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \mathbf{w}^t - \mathbf{y}^{(t)}\|_2^2 + \lambda_1 \sum_{d=1}^D \theta_d + \lambda_2 \sum_{d=1}^D \|\gamma_d\|_1, \quad \text{s.t. } \mathbf{W} = \vec{\theta} \Lambda \Gamma, \vec{\theta} \geq \mathbf{0}.$$

This model accommodates variations in support across multiple tasks while preserving common structures. The optimization process involves iteratively solving for either  $\theta$  or  $\gamma$  while keeping the other fixed, which is proved to be converged in Lozano and Swirszcz (2012). The limitation lies in the alternate optimization procedure of Multi-level Lasso, where learning  $\gamma$  while fixing  $\theta$  reduces to solving a classical Lasso problem. Although easily solvable, obtaining the solution for the global problem can be time-consuming (Friedman et al., 2007).

### Remarks

- (i) Decomposition methods facilitate the learning of additional task relatedness via imposing different regularizations on the weight components from the decomposition.
- (ii) Regularizations applied to different components may introduce new challenges for solving the problem.

**2.1.4. Low-Rank Factorization.** In MTL, as discussed before, information sharing among multiple tasks can be achieved by assuming that all the tasks are impacted by the same small subset of predictors. On the other hand, low-rank structures imposed on the coefficient matrices or tensors can induce a different type of information sharing among tasks, i.e., the tasks are affected by the predictors through a shared small set of latent variables or directions, which are extracted from the original feature space and are the most relevant subspace to the outcomes. Depending on the way of indexing multiple learning tasks, one can choose to organize the coefficient vectors from multiple learning tasks into a matrix of dimension  $D \times T$  or a tensor with a more delicate structure. In general, the multi-dimensional indices of tasks commonly imply that there are multi-layer relationships among multiple tasks, and the tensor form can help keep this inherent structure which allows leveraging information from different dimensions of task similarities.

**Matrix Factorization.** The most commonly seen situation is when we organize the coefficient vectors from multiple tasks into a matrix  $\mathbf{W}$ , and the rank penalized problem can be formulated as

$$(2.28) \quad \min_{\mathbf{W}} \sum_{t=1}^T \mathcal{L}^{(t)}(f(\mathbf{X}^{(t)}, \mathbf{w}^t), \mathbf{y}^{(t)}) + \lambda \operatorname{rank}(\mathbf{W}).$$

However, to minimize the rank of a matrix is NP-hard (Vandenberghe & Boyd, 1996) due to the combinatorial nature of the rank function (Han & Zhang, 2016; Ji & Ye, 2009). An alternative is to substitute the rank penalty with the trace of the rank for the symmetric positive semidefinite matrix (Mesbahi, 1999), but it excludes non-symmetric or even non-square matrices in real-world applications. Fazel et al. (2001) generalized the trace heuristic to any matrix by introducing the trace norm (a.k.a, nuclear norm or Ky-Fun k-norm) (Horn & Johnson, 2012), which is defined as the sum of a matrix's all singular values (See Table 3).

Low Rank Multi-Task Learning (Ji & Ye, 2009) first introduces the trace norm optimization problem into MTL, which yields a low-rank solution that maps to a low-dimensional feature subspace. The problem can be written as

$$(2.29) \quad \min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \mathbf{w}^t - \mathbf{y}^{(t)}\|_2^2 + \lambda \|\mathbf{W}\|_*,$$

where  $\|\cdot\|_*$  denotes the trace norm of the weight matrix  $\mathbf{W}$ . The technical challenge for the problem above is the non-smooth nature of the trace norm, which makes it converge slowly ( $O(\frac{1}{\sqrt{k}})$ ,  $k$  is the iterations). Ji and Ye (2009) developed an accelerated gradient method that boosts the learning process of trace norm minimization from  $O(\frac{1}{\sqrt{k}})$  to  $O(\frac{1}{k})$ , even to  $O(\frac{1}{k^2})$  with the help of Nesterov's method (Y. E. Nesterov, 1983). It is noticed that a dual reformation (Pong et al., 2010) of problem (2.29) can make it more solvable. In fact, both the rank penalty and the trace norm can be written in a more general form  $\sum_{r=1}^{\min(D,T)} \rho(\sigma_r(\mathbf{W}))$  where  $\rho(\cdot)$  is a penalty function and  $\sigma_r(\mathbf{W})$  is the  $r$ -th largest singular value of  $\mathbf{W}$ . When  $\rho(\sigma_r(\mathbf{W})) = I(\sigma_r(\mathbf{W}) \neq 0)$ , where  $I(\cdot)$  is the indicator function, we get the rank penalty which is also the  $\ell_0$  norm of the singular values. When  $\rho(\sigma_r(\mathbf{W})) = \sigma_r(\mathbf{W})$ , we get the nuclear norm penalty, i.e., the  $\ell_1$  norm of the singular values. For  $0 \leq h \leq 1$ , the properties of the  $\ell_h$  norm of the singular values, i.e., the Schatten- $h$  quasi-norm penalty, have been investigated in Rohde and Tsybakov (2011).

Instead of using different power functions of singular values as penalty functions, there are some other variants of the nuclear norm penalty that can lead to more delicate learning of a low-rank matrix.

The rank of a matrix is defined by the count of its non-zero singular values, meaning that a lower rank corresponds to fewer non-zero singular values. Unlike penalizing all singular values, which the trace norm avoids, it is more desirable and reasonable. This is because the trace norm specifically shrinks only small singular values toward zero, contributing to a more focused and effective regularization approach. To leave the larger singular values un-penalized, Reduced Rank Multi-Stage Multi-Task Learning (RAMUSA) (Han & Zhang, 2016) considers the objective function with truncated trace norm (D. Zhang et al., 2012) as

$$(2.30) \quad \min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \mathbf{w}^t - \mathbf{y}^{(t)}\|_2^2 + \lambda \sum_{r=1}^{\min(D,T)} \min\{\sigma_r(\mathbf{W}), \tau\}.$$

The parameter  $\tau$  serves as a threshold of the singular value magnitude, and only those singular values smaller than  $\tau$  will get penalized. When  $\tau \rightarrow \infty$ , problem (2.30) is reduced to the low-rank MTL problem (2.29). To address this non-convex problem, Han and Zhang (2016) introduce a multi-stage algorithm designed to learn a surrogate upper-bound function. Theoretical proofs affirm its capability for shrinkage, making it an effective approach to tackle the non-convex optimization challenge.

An alternative to the truncated trace norm to relieve the shrinkage on large singular values is the adaptive nuclear norm penalization  $\lambda \sum_{r=1}^{\min(D,T)} \alpha_r \sigma_r(\mathbf{W})$  proposed by K. Chen et al. (2013). The weights  $\{\alpha_r\}_{r=1}^R$  are used to adjust for the level of penalization on each singular value, which should be non-negative values and satisfy  $\alpha_1 \leq \dots \leq \alpha_R$ . The explanation is straightforward, i.e., the larger weights on the smaller singular values ensure a greater shrinkage towards 0, while the smaller weights on the larger singular values are helpful in reducing the shrinkage magnitude.

Low-rank methods are useful to achieve dimension reduction by learning a small set of latent variables. However, low-rank methods alone cannot identify which variables are truly predictive of the outcomes. To obtain a more interpretable model, one can assume that not all predictors are affecting the outcomes by adding a sparsity-inducing penalty in addition to a low-rank restriction. In the field of statistics, this line of research has received lots of attention, and variable selection can be achieved by adding a row-wise penalization on the coefficient matrix in a rank-restricted model. For example, L. Chen and Huang (2012) apply a group-lasso type penalty on the rows of the coefficient matrix. Similar work include Bunea et al. (2012) and She (2017). One of the other forms of sparsity structure considered in low-rank models is sparse SVD discussed in K. Chen et al. (2012) and Uematsu et al. (2019). Sparse SVD achieves predictor and response selection simultaneously. With a rank  $r$ , SVD dissects the correlation between responses and predictors, i.e., the coefficient matrix, into  $r$  orthogonal channels. The importance of each channel is measured by a singular value, and within each channel, the weights on predictors (responses) are in the corresponding right (left) singular vectors. The sparse SVD can achieve both SVD layer-specific sparsity pattern, by imposing sparsity on elements of each singular vector to find different subsets of predictors/responses that are making effects in each correlation pathway (K. Chen et al., 2012), and global variable selection, by shrinking all weights related to a certain variable contained in singular vectors to be zeroes (Uematsu et al., 2019).

Tensor Factorization. When we have multiple learning tasks that can be indexed by multi-dimensional indices, instead of stacking all the weight vectors into a matrix of dimension features  $\times$  tasks, keeping the structure of the index of tasks by saving the weight vectors into a tensor leads to MultiLinear

Multi-Task learning (MLMT) (Wimalawarne et al., 2014). MLMT brings us with several advantages compared with the conventional MTL. Firstly, it allows us to keep the inherent structure of the learning tasks so that different dimensions of task similarities can be learned, and the higher-order structures among tasks can be recovered as well. What's more, task imputation (i.e., TL) is made available with MLMT for tasks with no training data (Wimalawarne et al., 2014). The learning problem can be written as

$$(2.31) \quad \min_{\mathbf{W}} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \mathbf{w}^t - \mathbf{y}^{(t)}\|_2^2$$

where  $\mathbf{W} \in \mathbb{R}^{D \times I_2 \times \dots \times I_N}$  is a tensor consisting of learning weights  $\mathbf{w}^t \in \mathbb{R}^D$ , and the total number of tasks  $T = \prod_{j=2}^N I_j$ .

To exploit task similarities at each dimension, similar to low-rank matrix-based MTL, a multi-linear rank restriction can be imposed on the weight tensor. In Romera-Paredes et al. (2013), the authors directly incorporated the rank restriction into the learning task by using a low-rank Tucker decomposition (Kolda & Bader, 2009) of the weight tensor, and the Frobenius norms of Tucker decomposition components are added as regularizations to reduce overfitting. This optimization problem is solved by alternating minimization.

Alternatively, tensor trace norms are commonly used as a convex approximation to rank restrictions. However, not like the matrix rank, since a tensor rank has no unique definition, various trace norms are developed to fulfill different analysis demands for different anticipated information sharing mechanisms among tasks (Y. Zhang, Zhang, & Wang, 2022). With  $R(\mathbf{W})$  denoting a tensor trace norm, the learning task is

$$(2.32) \quad \min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \mathbf{w}^t - \mathbf{y}^{(t)}\|_2^2 + \lambda R(\mathbf{W})$$

where  $\lambda$  is the tuning parameter to control the magnitude of penalization.

In general, in the sense of Tucker decomposition or multi-linear SVD (Kolda & Bader, 2009; Tomioka & Suzuki, 2013), tensor trace norms include two categories: the overlapped tensor trace norms and the latent tensor trace norms. The latent trace norm (Tomioka & Suzuki, 2013; Wimalawarne et al., 2014) can be written as

$$(2.33) \quad \|\mathbf{W}\|_{*,latent} = \inf_{\mathbf{W}^{(1)} + \dots + \mathbf{W}^{(N)} = \mathbf{W}} \sum_{k=1}^N \|\mathbf{W}_{(k)}^{(k)}\|_*$$

where  $\mathbf{W}^{(k)}$  are latent tensors of  $\mathbf{W}$  and  $\mathbf{W}_{(k)}^{(k)}$  denotes a flattened tensor  $\mathbf{W}^{(k)}$  along its  $k$ th axis. Thus, the latent trace norm is the infimum of the summation of the matrix trace norm of flattened latent tensors of  $\mathbf{W}$ . To account for the heterogenous multilinear rank and dimensions, Wimalawarne et al. (2014) propose a scaled latent trace norm by adding a weight  $I_k^{-1/2}$  to each component  $\|\mathbf{W}_{(k)}^{(k)}\|_*$ . It can identify the dimension with the lowest rank  $r_k$  relative to its dimensionality  $I_k$ . The overlapped tensor trace norm (Romera-Paredes et al., 2013) of a tensor is defined as the weighted sum of nuclear norm of its flattened tensors. With different ways of tensor flattening, the overlapped tensor trace norms have different forms, including the Tucker trace norm (Romera-Paredes et al., 2013) that is a convex combination of matrix trace norms of tensor flattening along each axis in the tensor and the Tensor-Train trace norm (Oseledets, 2011) that conducts tensor flattening along successive axes starting from the first axis. Given that the feature representation can be factorized into semantic basis vectors and linear coefficients mapping the basis vector space to the original feature vector space, Y. Yang and Hospedales (2016) introduce the utilization of low-rank tensors in MTL through deep representation learning.

**Table 4.** Summary of regularization technique used in MTL.

| Model Name  | Origin                  | Year  | Type  | Matrix Regularizer   | Vector Regularization  |
|---|-------------------------|-------|---|--|--|
| Regularized MTL                                   | KDD                     | 2004  | Group regularization                              | Frobenius norm   | $\frac{1}{W} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^{(t)}\ _2^2 + \lambda_1 \sum_{t=1}^T \ w^t\ _2^2 + \lambda_2 \sum_{t=1}^T \ w^t\ _2^2$   |
| Learning Multiple Tasks with Kernel Methods       | JMLR                    | 2005  | Prior Sharing                                     | Adaptive penalty   | $\min_{\mathbf{W}, \mathbf{V}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^{(t)}\ _2^2 + \lambda \sum_{t=1}^T w^t v^t w^t v^t, \text{s.t. } \mathbf{V} \in S^D, \mathbf{V} \in S^D$   |
| Alternating structure optimization                | JMLR                    | 2005  | Decomposition                                     | Frobenius norm   | $\min_{(\mathbf{W}, \mathbf{V})} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} (w^t + \Theta^t v^t) - y^t\ _2^2 + \lambda \sum_{t=1}^T \ w^t\ _2^2 + \lambda \sum_{t=1}^T \ v^t\ _2^2$  |
| Multi-task feature selection                      | Tech. Rep. <sup>1</sup> | 2006  | Group-sparse learning                             | $\ell_{2,1}$ norm  | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^{(t)}\ _2^2 + \lambda \sum_{t=1}^T \ w^t\ _2^2$  |
| Multi-task Lasso                                  | Theis <sup>2</sup>      | 2006  | Group-sparse learning                             | $\ell_{0,1}$ norm  | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^{(t)}\ _2^2 + \lambda \sum_{t=1}^T \ w^t\ _{\infty}$   |
| Multi-task feature learning                       | NeurIPS                 | 2006  | Group-sparse learning, feature learning           | $\ell_{2,1}$ norm  | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} (U w^t - y^t)\ _2^2 + \lambda (\sum_{d=1}^D \ w_d^t\ _2)$ , s.t. $\mathbf{U} \in S^D$  |
| Convex multi-task feature learning                | Mach., Lea.             | 2008  | Feature learning                                  | Adaptive penalty   | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \lambda \sum_{t=1}^T \ w^t\ _2^2$ , s.t. $\mathbf{V} \in S_s^D, \text{tr}(\mathbf{V}) \leq 1, \text{col}(\mathbf{W}) \leq \text{col}(\mathbf{V})$  |
| Low rank MTL                                      | ICML                    | 2009  | Low-rank learning                                 | Trace norm   | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \lambda \ W\ _*$   |
| Convex MTL  | ICML                    | 2009  | —   | —  | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} (w^t - y^t)\ _2^2 + \lambda \ W\ _*$   |
| Dirty block-sparse model                          | NeurIPS                 | 2010  | Group-sparse learning, decomposition              | $\ell_{0,1}$ norm + $\ell_{1,1}$ norm                                      | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} (w^t + \theta^t v^t) - y^t\ _2^2 + \lambda_1 \sum_{d=1}^D \ w_d^t\ _2 + \lambda_2 \sum_{d=1}^D \ v_d^t\ _2$  |
| Sparse multi-task Lasso                           | NeurIPS                 | 2010  | Group-sparse learning                             | Weighted $\ell_{2,1}$ norm + weighted $\ell_{1,1}$ norm                    | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \lambda_1 \sum_{d=1}^D \rho_d \ w_d^t\ _2 + \lambda_2 \sum_{d=1}^D \rho_d \ w_d^t\ _1$   |
| Adaptive multi-task Lasso                         | NeurIPS                 | 2010  | Group-sparse learning                             | Weighted $\ell_{2,1}$ norm + weighted $\ell_{1,1}$ norm + adaptive penalty | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \lambda_1 \sum_{d=1}^D \rho_d \ w_d^t\ _2 + \lambda_2 \sum_{d=1}^D \rho_d \ w_d^t\ _1 + \log Z(\rho, \theta)$  |
| Large margin multi-task metric learning           | NeurIPS                 | 2010  | Prior Sharing                                     | Frobenius norm   | $P(\mathbf{W}   \mathbf{P}, \theta) = \frac{1}{Z(\mathbf{W}, \theta)} \prod_{t=1}^T \prod_{j=1}^N \exp(-\theta_d \ w_{d,t}\ _2) \times \prod_{d=1}^D \exp(-\rho_d \ w_d\ _2)$  |
| Hierarchical multitask structured output learning | NeurIPS                 | 2011  | Prior Sharing                                     | Frobenius norm   | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} (w^t - \bar{w})\ _2^2 + \sum_{t=1}^T \left[ \gamma_0 (\ \mathbf{M}_t\ _F^2 + \sum_{(i,j) \in \mathcal{C}_t} d_{ij}^2 (\mathbf{x}_i, \mathbf{x}_j) + \sum_{(i,j) \in \mathcal{S}_t} \xi_{ij}^2) \right]$ , s.t. $\forall t, \forall (i, j, k) \in \mathcal{C}_t, d_{ij}^2 (\mathbf{x}_i, \mathbf{x}_j) - d_{ik}^2 (\mathbf{x}_i, \mathbf{x}_k) \geq 0; \forall (i, j) \in \mathcal{S}_t, \xi_{ij}^2 \geq 0; \mathbf{M}_0, \mathbf{M}_1, \dots, \mathbf{M}_T \geq 0$ . |
| Robust MTL  | KDD                     | 2011  | low-rank learning                                 | Trace norm + $\ell_{2,1}$ norm   | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} (I^t + \delta^t) w^t - y^t\ _2^2 + \lambda_1 \ L\ _2 + \lambda_2 \sum_{t=1}^T \ w^t\ _2$ , s.t. $\mathbf{W} = \mathbf{P} + \mathbf{S}$   |
| Temporal group Lasso                              | KDD                     | 2011  | Group-sparse learning                             | Frobenius norm + $\ell_{2,1}$ norm   | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \lambda_1 \sum_{d=1}^D \ w_d^t\ _2 + \lambda_2 \sum_{t=1}^T \ w^t\ _2 + \lambda_3 \sum_{d=1}^D \ w_d^t\ _1$  |
| Clustered MTL                                     | NeurIPS                 | 2011  | task clustering                                   | Clustering penalty + $\ell_{2,2}$ norm                                     | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \lambda_1 (\text{tr}(\mathbf{W}^\top \mathbf{W}) - \text{tr}(\mathbf{F}^\top \mathbf{W}' \mathbf{W}')) + \lambda_2 \sum_{t=1}^T \ w^t\ _2^2$ , s.t. $\mathbf{F}_{d,j} = 1/\sqrt{n_j}$ if $j \in \mathcal{C}_d$ otherwise $0, t = 1, \dots, T$ , where $n_j$ is the #task in the $j$ -th cluster $\mathcal{C}_j$ .  |
| Sparse and low rank MTL                           | TKDD                    | 2012  | Decomposition, sparse learning, low-rank learning | $\ell_{1,1}$ norm + trace norm   | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \lambda \sum_{d=1}^D \ w_d^t\ _1, \text{s.t. } \mathbf{W} = \mathbf{P} + \mathbf{Q}, \ \mathbf{Q}\ _* \leq \tau$   |
| Convex fused sparse group Lasso                   | KDD                     | 2012  | Group-sparse learning                             | $\ell_{1,1}$ norm + $\ell_{2,1}$ norm                                      | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \lambda_1 \sum_{d=1}^D \ w_d^t\ _1 + \lambda_2 \sum_{t=1}^T \ w^t - w^{t+1}\ _1 + \lambda_3 \sum_{d=1}^D \ w_d^t\ _2$  |
| Adaptive multi-task elastic-net                   | SDM                     | 2012  | Group-sparse learning                             | $\ell_{2,1}$ norm + Frobenius norm   | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \lambda_1 \sum_{d=1}^D \ w_d^t\ _2 + \lambda_2 \sum_{t=1}^T \ w^t\ _2 + \lambda_3 \sum_{d=1}^D \ w_d^t\ _1$  |
| Multi-level Lasso                                 | ICML                    | 2012  | Decomposition, sparse learning                    | $\ell_{1,1}$ norm + adaptive penalty                                       | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \lambda_1 \sum_{d=1}^D \ w_d^t\ _2 + \lambda_2 \sum_{d=1}^D \rho_d \ w_d^t\ _1, \text{s.t. } \mathbf{W} = \theta \mathbf{A} \Gamma, \theta \geq 0$   |
| Robust multi-task feature learning                | KDD                     | 2012  | Decomposition, group-sparse learning              | $\ell_{2,1}$ norm + $\ell_{1,2}$ norm                                      | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} (I^t + \delta^t) w^t - y^t\ _2^2 + \lambda_1 \sum_{d=1}^D \ w_d^t\ _2 + \lambda_2 \sqrt{\rho_{d,t}} \ w_d^t\ _1^2, \text{s.t. } \mathbf{W} = \mathbf{P} + \mathbf{Q}$  |
| Multi-stage multi-task feature learning           | NeurIPS                 | 2012  | Spars learning                                    | Capped $\ell_1$ norm (T. Zhang, 2010)                                      | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \lambda \sum_{d=1}^D \ w_d^t\ _1, \min(\ w_d^t\ _1, \tau)$   |
| Convex formulation for MTL                        | IJCAI                   | 2012a | Prior sharing                                     | Clustering penalty   | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \frac{1}{2} \sum_{t=1}^T \text{tr}(\mathbf{W} \mathbf{W}') + \frac{1}{2} \sum_{t=1}^T \text{tr}(\mathbf{W} \mathbf{P}^{-1} \mathbf{W}')$ s.t. $\mathbf{O} \in S^D, \text{tr} \mathbf{O} = 1$   |
| Multi-linear multi-task learning                  | ICML                    | 2013  | Low-rank learning                                 | Overlapped tensor trace norm   | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \lambda \sum_{d=1}^D \ w_d^t\ _2$ , where $\mathbf{W}_{(k)}$ is the mode- $k$ unfolding of tensor $\mathbf{W} \in \mathbb{R}^{D \times I_2 \times \dots \times I_N}$ .   |
| Regularization approach to learn MTL              | TKDD                    | 2014  | Prior sharing                                     | Clustering penalty + $\ell_{2,2}$ norm                                     | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \lambda \sum_{d=1}^D \ w_d^t\ _2^2 + \lambda \sum_{d=1}^D \ w_d^t\ _2 + \text{dft}(s, \Omega) \cdot \ \mathbf{W}\ _F^2$ , where $\mathbf{dft}(\mathbf{s}, \Omega) \in S^D$ , where $d$ represents the number of features   |
| Multi-linear multi-task learning                  | NeurIPS                 | 2014  | Low-rank learning                                 | Scaled latent tensor trace norm  | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \text{inf}_{\mathbf{W} \in \mathbb{R}^{N \times I_2 \times \dots \times I_N}} \lambda \sum_{d=1}^D \ w_d^t\ _2^{1/2} \ \mathbf{W}_{(d)}\ _F^2$ , where $\mathbf{W} \in \mathbb{R}^{D \times I_2 \times \dots \times I_N}$ is a tensor.   |
| Task Tree model                                   | KDD                     | 2015  | task clustering                                   | $\ell_{2,2}$ norm  | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \sum_{d=1}^D \lambda_d \ w_d^t\ _2^2 + \lambda \sum_{d=1}^D \ w_d^t\ _2^2 + \text{dft}(s, \Omega) \cdot \ \mathbf{W}\ _F^2$ , where $\mathbf{W} \in \mathbb{R}^{D \times I_2 \times \dots \times I_N}$   |
| Reduced rank multi-stage MTL                      | AAAI                    | 2016  | Low-rank learning                                 | Capped trace norm (Q. Sun et al., 2010)                                    | $\min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \ X^{(t)} w^t - y^t\ _2^2 + \sum_{d=1}^D \lambda_d \ w_d^t\ _2^2 + \lambda \sum_{d=1}^D \ w_d^t\ _2^2 + \min_{\tau} \sigma_\tau(\mathbf{W}, \tau)$   |

<sup>1</sup> This work is published in Technical Report, the Department of Statistics, UC Berkeley.  
<sup>2</sup> This work is published in Jian Zhang's Ph.D. Thesis, CMU Technical Report CMU-LTI-06-006, 2006.

Since most of the overlapped tensor trace norms only make use a subset of all possible flattening of a tensor that reflect different beliefs of the information sharing mechanism among tasks, to search for all the low-rank structures in a weight tensor and unify various overlapped tensor trace norms, Y. Zhang, Zhang, and Wang (2022) propose a Generalized Tensor Trace Norm (GTTN) which is the convex sum of matrix trace norms of all possible tensor flattening. The combination weights of matrix trace norms of tensor flattenings are treated as unknown variables in the optimization problem to accommodate different levels of importance of each flattening.

When nonlinear low-rank structures among tasks are expected to achieve better learning performance, Y. Zhang, Zhang, and Wang (2022) propose the nonlinear GTTN that firstly transforms the rows or columns of each flattened tensor nonlinearly via a neural network and then performs GTTN on the transformed parameters to capture the nonlinear low-rank structure among all the tasks. For models that are nonlinear in the data, Signoretto et al. (2013) also provide a kernel-based method for MLMT.

### Remarks

- (i) Low-rank structures can achieve both information sharing among tasks and dimension reduction by enforcing all the tasks being affected by the same small set of latent variables extracted from the original feature space.
- (ii) Sparsity-inducing penalties can be added in addition to the rank restriction to achieve variable selection.
- (iii) Keeping the multi-dimensional indices of multiple tasks by saving the weight vectors into a tensor allows us to keep the inherent structure of the learning tasks so that: a. different dimensions of task similarities can be learned; b. the higher-order structures among tasks can be recovered; c. task imputation is made available for tasks with no training data.

**2.1.5. Prior Sharing.** Multi-task prior sharing focuses on understanding and exploiting the relationships between different tasks to improve learning efficiency and performance. This approach

is predicated on the idea that tasks, especially those that are related, can provide complementary information that enhances learning when approached collectively rather than in isolation. By identifying and leveraging the priori interconnections among tasks, priori sharing aims to achieve better generalization, more robust models, and improved predictions for each task.

The typical formulation of priori sharing in MTL is given in the same form as equation (2.4) This optimization objective function seeks to minimize a cumulative loss function over  $T$  tasks, which is a summation of individual losses for each task's predictions against its true values, adjusted by a global regularization term. The regularization term,  $\lambda\Omega(\mathbf{W})$  is then applied to the combined weight vector  $\mathbf{W}$  which concatenates all task-specific weights  $\mathbf{w}^{(t)}$ , thereby incorporating shared information across tasks into the model. It is designed based on a priori knowledge of task interrelations and enforces certain structure of constraints on  $\mathbf{W}$  to reflect the assumed relationships between tasks within the model. This formulation allows for the integration of similarities and differences across tasks to inform the learning process, aiming to improve the generalization of the model by leveraging shared patterns and task-specific peculiarities. The categorization of multi-task prior sharing can be broadly understood in the following ways:

Task similarity. There is compelling evidence supporting the advantages of learning information from multiple task domains compared to single-task data. In earlier studies, such as Evgeniou and Pontil (2004), and Parameswaran and Weinberger (2010), the formulation proposed by multi-task relationship learning was all generated based on prior assumptions of task relatedness. Specifically, Evgeniou and Pontil (2004), and Parameswaran and Weinberger (2010) assumed that the learning tasks are similar to each other and employed task-coupling parameters to model the target average task. In Regularized MTL (Evgeniou & Pontil, 2004), task-coupling parameters were utilized to model the relationships between tasks and extend existing kernel-based single-task methods like support vector machine (SVM) through a novel kernel function. Their formulation is

$$(2.34) \quad \begin{aligned} & \min_{\mathbf{w}_0, \mathbf{v}_0, \xi_{it}} \left\{ \sum_{t=1}^T \sum_{i=1}^m \xi_{it} + \frac{\lambda_1}{T} \sum_{t=1}^T \|\mathbf{v}_t\|_2^2 + \lambda_2 \|\mathbf{w}_0\|_2^2 \right\}, \\ & \text{s.t. } y_{it}(\mathbf{w}_0 + \mathbf{v}_t) \cdot \mathbf{x}_{it} \geq 1 - \xi_{it}, \quad \xi_{it} \geq 0, \quad \forall i \in \{1, 2, \dots, m\} \text{ and } t \in \{1, 2, \dots, T\} \end{aligned}$$

where  $m$  represents sample size of data points for each task,  $\xi_{it}$  represents the error for each estimation of parameter  $\mathbf{w}_0 + \mathbf{v}_t$  generated from the data distribution. They followed the formulation from Hierarchical Bayes (Allenby & Rossi, 1998; Arora et al., 1998; Heskes, 2000) and described the target  $T$  functions as hyperplanes  $f_t(x) = \mathbf{w}_t \cdot \mathbf{x}$ , where  $\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t$  denotes each corresponding target model. In their approach, the authors assume that when learning from tasks that are similar to each other, the discrepancies between different tasks  $\mathbf{v}_t$  are small, and the task relationships are linked to a common model  $\mathbf{w}_0$ . Additionally, Evgeniou et al. (2005) and Kato et al. (2007) provide prior information on the similarities between pairs of tasks and incorporate regularization terms to adjust the learning of multiple tasks in a manner that aligns the distance between model parameters with the distance between tasks. Furthermore, Görnitz et al. (2011) describes the relationship between tasks using a tree structure, and the model parameters learn the similarity from their parent nodes.

Task correlation. Nevertheless, simply assuming the relationship among tasks without evidence support is somewhat detrimental and may extrapolate the results. By proposing a model that learns task relatedness directly from the data, Bayesian models like Bonilla et al. (2007) defines prior information over all the unobserved functions for each task and adapts the model parameters

regarding the task identities as well as observed information without giving much model assumptions. Particularly, they use multi-task Gaussian Process (GP) prediction techniques to model the correlation among tasks, the formulation is

$$(2.35) \quad \begin{aligned} & \langle f_l(\mathbf{x})f_k(\mathbf{x}^\top) \rangle = K_{lk}^f k^x \langle \mathbf{x}, \mathbf{x}^\top \rangle, y_{il} \sim \mathcal{N}(f_l(x_i), \sigma_l^2), l, k \in \{1, \dots, T\}, i \in \{1, \dots, N\} \\ & \min_{\boldsymbol{\theta}_x} \left( N \log |\langle F^T(\mathbf{K}^x(\boldsymbol{\theta}_x))^{-1} F \rangle| + T \log |\mathbf{K}^x(\boldsymbol{\theta}_x)| \right), \end{aligned}$$

where they approach this problem by placing a GP prior over the latent functions  $\{f_l\}$  to directly induce correlations between tasks,  $\mathbf{K}^f$  denotes the inter-task dependency via a positive semi-definite (PSD) matrix,  $k^x$  denotes the covariance between input data points, and  $\sigma_l^2$  refers to the random noise of the  $l$ -th task,  $\mathbf{F}$  is the vector of function values corresponding to  $\mathbf{Y}$ . Bonilla et al. (2007) introduces a novel approach that employs a common covariance function for input features and a 'free-form' covariance matrix for different tasks, offering significant flexibility in modeling diverse data forms and task relationship. Furthermore, the utilization of this 'free-form' covariance matrix mitigates the need for extensive observed data, enhancing the efficiency of the method. To address the overfitting concern stemming from the point estimation approach in Bonilla et al. (2007), Y. Zhang and Yeung (2010) extended multi-task GP to a weight-space view for the multi-task  $t$  process, incorporating an inverse-Wishart prior to modeling the covariance matrix. This adaptation helps mitigate overfitting and enhances the robustness of the method.

Task covariance. In addition to learning through task correlation and task similarities, Y. Zhang and Yeung (2012a, 2014) introduced the concept of Multi-Task Relationship Learning (MTRL) by utilizing a task covariance matrix to capture task relatedness. Within the regularization framework, they derived a convex formulation for multi-task learning, enabling simultaneous learning of model parameters and task relationship. Their innovation lies in the application of a matrix-variate normal prior on the weight matrix  $\mathbf{W}$ , lending a structured prior, alongside certain likelihood functions, to guide the formulation of an objective function that seeks for a posterior solution maximizing the likelihood. The objective function they employed is

$$(2.36) \quad \begin{aligned} & \min_{\mathbf{W}, \boldsymbol{\Omega}} \mathcal{L}(\mathbf{W}) + \lambda_1 \|\mathbf{W}\|_F^2 + \lambda_2 \text{tr}(\mathbf{W} \boldsymbol{\Omega}^{-1} \mathbf{W}^T) \\ & \text{s.t. } \boldsymbol{\Omega} > 0, \text{tr}(\boldsymbol{\Omega}) \leq 1, \end{aligned}$$

where the optimization target they proposed can be expressed as the minimization of a loss function  $\mathcal{L}(\mathbf{W})$  augmented by a regularization term scaled by  $\lambda_1$  that penalizes the Frobenius norm of  $\mathbf{W}$ , and an additional term scaled by  $\lambda_2$  involving the trace of  $\mathbf{W} \boldsymbol{\Omega}^{-1} \mathbf{W}^T$ , reflecting the matrix-variate normal prior. Here,  $\boldsymbol{\Omega}$  denotes a positive definite matrix capturing task covariance, and its complexity is controlled through constraints ensuring its positive definiteness and bounded trace. This formulation has been established as jointly convex in  $\mathbf{W}, \boldsymbol{\Omega}$ , allowing for simultaneous optimization of model parameters and task covariance matrix.

In essence, their approach extends the principles of single-task learning with regularization while incorporating alternative optimization techniques to achieve a convex objective function. Further developments have extended this framework to enhance multi-task boosting (Y. Zhang & Yeung, 2012b) and multi-label learning (Y. Zhang & Yeung, 2013), illustrating its adaptability and potential for a broad spectrum of applications. The approach also offers an interpretative angle from the viewpoint of reproducing kernel Hilbert spaces for vector-valued functions (Ciliberto et al., 2015; Jawanpuria et al., 2015), showcasing its theoretical elegance and practical utility. Also, in the context of MTL with a considerable number of tasks, it becomes evident that not all tasks are

equally interrelated; many display a tendency toward sparsity in their inter-task relationships. Recognizing that a task may not contribute meaningfully to every other task and that sparse task relationships can mitigate overfitting issues more effectively than dense relationships, there is a growing interest in models that can capture these sparse patterns. Y. Zhang and Yang (2017) pays attention to the elucidation of such sparse task relationships, and the objective function can be written as

$$(2.37) \quad \min_{\mathbf{W}, \Omega \geq 0} \sum_{t=1}^T \frac{1}{N_t} \sum_{j=1}^{N_t} \mathcal{L}(\mathbf{w}_t^\top \phi(x_j), y_j) + \frac{\lambda_1}{2} \text{tr}(\mathbf{W} \Omega^{-1} \mathbf{W}^\top) + \lambda_2 \|\Omega\|_1,$$

where  $\phi(\cdot)$  corresponds to the feature mapping, and the learning task refers to  $f_t(\mathbf{x}) = \mathbf{w}_t^\top \phi(\mathbf{x})$ . By adding an  $l_1$  regularization on the covariance matrix  $\Omega$ , their proposed approach, termed the Sparse covAriance based Multi-Task (SPATS) model, is designed to determine a sparse task covariance structure. This method embraces the  $l_1$  regularization, renowned for promoting sparsity, within a regularization framework tailored for MTL. The convex nature of the SPATS model's objective function facilitates the development of an efficient alternating optimization strategy to find the solution.

### Remarks

- (i) In environments where tasks are interdependent and data is limited or imbalanced, the ability to discern and exploit the latent task interrelations becomes crucial.
- (ii) Overestimating task similarity can lead to negative transfer, where learning one task may adversely affect the performance of another. Task similarities might change dynamically during training, requiring adaptive models that can adjust to these changes.
- (iii) Models that heavily rely on task covariances are at risk of overfitting to the specific relations present in the training data, reducing their generalization capabilities.

**2.1.6. Task Clustering/Grouping.** Task relationships can be elucidated through the clustering or grouping of associated tasks, whereby tasks within the same cluster exhibit greater similarities. Executing clustering algorithms at the task level proves particularly advantageous in scenarios with numerous tasks. Typically, task clustering requires leveraging shared structural information across tasks, such as task similarity or distance. These are termed horizontal methods contrasting with hierarchical methods that harness inherent task structures, such as tree formations, to achieve MTL. Task prior sharing and clustering are closely related as both share the commonness across tasks, but clustered structure is an unknown priori that needs to be learned. For example, the problem defined in Eq. (2.34) could also be equivalent to solving the following optimization problem (See proof in Evgeniou and Pontil (2004, Page 3)):

$$(2.38) \quad \begin{aligned} & \min_{\mathbf{w}_t, \xi_{it}} \left\{ \sum_{t=1}^T \sum_{i=1}^m \xi_{it} + \frac{\lambda_1 \lambda_2}{T(\lambda_1 + \lambda_2)} \sum_{t=1}^T \|\mathbf{w}_t\|^2 + \frac{\lambda_1^2}{T(\lambda_1 + \lambda_2)} \sum_{t=1}^T \|\mathbf{w}_t - \frac{1}{T} \sum_{s=1}^T \mathbf{w}_s\|^2 \right\}, \\ & \text{s.t. } y_{it} \cdot \mathbf{w}_t \cdot \mathbf{x}_{it} \geq 1 - \xi_{it}, \quad \xi_{it} \geq 0, \end{aligned}$$

where  $\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t$  (see Eq. (2.34)). The second regularization term in Eq. (2.38) implies that all tasks are clustered into a single group, and the parameters across all tasks are constrained to exhibit maximum similarity. This special case shows that all tasks are clustered into one group. In practice, however, it is worth noting that certain related tasks might frequently be clustered into different groups.

Horizontal Methods. Clustered Multi-Task Learning (CMTL) (J. Zhou, Chen, & Ye, 2011) assumes that multiple tasks in the same cluster are similar to each other, and provides the insights of inherent relationships between ASO (Ando et al., 2005) and CMTL. Specifically, the CMTL is non-convex,

and the proposed convex relaxation of CMTL is equivalent to an existing convex relaxation of ASO. The objective function of CMTL can be formulated as

$$(2.39) \quad \begin{aligned} & \min_{\mathbf{W}, \mathbf{F}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \mathbf{w}^t - \mathbf{y}^t\|_2^2 + \lambda_1 (\text{tr}(\mathbf{W}^\top \mathbf{W}) - \text{tr}(\mathbf{F}^\top \mathbf{W}^\top \mathbf{W} \mathbf{F})) + \lambda_2 \sum_{t=1}^T \|\mathbf{w}^t\|_2^2, \\ & \text{s.t. } \mathbf{F}_{t,j} = 1/\sqrt{n_j} \text{ if } t \in \mathcal{C}_j \text{ otherwise } 0, t = 1, \dots, T, \end{aligned}$$

where  $n_j$  is the #task in the  $j$ -th cluster  $\mathbf{C}_j$ .

Hierarchical Methods. TTask Tree (TAT) (Han & Zhang, 2015) model is the first method for MTL to learn the tree structure under the regularization framework. By specifying the number of tree layers as  $H$ , Han and Zhang (2015) utilizes matrix decomposition to learn model weights for each layer, i.e.,  $\{\mathbf{W}_h\}_{h=1}^H$ . TAT devises sequential constraints on the distance between the consecutive weight matrices over tree layers. By combining the loss functions, its learning objective can be shown as:

$$(2.40) \quad \begin{aligned} & \min_{\mathbf{W}} \frac{1}{2} \sum_{t=1}^T \frac{1}{N_t} \|\mathbf{X}^{(t)} \sum_{h=1}^H \mathbf{w}_h^t - \mathbf{y}^t\|_2^2 + \sum_{h=1}^H \lambda_h \sum_{i < j}^T \|\mathbf{w}_h^i - \mathbf{w}_h^j\|_2^2, \\ & \text{s.t. } |\mathbf{w}_{h-1}^i - \mathbf{w}_{h-1}^j| \succeq |\mathbf{w}_h^i - \mathbf{w}_h^j|, \forall h \geq 2, \forall i < j, \end{aligned}$$

where the hyperparameters  $\{\lambda_h\}_{h=1}^H$  indicate the importance of different tree layers, and  $|\cdot|$  and  $\succeq$  denotes the elementwise operation. This sequential constraint encourages a non-increasing order for the pair distance between tasks from bottom to top.

### Remarks

- (i) Task clustering methods are scalable with respect to the number of tasks in MTL.
- (ii) Both clustering and priori sharing methods in MTL carry similar underlying meanings as they inherently decipher task relationships.
- (iii) Task clustering complements other MTL strategies, as any MTL approach can be implemented within the task clusters.
- (iv) Solutions in this section tend to be suboptimal, given that task clustering is not exclusive.

## 2.2. Deep Learning Era: Effective and Diversified.

With the advent of deep learning, more powerful computational units and more effective memory bandwidth, e.g., Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs), have made it possible to learn richer features for challenging tasks. Deep MTL methods, unlike traditional MTL methods imposing parameter regularizations or decompositions, can handle large-scale parameter sharing, feature propagation, NAS, task balancing, and optimization intervention, to name a few. The traditional techniques often involve complicated mathematical analysis but fail to learn a satisfactory performance in the real-world scenario with noise-polluted data or loosely-related tasks. However, deep MTL methods can overcome these issues by (1) directly extracting features in raw data and gradually elevating features layer-by-layer from low-level textures to mid-level semantics to high-level responses; and (2) progressively learning activations by stochastic gradients descent (SGD) (LeCun et al., 2002; Robbins & Monro, 1951) that is provably efficient and practical in obtaining an expressive networks (Livni et al., 2014). In this manner, hierarchical features can be efficiently communicated at different levels for jointly learning of multi-task objectives.

This section begins with a discussion of the architecture taxonomy commonly adopted in deep MTL, which serves as the backbone for the rest of the method overview. In the following, we summarize the feature propagation techniques that include feature fusion (see § 2.2.1), cascading (see § 2.2.2), distillation (see § 2.2.3), and cross-task attention (see § 2.2.4). These techniques

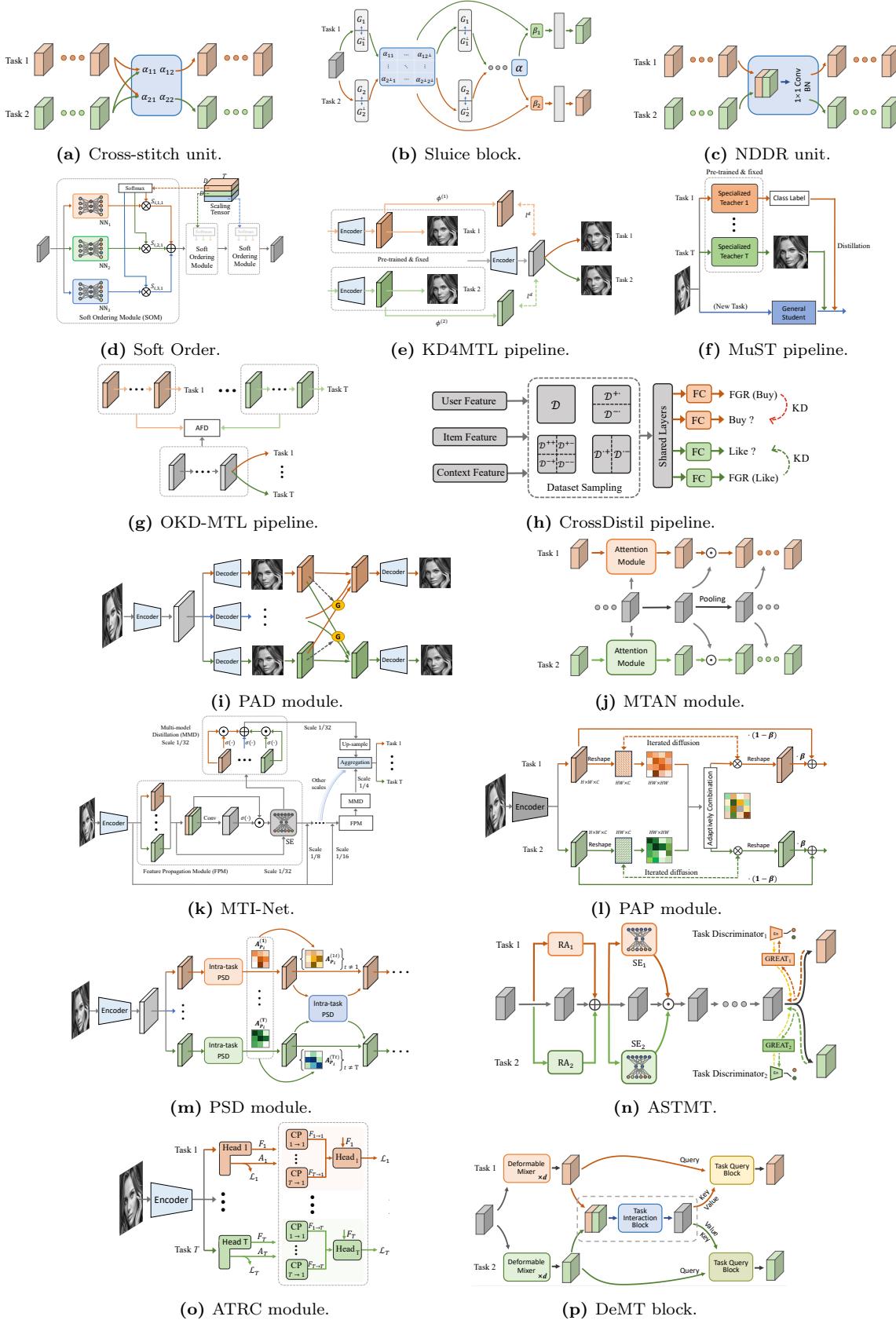
**Table 5.** Summary of deep MTL models.

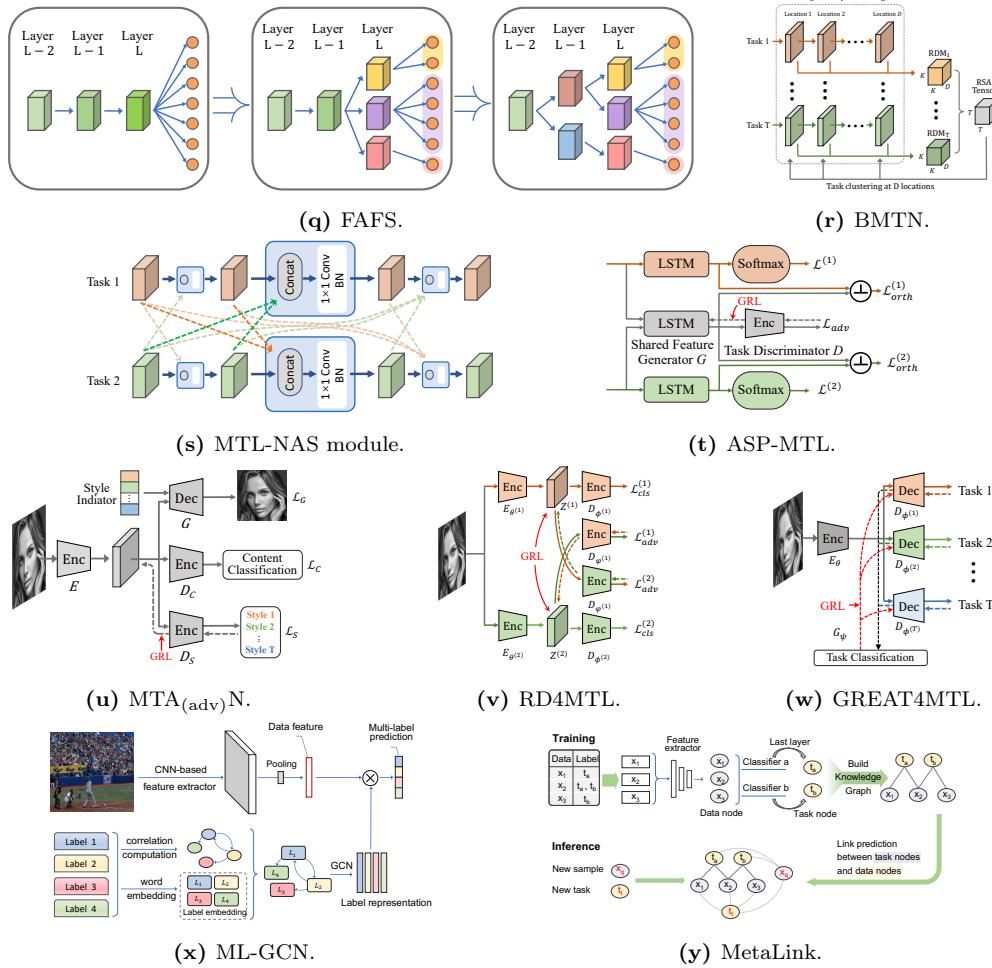
| Model Name                       | Origin        | Year | MTL Strategy  | Backbone          | Sharing     | Modality     | Task  | Measurement   | Loss Function  | Availability <sup>1</sup> |          |
|----------------------------------|---------------|------|---|-------------------|-------------|--------------|---|---|--|---------------------------|----------|
| TCDCN                            | ECCV          | 2014 | Early stopping  | CNN               | Hard        | Image        | Facial landmark detection/head pose estimation/gender classification/age estimation/expression recognition/facial attribute inference | Mean error (mErr) (Burgos-Artizzu et al., 2013), failure rate (Dantone et al., 2012)                                    | Mean squared error (MSE), cross-entropy (CE) loss                        | Official                  |          |
| MTL-ML                           | ACL-LICNLNP   | 2015 | —   | RNN               | Hard        | Text         | Multiple-target language translation  | BLEU-4 (Papineni et al., 2002), Delta   | CE loss  | —                         |          |
| Vanilla Cascading                | ACL           | 2016 | Cascading   | LSTM              | Hard        | Text         | Part-Of-Speech (POS)/Chunking/Combinatory Categorical Grammar (CCG) Supertagging  | F1 score, Micro-F1 score  | CE loss  | —                         |          |
| Cross-stitch networks            | CVPR          | 2016 | —   | CNN               | Soft        | Image        | Surface normals estimation (normals)/semantic segmentation (semseg), object detection/attribute prediction                            | mErr/ median error (medErr)/within $t^*$ in angular distance (within $t^*$ ), pixel accuracy (pixacc), mIoU, f1IoU, mAP | CE loss  | Unofficial                |          |
| ASP-MTL (aka AdvMTL)             | arXiv         | 2017 | Adversarial training  | LSTM              | Hard & Soft | Text         | Text classifications  | Error rate  | CE loss, adversarial loss, orthogonality constraint                      | Official                  |          |
| JMT                              | EMNLP         | 2017 | Cascading, adding constraints                                       | LSTM              | Soft        | Text         | Part-Of-Speech (POS) tagging/chunking/parsing/semantic relatedness/textual entailment   | Accuracy (acc), F1, MSE, unlabeled attachment score (UAS)/labeled attachment score (LAS)                                | CE loss, softmax loss, KL-divergence loss                                | Unofficial                |          |
| MNCs                             | CVPR          | 2016 | Cascading   | CNN               | Hard        | Image        | Object detection/mask estimation/object categorization  | mAP@mIoU  | Mask regression loss, softmax loss                                       | Official                  |          |
| FAFS                             | CVPR          | 2017 | NAS   | CNN               | Hard        | Image        | person attribute classification   | Acc/recall  | CE loss  | Official                  |          |
| MRN                              | NeurIPS       | 2017 | Task conditioning   | CNN               | Hard & Soft | Image        | classifications on different domains  | rel   | CE loss  | Official                  |          |
| PAD-Net                          | CVPR          | 2018 | Mutual distillation   | CNN               | Hard        | Image        | Depth/scene parsing/contour prediction/normals  | rel (Eigen et al., 2014)/RMSE/ log10 mErr/ acc with threshold $\delta$ (acc- $\delta$ ), IoU/acc                        | CE loss, softmax loss, Euclidean loss                                    | —                         |          |
| MTA <sub>(adv)</sub> N           | CVPR          | 2018 | Adversarial training  | CNN               | Hard        | Image        | font/glyph/identity/pose/illumination   | Recognition rate  | CE loss, adversarial loss  | —                         |          |
| TRL                              | ECCV          | 2018 | cross-task attention  | CNN               | Hard        | Image        | Depth estimation (depth)/semseg   | rel/ RMSE/acc- $\delta$ , pixacc/mean acc/mIoU  | berHu loss (Laina et al., 2016), CE loss, uncertainty loss               | —                         |          |
| MMoE                             | KDD           | 2018 | MoE   | MLP               | Hard & soft | Tabular      | Income/education/marriage prediction, engagement/satisfaction in recommendation   | Area Under the Curve (AUC)  | CE loss  | Unofficial                |          |
| Soft Order                       | ICLR          | 2018 | feature fusion  | CNN, MLP          | Soft        | Tabular      | data, image   | mErr  | CE loss  | —                         |          |
| GREAT4MTL                        | arXiv         | 2018 | adversarial training  | CNN               | Hard        | Image        | classification/colorization/edge/denoised reconstruction, depth/normal/keypoint   | Err, RMSE, $1 -  \cos(\cdot, \cdot) $   | CE loss  | —                         |          |
| Sluice networks                  | AAAI          | 2019 | Adding constraints, early stopping                                  | LSTM              | Hard & Soft | Text         | Clicking/crafty recognition (NER)/semantic role labeling/POS tagging  | Acc   | CE loss  | Official                  |          |
| HMTL                             | AAAI          | 2019 | cascading   | LSTM              | Hard        | Text         | NER Entity Mention Detection (EMD)/Relation Extraction (RE)/Coreference Resolution (CR)   | F1 score/precision/recall, MUC/B3/CEAF <sub>e</sub> (Moosavi & Strubbe, 2018)   | CE loss  | Unofficial                |          |
| DCMTL                            | AAAI          | 2019 | cascading   | CNN, LSTM         | Hard        | Text         | Segment labeling/Named Entity Labeling (NEL)/slot filling   | F1 score/precision/recall   | CRF loss, CE loss, ranking loss (N. T. Vu et al., 2010)                  | Official                  |          |
| NDDR-CNN                         | CVPR          | 2019 | feature fusion  | CNN               | Soft        | Image        | Normals/semseg, age estimation/gender classification  | mErr (medErr/within $t^*$ , mIoU, pixacc, mean/median absolute error (absErr), acc)                                     | CE loss  | Official                  |          |
| PAP                              | CVPR          | 2019 | cross-task attention  | CNN               | Hard        | Image        | Semseg/depth/normals  | RMSE/rel/acc with t, mErr/medError/within $t^*$ , mIoU/mean accuracy (mAcc)/pixacc                                      | CE loss, $\ell_1$ loss, berHu loss affinity loss (Z. Zhang et al., 2019) | —                         |          |
| MTA <sub>(cater)</sub> N (& DWA) | CVPR          | 2019 | Adaptive weighting  | CNN               | Hard & Soft | Image        | Semseg/depth/normals, 10 classifications (visual domain decathlon <sup>2</sup> )  | mIoU/pixacc, mErr/medErr/within $t^*$ , absErr/real error, accuracy   | CE loss, $\ell_1$ loss, dot product                                      | Official                  |          |
| ASTMT                            | CVPR          | 2019 | attention, single-tasking   | CNN               | Hard        | Image        | Semseg/depth/edges/normals/human parts/saliency estimation/allbedo  | mIoU/odF/mErr/maximum F-measure (maxF)/RMSE/ $\Delta_m$   | CE loss, $\ell_1$ loss   | Official                  |          |
| ML-GCN                           | CVPR          | 2019 | Graph based   | CNN, GCN          | Hard        | Image        | Multi-label recognition   | precision, recall, F1   | CE loss  | Official                  |          |
| RD4MTL                           | arXiv         | 2019 | Adversarial training  | CNN               | Hard        | Image        | Classifications   | mIoU/medErr/Within $t^*$ , mIoU/pixacc, Acc   | CE loss, adversarial loss  | Official                  |          |
| MTL-NAS                          | CVPR          | 2020 | NAS   | CNN               | Adaptive    | Image        | Semseg/normals, object classification/scene classification  | mIoU/medErr/Within $t^*$ , mIoU/pixacc, Acc   | CE loss, $\ell_2$ loss   | Official                  |          |
| BMTN                             | BMVC          | 2019 | NAS   | CNN               | Adaptive    | Image        | Semseg/edge/depth/keypoint detection (point), attribute classification  | mIoU, pixacc, $\ell_1$ , Acc  | CE loss, $\ell_1$ loss, $\Delta_m$                                       | Official                  |          |
| PSD                              | CVPR          | 2020 | Distillation  | CNN               | Hard & soft | Image        | Semseg/depth/normals  | RMSE/rel/acc with t, mIoU/mean accuracy/pixacc, mErr/medErr/within $t^*$  | CE loss, $\ell_1$ loss, berHu loss                                       | —                         |          |
| KD4MTL                           | ECCV Workshop | 2020 | distillation/knowledge  | CNN               | Hard & soft | Image        | Semseg/depth/normals, classification  | mIoU/pixacc, absErr/rel, mErr/medErr/within $t^*$   | CE loss, $\ell_1$ loss, dot product                                      | Official                  |          |
| MTI-Net                          | ECCV          | 2020 | multi-task distillation   | CNN               | Hard & Soft | Image        | Semseg/depth/edges detection (edges)/normals/saliency estimation/human parts  | mIoU, RMSE, mErr, optimal dataset-scale F-measure (odF) (Martin et al., 2004), $\Delta_m$                               | CE loss, $\ell_1$ loss   | Official                  |          |
| LTB                              | ICML          | 2020 | NAS, task grouping  | CNN               | Soft        | Image        | Regression, face attribute prediction, semseg/normals/depth/keypoints/edges   | Acc, CE, cos, mean absErr   | CE loss, $\ell_1$ loss, cosine loss                                      | —                         |          |
| AAMTRL                           | ICML          | 2020 | adversarial training  | CNN & LSTM        | Hard        | Text         | Classifications   | Relatedness evolution, acc, influence of #task  | Any 1-Lipschitz loss   | —                         |          |
| CGC & PLE                        | RecSys        | 2020 | MoE   | MLP               | Hard & soft | Tabular      | Sub-tasks in the recommendation systems, income/education/marriage prediction   | AUC/MSE, MTL gain   | CE loss, $\ell_2$ loss   | Unofficial                |          |
| TSNs                             | ICCV          | 2021 | task relationship learning, task conditioning                       | CNN               | Hard        | Image        | Semseg/depth/edges/normals/saliency estimation/human parts  | mIoU, RMSE, mErr, odsF, $\Delta_m$  | CE loss, $\ell_1$ loss   | Official                  |          |
| MuST                             | ICCV          | 2021 | knowledge distillation, task conditioning                           | CNN               | Hard        | Image        | Classification/detection/semseg/depth/normals   | Acc, mIoU, RMSE, odsF   | CE loss, $\ell_1$ loss   | —                         |          |
| AuxSegNet                        | ICCV          | 2021 | cross-task attention  | CNN               | Hard & Soft | Image        | Semseg/classification/saliency detection  | mIoU/precision/recall   | Multi-label softmax loss, CE loss  | Official                  |          |
| ATRC                             | ICCV          | 2021 | cross-task attention  | CNN               | Hard & soft | Image        | Semseg/depth estimation/edges/normals/saliency estimation/human parts   | mIoU, RMSE, mErr, odsF, maxF, $\Delta_m$  | CE loss, $\ell_1$ loss   | Official                  |          |
| DSelect-k                        | NeurIPS       | 2021 | MoE   | MLP, CNN          | Hard & soft | Tabular      | data, Image   | engagement/satisfaction task, classification  | Total loss, Acc, AUC/RMSE, #expert                                       | CE loss, $\ell_2$ loss    | Official |
| MT-TaG                           | ArXiv         | 2022 | MoE   | Transformer       | Hard & soft | Text         | 16 Language understanding tasks, e.g. textual entailment, sentiment classification, etc.  | Acc, Spearman correlation (Spearman correlation coefficient (Matthews, 1975))   | CE loss, MSE   | —                         |          |
| CrossDistil                      | AAAI          | 2022 | distillation  | MLP               | Hard & soft | Tabular      | Finish watching/like  | AUC, multi-AUC (Hand & Till, 2001)  | CE loss  | —                         |          |
| MuLT                             | CVPR          | 2022 | cross-task attention  | CNN & Transformer | Hard        | Image        | Semseg/depth/reshading/normals/keypoints/edges  | MTL gain, mErr of domain generalization   | CE loss, $\ell_1$ loss, rotate loss (Zamir et al., 2018)                 | Official                  |          |
| MTFormer                         | ECCV          | 2022 | cross-task attention, task balancing (spcc., Kendall et al. (2013)) | Transformer       | Hard & soft | Image        | Semseg/depth/saliency detection/human parts   | mIoU, RMSE, $\Delta_m$  | CE loss, $\ell_1$ loss, cross-task contrastive loss, uncertainty loss    | —                         |          |
| MQTransformer                    | arXiv         | 2022 | cross-task attention  | Transformer       | Hard & Soft | Image        | Semseg/depth/edges/normals/saliency estimation/human parts  | mIoU, RMSE, mErr, odsF, maxF  | CE loss, $\ell_1$ loss   | —                         |          |
| MetaLink                         | ICLR          | 2023 | Graph based   | MLP, GNN          | Hard        | Image, Graph | Classification  | mAP, ROC AUC  | CE loss  | Official                  |          |
| DeMT                             | AAAI          | 2023 | cross-task attention  | CNN & Transformer | Hard & Soft | Image        | Semseg/depth/edges/normals/saliency estimation/human parts  | $\Delta_m$ , mIoU, RMSE, mErr, odsF, maxF   | CE loss, $\ell_1$ loss   | Official                  |          |
| mTEB                             | WACV          | 2023 | cross-task attention  | CNN               | Hard & soft | Image        | Semseg/depth/normals/edges  | $\Delta_m$ , mIoU, RMSE, mErr, F1   | CE loss, berHu loss, cosine loss (Guizilini et al., 2021)                | Official                  |          |
| OKD-MTL                          | WACV          | 2023 | distillation, task weighting  | Transformer       | Hard & Soft | Image        | Semseg/depth/normals  | $\Delta_p$ , mIoU/pixacc, absErr/rel, mErr/medErr/within $t^*$  | Adaptive feature distillation loss, CE loss, $\ell_1$ loss, cosine loss  | —                         |          |
| AdaMV-MoE                        | ICCV          | 2023 | MoE   | Transformer       | Hard & soft | Image        | classification/detection/Seg  | Acc, Average Precision (AP)   | CE loss  | Official                  |          |

<sup>1</sup>This column provides the link to the implementation or execution. Click on “Official” or “Unofficial” to access the website.<sup>2</sup>Part of PASCAL in Detail Workshop Challenge, CVPR 2017, July 26th, Honolulu, Hawaii, USA. <https://www.robots.ox.ac.uk/~vgg/decahlon>.<sup>3</sup>We use “state<sup>+</sup> here to represent the domain of reinforcement learning, including the observations of states of environment, the positions of object, the actions made by agent, etc.<sup>4</sup>The average rank of MTL on all different tasks. MR = 1 if a method ranks first across all tasks.

encourage networks to automatically combine the features learned from different tasks, addressing the crucial challenge of effectively and efficiently utilizing the rich features enabled by deep learning. § 2.2.5 presents an overview of task balancing techniques in deep MTL, incorporating the linear combination of different tasks through three essential factors: gradient, loss, and learning speed. The comparison and recalibration of these factors aim to coordinate diverse tasks during the model weight update process. We will discuss this section from the point of gradient correction and dynamic weighting. In contrast, § 2.2.6 explores MOO in the context of MTL, which aims to simultaneously optimize potentially conflicting objective functions. Other promising topics covered include adversarial multi-task training (see § 2.2.7), MoE (see § 2.2.8), GCN-based MTL (see

§ 2.2.9), and NAS for MTL (see § 2.2.10). The summary of deep MTL models is presented in Table 5, and representative DL frameworks in MTL are illustrated in Fig. 8.

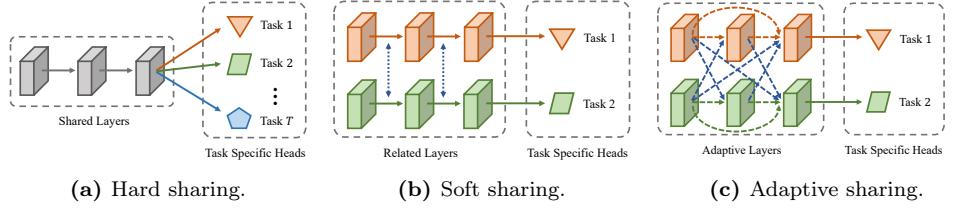




**Figure 8.** Frameworks of deep learning techniques used in MTL. (a-d) Feature fusion: cross-stitch networks, Sluice Network, NDDR-CNN, and Soft Order. (e-h) Knowledge distillation: KD4MTL, MuST, OKD-MTL, and CrossDistill. (i-p) Attention: PAD, MTAN, MTI-Net, PAP, PSD, ASTMT, ATRC, and DeMT. (q-s) NAS: FAFS, BMTN, and MTL-NAS. (t-w) Adversarial MTL: ASP-MTL, MTAN, RD4MTL, and AAMTRL. (x-y) Graph: ML-GCN and MetaLink.

**Architecture Taxonomy.** The remarkable success of deep MTL can be attributed to the rich extracted representations and their efficient sharing. Multi-task sharing relies on the basic splitting ways of architectures among involved tasks. P. Liu et al. (2016) first discuss three different sharing mechanisms based on text classification in Recurrent Neural Networks (RNNs): uniform-, coupled-, and shared-layer architectures. Ruder (2017) first organize it into two categories: hard parameter sharing and soft parameter sharing. According to this taxonomy, the uniform-layer architecture falls under hard-parameter sharing, while coupled- and shared-layer architectures are considered soft-parameter sharing. In general, Ruder (2017)'s taxonomy has been widely accepted by the research community (Vandenhende et al., 2021). We carry forward this taxonomy and enrich it with more details.

In hard parameter sharing, as shown in Fig. 9a, different tasks can share identical parameters in shallow layers and maintain their own specific parameters in the splitting heads. As shown in Fig. 7a, this idea can be dated back to 1990s (Bromley et al., 1993; R. Caruana, 1997; R. A. Caruana, 1993) when high-related tasks are introduced into a shared FNNs to serve as inductive bias for each other. Fig. 7b shows this idea used in RNNs in a modern way (D. Dong et al.,



**Figure 9.** Architecture taxonomy proposed by Ruder (2017) for deep multi-task sharing: (a) Hard parameter sharing, (b) soft parameter sharing, and (c) adaptive sharing. The 1D arrows indicate computations within the neural networks involving learnable parameters. The 2D shapes and 3D cubes represent the final responses and extracted features, respectively.

2015). CNNs can also adopt hard parameter sharing to perform multiple related tasks. As shown in Fig. 10, TCDCN (Z. Zhang et al., 2014) and Fast RCNN (Girshick, 2015; Girshick et al., 2014) are the earliest practice of this idea in computer vision. From a representation learning perspective, shallow layers are typically shared as a feature encoder that extracts common features such as edges and textures. By enriching these common features with more related tasks, deeper layers can help enable multitasking on task-specific heads.

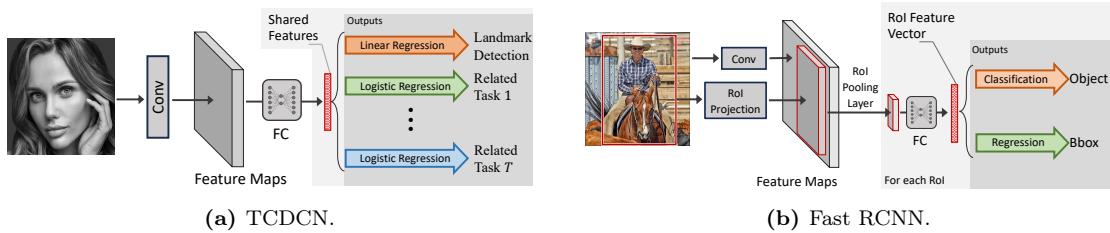
Misra et al. (2016) argue that there is no principled way of architecture splitting in hard parameter sharing, and conducted the first empirical study to investigate the performance trade-offs amongst varieties of involved tasks and splitting ways in CNNs. The dependence between involved tasks and the splitting ways of architecture motivates the exploration of an architecture that can capture all possible splittings and thus learn an optimal combination of task-shared and task-specific representations, i.e., soft parameter sharing shown in Fig. 9b. While hard-parameter sharing requires shallow layers to be identical across tasks, soft-parameter sharing encourages each task to maintain its own shallow layers and leverage features from related tasks during the propagation to capture similarities. These feature propagation techniques include but are not limited to fusion, aggregation, attention, etc. However, whether employing hard or soft parameter sharing, exploring the MTL architecture space still remains error-prone.. First of all, this space for deep neural architectures grows exponentially with depth, and incorporating more tasks significantly expands the range of optimal solutions. On the other hand, while hard parameter sharing compresses the model size, leading to a sub-optimal solution, soft parameter sharing ensures advancement by maintaining the maximum total model size, allowing each task to learn a specific architecture in contrast to STL. An adaptive architecture search in a greedy manner during the neural network training process shows promise. As shown in Fig. 9c the adaptive parameter sharing, each path from the different layers of different tasks is active before training. The connections vanish with the pursuit of model compression in the process of multi-task optimization, and usually, a thin network is finalized after this dynamic branching procedure.

Unless explicitly stated otherwise, we employ the notation provided in Tab. 6 within the context of deep learning settings to expand upon and complement the information presented in Tab. 3.

**2.2.1. Feature fusion.** Feature fusing is a common technique used in MTL to fuse features extracted under the supervision of different tasks, which can leverage shared and private knowledge across tasks. This technique allows each network to better exploit the relationships between tasks and thus improve overall performance. In general, feature fusion in MTL involves weighted summation, concatenation, or a combination of both. We categorize the feature fusion methods into two classes: parallel sharing, where the feature fusion happens at the same position of layers between tasks, and Non-parallel sharing, in which the permutation of sharing layers may exist. The representative

**Table 6.** Summary of notations used in Sec. 2.2.

| Notation   | Description   |
|--|---|
| $b, B$   | Batch size.   |
| $lr$   | Learning rate.  |
| $\mathcal{X}_t^l \in \mathbb{R}^{(B \times) H \times W \times C}$                    | Feature maps output from $l$ -th layer of $t$ -th task, where $(B, H, W, C$ are (batch size,) #height, #width, and #channel.                                |
| $\mathcal{W} \in \mathbb{R}^{S \times S \times C_{\text{in}} \times C_{\text{out}}}$ | Convolution filter, where $S$ denotes the size of filter, and $C_{\text{in}}, C_{\text{out}}$ denote the number of input and output channels, respectively. |
| $\exp(\cdot)$  | Exponential function.   |
| $\sigma(\cdot)$  | Sigmoid function, where $\sigma(x) = 1/(1 + \exp(-x))$ .  |
| $\text{softmax}(\cdot)$  | Softmax function, where $[\text{softmax}(\mathbf{x})]_j = \exp(x_j)/\sum_i \exp(x_i)$ for any entry index $j$ .   |
| $\text{sim}(\cdot, \cdot)$   | An arbitrary similarity function, e.g. cosine similarity $\cos(\cdot, \cdot)$ .   |
| $\odot$  | The element-wise dot product.   |
| $LN(\cdot)$  | Layer norm.   |
| $MHSA(q, k, v)$  | Multi-head self-attention operator.   |
| $CONV_{\mathcal{W}}(\cdot)$  | Convolution operation parametrized by $\mathcal{W}$ .   |
| $\text{RESHAPE}(\cdot)$  | Reshape operation to rearrange the original feature maps in $\mathbb{R}^{H \times W \times C}$ space into a new $\mathbb{R}^{HW \times C}$ space.           |



**Figure 10.** Two of the earliest applications of hard-parameter sharing in CNNs: (A) the Tasks-Constrained Deep Convolutional Network (TCDCN), which jointly extracts common features from human faces for multiple tasks such as landmark detection, head pose estimation, and facial attribute inference; and (B) the Fast Region-based Convolutional Network method (Fast R-CNN), where each region of interest (RoI) is projected into a fixed-size feature map first and then mapped to a feature vector used for both object probability prediction and bounding-box offsets regression.

works in the line of parallel sharing include Cross-Stitch Networks (Misra et al., 2016), Sluice Networks (Ruder et al., 2019), and Neural Discriminative Dimensionality Reduction in Convolutional Neural Networks (NDDR-CNN) (Y. Gao et al., 2019). As research in this direction progresses, an increasing number of learnable parameters are being used to control the fusion process. For example, Cross-Stitch Networks utilize four task-aware parameters, Sluice Networks capture latent subspaces of features via extra parameters, and NDDR-CNN models layer-wise fusion by using  $1 \times 1$  convolutions. However, expecting task feature hierarchies to align perfectly, even among closely related tasks, is unreasonable. Imposing parallel sharing in these unmatched layers could lead to negative transfer. To remedy this dilemma, Soft Order (Meyerson & Miikkulainen, 2018) uses a more flexible ordering of shared layers to assemble them in different ways for different tasks.

Parallel sharing. Cross-Stitch Networks (Misra et al., 2016) is a soft parameter-sharing architecture that can learn an optimal combination of task-shared and task-specific representations via four learnable parameters, which is named cross-stitch unit. As shown in Fig. 8a, the activations from different tasks are linear combined via four parameters  $(\alpha_{11}, \alpha_{12}, \alpha_{21}, \alpha_{22})$ . We denote by  $\mathcal{X}_l^i$  the feature maps in the  $l$ -th layer of task  $i$ . Then the formalization of the Cross-Stitch unit is

$$(2.41) \quad \begin{bmatrix} \mathcal{X}_{l+1}^1 \\ \mathcal{X}_{l+1}^2 \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \begin{bmatrix} \mathcal{X}_l^1 \\ \mathcal{X}_l^2 \end{bmatrix}$$

Specifically, the extreme setting of  $(\alpha_{11}, \alpha_{12}, \alpha_{21}, \alpha_{22}) = (1, 0, 0, 1)$  can make certain layers to be non-sharing. From this perspective, the separate STL is a special case of cross-stitch combinations. By varying  $\alpha_{11}$  and  $\alpha_{22}$  values, this proposed unit can move between task-shared and -specific representations, and even choose a middle ground if necessary.

Sluice Networks (Ruder et al., 2019) learns shared parameters between two BiLSTM-based sequence labeling networks (Plank et al., 2016). This work aims to model loosely related tasks with

non-overlapping datasets. As shown in Fig. 8b a sluice meta-network with two tasks, of which each layer is partitioned into two orthogonal subspaces  $\mathbf{G}$  and  $\mathbf{G}^\perp$ . Accordingly, the activations in the  $l$ -th layer of task  $i$  are also partitioned into  $\mathcal{X}_l^i$  and  $\mathcal{X}_l^{i^\perp}$ , thus leading to a matrix in  $\mathbb{R}^{4 \times 4}$  to combine activations from two tasks:

$$(2.42) \quad \begin{bmatrix} \mathcal{X}_{l+1}^1 \\ \mathcal{X}_{l+1}^{1^\perp} \\ \mathcal{X}_{l+1}^2 \\ \mathcal{X}_{l+1}^{2^\perp} \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{11^\perp} & \alpha_{12} & \alpha_{12^\perp} \\ \alpha_{1^\perp 1} & \alpha_{1^\perp 1^\perp} & \alpha_{1^\perp 2} & \alpha_{1^\perp 2^\perp} \\ \alpha_{21} & \alpha_{21^\perp} & \alpha_{22} & \alpha_{22^\perp} \\ \alpha_{2^\perp 1} & \alpha_{2^\perp 1^\perp} & \alpha_{2^\perp 2} & \alpha_{2^\perp 2^\perp} \end{bmatrix} \begin{bmatrix} \mathcal{X}_l^1 \\ \mathcal{X}_l^{1^\perp} \\ \mathcal{X}_l^2 \\ \mathcal{X}_l^{2^\perp} \end{bmatrix}$$

Inspired by Cross-stitch networks, these  $\alpha$  values are learnable to control how much to share for task-shared information and how much to preserve for task-specific information. Finally,  $\beta$  parameter (see Fig. 8b), through the skip-connections, linearly summarizes the multi-task representations at various levels of the network architecture.

Neural Discriminative Dimensionality Reduction in Convolutional Neural Networks (NDDR-CNN) (Y. Gao et al., 2019) further concatenates feature maps from different tasks in a channel-wise manner. This NDDR, as shown in Fig. 8c, can be fulfilled by using simple  $1 \times 1$  convolutional layer plus batch normalization layer, and be extended to any end-to-end training CNN in a “plug-and-play” fashion. Considering the number of tasks being  $T$ , we can denote  $1 \times 1$  convolution by  $\mathcal{W} \in \mathbb{R}^{1 \times 1 \times TC \times TC}$ , where  $TC$  is the depth of combined feature maps from all tasks. We concatenate feature maps according to the channel dimension and divide  $1 \times 1$  convolution according to the output dimension by  $T$  tasks as follows:

$$\mathcal{X}_l = [\mathcal{X}_l^1, \dots, \mathcal{X}_l^T], \mathcal{W} = [\mathcal{W}^1, \dots, \mathcal{W}^T],$$

where  $\mathcal{X}_l \in \mathbb{R}^{H \times W \times TC}$  and  $\mathcal{W}^t \in \mathbb{R}^{1 \times 1 \times TC \times C}$ . Then, the output feature maps at the  $(l+1)$ -th layer for the  $t$ -th task can be calculated as

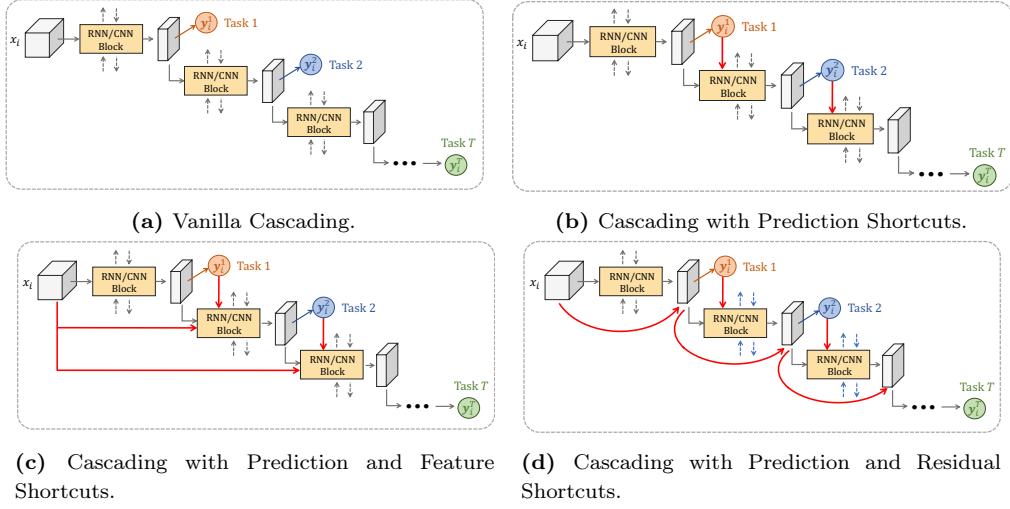
$$(2.43) \quad \mathcal{X}_{l+1}^t = CONV_{\mathcal{W}^t}(\mathcal{X}_l), t = 1, \dots, T.$$

The NDDR layer defined by Eq. (2.43) is a standard  $1 \times 1$  convolution operation in CNNs. To avoid a trivial solution on  $\mathcal{W}$  and the noise directions of learned features, the batch normalization layer is followed after each NDDR layer, and the  $\ell_2$  weight decay is applied on the weights of the NDDR layer, respectively.

Unparallel sharing. Soft Order (Meyerson & Miikkulainen, 2018) learns how shared layers are assembled in permuted ways for different tasks. Specifically, a learnable tensor of scalars  $S \in \mathbb{R}^{L \times L \times T}$ , is used to implement the soft ordering, where  $L$  is #layer and  $T$  is #task. For simplicity, consider a hard sharing network with  $L$  shared layers  $\{f_{\mathcal{W}_l}\}_{l=1}^L$  ( $f$  can be  $CONV$  or Linear function), then the soft ordering of this hard sharing for the  $t$ -th task is:

$$(2.44) \quad \mathcal{X}_l^t = \sum_{j=1}^L s_{t,j,l} f_{\mathcal{W}_j}(\mathcal{X}_{l-1}^t), l = 1, \dots, L, t = 1, \dots, T, \quad \text{s.t. } \sum_{j=1}^L s_{t,j,l} = 1 \text{ with } \forall(t, l),$$

where  $s_{t,j,l}$  is the  $(t, j, l)$ -th entry of the tensor  $S$ . Fig. 8d visualizes this layer permutation operation. It is noticed that the constraint on  $s_{t,j,l} = 1$  for  $\forall(t, l)$  can be easily implemented via a softmax function. In practice, a dropout operation is beneficial to increasing the generalization capacity of shared representations.



**Figure 11.** The taxonomy of cascading structures into four categories: (A) the vanilla cascading structure, (B) the cascading structure with prediction shortcuts, (C) the cascading structure with prediction and feature shortcuts, and (D) the cascading structure with prediction and residual shortcuts.

### Remarks

- (i) Feature fusion enables the exploration of multi-task interactions in a "plug and play" manner, making it a general-purpose MTL solution that can be generalized to any backbones.
- (ii) The feature-level relationships between tasks can be investigated by examining the introduced learnable parameters after training.
- (iii) Feature fusion cannot reveal what information is propagated during the multitasking process, highlighting the need for design guidelines that go beyond common practices.
- (iv) Feature fusion inherently imposes constraints on the SIMO setting, as it allows features to be fused only within the same context.
- (v) Feature fusion creates task-specific branches that also need to learn shared features across tasks, which can hinder task-awareness compared to STL, which focuses on capturing representations specific to the target task.

2.2.2. *Cascading*. Having supervision from all tasks at the outermost level is shown to be sub-optimal, another avenue of investigation for mitigating this parallel sharing is through the implementation of multi-task cascaded learning (Søgaard & Goldberg, 2016). This field of study involves supervising tasks at different levels within their respective layers, facilitating higher-level tasks to effectively leverage the shared representation derived from lower-level tasks. In practice, multi-task cascading can be applied to 1) the complicated task that can be decomposed into several sub-tasks, e.g., instance-aware semantic segmentation decomposed into differentiating instances, estimating masks and categorizing objects in CV (J. Dai et al., 2016), and 2) a group of hierarchical tasks, e.g., part-of-speech (POS) tagging (word-level), dependency parsing (syntactic-level) and question answering (QA) (semantic-level) in NLP (Hashimoto et al., 2017; Søgaard & Goldberg, 2016). In this line of research, early work (Søgaard & Goldberg, 2016) realize cascading by having low-level tasks supervised at shallow layers, and then reusing representations from shallow layers for higher-level tasks. The Joint Many-Task (JMT) model (Hashimoto et al., 2017) adds shortcut connections from each lower-level task prediction to higher-level tasks, which can further reflect task hierarchies. Furthermore, shortcut connections in Multi-task Network Cascades (MNCs) (J. Dai et al., 2016) and Deep Cascade Multi-Task Learning (DCMTL) (Y. Gong et al., 2019) come from both cascade

connection (predictions) and residual connection (features). Hierarchical MTL (HMTL) (Sanh et al., 2019) introduces more semantic tasks to share both common embeddings and encoders in a hierarchical cascading architecture.

Vanilla Cascading (Søgaard & Goldberg, 2016) first presents a multi-task learning architecture that utilizes bi-directional RNNs. This architecture enables the supervision of different tasks at various layers, as shown in Fig. 11a. In this study, the POS task is supervised at the innermost layer, and the syntactic chunking and Combinatory Categorical Grammar (CCG) supertagging join at the outermost layer to utilize the shared representation of the lower-level tasks via a hard parameter sharing. In this case, the incorporation of lower-level task supervision affects the shallow layer parameter updating, which is beneficial to all involved tasks in MTL.

Multi-task Network Cascades (MNCs) (J. Dai et al., 2016) performs three sub-tasks of the instance-aware semantic segmentation at the different stages and reuses the features of these tasks at different layers. Each of the three stages involves its own predictions of box-level instance proposals, mask-level instance regression, and instance categorization, respectively, and the later task learning relies on previous prediction output. As shown in Fig. 11b, the innermost features are utilized by all sub-tasks, which is beneficial to both the accuracy and speed in an end-to-end training manner.

Joint Many-Task (JMT) Model (Hashimoto et al., 2017) is another cascading model to predict NLP tasks with different linguistic levels of morphology, syntax, and semantics. JMT shares a similar architecture with MNCs, as shown in Fig. 11c, but each higher-level task contains the shortcut connections from the predictions of all lower-level tasks. In addition, the naïve  $\ell_2$  regularization term is imposed on model weights to allow the improvement of one task without exhibiting catastrophic interference with the other tasks.

Deep Cascade Multi-Task Learning (DCMTL) (Y. Gong et al., 2019) first incorporates both cascade and residual connections. As shown in Fig. 11d, the cascade connections transmit predictions from lower tasks, while the residual connections transmit inputs from lower layers. It has been validated that these skip connections are effective for strictly ordering tasks. The cascading structure alone proves inadequate for high-level tasks that heavily rely on low-level tasks. In addition, DCMTL can outperform previous SOTA methods and has been deployed on the online shopping assistant of a dominant Chinese E-commerce platform.

Hierarchical Multi-Task Learning (HMTL) (Sanh et al., 2019) is a parallel method trained in a hierarchical fashion. This model can supervise a set of low-level tasks at the bottom layers and more complex tasks at the top layers. Similar to MNCs (J. Dai et al., 2016), representations extracted at the very beginning are fed into all the successive encoders for different tasks, which is beneficial to the training stability and acceleration. Also shown in Fig. 11d, HMTL is a variation that parallels high-level tasks could exist, e.g., Coreference Resolution (CR) and Relation Extraction (RE), and more types of word representations like pre-trained GloVe (Pennington et al., 2014) and ELMo (Peters et al., 2018) embeddings, are combined to achieve the best performance.

### Remarks

- (i) Cascading facilitates feature communication across different layers.
- (ii) Cascading enhances the utility of features for tasks at different levels.

**2.2.3. Knowledge Distillation (KD).** Motivated by KD (Hinton et al., 2015) where a teacher model can guide a student model via passing meaningful knowledge (e.g., soft labels), separate models in MTL for different tasks can utilize definite information. Specifically, a teacher model can be trained on multiple tasks that are of interest and then serves as an expert in performing those tasks

and possessing versatile knowledge. The knowledge from the teacher model is then transferred to a student model. This can be done by training the student model to mimic the behavior of the teacher model, e.g., the student model learns to predict the outputs or pattern structures of the teacher model on the shared tasks. On the other hand, the student model can be trained jointly on multiple tasks, using both the labeled data for each task and the guidance from the teacher model. The shared information and generalizable representations learned from the teacher model can benefit the student model’s performance on all the tasks. In this manner, the teacher model performs auxiliary tasks to assist the student model in target tasks. For example, the depth prediction from a customized CNN can help the segmentation task via multi-modal distillation (i.e., train with RGB-Depth data instead of RGB data), while the depth prediction is an intermediate auxiliary task to the target segmentation task (D. Xu et al., 2018). The research in this subfield can be classified into two categories that correspond to the knowledge encompassed within a teacher model: feature-level and response-level. KD4MTL (W.-H. Li & Bilen, 2020) carries forward FitNets (Romero et al., 2014) via optimizing the distance between the features of the offline task-specific networks and the online multi-task network. MuST (Ghiasi et al., 2021) and OKD-MTL (Jacob et al., 2023) distill the knowledge (i.e., pseudo labels) from pre-trained specialized teachers to general-purpose students. MuST (Ghiasi et al., 2021) pretrains several specialized teachers capable of generating multi-task labels for the target dataset. CrossDistil (C. Yang et al., 2022) distills the responses of item preference across different tasks in the recommender system.

Feature-Level. Knowledge Distillation for Multi-task Learning (KD4MTL) (W.-H. Li & Bilen, 2020), as shown in Fig. 8e, first trains an offline task-specific network for each task, and then learns the multi-task network via adding the loss to minimize the distance between the task-specific network and the multi-task network. As the multi-task purpose network is capable of multiple tasks while the task-specific network is more professional at its own task, the two output features cannot be completely matched. Instead, the feature map from multi-task network, denoted by  $\mathcal{X} \in \mathbb{R}^{H \times W \times C}$ , is transformed via an adaptor  $\phi^{(t)} : \mathbb{R}^{H \times W \times C} \xrightarrow{1 \times 1 \times C \times C \text{ CONV}} \mathbb{R}^{H \times W \times C}$ ,  $t = 1, \dots, T$ . These adaptors are jointly learned with the multi-task network via the loss function defined as

$$(2.45) \quad \mathcal{L}^d = \sum_{t=1}^T \ell^d(\phi^{(t)}(\mathcal{X}), \tilde{\mathcal{X}}^{(t)}),$$

where  $\tilde{\mathcal{X}}^{(t)}$  is the feature map from an offline single network corresponding to the task  $t$ , and  $\ell^d$  is defined as the Euclidean distance between the two feature maps that is  $\ell_2$  normalized.

Online KD for MTL (OKD-MTL) (Jacob et al., 2023) proposes an online knowledge distillation method to mitigate negative transfer across tasks. The adaptive feature distillation (AFD) loss with an online task weighting (OTW) scheme is designed to selectively train layers for each task. As shown in Fig. 8g, the critical component AFD is an online weighted knowledge distillation performed on intermediate features from the shared ViT backbone of MTL, and the distilled features are from the teacher model that performs STL on each task. We denote by  $L$  the total number of layers of the ViT encoder backbone and let  $T$  denote the number of tasks. Then the AFD loss is defined as

$$(2.46) \quad \mathcal{L}_{\text{AFD}} = \sum_{l=1}^L \|\bar{\mathcal{X}}_l - \sum_{t=1}^T w_l^t \mathcal{X}_l^t\|_2^2$$

where  $w_l^t$  denotes the learnable parameters for the  $t$ -th task in the  $l$ -th layer, which balances the multiple tasks.  $\bar{\mathcal{X}}_l$  is the shared features learned from the teacher model at  $l$ -th layer. The shared features can be distilled for each task features  $\mathcal{X}_l^t$  through Eq. (2.46) above. In the framework of OKD-MTL, the STL teacher and MTL students are trained in an end-to-end manner through the

total loss

$$(2.47) \quad \mathcal{L}_{total} = \mathcal{L}_{AFD} + \sum_{t=1}^T (\mathcal{L}_{STL}^t + \lambda_t \mathcal{L}_{MTL}^t).$$

To mitigate the gap between the MTL and STL losses, OTW adjusts the task weight  $\lambda_t$  for the  $t$ -th task at iteration  $i$  as follows:

$$(2.48) \quad \lambda_t(i) = T \frac{\exp(r^t(i)/k)}{\sum_{s=1}^T \exp(r^s(i)/k)}, r^t(i) = \frac{\mathcal{L}_{MTL}^t(i)}{\mathcal{L}_{STL}^t(i)}, t = 1, \dots, T,$$

where  $k$  serves as the temperature hyperparameter to control this task weighting process, and  $i$  represents the iteration index.

Response-Level Multi-Task Self-Training (MuST) (Ghiasi et al., 2021) first trains<sup>5</sup> the classification, detection, and segmentation teacher models from scratch on ImageNet (J. Deng et al., 2009; Russakovsky et al., 2015)/JFT-300M (C. Sun et al., 2017), Objects365 (Shao et al., 2019), and COCO (Kirillov et al., 2019), respectively. The knowledge is then transferred from these specialized teachers to a general-purpose student model via pseudo-labeling. Fig. 8f shows us an overview of MuST, every image in the shared dataset has supervision for all tasks, either supervised or pseudo labels. To balance these loss functions are tricky (See § 2.2.5) and MuST adopts  $w_i = b^s lr_i^t / (b^t lr^s)$  (Goyal et al., 2017) for ImageNet experiments, where  $b$  denotes the batch size,  $lr$  denotes the learning rate, the superscript indicates the student or teacher, and the total loss of MTL is defined as  $\mathcal{L}_{total} = \sum_i w_i \mathcal{L}_i$ . For JFT300M, the algorithm in Kendall et al. (2018) was used to learn  $w_i$  for each task. For depth loss, the weight  $w_i$  was chosen by a parameter sweep. It has been validated that MuST can both rival supervised STL and enhance transfer learning performance.

CrossDistil (Cross-Task Knowledge Distillation) (C. Yang et al., 2022) proposes a recommender framework that can transfer the fine-grained ranking knowledge about user's preference towards items, as shown in Fig. 8h. To facilitate fine-grained ranking, the training samples are divided into multiple subsets, taking into account all possible combinations of the tasks. For instance, in a recommender system where two tasks involve predicting "Buy" and "Like" for an item, the potential task combinations include "Buy:1, Like:1", "Buy:1, Like:0", "Buy:0, Like:1", and "Buy:0, Like:0". For simplicity, the division of multiple subsets on two tasks are:

$$(2.49) \quad \begin{cases} \mathcal{D}^{++} = \{(\mathbf{x}_i, y_i^{(1)}, y_i^{(2)}) \in \mathcal{D} | y_i^{(1)} = 1, y_i^{(2)} = 1\}, \mathcal{D}^{+-} = \{(\mathbf{x}_i, y_i^{(1)}, y_i^{(2)}) \in \mathcal{D} | y_i^{(1)} = 1, y_i^{(2)} = 0\}, \\ \mathcal{D}^{-+} = \{(\mathbf{x}_i, y_i^{(1)}, y_i^{(2)}) \in \mathcal{D} | y_i^{(1)} = 0, y_i^{(2)} = 1\}, \mathcal{D}^{--} = \{(\mathbf{x}_i, y_i^{(1)}, y_i^{(2)}) \in \mathcal{D} | y_i^{(1)} = 0, y_i^{(2)} = 0\}, \\ \mathcal{D}^+ = \mathcal{D}^{++} \cup \mathcal{D}^{+-}, \mathcal{D}^- = \mathcal{D}^{-+} \cup \mathcal{D}^{--}, \mathcal{D}^+ = \mathcal{D}^{++} \cup \mathcal{D}^{-+}, \mathcal{D}^- = \mathcal{D}^{+-} \cup \mathcal{D}^{--}, \end{cases}$$

where  $\mathbf{x}$  represents the input feature vector from the whole dataset  $\mathcal{D}$ .

We denote by  $\mathbf{x}^{++} \in \mathcal{D}^{++}$  and so forth. The fine-grained ranking considers the corresponding multipartite order  $\mathbf{x}^{++} > \mathbf{x}^{+-} > \mathbf{x}^{-+} > \mathbf{x}^{--}$  instead of bipartite orders, e.g.,  $\mathbf{x}^+ > \mathbf{x}^-$  or  $\mathbf{x}^+ > \mathbf{x}^-$ , which may be contradictory among different tasks. Based on the fine-grained ranking, an augmented loss is introduced for each task as

$$(2.50) \quad \mathcal{L}_{aug} = - \sum_{(\mathbf{x}^{++}, \mathbf{x}^{+-}, \mathbf{x}^{-+}, \mathbf{x}^{--})} [\beta_1 \ln \sigma(\hat{r}_{++>+-}) + \beta_2 \ln \sigma(\hat{r}_{-+>--})] - \sum_{(\mathbf{x}_+, \mathbf{x}_-)} \ln \sigma(\hat{r}_{+>-}),$$

where  $\beta_1$  and  $\beta_2$  are two hyper-parameters to balance the importance of pair-wise ranking relations and  $\hat{r}$  is the logit value before the sigmoid function  $\sigma$ . Additionally,  $\hat{r}_{++>--} = \hat{r}_{++} - \hat{r}_{--}$  and so forth. In contrast, the original regression-based loss function for each task is

$$(2.51) \quad \mathcal{L}_{CE} = - \sum_{\mathbf{x}_i \in \mathcal{D}} [y_i \ln \sigma(\hat{r}_i) + (1 - y_i) \ln(1 - \sigma(\hat{r}_i))].$$

---

<sup>5</sup>Pre-trained checkpoints are also recommended to alleviate computational burdens.

Based on Eqs. (2.50) and (2.51), CrossDistil regards the learning task of augmented loss as teachers and the learning task of regression-based loss as students, the distillation loss for each of task is

$$(2.52) \quad \mathcal{L}_{\text{KD}} = - \sum_{x_i \in \mathcal{D}} [\sigma(\tilde{r}_i/\tau) \ln \sigma(\hat{r}_i/\tau) + (1 - \sigma(\tilde{r}_i/\tau)) \ln(1 - \sigma(\hat{r}_i/\tau))],$$

where  $\tilde{r}_i$  is learned and calibrated from Eq. (2.50), and an error correction mechanism is applied to ensure its alignment with the hard label  $y_i$ . The original regression loss and knowledge distillation loss contribute to the learning of students for multiple tasks as

$$(2.53) \quad \mathcal{L}_{\text{MT}} = \sum_{t=1}^T [(1 - \alpha^{(t)}) \mathcal{L}_{\text{CE}}^{(t)} + \alpha^{(t)} \mathcal{L}_{\text{KD}}^{(t)}],$$

where  $\alpha^{(t)}, t = 1, \dots, T \in [0, 1]$  is a hyper-parameter to balance two loss functions. In this manner, by distilling the fine-grained ranking of task combinations, cross-task knowledge is effectively transferred.

**2.2.4. Cross-Task Attention.** Attention mechanism (Brauwers & Frasincar, 2021; M.-H. Guo et al., 2022; Niu et al., 2021) has been one of the most crucial concepts in RNNs, CNNs, and Transformers over the past decade in deep learning. Generally, attention is an information aggregation technique inspired by a human recognition system that tends to prioritize part of local regions over others when processing rich information. Under MTL settings, features from different tasks are more abundant than in STL, thus leading to a natural integration of the attention mechanism. Cross-task attention (Brüggemann et al., 2021), encoding task-aware features into cross-task queries, can perform task-association via refinement of multi-source features. Unlike feature fusion methods (Y. Gao et al., 2019; Misra et al., 2016; Ruder et al., 2019) that propagate task-shared information among different task-specific branches, cross-task attention calculates what/how to share based on cross-task comparison between source tasks and target task. Considering the "morphological" aspect, the hard compartmentalization effect caused by a block-structured communication matrix in feature fusion methods could preserve the interference of features in some cases for tasks. This dilemma could be alleviated with a soft, learnable form of task-aware feature attention. Early works (Brüggemann et al., 2021; S. Liu, Johns, & Davison, 2019; D. Xu et al., 2018; Z. Zhang et al., 2019; L. Zhou et al., 2020) build naïve attention modules (e.g., sigmoid function or inner product) to refine feature affinity or capture relational contexts across tasks, and then locate/diffuse features according to the attention map. PAD-Net (D. Xu et al., 2018) and MTAN (S. Liu, Johns, & Davison, 2019) select attentive features via an attention mask after the sigmoid activation. PAP (Z. Zhang et al., 2019) and PSD (L. Zhou et al., 2020) iteratively diffuse features based on a cross-task affinity matrix. MTI-Net (Vandenhende, Georgoulis, & Van Gool, 2020) first considers task interactions at multiple scales using both Sigmoid function and squeeze-and-excitation block (J. Hu et al., 2018).

Transformer-based works exploit long-range dependencies using self-attention mechanisms.

### Remarks

- (i) Cross-task attention is a lightweight module that can leverage source-target pairwise similarity to refine task-specific features.

**Feature Filtering.** Multi-Task Guided Prediction-And-Distillation Network (PAD-Net) (D. Xu et al., 2018) utilizes the predictions from hierarchical auxiliary tasks as multi-modal inputs to distill knowledge for the final tasks. As shown in Fig. 8i, the framework of PAD-Net, a hard parameter sharing-based encoder, extracts common feature maps that can be used for different tasks, and then the decoder for each auxiliary task generates intermediate predictions for the usage of multi-modal distillation. The source paper proposes three distillation modules to incorporate useful multi-modal

information for the final tasks. Suppose the feature maps from  $s$ -th task at  $l$ -th layer is denoted as  $\mathbf{X}_l^s \in \mathbb{R}^{H \times W \times C}$ ,  $s = 1, \dots, T$ , which are transformed from predictions of  $s$ -th task via convolutional layers. The output feature maps for the usage of  $t$ -th task after the multi-modal distillation is represented as  $\mathcal{X}_{l+1}^{o,t}$ .

The first way to perform cross-modal distillation is a naïve concatenation via  $\mathcal{X}_{l+1}^o = [\mathcal{X}_l^1, \dots, \mathcal{X}_l^T] \in \mathbb{R}^{H \times W \times TC}$ , which is then fed into the separate decoders for each task. Differently, the second way refines feature  $\mathcal{X}_l^t$  via passing knowledge from other tasks as below:

$$(2.54) \quad \mathcal{X}_{l+1}^{o,t} = \mathcal{X}_l^t + \sum_{s \neq t}^T CONV_{\mathcal{W}^{s \rightarrow t}}(\mathcal{X}_l^s), t = 1, \dots, T,$$

where  $\mathcal{W}^{s \rightarrow t}$  denotes the weight tensor of convolutions that maps the  $s$ -th task to the  $t$ -th task. Furthermore, the third way utilizes the sigmoid function to filter the passing knowledge, which learns an attention map  $\mathbf{G}^t$  for the  $t$ -th task as follows:

$$(2.55) \quad \mathbf{G}^t = \sigma(CONV_{\mathcal{W}^t}(\mathcal{X}_l^t)), t = 1, \dots, T.$$

Then the knowledge is filtered via this attention map as follows:

$$(2.56) \quad \mathcal{X}_{l+1}^{o,t} = \mathcal{X}_l^t + \sum_{s \neq t}^T \mathbf{G}^t \odot CONV_{\mathcal{W}^{s \rightarrow t}}(\mathcal{X}_l^s), t = 1, \dots, T.$$

After the multi-modal distillation, the distilled feature maps are up-sampled for the final pixel-level prediction tasks.

Multi-Task Attention Network (MTAN) (S. Liu, Johns, & Davison, 2019) presents a novel MTL architecture based on task-specific feature-wise attention, while global features are shared across different tasks. Suppose the shared global features are denoted by  $\mathcal{X}_l$  at the  $l$ -th layer, and the features learned from task  $t$  are denoted by  $\mathcal{X}_l^t$ . Then the feature-wise attention on the global feature pool is computed as follows:

$$(2.57) \quad \mathcal{X}_{l+1}^t = \sigma(\mathcal{X}_l^t) \odot \mathcal{X}_l,$$

where  $\mathcal{X}_{l+1}^t$  is then concatenated with the features from the global pool again and fed into the task-specific convolution blocks. The attention map  $\sigma(\mathcal{X}_l^t)$  is learned in an end-to-end fashion as a parameter-free activation function.

To make the learning process more balanced between different tasks, S. Liu, Johns, and Davison (2019) also suggests a simple yet effective Dynamic Weight Average (DWA) strategy (See § 2.2.5) to adjust losses according to their magnitudes in different epochs.

Multi-Scale Task Interaction Networks (MTI-Net) (Vandenhende, Georgoulis, & Van Gool, 2020) aggregates multi-modal features at different scales from the decoder. As shown in Fig. 8k, features at each scale are transformed and distilled by the feature propagation module and multi-modal distillation, respectively. This allows the model to capture task interactions at multiple scales. As the higher resolution scales have a limited receptive field, low-quality task-related features are presented. Simple upsampling and passing of task-related features from lower scales to higher scales (Ronneberger et al., 2015) inspire the design of the Feature Propagation Module (FPM). In this manner, features from different tasks at each scale are harmonized via the traditional convolutions and activation functions. To obtain the task-attentive features, a Sigmoid function along the task dimension is inserted to generate a task attention mask. To remedy the negative transfer among unrelated tasks, a per-task channel gating mechanism (SE, i.e. Squeeze-And-Excitation module (J. Hu et al., 2018)) is used to refine the shared representations.

Furthermore, suppose the feature maps for the task  $s$  at scale  $l$  ( $\in \{1/4, 1/8, 1/16, 1/32\}$ ) represented by  $\mathcal{X}_l^s, s = 1, \dots, T$ , then the per-scale multi-modal distillation process for task  $t$  is repeated

as follows:

$$(2.58) \quad \mathcal{X}_l^t = \mathcal{X}_l^t + \sum_{s \neq t} \sigma( \text{CONV}_{\mathcal{W}_l^{s \rightarrow t}}(\mathcal{X}_l^s)) \text{CONV}_{\hat{\mathcal{W}}_l^{s \rightarrow t}}(\mathcal{X}_l^s), t = 1, \dots, T,$$

where the Sigmoid function  $\sigma$  produces a spatial-wise attention mask to filter the features at different scales.  $\mathcal{W}_l^{s \rightarrow t}$  and  $\hat{\mathcal{W}}_l^{s \rightarrow t}$  denote the weights to map features before attention. The FPM and multi-scale multi-modal distillation result in distilled cross-task features at every scale, which are then fed into the final aggregation module. The predictions are based on decoding these final representations via a task-specific head for each task.

Feature Diffusion. Pattern-Affinitive Propagation (PAP) (Z. Zhang et al., 2019) builds a cross-task affinity matrix based on a spatial-wise attention mechanism and then iteratively diffuses features on each of the tasks to refine affinitive patterns among tasks. The detailed architecture is shown in Fig. 81. Suppose the feature maps before the computing of task-specific affinity matrix are denoted by  $\mathcal{X}_l^t \in \mathbb{R}^{H \times W \times C}$ , the affinity matrix for each task is computed using the inner product between each pair of spatial-wise feature vector with the length of  $C$ :

$$(2.59) \quad \mathbf{X}_l^t = \text{RESHAPE}(\mathcal{X}_l^t) \in \mathbb{R}^{HW \times C}, \mathbf{M}^t = \mathbf{X}_l^t \mathbf{X}_l^{t \top} \in \mathbb{R}^{HW \times HW}, t = 1, \dots, T,$$

where  $\text{RESHAPE}(\cdot)$  is used to preserve the channel dimension. If the affinity matrix of each task is weighted by a learnable parameter  $\alpha_t$  ( $t = 1, \dots, T$ , and  $\sum_{t=1}^T \alpha_t = 1$ ), then the final affinity matrix for the task  $t$  can be adaptively combined as follows:

$$(2.60) \quad \hat{\mathbf{M}}^s = \sum_{t=1}^T \alpha_t^s \mathbf{M}^t, s = 1, \dots, T,$$

which is an adaptive combination process that can propagate the cross-task affinitive patterns for the target  $s$ -th task. Furthermore, the cross-task affinitive patterns are used to iteratively diffuse features for each task:

$$(2.61) \quad \mathbf{X}_l^t(i+1) = \hat{\mathbf{M}}^t \cdot \mathbf{X}_l^t(i), t = 1, \dots, T, i = 0, 1, \dots, i_{\max},$$

where  $i$  denotes the diffusion step. In general, the multi-step iterative diffusion process propagates the affinity information best. Suppose the maximum of step is  $i_{\max}$ , finally the feature maps in the next layer are computed as follows:

$$(2.62) \quad \mathbf{X}_{l+1}^t = \beta \cdot \mathbf{X}_l^t(i_{\max}) + (1 - \beta) \cdot \mathbf{X}_l^t(0), \mathcal{X}_{l+1}^t = \text{RESHAPE}(\mathbf{X}_{l+1}^t) \in \mathbb{R}^{H \times W \times C}, t = 1, \dots, T,$$

where  $\beta$  is a hyperparameter to control the feature consistency.

Pattern-Structure Diffusion (PSD) (L. Zhou et al., 2020) utilizes a shared CNN encoder to extract feature maps that can be fed into the task-specific decoders, where the pattern structures are distilled within intra-task and across inter-task. As shown in Fig. 8m, the intra-task PSD is used to transmit pattern structure within each task to enhance the task-specific patterns and then connect with inter-task PSD to correlate relations of pattern structures across different tasks. Without loss of generality, we assume a  $l \times l$  patch cropped at each position of feature maps  $\mathcal{X} \in \mathbb{R}^{H \times W \times C}$  as  $\mathcal{X}_{P_i} \in \mathbb{R}^{l \times l \times C}$ , where  $P_i$  means the pattern at position  $i$ . Then the pattern structure can be defined from the KNN graph on  $l \times l$  points within  $\mathcal{X}_{P_i}$  as follows:

$$(2.63) \quad [\mathbf{A}_{P_i}]_{j,k} = \exp \{-\|\text{RESHAPE}(\mathcal{X}_{P_i})_j - \text{RESHAPE}(\mathcal{X}_{P_i})_k\|_2^2 / \tau^2\}, i = 1, \dots, HW, j, k = 1, \dots, l^2,$$

where  $\tau$  is a fixed hyper-parameter set by user. To make pattern structure at different scale comparable,  $\mathbf{A}_{P_i}$  is further normalized as follows:

$$(2.64) \quad \mathbf{A}_{P_i} \leftarrow \mathbf{A}_{P_i} / (\mathbf{1}^\top \mathbf{A}_{P_i} \mathbf{1}).$$

Then the intra-task PSD can be formulated as a recursive process:

$$(2.65) \quad [RESHAPE(\mathcal{X}^{i+1})]_j = [RESHAPE(\mathcal{X}^i)]_j + \beta \sum_{k \in \mathcal{N}(v_j)} \mathbf{A}_{j,k} \times [RESHAPE(\mathcal{X}^i)]_k,$$

where  $\mathbf{A}$  denotes the pattern structure of the whole feature map,  $\mathcal{N}(v_j)$  is the neighbor set of the target pixel  $v_j$ , and  $\beta$  is a fixed hyper-parameter to control the residual connection. The iteration above contains multiple steps to guarantee that each local pattern is spread into distant regions, which is a diffused process.

To achieve cross-task pattern-structure propagation, inter-task PSD transfers the patterns from other tasks as follows:

$$(2.66) \quad \begin{aligned} [RESHAPE(\mathcal{X}^{(t)})]_j &= [RESHAPE(\mathcal{X}^{(t)})]_j + \sum_{s \neq t} \sum_{k \in \mathcal{N}(v_j)} \beta_{s \rightarrow t} \mathbf{A}_{j,k}^{s \rightarrow t} \times [RESHAPE(\mathcal{X}^{(t)})]_k, \\ \text{s.t. } \mathbf{A}_{P_i}^{s \rightarrow t} &= \mathbf{A}_{P_i}^{(t)} \odot \mathbf{A}_{P_i}^{(s)} / [\mathbf{1}^\top (\mathbf{A}_{P_i}^{(t)} \odot \mathbf{A}_{P_i}^{(s)}) \mathbf{1}], s, t = 1, \dots, T, \end{aligned}$$

where  $\{\mathbf{A}_{P_i}^{ts}\}_{s \neq t}$  represent the transferred pattern-structures from task  $s$  to the target task  $t$ . In this manner, the PSD method distills feature similarity across different tasks.

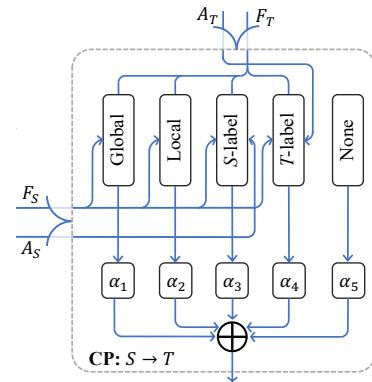
Soft Attention. Attentive Single-Tasking of Multiple Tasks (ASTMT) (Maninis et al., 2019) argues the dilemma that the critical information from one task to another could be a nuisance while inferring multiple tasks together. ASTMT addresses it by single-tasking, a strategy that executes one task at a time instead of inferring all of them simultaneously. Technically, every task shares a backbone network in a hard manner but adapts its specificity with residual adapter (RA) branches, which is shown in Fig. 8n. Suppose the RA operation is represented by  $RA_t$  for the  $t$ -th task, and its original residual skip connection is  $R$ . Then the single-tasking process by RA is calculated as below:

$$(2.67) \quad \mathcal{X}_{l+1}^t = \mathcal{X}_l^t + R(\mathcal{X}_l^t) + RA_t(\mathcal{X}_l^t), t = 1, \dots, T,$$

where  $R$  denotes the residual connection that is not influenced by the task.  $RA_t$  can be naïve bottleneck convolutions or transformed to an attentive block  $SE_t$  (e.g. SE-ResNet block (J. Hu et al., 2018)). In order to address the limitation of this adaptation failing to disentangle the shared and task-specific space, a GRadiEnt Adversarial Training (GREAT) process (Sinha et al., 2018) is introduced for different tasks to ensure that the shared backbone learns the shared representations and maintains this quality during the single-tasking process. More details of multi-task adversarial training are shown in § 2.2.7.

Adaptive Task-Relational Context (ATRC) module (Brüggemann et al., 2021) enables global cross-task and local spatial-wise attention mechanisms to refine each task prediction, which is a general module that can be applied to any backbones across any supervised dense prediction tasks. The ATRC refinements begin with a hard-parameter sharing encoder, of which each task head can generate task-specific features  $\mathcal{X}_t$  and auxiliary predictions  $\mathcal{P}_t$ , where  $t = 1, \dots, T$ . Specifically, the features  $\mathcal{X}_t$  of each target task  $\mathcal{T}_t$  is refined by attending to the features  $\mathcal{X}_s$  of every available task  $\mathcal{T}_s$ ,  $s \in \{1, \dots, T\}$  within a separate Context Pooling (CP) block. As shown in Fig. 8o, the original features  $\mathcal{X}_t$  and refined features  $\{\mathcal{X}_{s \rightarrow t}\}_{s=1}^T$  are combined to predict the target task  $\mathcal{T}_t$ .

There are three categories of context information (global context, local context, and label context) to be learned via refining



**Figure 12.** The computational details of Context Pooling (CP).

features from the source task to the target task. The detailed illustration can be observed in Fig. 12 positioned to the right. Each CP block accepts the features  $\mathcal{X}_s, \mathcal{X}_t$  and predictions  $\mathcal{P}_s, \mathcal{P}_t$  from the source task and target task, respectively.  $\mathcal{X}_t$  and  $\mathcal{X}_s$  are transformed into queries  $\mathbf{Q}$ , keys  $\mathbf{K}$  and values  $\mathbf{V}$  (flattening along the spatial dimension and preserving channel dimension) as below:

$$(2.68) \quad \begin{aligned} \mathbf{Q} &= \text{RESHAPE}(\text{CONV}_{\mathcal{W}_q}(\mathcal{X}_t)), \mathbf{K} = \text{RESHAPE}(\text{CONV}_{\mathcal{W}_k}(\mathcal{X}_s)), \\ \mathbf{V} &= \text{RESHAPE}(\text{CONV}_{\mathcal{W}_v}(\mathcal{X}_s)), \end{aligned}$$

where  $\text{CONV}_*(\cdot)$  is a  $1 \times 1$  CONV-BN-ReLU operation, and  $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{HW \times C}$ . In the attention of global context, a target feature value  $v_i$  at position  $i$  is substituted with

$$(2.69) \quad \mathbf{v}'_i = \sum_{j=1}^L \text{sim}(\mathbf{q}_i, \mathbf{k}_j) \mathbf{v}_j / \sum_{j=1}^L \text{sim}(\mathbf{q}_i, \mathbf{k}_j), i = 1, \dots, L,$$

where  $L$  denotes the number of total pixels (i.e. feature values) and  $\text{sim}(\cdot, \cdot)$  denotes an arbitrary similarity function. For the local context attention, let us denote by  $\mathcal{N}_p(i)$  the 2D spatial neighborhood of target pixel at position  $i$  with the patch extent  $p$ , then the spatial-wise local attention is formulated as below:

$$(2.70) \quad \mathbf{v}'_i = \sum_{j \in \mathcal{N}_p(i)} \text{softmax}(\mathbf{q}_i \mathbf{k}_j / \sqrt{C}) \mathbf{v}_j, i = 1, \dots, L,$$

where  $C$  is the channel dimension of  $\mathbf{K}$ . For the  $T$ -label context and  $S$ -label context defined in the label space that is partitioned into a set of disjoint label regions. The aim is to find a prototypical representation for each pixel. Suppose  $\mathcal{P}_t \in HW \times R_t$ , where each entry of the last dimension indicates the degree that a pixel belongs to a label region  $r \in \{1, \dots, R_t\}$ . For the  $T$ -label context, the keys  $\mathbf{K}$  and values  $\mathbf{V}$  are calculated via the region prototypes as below:

$$(2.71) \quad \mathbf{K} = \text{CONV}_{\mathcal{W}_k}(\hat{\mathcal{P}}_t^\top \text{RESHAPE}(\mathcal{X}_s)), \mathbf{V} = \text{CONV}_{\mathcal{W}_v}(\hat{\mathcal{P}}_t^\top \text{RESHAPE}(\mathcal{X}_s)),$$

where  $\hat{\mathcal{P}}_t$  denotes the softmax normalization over the spatial dimension, and the matrix  $\hat{\mathcal{P}}_t^\top \text{RESHAPE}(\mathcal{X}_s) \in \mathbb{R}^{R_t \times C}$  represents the region prototypes. Alternatively,  $\mathcal{P}_t$  is substituted with the source task prediction maps  $\mathcal{P}_s$  in the  $S$ -label context. The outputs of both are attention-weighted combinations of features  $\mathbf{v}$ :

$$(2.72) \quad \mathbf{v}' = \text{softmax}(\mathbf{q} \mathbf{k}^\top / \sqrt{C}) \mathbf{v}.$$

Deformable Mixer Transformers (DeMT) (L. Zhang et al., 2023) is an encoder-decoder architecture that combines the merits of deformable CNNs (J. Dai et al., 2017; X. Zhu et al., 2019) and attention-based ViT (Dosovitskiy et al., 2021) to model multiple tasks, the details are shown in Fig. 8p. The encoder, aka the deformable mixer in L. Zhang et al. (2023), is aware of feature mixing across channels through  $1 \times 1$  convolutions and captures the deformable spatial features through learnable offsets. After task-specific features are learned by the encoder part, the task-aware transformer decoder first applies the task interactions based on the attention mechanism (MHSA + MLP) and then constructs the task query block to decode the task awareness features for each task. Suppose the transformer operator inside the task interaction block can be abstracted as

$$(2.73) \quad \mathcal{X}_{l+1} = \text{MHSA}_{\text{inter}}(q = LN(\mathcal{X}_l), k = LN(\mathcal{X}_l), v = LN(\mathcal{X}_l)),$$

where  $LN$  denotes the layer norm on fused feature  $\mathcal{X}_l$ , and the subscripts  $l$  and  $l+1$  denote the feature index before and after the task interaction block, respectively. To decode task awareness in the task query block, another transformer involves task-specific query before  $\text{MHSA}_{\text{inter}}$  (i.e.,  $\mathcal{X}_l^t$ ):

$$(2.74) \quad \mathcal{X}_{l+2}^t = \text{MHSA}_{\text{query}}(q = LN(\mathcal{X}_l^t), k = LN(\mathcal{X}_{l+1}), v = LN(\mathcal{X}_{l+1})), t = 1, \dots, T,$$

where the subscript  $l + 2$  denotes the feature index after the task query block.

### Remarks

- (i) Knowledge distillation can utilize and transfer interpretable patterns across multiple tasks, resulting in meaningful principles that can provide guidance for architectural design.
- (ii) Knowledge distillation has the capability to aggregate refined features from multiple tasks at various scales, thereby enhancing task generalization ability and significantly improving performance.
- (iii) Knowledge distillation allows for the creation of smaller and more efficient student models on target tasks. The distilled knowledge helps compress the complex teacher model into a more lightweight student model while retaining a comparable level of performance.
- (iv) Knowledge distillation enables the transfer of knowledge across tasks, even if they are different or loosely related. This flexibility allows for leveraging insights from related tasks to enhance the learning process, resulting in better performance on each individual task.
- (v) The overall performance heavily depends on the quality and capabilities of the teacher model. If the teacher model is not well-trained or lacks expertise in the specific tasks, the knowledge distillation process may not be effective, limiting the potential benefits.
- (vi) Implementing knowledge distillation adds extra computational complexity that often involves the processes of training, transferring, and fine-tuning, thus inevitably being time-consuming and resource-intensive.

**2.2.5. Scalarization Approach.** One of the most popular methods to solve multi-task learning problems is the scalarization approach, which formulates the problem as a linear combination of loss functions of different tasks (Z. Chen et al., 2018; Kendall et al., 2018; S. Liu, Johns, & Davison, 2019; Senushkin et al., 2023) as

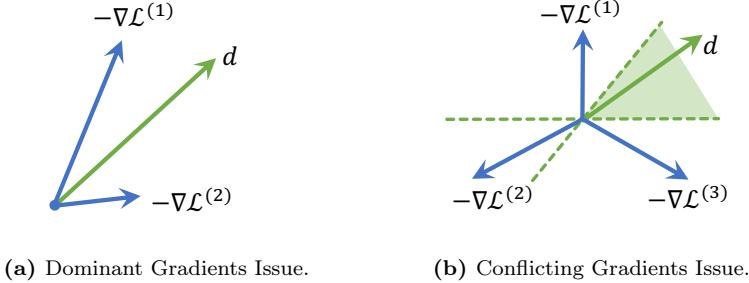
$$(2.75) \quad \min_{\mathbf{W}} \mathcal{L}_{\text{total}}(\mathbf{W}) = \sum_{t=1}^T \alpha^{(t)} \mathcal{L}^{(t)}(\mathbf{W})$$

where  $\{\alpha^{(t)}\}_{t=1}^T \subset \mathbb{R}_+$  are the tasks' weights and are used to encode preferences over different tasks.  $\mathbf{W}$  is the model parameter and  $\{\mathcal{L}^{(t)}\}_{t=1}^T$  are loss functions for different tasks. In each loss function  $\mathcal{L}^{(t)}$ , we drop the dependency on training samples  $\{\mathbf{X}^t, \mathbf{y}^t\}$  to avoid cluttered notations.

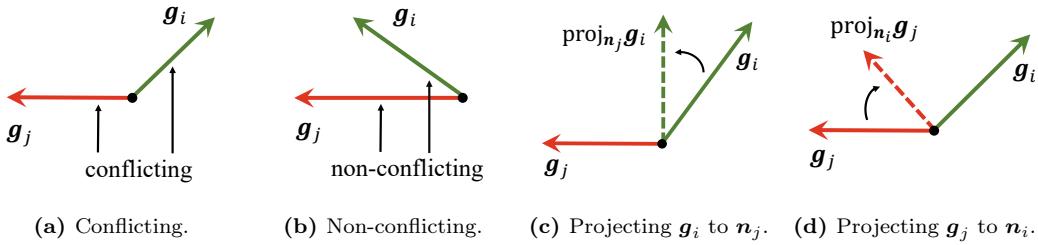
Gradient-based methods are perhaps the most popular choices to solve Eq. (2.75), whose update rule of  $\mathbf{W}$  takes the form of  $\mathbf{W} \leftarrow \mathbf{W} + \eta \mathbf{d}$ , where  $\eta > 0$  is the learning rate and  $\mathbf{d}$  is the search direction.  $\mathbf{d}$  is a function of  $\{\alpha^{(t)} \nabla \mathcal{L}^{(t)}\}_{t=1}^T$ , for example,  $\mathbf{d} = -\sum_{t=1}^T \alpha^{(t)} \nabla \mathcal{L}^{(t)}(\mathbf{W})$ . Aside from the challenge of choosing a proper learning rate  $\eta$ , there are two additional challenges, *dominant gradients* and *conflicting gradients*, see Fig. 13 for an illustration. Dominating gradient issue occurs when the norm of gradients of some tasks' losses are significantly larger than the others, hence the updating direction  $\mathbf{d}$  are biased towards to tasks with larger gradient norm. Conflicting gradients issue arises when one makes progress in one task, the performance of another task is degraded.

In the remainder of this section, we review some works with different philosophies to address dominant and conflicting gradients' challenges. These methods can be roughly characterized as *gradient correction* approach, where transformations are made to gradients to address the conflicting gradients issue and *dynamic weighting*, where  $\{\alpha^{(t)}\}$  are updated in each iteration to address the dominant gradients issue.

Gradient Correction. Projecting Conflicting Gradients (PCGrad) (T. Yu et al., 2020) proposes to mitigate the conflicting gradients issue by projecting the conflicting gradients in the orthogonal subspace. Formally, PCGrad (T. Yu et al., 2020) defines two gradients  $(g_i, g_j)$  to be conflicting if  $g_i^T g_j < 0$ . To address this issue, instead of forming the search direction as  $\mathbf{d} = -(g_i + g_j)$ , PCGrad



**Figure 13.** (a)dominant gradients issue. The update direction  $\mathbf{d}$  is dominated by the negative gradient of the loss of task 1. (b)conflicting gradients issue. When  $\{\alpha^{(t)}\}_{t=1}^3$  are not properly set, the update direction  $\mathbf{d}$  can decreases the loss of task 1 and 3 while increases the loss of task 2. Therefore, the pefromance on the task 2 is compromised.



**Figure 14.** Demonstration of gradient projection technique used in T. Yu et al. (2020).

suggested using  $\mathbf{d} = -(\mathbf{Proj}_{n_j}(g_i) + \mathbf{Proj}_{n_i}(g_j))$ , where  $n_i^T g_j = 0$  and  $n_j^T g_i = 0$  and  $\mathbf{Proj}$  is the Euclidean projection operator. See Fig. 14 for an illustration. This method, from the perspective of multi-objective optimization perspective (which will be discussed in the next section), is a particular choice of choosing a common descent direction. Gradient sign Dropout (GradDrop)(Z. Chen, Ngiam, et al., 2020) attributed conflicts to the differences in the signs of gradients along each coordinate direction. Motivated by the dropout, a probabilistic masking procedure is proposed to keep only gradients consistent in signs in each update. Conflict-Averse Gradient descent (CAGrad) (B. Liu et al., 2021) proposes to mitigate gradient conflicts by solving the problem

$$(2.76) \quad \max_{\mathbf{d}} \min_{t \in [T]} \nabla \mathcal{L}^{(t)}(\mathbf{W})^T (-\mathbf{d}) \text{ s.t. } \|\mathbf{d} - \nabla \mathcal{L}_{\text{total}}(\mathbf{W})\| \leq c \|\nabla \mathcal{L}_{\text{total}}(\mathbf{W})\|,$$

where  $c > 0$  is a prescribed parameter. The intuition is that  $-\min_{t \in [T]} \nabla \mathcal{L}^{(t)}(\mathbf{W})^T \mathbf{d}$  can be used as the approximated evaluation of the conflict among objectives, and one wants to find the direction  $\mathbf{d}$  that minimizes such a conflict while stays close to the original negative gradient of  $\mathcal{L}_{\text{total}}(\mathbf{W})$ .

Reducing conflicting gradient (Recon) (Shi et al., 2023) empirically observes that PCGrad, GradDrop, and CAGrad (Z. Chen, Ngiam, et al., 2020; B. Liu et al., 2021; T. Yu et al., 2020) can only slightly reduce the occurrence of conflicting gradients (compared to joint-training<sup>6</sup>) in some cases, and in some other cases they even increase the occurrence. Therefore, Recon proposed to analyze parameters in a layer-wise fashion to pinpoint the shared parameters that are most likely to incur conflicting gradients. Concretely, let  $(g_i^k, g_j^k)$  be the gradients of the  $(i, j)$  task pair with respect to the  $k$ th layer's parameters.  $(g_i^k, g_j^k)$  is said to be  $S$ -conflicting if  $s^k := \frac{\langle g_i^k, g_j^k \rangle}{\|g_i^k\| \|g_j^k\|} < S$  for any  $s \in [-1, 0]$ . Recon first trained the models via any gradient-based method with  $E$  epochs, e.g., PCGrad, GradDrop, and CAGrad, and then derived the conflicting scores for each layer over  $E$

<sup>6</sup>The joint-training refers to the case that  $\alpha^{(t)} = 1$  for all  $t \in [T]$  in Eq. (2.75).

epochs to identify the top  $K$  layers with the highest (most negative) conflicting scores. Finally, Recon turned these  $K$  layers' parameters into task-specific parameters and retrained the network from scratch. As pointed out in Shi et al. (2023), while Recon is sensitive to the parameters  $K$  and  $S$ , one only needs to tune them once for a given network architecture.

Dynamic weighting. GradNorm proposed in Z. Chen et al. (2018) suggests to mitigate the dominant gradient issue so that gradients for each task have the proper magnitude. The strategy to adjust  $\{\alpha^{(t)}\}$  is based on the average gradient norm of each task and the relative progress achieved for each task. With this information, GradNorm constructs a reference point at each iteration,  $\{\alpha^{(t)}\}$  was then selected to minimize the  $\ell_1$  distance between the actual gradient of each task and the reference point. Concretely, let  $GN_{\mathbf{W}}^{(t)}(i) = \|\nabla_{\mathbf{W}}\alpha^{(t)}(i)\mathcal{L}^{(t)}(i)\|_2$  be the measure of  $\ell_2$  norm of the  $t$ th task's weighted gradient at iteration  $i^7$ . Next, the averaged gradient norm across all tasks was calculated as  $\overline{GN}_{\mathbf{W}}(i) = \mathbb{E}_t[GN_{\mathbf{W}}^{(t)}(i)]$ . To measure the training progress of each task,  $\tilde{\mathcal{L}}^{(t)}(i) = \frac{\mathcal{L}^{(t)}(i)}{\mathcal{L}^{(t)}(0)}$  was introduced, which inversely proportional to the training rate. Lastly, the relative inverse training rate for task  $t$  can be formulated as  $r^{(t)}(i) = \frac{\mathcal{L}^{(t)}(i)}{\mathcal{L}^{(t)}(0)} / \mathbb{E}_t[\tilde{\mathcal{L}}^{(t)}(i)]$ . The higher value of  $r^{(t)}(i)$  indicates a higher gradient magnitude for task  $t$  at iteration  $i$ , which encourages task  $t$  to learn more quickly. Finally, the weight  $\alpha^{t+1}$  was determined by solving the following problem

$$(2.77) \quad \min_{\{\alpha^{(t)}\}_{t=1}^T} \sum_{t=1}^T \|GN_{\mathbf{W}}^{(t)}(i) - \overline{GN}_{\mathbf{W}}(i) \cdot [r^{(t)}(i)]^\zeta\|_1,$$

where  $\zeta$  is introduced to avoid dramatically different learning dynamics between tasks caused by various task complexity. Inspired by GradNorm, Dynamic Weight Averaging (DWA) is another strategy proposed in S. Liu, Johns, and Davison (2019) to balance the task-specific losses. The updating process of  $\alpha^{(t)}(i)$  is defined as  $\alpha^{(t)}(i) = \frac{\sum_t \alpha^{(t)}(i) e^{r^{(t)}(i-1)/T}}{\sum_{t=1}^T e^{r^{(t)}(i-1)/T}}$  and  $r^{(t)}(i-1) = \frac{\mathcal{L}^{(t)}(i-1)}{\mathcal{L}^{(t)}(i-2)}$ , where  $r^{(t)}(i)$  is the relative progress for the task  $t$  at the iteration  $i$ . Reinforced MTL (RMTL) (S. Liu, 2018, Chapter 3) adjusts  $\{\alpha^{(t)}\}$  using the reinforcement learning strategy and Loss-Balanced Task Weighting. LBTW (S. Liu, Liang, & Gitter, 2019) combines GradNorm and RMTL in a way such that the weights  $\{\alpha^{(t)}\}$  were adapted to both samples and tasks. Impartial MTL (IMTL) (L. Liu et al., 2021) proposes to update  $\{\alpha^{(t)}\}$  in each iteration such that the aggregated gradient  $\sum_{t=1}^T \alpha^{(t)} \nabla \mathcal{L}^{(t)}(\mathbf{W})$  has equal projections onto the raw gradients of individual tasks. It achieves this goal by solving the following linear system (with respect to  $\{\alpha^{(t)}\}$ )

$$\left( \sum_{t=1}^T \alpha^{(t)} \nabla \mathcal{L}^{(t)}(\mathbf{W}) \right)^T \frac{\nabla \mathcal{L}^{(t)}(\mathbf{W})}{\|\nabla \mathcal{L}^{(t)}(\mathbf{W})\|} = \left( \sum_{t=1}^T \alpha^{(t)} \nabla \mathcal{L}^{(t)}(\mathbf{W}) \right)^T \frac{\nabla \mathcal{L}^{(1)}(\mathbf{W})}{\|\nabla \mathcal{L}^{(1)}(\mathbf{W})\|}, \text{ for } t \in \{2, \dots, T\}$$

$$\sum_{t=1}^T \alpha^{(t)} = 1.$$

Before solving for  $\{\alpha^{(t)}\}_{t=1}^T$ , IMTL also proposes a heuristic to scale  $\{\mathcal{L}^{(t)}(\mathbf{W})\}$  such that all losses are in the similar scales, which essentially is another scaling of the  $\{\nabla \mathcal{L}^{(t)}(\mathbf{W})\}$ . Achievement-based MTL (Yun & Cho, 2023) suggests defining the weights for each task by measuring the training progress as  $\alpha^{(t)} = (1 - \text{acc}_t / (\text{m} \cdot \text{maxacc}_t))^\gamma$  where  $\gamma > 0$ ,  $\text{m} > 1$ ,  $\text{acc}_t$  and  $\text{maxacc}_t$  are the current training accuracy (trained in the multitask setting) for the task  $t$  and the max training accuracy (trained in the single setting), respectively. And Achievement-based MTL considers using the geometric mean instead of arithmetic mean to define the loss function; namely, it solves  $\min_{\mathbf{W}} \left( \prod_{t=1}^T (L^{(t)}(\mathbf{W}))^{\alpha^{(t)}} \right)^{1/T}$ .

<sup>7</sup>We add addition index  $i$  to indicate their dependence on the iteration counter  $i$ .

Uncertainty Weighting (Kendall et al., 2018) takes a different perspective from the above dynamic weighting approaches. This work assumes there are underlying distributions for different tasks' labels, and different tasks are independent. The final loss function, deriving from the likelihood perspective, takes the same form as Eq. (2.75) with  $\{\alpha_k^{(t)}\}$  being specified as the reciprocal of the variance of each distribution used to modeling each task and loss function. Instead of just optimizing over the parameter  $\mathbf{W}$ , Kendall et al. (2018) optimizes  $\mathbf{W}$  and  $\{\alpha^{(t)}\}$  simultaneously

$$(2.78) \quad \min_{\mathbf{W}, \{\alpha^{(t)}\}_{t=1}^T} \mathcal{L}_{\text{total}}(\mathbf{W}) = \sum_{t=1}^T \alpha^{(t)} \mathcal{L}^{(t)}(\mathbf{W}).$$

At this point, one can observe that all aforementioned works under the dynamic weighting category, excluding Kendall et al. (2018), do not necessarily respect optimization problem formulation in Eq. (2.75) even though they empirically work well in producing useful solutions. Nonetheless, one can also regard the dynamic weighting approach as either solving Eq. (2.78) using different rule-based strategies to update  $\{\alpha^{(t)}\}_{t=1}^T$  or using gradient-based methods to inexactly solve a sequence of problems in the form of Eq. (2.75).

To conclude this section, we point out that there are some works that try to address two issues simultaneously (Javaloy & Valera, 2022; Senushkin et al., 2023). For example, Alignment for MTL (Aligned-MTL (Senushkin et al., 2023) considers the condition number of the linear system  $\mathbf{d} = \mathbf{G}\boldsymbol{\alpha}$  as a measure of the degree of the severeness of both gradient dominance and conflict, where  $\mathbf{G} = [-\nabla \mathcal{L}^{(1)}, \dots, -\nabla \mathcal{L}^{(T)}]$  and  $\boldsymbol{\alpha} = (\alpha^{(1)}, \dots, \alpha^{(T)})^T$ . Therefore, the authors propose to find well-conditioned  $\hat{\mathbf{G}}$  to approximate  $\mathbf{G}$  and, therefore, obtain a refined update direction  $\hat{\mathbf{d}}$ . Concretely, the author proposed to solve  $\min_{\hat{\mathbf{G}}} \|\hat{\mathbf{G}} - \mathbf{G}\|$  s.t.  $\hat{\mathbf{G}}^T \hat{\mathbf{G}} = I$ , by singular value decomposition (SVD) and use the refined direction  $\hat{\mathbf{d}} = \hat{\mathbf{G}}\boldsymbol{\alpha}$  instead of  $\mathbf{d} = \mathbf{G}\boldsymbol{\alpha}$ . The convergence rate of the proposed algorithm is established under the assumption that all loss functions are Lipschitz smooth and bounded below. Although the numerical results are promising, one should be aware of the computation cost of the SVD despite the existence of efficient algorithms (Bondhugula et al., 2006).

### Remarks

- (i) Scalarization approach features in its simplicity as it transforms a multi-objective problem into a single-objective one. Hence, it is easy to implement, and many off-shelf optimizers can be applied.
- (ii) Generally, the scalarization approach has computational efficiency advantages over multi-objective optimization approaches, as will be discussed in the next section.
- (iii) The solution found by the scalarization approach might lack diversity as it could be biased to a solution depending on prescribed weights<sup>a</sup>. Also, it is hard to conduct convergence analysis, especially for the dynamic weighting approach, since it attempts to solve a sequence of problems inexactly.

<sup>a</sup>We characterize the diversity through the Pareto Front, which will be discussed in the next section.

2.2.6. *Multi-objective Optimization (MOO)*.. In contrast to the scalarization approach, which converts different objective functions  $\{\mathcal{L}^{(1)}, \dots, \mathcal{L}^{(T)}\}$  into one aggregated objective function  $\mathcal{L}_{\text{total}}$  and then optimizes it, MOO, aims to *simultaneously* optimizing several objective functions (potentially conflicting). Concretely, MOO aims to solve the following problem

$$(2.79) \quad \min_{\mathbf{W}} \mathcal{L}(\mathbf{W}) = (\mathcal{L}^{(1)}(\mathbf{W}), \dots, \mathcal{L}^{(T)}(\mathbf{W}))^T \quad \text{s.t. } \mathbf{W} \in \mathcal{F},$$

where  $\mathcal{F}$  is the feasible domain for  $\mathbf{W}$  (examples will be given shortly). For a comprehensive background on the MOO topic, we refer readers to Ehrgott (2005); for readers who prefer a quick overview of this subject, we recommend S. Liu and Vicente (2020). Below, we just provide the

minimum backgrounds required to make the exposition accessible to readers with backgrounds in single objective optimization.

We begin with a few concepts that help readers understand the type of solutions that MOO algorithms can normally obtain.

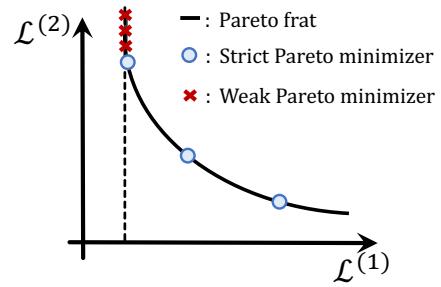
#### Definition 4.

- (1)  $\mathbf{W}^*$  is called a **weak Pareto minimizer** of  $\mathcal{L}$  over  $\mathcal{F}$  if there is no  $\mathbf{W} \in \mathcal{F}$  such that  $\mathcal{L}(\mathbf{W}) < \mathcal{L}(\mathbf{W}^*)$ . Here,  $<$  is the element-wise comparison. The set  $PF(\mathcal{L}) = \{\mathcal{L}(\mathbf{W}^*) \mid \mathbf{W}^* \text{ is a weak Pareto minimizer}\}$  is called the **Pareto front**.
- (2)  $\mathbf{W}^*$  is called a **strict Pareto minimizer** of  $\mathcal{L}$  over  $\mathcal{F}$  if there is no  $\mathbf{W} \in \mathcal{F}$  such that  $\mathcal{L}(\mathbf{W}) \leq \mathcal{L}(\mathbf{W}^*)$  and  $\mathbf{W} \neq \mathbf{W}^*$ . Here,  $\leq$  is the element-wise comparison.
- (3)  $\mathbf{W}^*$  is called a **Pareto stationary point** of  $\mathcal{L}$  over  $\mathcal{F}$  if  $\max_{t=1,\dots,T} (\mathbf{W} - \mathbf{W}^*)^T \nabla \mathcal{L}^{(t)}(\mathbf{W}^*) \geq 0$  for all  $\mathbf{W} \in \mathcal{F}$ . Intuitively, this definition implies that for the objective function, there exists at least one such that there does not exist any feasible direction  $d := \mathbf{W} - \mathbf{W}^*$  to further decrease it.

We give a graphical illustration of all these Pareto-related points in Fig. 15. In Fig. 15, the  $\mathbf{W}^*$ 's that correspond to circles and crosses are Pareto stationary points. However, when  $\{\mathcal{L}^{(t)}(\mathbf{W})\}_{t=1}^T$  are not convex, the Pareto stationary points can generate  $\{\mathcal{L}^t(\mathbf{W}^*)\}$  that are NOT sit on the Pareto font. An analogy for this phenomenon in single objective optimization would be that a stationary point of a nonconvex objective function may not be the global minimum. Due to the nonconvexity nature of neural networks, algorithms considered here (when the convergence analysis is provided), if not all, can only guarantee to find the **Pareto stationary point** instead of the weak/strict Pareto minimizers. However, if additional assumptions like (strong) convexity are assumed, then one can obtain solutions whose objective values are on the Pareto front. In the sequel, we review some works with different strategies to generate the a (set of) Pareto stationary point(s).

The first line of works, e.g., X. Lin et al. (2019), Navon et al. (2022), and Sener and Koltun (2018) were built upon and extended the seminal work, Multiple-Gradient Descent Algorithm (MGDA) (Fliege & Svaiter, 2000) to the neural network settings. The essence of MGDA is, at each iteration, to find a common descent direction  $d$  that decreases all objective functions  $\{\mathcal{L}^{(t)}\}$  simultaneously. If no such direction exists, the algorithm terminates and returns a (set of) Pareto stationary point(s). MGDA constructs the **common descent direction**  $d$  by solving the following optimization problem<sup>8</sup>

$$(2.80) \quad \max_{d \in \mathbb{R}^n} \min_{t=1,\dots,T} (-\nabla \mathcal{L}(\mathbf{W})^{(t)})^T d + \frac{1}{2} \|d\|^2.$$



**Figure 15.** An illustration of weak and strict Pareto minimizers and Pareto front. We emphasize that the circles and crosses on the curve are NOT weak and strict Pareto minimizers. Instead, those  $\mathbf{W}^*$ 's that generate circles and crosses are weak and strict Pareto minimizers, respectively. In this figure, all circles and crosses are Pareto stationary points. We remark that in this example, the Pareto front is convex and continuous. The Pareto front can also be non-convex and/or discontinued, for example, see S. Liu and Vicente (2020, Page 12).

<sup>8</sup>For simplicity, we now only consider the unconstrained case  $\mathcal{F} = \mathbb{R}^n$ ; we will discuss the constrained case  $\mathcal{F} \subset \mathbb{R}^n$  shortly.

In problem (2.80), if we drop the second order term  $\frac{1}{2} \|\mathbf{d}\|^2$ , it intuitively tries to find the search direction  $\mathbf{d}$  that can maximize the minimal progress<sup>9</sup> can be made. The second order term is added to guarantee the uniqueness of the solution of problem (2.80). The solution  $\mathbf{d}$  is known as the *steepest common descent direction* in the optimization literature. In deep neural network applications, however,  $n$  can be of the billion scale, so it is very challenging to solve problem (2.80) directly. Instead of solving (2.80), MGDA-MTL (Sener & Koltun, 2018) considers to the solve the dual problem

$$(2.81) \quad \min_{\beta \in \mathbb{R}^T} \frac{1}{2} \left\| \sum_{t=1}^T [\beta]_t \nabla \mathcal{L}^{(t)}(\mathbf{W}) \right\|^2 \quad \text{s.t.} \quad \sum_{t=1}^T [\beta]_t = 1 \text{ and } [\beta]_t \geq 0 \text{ for all } t,$$

where  $[\beta]_t$  is the  $t$ -th element of the vector  $\beta$ . One can see that the dual problem's dimension reduces to  $T$ , which is usually smaller than  $n$  in several orders of magnitude and can be solved efficiently, e.g., Frank-Wolfe algorithm (Jaggi, 2013) as is used in Sener and Koltun (2018). The solution  $\mathbf{d}^*$  to the problem (2.80) can be recovered by the solution to the problem (2.81)  $\beta^*$  as  $\mathbf{d}^* = -\sum_{t=1}^T [\beta]^*_t \nabla \mathcal{L}^{(t)}(\mathbf{W})$  and the model parameter is updated as  $\mathbf{W} \leftarrow \mathbf{W} + \eta \mathbf{d}^*$  with  $\eta \geq 0$ . With proper assumption, iterates or a subsequence of the iterates converge to a Pareto stationary point. If all  $\{\mathcal{L}^{(t)}\}$  are convex, then the point that the iterates converge to is not only a Pareto stationary point but also is a weak Pareto minimizer, meaning its corresponding function value vector is on the Pareto front. MGDA-MTL further developed an efficient variant of MGDA when the neural network's parameters can be decoupled as  $\mathbf{W} = (\mathbf{W}^{\text{shared}}, \mathbf{W}^{(1)}, \dots, \mathbf{W}^{(T)})$ , and the common descent direction only needs to be found with respect to the  $\mathbf{W}^{\text{shared}}$  part. Another work, Nash-MTL (Navon et al., 2022), formulates the problem of finding the common descent direction as a bargain game. Concretely, the common descent direction  $\mathbf{d}$  is obtained as  $\mathbf{d} = G\beta$  where  $G = [\nabla \mathcal{L}^{(1)}(\mathbf{W}), \dots, \nabla \mathcal{L}^{(T)}(\mathbf{W})]$  and  $\beta$  is a solution to the linear system<sup>10</sup>  $G^T G\beta = 1/\beta$ .

One potential issue with MGDA-MTL and Nash-MTL, more generally, MGDA-type methods are the algorithms that can only produce one Pareto stationary point instead of a set of Pareto stationary points. Producing a set of solutions has the advantage of allowing practitioners to choose one solution that best fits their needs. To address this issue, Pareto-MTL (X. Lin et al., 2019) considers restricting the solution  $\mathbf{W}^*$  produced by one run of MGDA in a certain domain such that  $\{\mathcal{L}^{(t)}(\mathbf{W}^*)\}$  is on a restricted region of the Pareto front<sup>11</sup>. By carefully crafting the regions, the algorithm can generate  $K$  well-separated solutions on the Pareto front. Specifically, assuming  $\mathcal{L}(\mathbf{W}) \geq 0$  for all  $\mathbf{W}$  and that a set of  $K$  preference vectors  $\{\mathbf{u}_k\}_{k=1}^K$  are given, Pareto-MTL considered to solve the  $K$  problems in parallel, where the  $k$ th problem is

$$(2.82) \quad \min_{\mathbf{W}} \mathcal{L}(\mathbf{W}) = (\mathcal{L}^{(1)}(\mathbf{W}), \dots, \mathcal{L}^{(T)}(\mathbf{W}))^T \quad \text{s.t.} \quad \mathbf{u}_i^T \mathcal{L}(\mathbf{W}) \leq \mathbf{u}_k^T \mathcal{L}(\mathbf{W}) \text{ for all } i \in [K] \setminus \{k\}.$$

where  $[K] = \{1, \dots, K\}$ . Intuitively, the constraints in Eq. (2.82) force the solution  $\mathcal{L}(\mathbf{W})$  to stay close to  $\mathbf{u}_k$  in the angular space. The problem (2.82) is more challenging than problem (2.79) since it has  $K - 1$  nonlinear inequality constraints. Consequently, problem (2.80) is changed to account for these additional  $K - 1$  constraints. For more details, we refer readers to Eq. (14) in X. Lin et al. (2019). However, as pointed out in exact Pareto Optimal Search (EPO search) (Mahapatra & Rajan, 2020), Pareto-MTL does not guarantee that the solution matches the exact preference, and  $K$  needs to grow exponentially fast as  $T$  increases. Therefore EPO search re-designs the constraints

<sup>9</sup>The progress is measured by the difference between of  $\mathcal{L}^{(t)}(\mathbf{W})$  and the first order Taylor approximation of  $\mathcal{L}^{(t)}$  at  $\mathbf{W} + \mathbf{d}$ .

<sup>10</sup> $1/\beta$  is the element-wise reciprocal.

<sup>11</sup>This is realizable only if the solution is a weak Pareto minimizer.

and develops a new algorithm to search for the exact solution that matches the preference. Formally, EPO search proposes to solve

$$(2.83) \quad \min_{\mathbf{W}} \mathcal{L}(\mathbf{W}) = (\mathcal{L}^{(1)}(\mathbf{W}), \dots, \mathcal{L}^{(T)}(\mathbf{W}))^T \quad \text{s.t. } \mathcal{L}^{(1)}(\mathbf{W})[u]_1 = \dots = \mathcal{L}^{(T)}(\mathbf{W})[u]_T,$$

where  $u \in \mathbb{R}^T$  is the user-specified preference vector,  $[\cdot]_i$  takes the  $i$ th elements, and  $\mathcal{L}^{(t)}(\mathbf{W})$  is non-negative for all  $t \in [T]$ . Geometrically, this constraint enforces the solution  $\mathbf{W}^*$  in a way such that the ray  $(1/u_1, \dots, 1/u_T)$  intersects with the Pareto front at  $\mathcal{L}(\mathbf{W}^*)$ . Given an iterate  $\mathbf{W}$ , EPO search forms a search direction that tries to balance the constraint violation (the new iterate can “better” satisfy the constraint) and decrease all objective functions. Formally, the paper borrows the uniformity to measure the constraint violation by defining the non-uniformity measure  $\mu(\mathbf{W}) = \sum_{t=1}^T \hat{\mathcal{L}}^{(t)}(\mathbf{W}) \log \left( \frac{\hat{\mathcal{L}}^{(t)}(\mathbf{W})}{1/T} \right) = \text{KL}(\hat{\mathcal{L}}(\mathbf{W}) | \frac{1}{T})$  with  $\hat{\mathcal{L}}^{(t)}(\mathbf{W}) = \frac{[u]_t \mathcal{L}^{(t)}(\mathbf{W})}{\sum_{t'=1}^m [u]_{t'} \mathcal{L}^{(t')}(\mathbf{W})}$ . One can easily check that  $\mu(\mathbf{W}) = 0$  if and only if  $\mathbf{W}$  satisfies the constraints. EPO search shows that taking a step along the direction  $\mathbf{d}_1 = \sum_{t=1}^T \nabla \mathcal{L}^{(t)}(\mathbf{W})[u_k] (\log(\hat{\mathcal{L}}(\mathbf{W})/(1/m)) - \mu(\mathbf{W}))$  can reduce the non-uniformity (constraint violation). Meanwhile, the common descent direction  $\mathbf{d}_2$  that reduces the all objective functions, if there exists any, takes form of  $G\beta$ , where  $G = [\nabla \mathcal{L}^{(1)}(\mathbf{W}), \dots, \nabla \mathcal{L}^{(T)}(\mathbf{W})]$  and  $\beta_t \geq 0$  for all  $t \in [T]$  and  $\mathbf{1}^T \beta = 1$ . Then EPO search designs a linear programming problem to find a search direction  $\mathbf{d}$  that balances reducing constraint violation and reducing the loss functions guided by  $(\mathbf{d}_1, \mathbf{d}_2)$ . For more details, please refer to Mahapatra and Rajan (2020, Equation 24). Built upon EPO search, PHN (Pareto hyperNetworks) (Navon, Shamsian, et al., 2021) proposes to use hypernetwork, which takes the preference vector  $\mathbf{u}$  as the input and outputs the neural network weights for the multi-tasking, to attempt to learn the whole Pareto-front. Although the training is more challenging, if the hypernetwork could be properly trained, then at the inference time, the user can supply any preference vector  $\mathbf{u}$ , and the hypernetwork can output a Pareto stationary solution that closely aligns with the preference vector  $\mathbf{u}$  without requiring any additional efforts.

All aforementioned algorithms, despite their actual implementation, assume access to true gradients  $\{\nabla \mathcal{L}(t)(\mathbf{W})\}_{t=1}^T$ . This assumption might fail when in deep neural network settings. MoCo (multi-objective gradient correction) (Fernando et al., 2023) is proposed to address this issue. It extends MGDA to the stochastic setting, providing convergence rates for both convex and non-convex cases. The most notable challenge with extending MGDA to the stochastic setting lies in the noise of stochastic gradient estimators of true gradients  $\{\nabla \mathcal{L}(t)(\mathbf{W})\}_{t=1}^T$ . The standard way to address the issue is through the variance reduction technique. Unlike the seminar work of S. Liu and Vicente (2021), which achieves the variance reduction via increasing batch sizes, MoCo (Fernando et al., 2023) reduces the variance via the momentum-based method, which has the advantage of keeping the batch size as small as one while still guarantee the convergence (under proper assumptions). Concretely, at the  $k$ th iteration, instead of solving problem (2.81), MoCo solves

$$(2.84) \quad \min_{\beta \in \mathbb{R}^T} \frac{1}{2} \left\| \sum_{t=1}^T [\beta]_t d_k^{(t)} \right\|^2 \quad \text{s.t. } \sum_{t=1}^T [\beta]_t = 1 \text{ and } [\beta]_t \geq 0 \text{ for all } t,$$

where  $d_{k+1}^{(t)} = \mathbf{Proj}_{L_t}[d_k^{(t)} - \zeta_t(d_k^{(t)} - \nabla \tilde{\mathcal{L}}^{(t)}(\mathbf{W}_k))]$ , where  $\mathbf{Proj}_{L_t}$  projects vector to a ball centered at origin with radius  $L_t$ ,  $L_t$  is the Lipschitz constant of  $\mathcal{L}^t$ ,  $\zeta_t$  is some positive constant, and  $\nabla \tilde{\mathcal{L}}^{(t)}(\mathbf{W}_k)$  is some approximation of  $\nabla \mathcal{L}^{(t)}(\mathbf{W}_k)$ . One can show that  $\|d_k^{(t)} - \nabla \mathcal{L}^{(t)}(\mathbf{W}_k)\| \rightarrow 0$  as  $k \rightarrow \infty$ , hence achieving the variance reduction.

To conclude this section, a comprehensive list, to our best knowledge, to include all existing optimization methods in § 2.2.5 & § 2.2.6, is summarized in Table 7.

**Table 7.** Algorithms for the MTL as a multi-objective optimization.

| Algorithm               | Venue   | Year | Method                                  | Convergence  | Highlight   | Availability |
|-------------------------|---------|------|---|--|---|--------------|
| Uncertainty Weighting   | CVPR    | 2018 | Dynamic Weighting                       | —  | Optimize $\{\alpha^{(t)}\}_{t=1}^T$ and $\mathbf{W}$ simultaneously.  | Official     |
| GradNorm                | ICML    | 2018 | Dynamic Weighting                       | —  | Adjust $\{\alpha^{(t)}\}$ is based on the average gradient norm of each tasks and the relative progress achieved for each tasks.  | Unofficial   |
| MGDA-MTL                | NeurIPS | 2018 | Multi-Objective Opt.                    | Asymptotic Convergence   | Seminal work, which proposes to use MOO to solve deep MTL problems based on multi-gradient descent algorithm.   | Official     |
| RMTL                    | Thesis  | 2019 | Dynamic Weighting                       | —  | Adjust $\{\alpha^{(t)}\}$ is based on the relative progress achieved for each tasks.  | Official     |
| LBTW                    | AAAI    | 2019 | Dynamic Weighting                       | —  | Adjust $\{\alpha^{(t)}\}$ using the reinforcement learning strategy.  | Official     |
| DWA                     | CVPR    | 2019 | Dynamic Weighting                       | —  | $\{\alpha^{(t)}\}$ is adapted to both samples and tasks.  | Official     |
| MLDT                    | CVPR    | 2019 | Dynamic Weighting                       | —  | $\{\alpha^{(t)}\}$ is adapted to the likelihood of a loss reduction.  | Official     |
| Pareto MTL              | NeurIPS | 2019 | Multi-Objective Opt.                    | Asymptotic Convergence   | Attempt to incorporate user's preference into the solution.   | Official     |
| Controllable Pareto MTL | arXiv   | 2020 | Multi-Objective Opt.                    | —  | Use a hypernetwork to learn the entire Pareto front.  | Official     |
| PCGrad                  | NeurIPS | 2020 | Gradient Correction                     | —  | Projecting onto orthogonal subspaces to mitigate the gradient conflicts.  | Official     |
| GradDrop                | NeurIPS | 2020 | Gradient Correction                     | —  | Only keep gradients are consistent in signs in each update.   | Official     |
| Continuous Pareto MTL   | ICML    | 2020 | Multi-Objective Opt.                    | —  | Construct a continuous, first-order approximation of the local Pareto set.  | Official     |
| EPO Search              | ICML    | 2020 | Multi-Objective Opt.                    | —  | Find a Pareto stationary solution to exactly match a user's preference. Require losses to be non-negative.  | Official     |
| AuxiLearn               | ICLR    | 2021 | Bi-level Opt.                           | —  | Learn to combine losses in a nonlinear fashion.   | Official     |
| IMTL                    | ICLR    | 2021 | Gradient Correction                     | —  | Find $\{\alpha^{(t)}\}_{t=1}^T$ , such that the aggregated gradient $\sum_{t=1}^T \alpha^{(t)} \nabla L^{(t)}(\mathbf{W})$ has equal projection onto the raw gradients of individual tasks. | Unofficial   |
| GradVac                 | ICLR    | 2021 | Dynamic Weighting                       | —  | Encourage more geometrically aligned parameter updates for close tasks.   | Unofficial   |
| PHN                     | ICLR    | 2021 | Multi-Objective Opt.                    | —  | Use a hypernetwork to learn the entire Pareto front.  | Official     |
| CAGrad                  | NeurIPS | 2021 | Gradient Correction                     | Asymptotic Convergence   | The search direction is find by solving a subproblem that is similar to MGDA.   | Official     |
| SVGD                    | NeurIPS | 2021 | Multi-Objective Opt.                    | Convergence rate for strongly convex and third-order continuously differentiable functions | Integrate MGDA with Stein variational gradient descent and Langevin dynamics to obtain diverse solutions.   | Official     |
| COSMOS                  | ICDM    | 2021 | —                                       | —  | A single optimization run to approximate the full set of the Pareto front by combining preferences vectors sampled from Dirichlet distribution and training data.                           | Official     |
| HV Maximization         | arXiv   | 2021 | —                                       | —  | Utilize hyper-volume to approximate sample level Pareto front.  | Official     |
| PNG                     | UAI     | 2022 | Multi-Objective Opt.                    | Convergence rate for convex losses   | Minimize preference loss over the Pareto front (manifold optimization) while only using the first order information.  | —            |
| RLW & RGW               | TMLR    | 2022 | Dynamic Weighting                       | —  | Converge to a neighborhood of the optimal solution under strongly convex assumption.  | Unofficial   |
| Nash-MTL                | ICML    | 2022 | Multi-Objective Opt.                    | Asymptotic Convergence   | Sample the weights $\{\alpha^{(t)}\}_{t=1}^T$ from a given distribution at each step.   | Official     |
| (X)WC-MGDA              | ICML    | 2022 | Dynamic Weighting                       | —  | Formulate the problem of finding a common descent direction as a bargaining game.   | Official     |
| Rotograd                | ICLR    | 2022 | Dynamic Weighting + Gradient Correction | —  | Lift the restriction of non-negativity requirement on losses in EPO search.   | —            |
| MoCo                    | 2023    | ICLR | 2023                                    | Multi-Objective Opt.   | Dynamic Weighting via gradient norm Gradient Correction via rotating the feature-space  | Official     |
| Recon                   | ICLR    | 2023 | Gradient Correction                     | —  | Convergence rates for convex & nonconvex losses Stochastic Gradient & Variance Reduction  | Official     |
| Aligned-MTL             | CVPR    | 2023 | Gradient Correction                     | —  | Turn shared parameters that most likely to cause gradient conflicts into task specific parameters.  | Official     |
| Achievement-based MTL   | ICCV    | 2023 | Dynamic Weighting                       | —  | Use the condition number of the linear system to measure the severity of gradient dominance and conflicting issues.   | Official     |
| FULLER                  | ICCV    | 2023 | Dynamic Weighting                       | —  | Use training progress to dynamically weight tasks and use geometric mean to average loss from tasks   | Official     |

### Remarks

- (i) The MOO approach, though it generally requires extra efforts to find the common descent directions, provides a solid framework to conduct convergence analysis.
- (ii) The MOO approach helps explore more diversified solutions over the Pareto front, whereas the scalarization approach cannot find the solutions on the concave part of the Pareto front. Obtaining diversified solutions helps users to understand trade-offs among a set of objectives.
- (iii) MOO approach gives the flexibility to incorporate user preferences in the solutions and does not require laborious tuning on task weights.

**2.2.7. Adversarial training.** In the era of deep learning, joint task modeling has shown promising success by employing feature propagation or task balancing. However, it is important to acknowledge that task-specific features do not consistently result in mutual benefits, and learning multiple loosely connected tasks simultaneously introduces irrelevant noise. While task balancing helps alleviate the negative impact of transfer learning, it neglects the information exchange between tasks, often leading to suboptimal solutions. To address this issue, adversarial training (Adhikarla et al., 2022), as an optimization approach, can effectively disentangle the space between task-shared and -specific features by inherently preventing feature interference. This approach involves introducing a task discriminator, which distinguishes features or gradients learned from different tasks. The discriminator is trained along with a shared feature extractor to converge to a saddle point where the discriminator is unable to differentiate features or gradients learned from different tasks. Research in this field can be categorized into two main approaches based on the type of information utilized for adversarial training: representation-based and gradient-based. ASP-MTL (aka AdvMTL) (P. Liu et al., 2017) first proposes an adversarial MTL framework to learn task-shared and -specific features independently and introduces adversarial training to make shared features invariant to the involved tasks. MTA<sub>(adv)</sub>N (Y. Liu et al., 2018) presents an adversarial MTL framework in the

image generation tasks, where multiple existing factors for image generation are considered as tasks and disentangled in an adversarial way with the training of shared encoder. RD4MTL (Meng et al., 2019) employs adversarial training to encourage the features from different tasks to be disentangled and the features of irrelevant tasks to be minimally informative. GREAT4MTL (Sinha et al., 2018) and AAMTRL (Y. Mao et al., 2020) utilize the gradients derived from different tasks and disentangle the space using gradient reversal procedure (Ganin et al., 2016).

Representation-Based. Adversarial Shared-Private Multi-Task Learning (ASP-MTL, aka AdvMTL) (P. Liu et al., 2017) first proposes an adversarial MTL framework to alleviate the interference of shared and specific feature spaces among involved tasks. The underlying observation is the fact that the same word in a sentence may indicate different sentiments in different tasks, e.g. the "*infantile*" in product reviews "*The infantile cart is simple and easy to use.*" and product review "*This kind of humor is infantile and boring.*". "*infantile*" is a potential backdoor word encoded in the shared feature space as it expresses a neutral attitude in the product review while it conveys a negative attitude in the movie review. ASP-MTL addresses this issue by dividing the feature space into shared and specific (private) space in a parallel manner, as shown in Fig. 8t, and disentangles them using orthogonality constraints and adversarial losses. Let  $\mathcal{X}_s^{(t)}$  and  $\mathcal{X}_p^{(t)}$  denote the representations of shared and private layers for the  $t$ -th task, respectively. The adversarial training process alternates between the shared feature generator  $G$  (parametrized by  $\mathbf{W}_s$ ) and the task discriminator  $D$  (parametrized by  $\mathbf{W}_d$ ) through a minimax optimization:

$$(2.85) \quad \mathcal{L}_{adv} = \min_{\mathbf{W}_s} \max_{\mathbf{W}_d} -\mathcal{L}_{CE}[D_{\mathbf{W}_d}(\mathcal{X}_s^{(t)}), \mathbf{t}^{(t)}], t = 1, \dots, T,$$

where  $\mathbf{t}^{(t)}$  denotes the ground-truth label to indicate the task type, and  $\mathcal{L}_{CE}$  means the use of Cross Entropy loss in practice. To further extract task invariant features from the shared layers, ASP-MTL introduces the orthogonality constraint as follows to disentangle the shared and private feature space.

$$(2.86) \quad \mathcal{L}_{orth}^{(t)} = \|\text{vec}(\mathcal{X}_s^{(t)})^\top \text{vec}(\mathcal{X}_p^{(t)})\|_F^2, t = 1, \dots, T,$$

where we abuse the vectorization  $\text{vec}(\cdot)$  to preserve the sample dimension of the output feature tensors. The final learning objective function consists of three components as below:

$$(2.87) \quad \mathcal{L}_{total} = \sum_{t=1}^T (\mathcal{L}_{spec}^{(t)} + \lambda \mathcal{L}_{adv}^{(t)} + \gamma \mathcal{L}_{orth}^{(t)}),$$

where  $\mathcal{L}_{spec}^{(t)}$  is the task-specific objective for the  $t$ -th task, and  $\lambda$  and  $\gamma$  are hyper-parameters to balance the learning terms. This total objective is trained with backpropagation via the advantage of gradient reversal layer (GRL) (Ganin & Lempitsky, 2015).

Multi-Task Adversarial Network (MTA<sub>(adv)</sub>N) (Y. Liu et al., 2018) targets the problem of multiple factors existing in image generation. The architecture of MTA<sub>(adv)</sub>N is shown in Fig. 8u, where the shared encoder  $E$  extracts the features that are disentangled across style factors for the use of content classification (discriminator  $D_C$ ) and generation (generator  $G$ ). Let the original image and the corresponding content label be represented by  $\mathcal{X}$  and  $\mathbf{y}$ , respectively. Then the training of the generation task entails the updation of shared feature extractor  $E$  and generator  $G$ :

$$(2.88) \quad \mathcal{L}_G = \min_{E,G} \sum_{(i,j):\mathbf{y}_j=\mathbf{y}_i,\mathbf{z}_j=\mathbf{z}'_i} \|G(E(\mathcal{X}_i), \mathbf{z}'_i) - \mathcal{X}_j\|_2^2,$$

where  $i$  and  $j$  are data indices.  $\mathbf{z}'$  is sampled from the style label codebook  $\mathcal{Z}$ . Eq. (2.88) means that the generator  $G$  tries to reconstruct the data  $\mathcal{X}_i$  itself if  $\mathbf{z}'_i = \mathbf{z}_i$  and tries to minimize the distance between the style-transferred  $\mathcal{X}_i$  and any sample  $\mathcal{X}_j$  with the same content and style labels (i.e.  $\mathbf{y}_j = \mathbf{y}_i, \mathbf{z}_j = \mathbf{z}'_i$ ) otherwise.

The key adversarial training of style labels is defined using Earth Mover's Distance (EMD) loss (Arjovsky et al., 2017) as follows:

$$(2.89) \quad \mathcal{L}_S = \min_E \max_{D_S} \sum_i -\mathcal{L}_{EMD}(\mathbf{x}_i, \mathbf{z}_i) - \lambda \Omega_{GP}(D_S),$$

where  $\Omega_{GP}(D_S)$  is a gradient penalty term (Gulrajani et al., 2017) for the purpose of training stability and  $\lambda$  serves as a trade-off hyper-parameter. To add the classification of content factor, the total training objective is formulated as follows:

$$(2.90) \quad \mathcal{L}_{total} = \min_{E,G} \mathcal{L}_G + \alpha \min_E \max_{D_S} \mathcal{L}_S + \beta \min_{E,D_C} \mathcal{L}_C,$$

where  $\mathcal{L}_C$  denotes the Cross-Entropy loss of the content classification task,  $\alpha$  and  $\beta$  both are the hyper-parameters.

Representation Disentanglement for Multi-Task Learning (RD4MTL) (Meng et al., 2019) aims to disentangle the indiscriminate mixing of properties in medical image analysis. As depicted in Fig. 8v, an adversarial training process encourages the features from different tasks to be disentangled and minimally informative. Let  $\mathbf{Z}^{(t)}$  represent the latent features extracted by the specific encoder  $E_{\theta^{(t)}}$  from the original image  $\mathcal{X}$ , then  $\mathcal{L}_{cls}^{(t)}$  as the  $t$ -th task-specific classification loss can be calculated as follows:

$$(2.91) \quad \mathbf{Z}^{(t)} = E_{\theta^{(t)}}(\mathcal{X}), \mathcal{L}_{cls}^{(t)} = \mathcal{L}_{CE}(D_{\phi^{(t)}}(\mathbf{Z}^{(t)}), \mathbf{y}^{(t)}), t = 1, \dots, T,$$

where  $\mathbf{y}^{(t)}$  is the ground truth label of the  $t$ -th task, and  $\mathcal{L}_{CE}$  is the Cross Entropy loss in practice. Furthermore, the adversarial regularization uses a minimax competition process as below:

$$(2.92) \quad \mathcal{L}_{adv}^{(t)} = \min_{\{\theta^{(s)}, \phi^{(s)}\}_{s \neq t}^T} \max_{\psi^{(t)}} \sum_{s \neq t}^T -\mathcal{L}_{CE}(D_{\psi^{(t)}}(\mathbf{Z}^{(s)}), \mathbf{y}^{(s)}), t = 1, \dots, T,$$

then the total training objective can be formulated as follows:

$$(2.93) \quad \mathcal{L}_{total} = \sum_{t=1}^T (\mathcal{L}_{cls}^{(t)} + \lambda \mathcal{L}_{adv}^{(t)}),$$

where  $\lambda$  balances the two loss terms.

Adaptive Adversarial Multi-Task Representation Learning (AAMTRL) (Y. Mao et al., 2020) investigates the theoretical mechanism of adversarial MTL via using Lagrangian duality, and further proposes the AAMTRL that can improve the performance of classical adversarial MTL (aka AMTRL methods in (Y. Mao et al., 2020)). For simplicity, if the shared and -private features for the  $t$ -th are represented by  $\mathcal{X}_s^{(t)}$  and  $\mathcal{X}_p^{(t)}$ , aligning with the formalization in Eq. (2.85). Assume the shared feature extractor  $E$  (parametrized by  $\mathbf{W}_s$ ) and task discriminator  $D$  (parametrized by  $\mathbf{W}_d$ ) to be Bayes-optimal, AAMTRL introduces the matrix  $\mathbf{R}$  to measure the task relatedness, where

$$(2.94) \quad r_{i,j} = \frac{D_j(\mathcal{X}_s^{(i)}) + D_i(\mathcal{X}_s^{(j)})}{D_i(\mathcal{X}_s^{(i)}) + D_j(\mathcal{X}_s^{(j)})},$$

where  $r_{i,j}$  is the  $(i,j)$ -th entry of the matrix  $\mathbf{R}$ , and  $D_i$  represents the probability that the discriminator  $D$  classify the input representations as  $i$ -th task type. In AAMTRL, the adaptation is realized by the weighting strategy of task-specific objectives  $\{\mathcal{L}_{spec}^{(t)}\}_{t=1}^T$ :

$$(2.95) \quad \mathcal{L}_{spec} = \sum_{t=1}^T \alpha_t \mathcal{L}_{spec}^{(t)}, \alpha_t = \mathbf{1} \mathbf{R} / (\mathbf{1} \mathbf{R} \mathbf{1}^\top).$$

The classic adversarial MTRL problem can be regard as the Lagrangian dual function of the following equality-constrained optimization problem:

$$(2.96) \quad \min_{\{\mathbf{W}_s, \mathbf{W}_d\}} \mathcal{L}_{spec}, \quad \text{s.t.} \quad \mathcal{L}_{adv} = 0.$$

To avoid the sub-optimal solution of the traditional Lagrangian duality in solving the problem above, an augmented Lagrangian with a quadratic form is proposed as follows:

$$(2.97) \quad \min_{\{\mathbf{W}_s, \mathbf{W}_d\}} \mathcal{L}_{spec} + \lambda \mathcal{L}_{adv} + r/2 \mathcal{L}_{adv}^2,$$

where  $\lambda$  is the Lagrangian multiplier, and  $r$  is the penalty hyper-parameter that can balance the duality gap. By using Lagrangian duality, AAMTRL can have an exact generalization error bound that is minimally investigated in the classic AMTRL.

Gradient-Based. GRadiEnt Adversarial Training for MTL (GREAT4MTL) (Sinha et al., 2018) is one of the scenarios of GRadiEnt Adversarial Training (GREAT) that tries to make the gradients indistinguishable across involved tasks. As depicted in Fig. 8w, the encoder  $E_\theta$  extracts shared features for multiple tasks, and the decoders  $\{D_{\phi^{(t)}}\}_{t=1}^T$  are used to perform  $T$  involved tasks. Thus, the basic learning objectives for specific tasks are:

$$(2.98) \quad \mathcal{L}_{spec}^{(t)} = \min_{\theta, \{\phi^{(t)}\}_{t=1}^T} \mathcal{L}^{(t)}(D_{\phi^{(t)}}(E_\theta(\mathcal{X}^{(t)})), \mathbf{y}^{(t)}), t = 1, \dots, T,$$

where  $\{(\mathcal{X}^{(t)}, \mathbf{y}^{(t)})\}_{t=1}^T$  is the total dataset containing  $T$  tasks, and  $\mathcal{L}^{(t)}$  is dependent on the task type. In GREAT4MTL, the Gradient-Alignment Layer (GAL)  $G_\psi$  is placed after the shared encoder and before the task-specific decoders to perform task discrimination. Unlike representation-based methods that attend to the features,  $G_\psi$  is trained using gradients from different tasks as inputs:

$$(2.99) \quad \mathcal{L}_{adv} = \min_{\{\phi^{(t)}\}_{t=1}^T} \max_{\psi} \sum_{t=1}^T -\mathcal{L}_{CE}(G_\psi(\nabla_{E_\theta(\mathcal{X}^{(t)})} D_{\phi^{(t)}} \mathcal{L}_{spec}^{(t)}), \mathbf{y}^{(t)}), \mathbf{t}^{(t)},$$

where  $\mathbf{t}$  is the ground truth label to indicate the task type, and the Cross-Entropy loss  $\mathcal{L}_{CE}$  is used to calculate the task classification error. Then the total training objective function is:

$$(2.100) \quad \mathcal{L}_{total} = \sum_{t=1}^T \mathcal{L}_{spec}^{(t)} + \mathcal{L}_{adv}.$$

The GRL is inserted before the GAL to streamline the minimax optimization process above. The trade-off hyper-parameter is eliminated in Eq. (2.100) by using different learning rates during the training process of  $\mathcal{L}_{spec}$  and  $\mathcal{L}_{adv}$ .

ASTMT (Maninis et al., 2019) also employs the GREAT strategy to effectively disentangle the task-shared and task-specific features acquired from the shared backbone and single-tasking components, as illustrated in the right portion of Fig. 8n. It highlights the compatibility of GREAT to be seamlessly integrated with other frameworks.

### Remarks

- (i) Adversarial training effectively disentangles the feature space into shared and task-specific components, ensuring that shared features remain indistinguishable across multiple tasks, while specific features retain their distinctiveness.

**2.2.8. Mixture of Experts (MoE).** Deep neural-based architectures have been extensively utilized in real-world MTL problems. However, the challenge of scaling high-capacity deep neural networks to adapt to multi-task settings remains conceptually appealing. The MoE (Jacobs et al., 1991) framework inherently incorporates multiple expert networks, each of which can be selected for learning different tasks. The modern MoE layer (Eigen et al., 2013; Shazeer et al., 2017) has transformed the MoE module into a universally adaptable component that seamlessly integrates into various systems, including CNNs, RNNs, and Transformers, enabling plug-and-play functionality. The MoE layer, as depicted in Fig. 16a, generally comprises a set of  $N$  expert networks  $\{E_n\}_{n=1}^N$  and a gating network  $G$ , whose output depends on the input data  $\mathcal{X}$ . This gating network generates a sparse  $N$ -dimensional vector that selects the necessary expert networks to compute the final

prediction as follows:

$$(2.101) \quad \tilde{\mathbf{y}} = \sum_{n=1}^N G(\mathcal{X})_n E_n(\mathcal{X}),$$

where  $G(\mathcal{X})_n \in \{0, 1\}$  is the  $n$ -th entry of the sparse vector generated by the gating network  $G$ , and  $\tilde{\mathbf{y}}$  represents the. Beyond MoE for STL, Multi-gate Mixture-of-Experts (MMoE) (J. Ma et al., 2018) explicitly introduces multiple gates/routers ( $\{G_t\}_{t=1}^T$ ) for each task, as shown in Fig. 16b. The final prediction for the  $t$ -th task is calculated as

$$(2.102) \quad \mathbf{y}^{(t)} = \sum_{n=1}^N G_t(\mathcal{X}^{(t)})_n E_n(\mathcal{X}^{(t)}), t = 1, \dots, T,$$

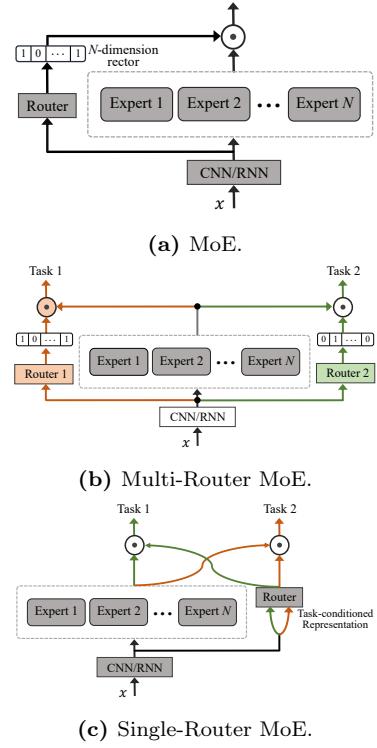
where  $(\mathcal{X}^{(t)}, \mathbf{y}^{(t)})$  represents the sampled data from  $t$ -th task. This prior research has inspired the development and utilization of multi-router MoE for MTL. It includes DSelect-k that selects top  $k$  experts for each task, MT-Tag (Gupta et al., 2022), demonstrating the robustness of Multi-Router MoE to the loosely related tasks, CmoIE (S. Wang et al., 2022), which constructs more insightful experts instead of incompetent ones, Mod-Squad (Z. Chen et al., 2023), specializing experts for specific tasks by measuring the mutual information (MI) between tasks and experts, and SummaReranker (Ravaut et al., 2022), performing re-ranking on a set of summary candidates to select the best one. On the other hand, task-conditioned routing with a shared router/gate is another variant where task-dependent representations are fed into the only existing router, making their expert selections, as depicted in Fig. 16c for comparison. The shared-router MoE is discussed separately from the Multi-router MoE in M<sup>3</sup>ViT(Fan et al., 2022). Task-level MoE (Q. Ye et al., 2022) designs different router architectures with varying complexities under shared-router settings, including MLP, LSTM, and Transformer. In both ways, task relationships are captured in different mixture patterns of experts assembling.

Multi-Router MoE. Multi-gate Mixture of Experts (MMoE) (J. Ma et al., 2018) replaces the shared layers in the hard parameter architecture with multiple MoE layers and retains individual routers for each task, resembling the soft parameter architecture. The computational process of predicting  $t$ -th task is shown in Eq. (2.102). The router networks of MMoE is the softmax of the linear transformations of the input data representation:

$$(2.103) \quad G_t(\mathbf{X}^{(t)}) = \text{softmax}(\mathbf{W}_t \mathbf{X}^{(t)}), t = 1, \dots, T,$$

where  $\mathbf{W}_t \in \mathbb{R}^{N \times D}$ , and  $N, D$  is the number of experts and the number of features. In comparison to the soft parameter sharing architecture, MMoE features routers solely for each task, resulting in a lighter size and enhanced scalability with an increasing number of tasks. In addition, the conditional computation (Bengio et al., 2013; Shazeer et al., 2017) of the MoE layer requires the activation of only specific parts of the experts on a per-example basis. While Shazeer et al. (2017) offers a top- $k$  gating function by adding tunable Gaussian noise, the theoretically scary discontinuities can lead to convergence issues if learning via gradient-based optimization.

Differentiable Selection of top- $k$  experts(DSelect- $k$ ) (Hazimeh et al., 2021) bridges this gap by proposing



**Figure 16.** The taxonomy of (a) MoE into two categories: (b) Multi-Router MoE (c) Single-Router MoE.

a continuously differentiable and sparse gate in the context of MMoE. Obviously, the direct cardinality constraint ( $\ell_0$  norm) on the output vector of the gate function is not amenable to SGD. To address this issue, a binary encoding scheme is introduced to realize top- $k$  selection via unconstrained minimization. Let  $\mathbf{Z} \in \mathbb{R}^{k \times m}$  denote a matrix that selects the top- $k$  experts, whose  $i$ -th row  $\mathbf{z}_i$  is a  $m$ -dimensional binary encoding of the index of any single expert, where  $m = \log_2 N$  and  $N$  is the number of total experts. The gate output vector  $\mathbf{q}$  is defined as follows:

$$(2.104) \quad \mathbf{q}_{\alpha, \mathbf{Z}} = \sum_{i=1}^k \sigma(\alpha)_i r(\mathbf{z}_i),$$

where  $\alpha \in \mathbb{R}^k$  is a learnable vector to control the importance of the final selected top- $k$  experts, and  $r(\mathbf{z}_i) \in \mathbb{R}^N$  defines the single expert selector that returns a one-hot encoding of the index of some selected expert. It is noticeable that  $\|\mathbf{q}(\alpha, \mathbf{Z})\|_0 \leq k$  and  $\sum_{i=1}^N q(\alpha, \mathbf{Z})_i = 1$ , which realize the similar property for the gate output without any constraint involved. Furthermore, DSelect-k using a element-wise smoothing function  $S : \mathbb{R} \rightarrow \mathbb{R}$  to relax every binary variable in  $\mathbf{Z}$  to be continuous in the range  $(-\infty, +\infty)$ :

$$(2.105) \quad \tilde{\mathbf{q}}_{\alpha, \mathbf{Z}} \approx \mathbf{q}_{\alpha, S(\tilde{\mathbf{Z}})}, S(z) = \begin{cases} 0, & \text{if } z \leq -\gamma/2, \\ (-2/\gamma^3)z^3 + (3/(2\gamma))z + 1/2, & \text{if } -\gamma/2 \leq z \leq \gamma/2, \\ 1, & \text{if } z \geq \gamma/2, \end{cases}$$

where  $\gamma$  is a hyper-parameter that controls the width of the fractional region. Eqs. (2.104) and (2.105) transform the top- $k$  selection to be unconstrained and first-order differentiable.

Multi-Task Task-aware Gating (MT-TaG) (Gupta et al., 2022) designs the task-aware sparse gating function to route expert selection for each task. The incorporation of task-conditioned information into the routing mechanism is realized by constraining each embedding to only the top-1 expert selection. Let  $\mathbf{x}_i^{(t)}$  be the token/embedding representation in the  $i$ -th position of the input sequence for the  $t$ -th task. A linear mapping process is first applied to obtain the touting logits below:

$$(2.106) \quad \tilde{\mathbf{x}}_i^{(t)} = \mathbf{W}^{(t)} \mathbf{x}_i^{(t)}, t = 1, \dots, T,$$

then the only expert routing is as follows through a softmax process:

$$(2.107) \quad h(\mathbf{x}_i^{(t)}) = \max_j \left( \frac{e^{\tilde{\mathbf{x}}_i^{(t)}}}{\mathbf{1}^\top e^{\tilde{\mathbf{x}}_i^{(t)}}} \right)_j \cdot E_j(\mathbf{x}_i^{(t)}), t = 1, \dots, T,$$

where  $h$  denotes the task-conditioned representation calculated by the selected experts. Noticeably, the task relationship is implicitly encompassed within the variable  $h$ , thereby remaining independent of the experts involved. SummaReranker (Ravaut et al., 2022) targets only the abstractive summarization task but utilizes different metrics to measure it. The re-ranking on a set of summary candidates generated by MMoE can consistently promote the base model.

However, the promise of MMoE has been validated in MTL with the explicit task relationship backups. Calibrated Mixture of Insightful Experts(CMoIE) (S. Wang et al., 2022) investigates the negative transfer in MMoE caused by incompetent experts in certain applications. Specifically, a conflict resolution module between each pair of experts and the expert communication among the layers of different experts are introduced to advocate the diversity and capacity of experts. Additionally, a mixture calibration structure employed in the routing networks encourages the expert responsibilities to handle more tasks without losing their specialty. For any input data  $\mathcal{X}$ , the conflict resolution employs the Euclidean distance to measure the outputs from each pair of experts:

$$(2.108) \quad \mathbf{D}_{i,j} = \ell_2(E_i(\mathcal{X}), E_j(\mathcal{X})), i, j = 1, \dots, N,$$

where  $N$  is the number of total experts, and  $\mathbf{D} \in \mathbb{R}^{N \times N}$  denotes the distance matrix between each pair of experts. Based on the max-margin  $t$ -distribution, the corresponding conflict attention matrix for each pair of experts is calculated to highlight the excessively similar expert pairs:

$$(2.109) \quad \mathbf{A}_{i,j} = 1/(1 + \max(0, \mathbf{D}_{i,j} - R_i)), i, j = 1, \dots, N,$$

where  $R_i$  is the conflict radius of the expert  $E_i$  that defines the upper quartile of  $\{\mathbf{D}_{i,j}\}_{j=1}^N$ . Furthermore, the conflict loss is proposed as follows:

$$(2.110) \quad \mathcal{L}_{conflict} = - \sum_{i=1}^N \sum_{j=1}^N (\mathbf{A} \odot \mathbf{D})_{i,j},$$

where  $\mathcal{L}_{conflict}$  is combined with multi-task loss in an end-to-end training process. To capture implicit task relationships by constructing task-aware representations, the fusion matrix  $\mathbf{F} \in \mathbb{R}^{N \times N}$  is defined using multilinear map as follows:

$$(2.111) \quad \mathbf{F}_l^{(t)} = G^{(t)}(\mathcal{X}^{(t)}) \otimes H_l(\mathcal{X}^{(t)}), t = 1, \dots, T,$$

where  $G^{(t)}$  and  $H_l$  are the routing networks and another hidden-layer gating network before the  $l$ -th layer for the experts. Let the hidden representations at the  $l$ -th layer of  $E_n$  denote by  $\mathbf{z}_l^n, n = 1, \dots, N$ , and then stack all of them by the way of  $[\mathbf{z}_l^1, \dots, \mathbf{z}_l^n]^\top$  to be the hidden representation matrix  $\mathbf{Z}_l$ . Through the fusion process defined in Eq. (2.111), the input of  $(l+1)$ -th layer of  $\{E_n\}_{n=1}^N$  is diffused by:

$$(2.112) \quad \tilde{\mathbf{Z}}_{l+1}^{(t)} = \mathbf{F}_l^{(t)} \mathbf{Z}_l^{(t)} + \mathbf{Z}_l^{(t)}, t = 1, \dots, T,$$

where the representation is tailored by the task-specific fusion matrix. The residual block  $(+\mathbf{Z}_l^{(t)})$  above can suppress the individuality ruin of experts during the fusion process. To further enhance the specialization and concentration of experts on specific tasks, the mixture calibration introduces a dynamic temperature  $\tau^{(t)}$  to control the logits for each routing network:

$$(2.113) \quad G^{(t)}(\mathcal{X}^{(t)}) = \text{softmax}(g^{(t)}(\mathcal{X}^{(t)})/\tau^{(t)}), t = 1, \dots, T,$$

where the temperature parameters are progressively decreased from 1 during the training process.

Mod-Squad (Z. Chen et al., 2023) also allows cooperation and specialization in the process of matching experts and tasks. To make the experts dependent on tasks, the mutual information between them is first measured as below:

$$(2.114) \quad \mathcal{I}(\mathcal{T}; E) = \sum_{t=1}^T \sum_{n=1}^N P(\mathcal{T}_t, E_n) \log \frac{P(\mathcal{T}_t, E_n)}{P(\mathcal{T}_t)P(E_n)},$$

where the joint probability will be decided by the number of data that are routed inside a task to the target expert. Then the total loss can be formulated as follows:

$$(2.115) \quad \mathcal{L}_{total} = \sum_{t=1}^T \lambda_t \mathcal{L}_{\mathcal{T}_t} - \gamma \sum_{\forall \text{ MoE layers } l} I(T; E_l),$$

where  $\lambda_t$  is the hyper parameter to control the  $t$ -th task-specific loss  $\mathcal{T}_t$ , and  $\gamma$  balances the multi-task loss term and mutual information term.

Shared-Router (Task-Conditioned) MoE. Task-Level MoE (Q. Ye et al., 2022) first uses a shared router that takes the task representation as input, which is selected from a look-up embedding table. Moreover, Task-Level MoE first investigates the combinations of different backbone (MLP, LSTM, and Transformer) and softmax (softmax, Gumbel-Softmax, and ST Gumbel-Softmax (Jang et al., 2016)) variations of routers. M<sup>3</sup>ViT (Fan et al., 2022) customizes MoE into a ViT backbone, which compares the multi-router MoE and shared-router MoE. ViT-based MMoE can feature hardware memory efficiency, as certified in Edge-MoE (Sarkar et al., 2023).

To circumvent the limitations associated with a fixed single expert, the AdaMV-MoE (T. Chen et al., 2023), denoted as the Adaptive Mixture of Experts framework for Multi-task Vision Recognition, possesses the capacity to autonomously ascertain the number of sparsely activated MoE based on input token embeddings. Task-specific router networks are employed to select the most relevant experts for individual tasks. This process can be mathematically expressed as:

$$(2.116) \quad G_t(\mathbf{x}^{(t)}) = \sum_{n=1}^{N_t} \mathcal{R}_t(\mathbf{x}) \cdot E_n(\mathbf{x}), t = 1, \dots, T,$$

where  $\mathcal{R}_t$  is the router for  $t$ -th task. It should be noted that the number of experts ( $N_t$ ) engaged is not predefined. AdaMV-MoE incorporates an adaptive mechanism, specifically the Adaptive Expert Selection (AES) technique, to dynamically adjust this quantity based on task-specific loss values observed during validation on datasets ( $\mathcal{L}_{\text{val}}^{(t)}$ ). If  $\mathcal{L}_{\text{val}}^{(t)}$  exhibits no signs of decline over several iterations, the number of experts ( $N_t$ ) should be augmented by 1. In contrast, if it exceeds the best loss value above, the number of experts should be reduced. Ultimately, after numerous iterations, the number of experts can be stabilized.

### Remarks

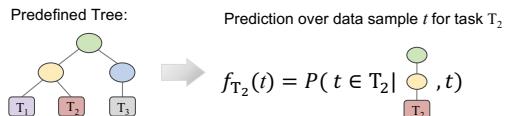
- (i) MoE seamlessly accommodates tasks with different backbones, making it well-suited for scenarios with diverse task requirements and complexities.
- (ii) MoE efficiently allocates resources by assigning different experts to different tasks, avoiding redundancy and optimizing computational resources.
- (iii) MoE exhibits remarkable scalability, rendering it highly suitable for large-scale industry applications on the ground.

**2.2.9. Graph based.** Graphs have been widely used in data mining and machine learning due to their unique representation of objects and their interactions. Graph neural networks (GNNs) (Gori et al., 2005; Scarselli et al., 2008; Sperduti & Starita, 1997; Z. Wu et al., 2020), which leverage nodes and edges among their connected nodes in graphs to conduct inference, have gained applause with impressing performance in capturing the inter-nodes relations on graphs. It is natural to consider the tasks and corresponding data samples in MTL as nodes and their relations as the edges to construct a graph for MTL (Alon et al., 2017). Via conducting graph mining on such graphs, relations among tasks or data samples in MTL can be better understood so as to assist the final MTL model in conducting inference (Cao et al., 2022; Z.-M. Chen et al., 2019; C. Liu et al., 2020; S. Liu et al., 2022)

MultiKernel (Widmer et al., 2010) conducts MTL over a series of classification tasks with predefined hierarchical relations, which is often the case for biological problems. Notably, it constructs a tree that reflects the hierarchical relations between tasks and domains, where leaf nodes are the tasks it studies (e.g., dog), whose parent and ancestors (non-leaf nodes) are the corresponding biological domains (e.g., mammals and animals).

For a queried task  $\mathbf{x}$ , MultiKernel classifies it over every task  $t$ 's predictor  $f_t$  by

$$(2.117) \quad f_t(\mathbf{x}) = (\mathbf{u}_t + \sum_{r \in \{t\}'s \text{ ancestors}} \lambda_{t,r} \mathbf{u}_r)^\top \mathbf{x} + b_t,$$



**Figure 17.** An example of MultiKernel predicting the probability of a data sample  $t$  belonging to task  $T_2$ . MultiKernel conducts the prediction based on  $T_2$  and its domains, whose hierarchical relation is extracted from the predefined tree. Specifically, ellipses are domains, and squares are tasks.

where  $\lambda_{t,r}$  is a pre-calculated constant inversely related to the distance between task  $t$  and its ancestors  $r$ .  $\mathbf{u}_t$  is the representation of task  $t$ . The representations of nodes within the predefined tree are learned by minimizing the task error.  $b_t$  is a learnable variable.

ML-GCN (Z.-M. Chen et al., 2019) is a graph convolutional network (GCN)-based MTL model for capturing the label correlations in multi-label image recognition. Specifically, different from traditional MTL, ML-GCN pre-constructs a correlation matrix that reflects labels' co-occurrence patterns within datasets. This matrix enables the system to build a label graph, where each node represents a label, and whose feature is the corresponding word embedding.

On retrieving the label graph, ML-GCN jointly trains a CNN and a GCN for the MTL. The CNN learns from image datasets to retrieve image representations, and the GCN learns from the label graph to generate label representations. ML-GCN retrieves multi-label prediction  $\hat{\mathbf{y}}$  for an input image  $\mathbf{x}$  by computing dot products between image representations and label representations as  $\hat{\mathbf{y}} = \mathbf{W} \cdot f(\mathbf{x}; \theta)$ , where  $f(\cdot)$  and  $\theta$  are the CNN model and its parameters respectively.  $\mathbf{W} = \{\mathbf{w}^{(i)}\}_{i=0}^C$  is the set of label representations output the GCN.

ML-GCN resorts to the traditional multi-label classification loss for training. The entire construction of ML-GCN is shown in Fig. 8x.

MetaLink (Cao et al., 2022) assumes that, for a given data point, at the inference time, the multi-task model has access to its labels from auxiliary tasks. Based on this assumption, MetaLink leverages labels from other tasks to improve the predictive performance. Particularly, MetaLink constructs a knowledge graph to capture not only the task-task relations as in ML-GCN but also the inter- and intra-relations between tasks and data.

The knowledge graph consists of two types of nodes: (1) data nodes, whose features are embeddings computed by the neural networks, and (2) task nodes, whose features are the last layer weights of the corresponding task-specific neural networks. Whenever a data sample belongs to a task, an edge is connected between these two nodes, and the label of the edge describes how the data point is classified in the particular task. In this way, MetaLink transfers the traditional MTL to a link prediction task between data nodes and task nodes, as shown in Fig. 8y.

In terms of updating the entire model, MetaLink does not specify the criterion or introduce any particular regularizing terms.

### Remarks

- (i) Graph-based representations allow tasks to be modeled as nodes and their relationships as edges. This enables the capturing of intricate dependencies and relationships among tasks, providing a more nuanced understanding compared to simplistic task relationship learning.
- (ii) GCN exhibits scalability in handling MTL scenarios with large number of tasks.
- (iii) GCN excels in information propagation across tasks within a graph structure. The interconnected nature of tasks in a graph allows for the sharing of relevant information, fostering collaborative learning.

**2.2.10. Neural Architecture Search (NAS).** NAS is a popular method in designing deep neural networks automatically, which has the potential to revolutionize the way neural networks are designed and used in many different fields, including MTL. NAS in MTL refers to the use of NAS to design neural networks that can perform multiple tasks simultaneously. This is different from traditional neural network design, where a separate network is typically trained for each task. In MTL, the goal is to learn a shared representation that can be used to perform multiple tasks effectively. Conventional architecture realizes multi-tasking by hard-parameter sharing that trains multiple task heads

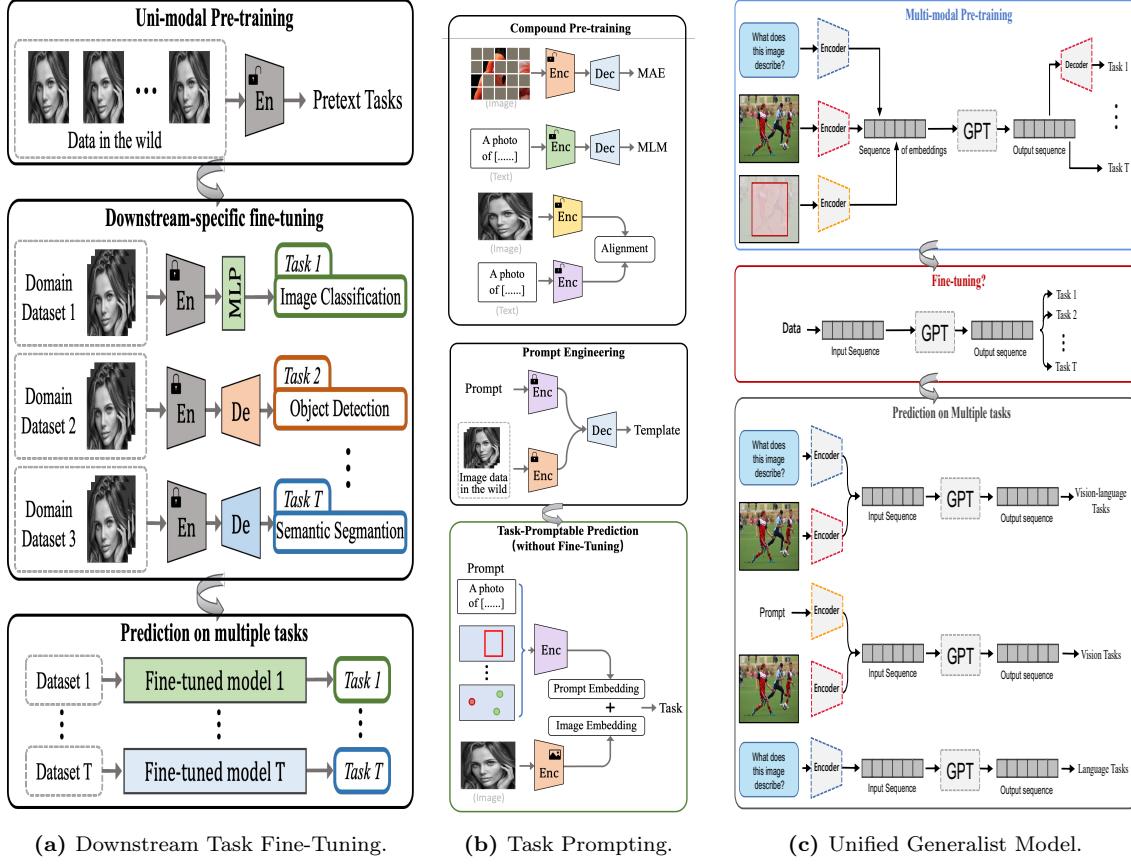
that share shallow feature extractors, e.g., TCDCN (Z. Zhang et al., 2014) and Fast RCNN (Girshick, 2015; Girshick et al., 2014), or by training separate neural network to perform all each task with the shared trunk, e.g., Cross-Stitch Networks (Misra et al., 2016) and NDDR-CNN (Y. Gao et al., 2019). However, the potential design space for deep multi-task neural architectures grows exponentially with the depth, and incorporating more tasks significantly expands the range of optimal solutions.

NAS can be used as an automatic approach to search for the optimal architecture for an MTL system. This involves defining a search space that includes a range of possible architectures and using a search algorithm to explore this space and identify the best-performing architecture. The search algorithm can be based on techniques such as reinforcement learning, evolutionary algorithms, or gradient-based optimization. There are several benefits to using NAS in multi-task learning. For example, it can reduce the need for manual design of the network architecture, improve the performance of the multi-task system, and reduce the amount of data and computation required to train the network. It can also be used to identify architectures that are more efficient and easier to implement in practice.

Fully-Adaptive Feature Sharing (FAFS) (Y. Lu et al., 2017) is the earliest method that trains networks with an adaptive widening process. The initial network is a slimmed-down version from reducing the number of convolutional filters in CNN or neurons in MLP. It gradually expands through a multi-round widening and training procedure, facilitated by a top-down splitting algorithm. In practice, the original active layer, depicted as the  $L$ -th layer in Fig. 8q, consists of numerous branches. These branches are then grouped together in the lower  $(L - 1)$ -th layer. Subsequently, the  $(L - 1)$ -th layer becomes the new active layer, and this iterative process continues from the top layers until the convergence.

Branched Multi-Task Networks (BMTN) (Vandenhende, Georgoulis, Gool, & Brabandere, 2020) argues that learning layer sharing level in the early soft parameter sharing methods suffer from sub-optimal solutions, and relying solely on NAS to design the MTL architecture is significantly cumbersome. By leveraging the affinities of involved multiple tasks using Representation Similarity Analysis (RSA) (Dwivedi & Roig, 2019), BMTN can automatically cluster the tasks at shared locations, in which bottom layers are task-agnostic and top layers gradually grow to be task-specific. For each task, as depicted in Fig. 8r, BMTN initially computes the representation dissimilarity matrices (RDMs) between  $K$  images at  $D$  locations. The RDMs are defined as  $1 - \rho$ , where  $\rho$  represents the Pearson correlation coefficient (Pearson, 1895). Subsequently, the task affinity tensor  $\mathbf{A} \in \mathbb{R}^{D \times T \times T}$  is established based on the RDMs of all tasks using the Spearman's correlation coefficient (Spearman, 1961). Finally, BMTN is established by minimizing the sum of these task dissimilarity scores (i.e.  $1 - \mathcal{A}_{d,i,j}$ ) between each pair of tasks  $i$  and  $j$  at every location  $d$ ,  $i, j = 1, \dots, T, d = 1, \dots, D$ .

Multi-Task Learning by Neural Architecture Search (MTL-NAS) (Y. Gao et al., 2020) is a method to search cross-task edges into fixed single-task network backbones. The framework is shown in Fig. 8s. It involves a single-shot gradient-based search algorithm that can optimize the architecture weights overall legal connections defined by the search space. Specifically, this search algorithm contains the continuous relaxation and the discretization procedures. This novel search algorithm is able to close the performance gap between search and evaluation and also generalizes the popular single-shot gradient-based methods such as DARTS (H. Liu et al., 2018) and SNAS (S. Xie et al., 2018).



(a) Downstream Task Fine-Tuning.

(b) Task Prompting.

(c) Unified Generalist Model.

**Figure 18.** The taxonomy of PFM of MTL into three categories: (A) Downstream Task Fine-Tuning (B) Task Prompting (C) Unified Generalist Model.

### Remarks

- 
- NAS facilitates the automatic and adaptive discovery of task-specific neural network architectures, departing from conventional hard or soft parameter sharing stereotypes.
- NAS not only searches for architectures but can also optimize hyperparameters during the search process. This automatic tuning ensures that the MTL model is configured with optimal settings for each task, reducing the need for manual fine-tuning.
- NAS can discover architectures that capture these dependencies effectively, allowing tasks to share information efficiently. This adaptability is crucial in scenarios where tasks have a significant influence on each other.

### 2.3. Foundation Model Era: Towards Unified and Versatile.

AI models are shifting their focus from deeper networks (e.g., ConvNets (Fukushima, 1980; K. He et al., 2016; LeCun et al., 1998; Z. Liu et al., 2022), GANs (Goodfellow et al., 2020), CapsNets (Sabour et al., 2017), RNNs (Hochreiter & Schmidhuber, 1997; Rumelhart et al., 1986)) to foundation (e.g., BERT (Devlin et al., 2018), GPT-4<sup>12</sup> (OpenAI, 2023), SAM (Kirillov et al., 2023), DALL-E 3<sup>13</sup> (Ramesh et al., 2021)). Such foundation models leverage (usually in self-supervised, unsupervised, and assisted-manual ways) web-scale pretraining data in the wild and then adapt

<sup>12</sup><https://openai.com/research/gpt-4>

<sup>13</sup><https://openai.com/dall-e-3>

their backbones to different downstream tasks (Bommasani et al., 2021; C. Zhou et al., 2023), thus inherently conflict-free towards MTL. In light of recent development of scalable learners, particularly Transformers, foundation models evolve from parameter-based transfer learning with new emergent capabilities. They facilitate the integration of multiple tasks into a pretrained backbone, achieved through only fine-tuning or even zero-shot learning (ZSL). In this context, the emergent properties in foundation models extend MTL from a fixed set of tasks (where training and test tasks are identical) to handling unknown tasks. When viewed from a task-oriented perspective, MTL, empowered by foundation models, can be categorized into three distinct types:

- (1) *(Downstream) Task-Generalizable Fine-tuning.* This category involves the uni-modal learning of inclusive representations in semi-supervised, self-supervised, and unsupervised learning manners. Notable examples include BiGAN (Donahue & Simonyan, 2019; Donahue et al., 2016), BERT (Devlin et al., 2018), MoCo (X. Chen et al., 2020, 2021; K. He et al., 2020), , SimCLR (T. Chen, Kornblith, Norouzi, & Hinton, 2020; T. Chen, Kornblith, Swerstky, et al., 2020), MAE (K. He et al., 2022), and GPT (Brown et al., 2020; OpenAI, 2023; Radford et al., 2018, 2019). The learned encoders should be transferable to a variety of downstream supervised tasks, thereby enabling them to be multi-task learners.
- (2) *Task-Promptable Engineering.* In this category, the original inputs are modified through task-specific prompts (e.g., SAM (Kirillov et al., 2023)) during the pretraining stage. Prompt engineering can affect the representation of data and facilitate the learners with few-shot and even zero-shot abilities toward new tasks.
- (3) *Task-Agnostic Unification.* This category highlights that the representations remain unbiased toward specific tasks and data modalities via employing a unified serialization/sequence of data tokens, including Pix2Seq (T. Chen, Saxena, Li, Fleet, & Hinton, 2022; T. Chen, Saxena, Li, Lin, et al., 2022), UniTAB (Z. Yang et al., 2022), Unified-IO (J. Lu et al., 2022), Uni-Perceiver (H. Li et al., 2023; J. Zhu et al., 2022; X. Zhu et al., 2022a), OFA (Bai et al., 2022; P. Wang et al., 2022), Gato (Reed et al., 2022), UnIVAL (Shukor et al., 2023), etc. As a result, multi-modal learners can obtain the generalizability from existing tasks to new ones, even those involving diverse data modalities.

2.3.1. *Downstream Task Fine-Tuning.* At the moment of Pretrained Foundation Models (PFMs) (C. Zhou et al., 2023) inception, the terminology “pre-training” remained somewhat ambiguous within the field of deep learning research. This practice involves the initial learning of model backbones on a general dataset, e.g., ImageNet (J. Deng et al., 2009; Russakovsky et al., 2015), followed by their transfer to other tasks that commence fine-tuning with a warm-up initialization. Consequently, a similar process of “fine-tuning” before PFMs pertains to the fine-tuning of model backbones. In our context, fine-tuning with the changes of backbone parameters refers to model tuning, unless otherwise specified. It matters since PFMs are costly to backpropagate, and the ability to generalize large frozen backbone to multiple downstream tasks referred to as downstream fine-tuning, can ease this burden. By confining our discussion to the context of downstream fine-tuning within the frozen model, we can extend the previous definition of MTL (refer to Definition 3). In this context, a single model can effectively handle a set of tasks. This approach also facilitates a clear separation from the domain of (parameter-based) TL.

In the context of fine-tuning for downstream tasks facilitated by PFMs, the process typically begins with the pre-training of a backbone foundation model on large data in the wild. This pre-training phase often employs unsupervised or self-supervised methods. Subsequently, the pretrained backbone is fine-tuned using task-specific domain datasets, as illustrated in Fig. 18a. Leveraging the task-unbiased representations acquired from the frozen backbone, fine-tuning of task-specific heads

(e.g., simple MLPs for classification tasks or mask decoders for dense prediction tasks) frequently yields competitive or even superior results when compared to prior supervised outcomes across a spectrum of diverse downstream tasks.

Nonetheless, it is important to note that the pre-training phase tends to restrict data modality due to the constraints of self-supervised techniques, which are inherently data-specific. For instance, methodologies like masked image modeling (MIM) in MAE are suitable for image data, while masked language modeling (MLM) in BERT is tailored for text data. Subsequent review provides an in-depth exploration of downstream task fine-tuning methods categorized by data modality. Specifically, we will discuss these methods within the domains of vision, language, and vision-language tasks.

Vision Tasks. Early pre-training techniques in computer vision primarily focus on learning from pretext tasks. Exemplar CNN (Alexey et al., 2016; Dosovitskiy et al., 2014), for instance, initially pretrains backbone models by discriminating various patches within unlabeled data. In the case of Inpainting (Pathak et al., 2016), the pretext task involves predicting the masked central parts of images. Colorization (R. Zhang et al., 2016), on the other hand, establishes mappings from grayscale images to their colored versions. Split-Brain Autoencoders (R. Zhang et al., 2017) forces the network to split into two disjoint sub-networks, each processing one-half of the input images while predicting the corresponding missing parts from the other sub-network. Recently, BEiT (Bao et al., 2021; Z. Peng et al., 2022) and MAE (K. He et al., 2022) simply reconstruct the random mask patches of the images to pretrain the backbones, i.e., masked image modeling (MIM). Other MIM methods contain iBOT (J. Zhou et al., 2021), CAE (X. Chen, Ding, et al., 2023), SimMIM (Z. Xie et al., 2022), BEVT (R. Wang et al., 2022), ConMIM (Yi et al., 2022), VideoMAE (Tong et al., 2022; L. Wang et al., 2023), to name a few. Jigsaw (Noroozi & Favaro, 2016) and Completing Damaged Jigsaw Puzzles (CDJP) (D. Kim et al., 2018) employ Jigsaw puzzles as pretext tasks during model pre-training. Counting (Noroozi et al., 2017) can also serve as a pretext task to facilitate representation learning. Noise As Targets (NAT) (Bojanowski & Joulin, 2017) focuses on learning representations by aligning the deep features of the backbone with predefined targets in a low-dimensional space. RotNet (Gidaris et al., 2018), however, is designed for predicting different image rotations. Notably, such early pre-training techniques of pretext tasks typically do not require manual annotations, allowing for fast training without the necessity of developing new loss functions. Downstream multiple tasks commonly include classification, object detection, and segmentation.

An alternative line of research aims to design a general representation learning algorithm that is unbiased to the pretext tasks, often referred to as contrastive self-supervised learning (SSL) (Jaiswal et al., 2020; X. Liu, Zhang, et al., 2021). This method unlocks the potential of representations by introducing a novel loss function that hinges on the concept of “contrast.” If we denote the sets of samples that are similar and dissimilar to  $\mathcal{X}$  as  $\mathcal{X}^+$  and  $\mathcal{X}^-$  respectively, the Noise Contrastive Estimation (NCE) loss (Gutmann & Hyvärinen, 2010) can be defined as

$$(2.118) \quad \mathbb{L}_{\text{NCE}} = \mathbb{E}_{\mathcal{X}, \mathcal{X}^+, \mathcal{X}^-} \left[ -\log(e^{f(\mathcal{X})^\top f(\mathcal{X}^+)}) / [e^{f(\mathcal{X})^\top f(\mathcal{X}^+)} + e^{f(\mathcal{X})^\top f(\mathcal{X}^-)}] \right],$$

where the function  $f(\cdot)$  represents the encoder function used to learn image embedding. It is worth noting that the cosine-based similarity measurement mentioned above can be customized to suit various scenarios. Additionally, the InfoNCE loss (Oord et al., 2018) extends this concept by incorporating a more extensive set of dissimilar pairs as

$$(2.119) \quad \mathbb{L}_{\text{InfoNCE}} = \mathbb{E}_{\mathcal{X}, \mathcal{X}^+, \mathcal{X}^b} \left[ -\log(e^{f(\mathcal{X})^\top f(\mathcal{X}^+)}) / [e^{f(\mathcal{X})^\top f(\mathcal{X}^+)} + \sum_{b=1}^{B-1} e^{f(\mathcal{X})^\top f(\mathcal{X}^b)}] \right],$$

where  $B$  represents the batch size, comprising  $B - 1$  negative pairs  $\{(\mathcal{X}, \mathcal{X}^b)\}_{b=1}^{B-1}$  and one positive pair  $(\mathcal{X}, \mathcal{X}^+)$ . These loss functions are closely linked to the maximization of mutual information (MI) between the encoded representations.

Many contrastive SSL methods draw from the loss functions (2.118) and (2.119) to acquire task-invariant representations. Non-parametric instance discrimination (NPID) (Z. Wu et al., 2018) can capture apparent similarity among instances using NCE. In contrast, contrastive predictive coding (CPC) (Henaff, 2020; Oord et al., 2018) first introduces the InfoNCE loss for the pre-training of RNN in an autoregressive manner. Deep InfoMax (DIM) (Hjelm et al., 2018), Deep Graph InfoMax (DGI) (Veličković et al., 2018), and Augmented Multiscale DIM (AMDIM) (Bachman et al., 2019) take a direct approach by maximizing the MI between representations. Contrastive multiview coding (CMC) (Tian et al., 2020) extends the concept of MI maximization to incorporate more than two views, MoCo (X. Chen et al., 2020, 2021; K. He et al., 2020) employs InfoNCE but introduces the momentum contrast based on a memory bank used in (Z. Wu et al., 2018). SimCLR (T. Chen, Kornblith, Norouzi, & Hinton, 2020; T. Chen, Kornblith, Swersky, et al., 2020) proposes a novel contrastive loss known as the normalized temperature-scaled cross-entropy loss (NT-Xent) for representation learning. Bootstrap Your Own Latent (BYOL) (Grill et al., 2020), conversely, takes a different approach by obviating the need for negative pairs. On the other hand, several other methods (Caron et al., 2018, 2020; Goyal et al., 2021; J. Li et al., 2020) endeavor to employ clustering algorithms that contrast data representations based on class prototypes.

Language Tasks. In the domain of language, initial pre-training approaches utilizing word embeddings (Mikolov et al., 2013; Pennington et al., 2014) to predict subsequent tokens for a warm start have shown potential in enhancing the performance of downstream NLP tasks (A. M. Dai & Le, 2015; McCann et al., 2017). Nonetheless, these methods often rely on a limited dataset for pre-training, which restricts their effectiveness and prevents consistently satisfactory outcomes across the spectrum of downstream NLP tasks. Current Transformer-based Pre-trained Foundations Models (PFMs) in natural language processing can be broadly classified into three types (H. Wang et al., 2022): *encoder-only*, *decoder-only*, and *encoder-decoder* architectures. Encoder-only architectures employ a bidirectional Transformer encoder designed to reconstruct masked tokens. Decoder-only models utilize a unidirectional Transformer decoder that predicts tokens in a left-to-right autoregressive fashion. Encoder-decoder models are crafted for sequence-to-sequence (seq2seq) generation tasks, pretrained by masking tokens in the source sequence and predicting them in the target sequence.

This taxonomy aligns with the constraints in terms of tasks. Since the encoder-only architectures, e.g., BERT (Devlin et al., 2018), ERNIE 1.0/2.0 (Y. Sun et al., 2019, 2020), SpanBERT (Joshi et al., 2020), DeBERTa (P. He et al., 2020), and GLaM (Du et al., 2022), are pretrained to predict masked tokens based on the bidirectional context, they are better suited for understanding tasks rather than generation tasks. They are adept at tasks like document classification, named entity recognition, and question answering where the full context is available and the task is to understand or extract information rather than generate it. Encoder-only models often have a fixed maximum sequence length, which limits their ability to handle very long documents directly. They are not designed for incremental token-by-token generation and thus are inefficient for tasks that require such predictions, like text completion or interactive text generation. Conversely, decoder-only architectures, e.g., GPT-3 (Brown et al., 2020), PanGu- $\alpha$  (W. Zeng et al., 2021), Turing-NLG, HyperCLOVA (B. Kim et al., 2021), Gopher (Rae et al., 2021), LaMDA (Thoppilan et al., 2022), PaLM (Chowdhery et al., 2022), Open Pre-trained Transformers (OPT) (S. Zhang et al., 2022),

LLaMA (Touvron et al., 2023), PanGu- $\Sigma$  (Ren et al., 2023) and PaLM-2 (Anil et al., 2023), are pre-trained in a unidirectional context, making them well-suited for generative tasks such as language modeling and text generation. However, this unidirectional training means they may be less effective for tasks that require understanding the full context of the input, as they can only condition on the left context. These models generate one-text token at a time, which can be slower compared to models that handle the entire input at once, and they might struggle with tasks requiring bidirectional context. Encoder-decoder Architectures, e.g., T5 (Raffel et al., 2020), BART (Lewis et al., 2020), ERNIE 3.0 (Y. Sun et al., 2021), Switch Transformers (Fedus et al., 2022) and Flan-T5 (Chung et al., 2022), are more flexible as they can handle both understanding and generation tasks. While they offer considerable advantages in terms of their adaptability to various tasks, they come with trade-offs in terms of model complexity, resource requirements, and potential issues with error propagation.

Vision-Language Tasks. PFMs effectively manage multiple tasks without requiring model tuning. However, the aforementioned methods remain constrained to a unimodal context. In real-world scenarios, there is a natural requirement for multimodal or cross-modal intelligence. Such intelligence should handle multiple tasks across diverse modalities and domains. Vision-Language (VL), as its name implies, bridges CV and NLP. It was among the first areas to be extensively explored by the research community for multi-modal learning in recent years. Given the intricacy and scope of VL tasks, foundation models employing vision-language pre-training (VLP) have rapidly gained prominence, showcasing notable performance. Initial VLP approaches (Y.-C. Chen et al., 2020; W. Kim et al., 2021; J. Li et al., 2021; L. H. Li et al., 2019; Su et al., 2019; H. Tan & Bansal, 2019) centered on task-specific tasks such as visual question answering (VQA), image captioning, visual grounding, etc.

The advent of the contrastive language-image pre-training (CLIP) (Radford et al., 2021), however, marks a significant leap forward in multiple downstream tasks, as it jointly refines dual encoders to align (image, text) pairs within latent embedding space, showcasing learning SOTA multimodal representations from unstructured image-text data. The general representations by cross-modal contrastive learning validate stellar performance in zero-shot transfer across various vision-language (VL) tasks. In a similar trajectory, the Large-scale Image and Noisy-text embedding (ALIGN) (C. Jia et al., 2021) method leverages uncurated data, amplifying the efficacy of VLP in downstream cross-modal retrieval tasks. Other contrastive VLP methods contain ALBEF (J. Li et al., 2021), WenLan (Huo et al., 2021), triple contrastive learning (TCL) (J. Yang et al., 2022), and BLIP (J. Li et al., 2022, 2023). All these methods contribute to the learning of general-purpose visual and linguistic representations, seamlessly adapting to a variety of downstream tasks ranging from cross-modal reasoning (e.g., VQA) and cross-modal matching (e.g., Image Text Retrieval and Visual Referring Expression), to vision and language generation tasks. Notably, DALL·E (Ramesh et al., 2021) stands out in its remarkable capability to perform text-to-image generation tasks in a zero-shot manner, meeting commercial application standards. This underscores the potential and versatility of VLP in facilitating generalist applications.

**Remarks**

- (i) Downstream fine-tuning reduces the data requirements for downstream tasks and also the training (fine-tuning) time and resources.
- (ii) Downstream fine-tuning eases the intensive training burden and enhances the accessibility of PFMs, rendering them a practical solution available to anyone.
- (iii) Downstream fine-tuning necessitates that the data modalities for downstream tasks remain consistent with those pretrained in pretext tasks.
- (iv) Due to PFMs containing pretext task biases, the full potential of multi-task performance remains unrealized.

**2.3.2. Task Prompting.** As the evolution of PFMs advances, the incorporation of prompting into the tuning process of frozen PFMs for downstream tasks has initially become widely recognized through the name of “prompt design” (Brown et al., 2020) and subsequently carried forward through the practice of “prompt tuning.” (Lester et al., 2021) Conceptually, prompts serve as carriers of task-descriptive information, enabling the adaptation of PFMs to various tasks in a manner that can be either manually crafted or automatically generated, as illustrated in Fig. 18b. The primary use of prompts lies in their built-in ability to significantly alleviate the demands of task-specific fine-tuning through freezing backbone parameters of PFMs and only learning task-indicating prompts, ultimately leading to enhanced few-shot or even zero-shot generalizability, all while requiring augmenting inputs and maintaining minimal to no parameter updates. A comprehensive examination of prompt taxonomy exceeds the scope of this section. Consequently, we adopt the notion of task prompting to encompass all prompt engineering methodologies within the framework of task adaptation and generalization.

The additional task-specific prompts augmented with the model can be hard and soft (J. Gu et al., 2023). The hard prompts contain task instructions or hints from human-interpretable natural language, including human instructions (Efrat & Levy, 2020; Radford et al., 2019) in the early stage and more advanced In-Context Learning (ICL) (Q. Dong et al., 2022) and chain-of-thought (CoT) (Chu et al., 2023; Z. Yu et al., 2023). The soft prompts are also referred to as continuous prompting or prompt tuning that optimizes prompts implicitly in the embedding space, which can be learned/propagated to align with specific tasks.

Hard Prompt Engineering. Large Language Models (LLMs), via making predictions based on a few examples in the context, i.e. ICL, can finally perform different tasks. This learning from demonstration and analogy are also presented as emergent abilities (J. Wei, Tay, et al., 2022) in LLMs. GPT-3 (Brown et al., 2020) first verified that LLMs are few-shot learners and that different tasks can be performed given a few examples in the form of demonstration context. InstructGPT (Ouyang et al., 2022) further aligned LLMs with user intent using reinforcement learning from human feedback (RLHF). The developments in ICL contain strategies both in training stage (M. Chen et al., 2022; Y. Gu et al., 2023; Iyer et al., 2022; Min et al., 2022; Y. Wang, Mishra, et al., 2022; J. Wei et al., 2021; J. Wei et al., 2023) and inference stage (Gonen et al., 2022; Hao et al., 2022; Honovich et al., 2022; X. Li & Qiu, 2023; J. Liu et al., 2021; Y. Lu et al., 2022; Rubin et al., 2022; Sorensen et al., 2022; B. Xu et al., 2023; C. Xu et al., 2023; Y. Zhang, Feng, & Tan, 2022; D. Zhou et al., 2022). FLAN (J. Wei et al., 2021) tuned LLMs via natural language instruction templates over 60 NLP tasks and surpassed zero-shot GPT-3 on some of the datasets. MetaICL (Min et al., 2022) introduced meta-training for ICL on a more broad spectrum (100-level) of NLP tasks. Sup-NatInst (Y. Wang, Mishra, et al., 2022) presented a benchmark of 1000-level NLP tasks and proposed Tk-Instruct that can outperform InstructGPT with fewer parameters. OPT-IML (Iyer

et al., 2022) Scales LLMs instruction meta-learning to 2000 NLP tasks through the lens of generalization. Symbol Tuning (J. Wei et al., 2023) targets the situation when instructions or natural language are insignificant in predicting the task. PICL (Y. Gu et al., 2023) enhanced the ICL ability for LLMs by pre-training to maintain task generalization, while previous investigations are how to select in-context examples for better few-shot capabilities during the testing stage (J. Liu et al., 2021). Other methods (Gonen et al., 2022; Hao et al., 2022; Honovich et al., 2022; X. Li & Qiu, 2023; Y. Lu et al., 2022; Sorensen et al., 2022; B. Xu et al., 2023; C. Xu et al., 2023; Y. Zhang, Feng, & Tan, 2022; D. Zhou et al., 2022) tried to understand why the performance varies from different prompts and how to pick better prompts from different angles. After prompt retriever (Rubin et al., 2022) is verified efficient for ICL, many efforts used the prompt pool as a tool to support retrieval-based prompting, where relevant prompts or context are retrieved for ICL (X. Li et al., 2023; Rubin et al., 2021; J. Ye et al., 2023; Y. Zhang, Zhou, & Liu, 2023).

Furthermore, chain-of-thought (CoT) prompts are a series of instructions with progressive orders, which can help LLMs perform complex reasoning tasks step by step (W. Chen et al., 2022; Y. Fu et al., 2022; Ho et al., 2022; Kojima et al., 2022; Trivedi et al., 2022; J. Wei, Wang, et al., 2022; Z. Zhang et al., 2022). Manual-CoT (J. Wei, Wang, et al., 2022) first explores how to improve the ability of LLM by generating CoT. Zero-Shot-CoT (Kojima et al., 2022) proposes a single task-agnostic zero-shot prompt to surpass ICL even without input-output demonstrations. Complex-CoT (Y. Fu et al., 2022) shows that complex reasoning chains excel simple chains. Auto-CoT (Z. Zhang et al., 2022) mitigates the mistakes that could happen in precious manual ways by automatically constructing demonstrations for different questions. Fine-tune-CoT (Ho et al., 2022) can use teacher-generated reasoning to fine-tune smaller models. IRCoT (Trivedi et al., 2022) interleaves retrieval with steps and, in turn, improves the ability of CoT by retrieved results. PoT (W. Chen et al., 2022) uses programming language statements to delegate math computations.

Soft Prompt Tuning. In comparison, soft prompt tuning can backpropagate prompt vectors using gradient descent. Lester et al. (2021) introduces the concept of “prompt tuning” and distinguishes it from previous model tuning and prompt design methods. During the training, prompt tuning can refine the prompts to improve learning performance on specific tasks. Thus, the multi-task setting can be realized by simply mixing training data across different tasks. Soft Prompt Transfer (SPoT) (T. Vu et al., 2021) pioneers the demonstration that prompt tuning can efficiently transfer from source to target tasks, offering a parameter-efficient approach to prompt-based transfer learning across diverse tasks. P-Tuning (X. Liu et al., 2022) empirically optimizes prompt tuning to be universally effective across a wide range of tasks. ATTENTIONal Mixtures of Prompt Tuning (ATTEMPT) (Asai et al., 2022) exemplifies this concept by combining multiple prompts trained on large-scale source tasks, generalizing instance-wise prompts on target tasks while keeping model parameters and source prompts frozen. Multi-task Pre-trained Modular Prompt ( $MP^2$ ) (T. Sun et al., 2023) enhances FSL for prompt tuning in multi-task settings. J. Liu et al. (2023a) is the first to showcase that prompt learning achieves SOTA performance for MTL in FSL settings, even surpassing ChatGPT. Hierarchical Prompt (HiPro) learning (J. Liu et al., 2023b) evaluates prompt tuning on standard MTL datasets and outperforms SOTA MTL methodologies by learning task-shared and task-individual prompts. Multitask Vision-Language Prompt Tuning (MVLPT) (Shen et al., 2024) incorporates cross-task knowledge into learning a single transferable prompt for vision-language models (VLMs). Prompt Guided Transformer (PGT) (Y. Lu et al., 2024) introduces a prompt-conditioned Transformer block, integrating task-specific prompts into the self-attention mechanism, achieving global dependency modeling and parameter-efficient feature adaptation across multiple

tasks. PromptonomyViT (PViT) model, as introduced in Herzog et al. (2024), leverages prompts to capture task-specific information in video Transformers.

Prefix-tuning X. L. Li and Liang (2021) is another lightweight alternative to fine-tune LLMs for different tasks while also keeping model parameters frozen. Prefix-tuning learns a continuous task-specific vector prefixed to the subsequent tokens. It can obtain comparable performance in the full data setting and outperform fine-tuning in low-data settings. Y. Chen et al. (2022) proposes a Unified few-shot Summarization (UniSumm) model pretrained on multiple text summarization tasks, which exhibits the capability to generalize to different few-shot tasks through the utilization of prefix-tuning. Chong et al. (2023) trains a prefix transfer module to selectively leverage the knowledge from various prefixes according to the input text. Collaborative domain-Prefix tuning for cross-domain NER (CP-NER) (X. Chen, Li, et al., 2023) utilizes text-to-text generation, grounding domain-related instructions to transfer knowledge to new domain tasks. Prefix-tuning approaches highlight the importance of leveraging prefixes and domain-specific information for improving performance in multiple tasks.

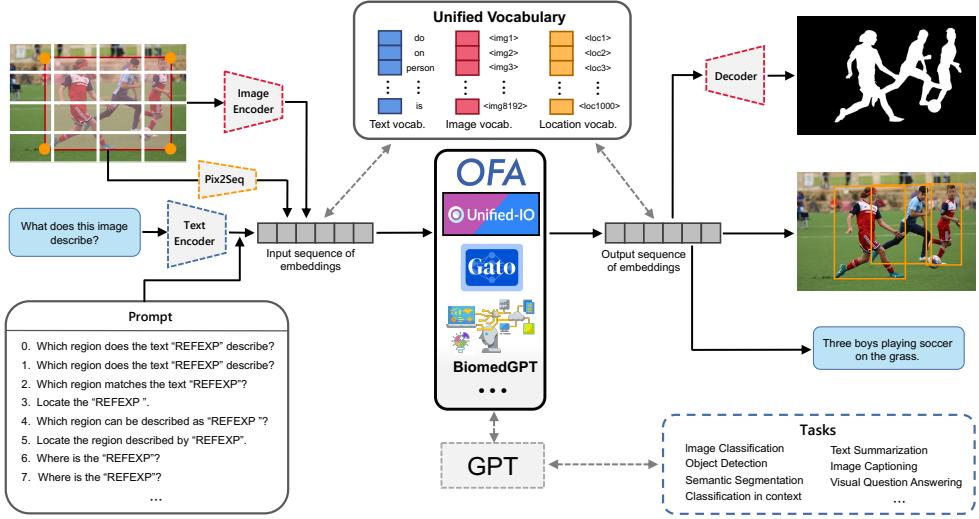
### Remarks

- (i) Task prompting stands out as highly parameter-efficient, demanding fewer than 0.01% of task-specific parameters even for models exceeding a billion parameters (Lester et al., 2021).
- (ii) Task-specific prompts exhibit a remarkable degree of adaptability, affording the capacity for on-the-fly customization to accommodate a diverse set of tasks, thus enhancing the flexibility in managing a multitude of heterogeneous tasks simultaneously.
- (iii) Task prompting facilitates the achievement of few-shot and even zero-shot learning, empowering PFMs to effectively perform tasks with minimal to no examples.
- (iv) Researchers/practitioners can have fine-grained control over how the model performs different tasks, as prompts can be customized to guide the model behavior precisely.
- (v) The prompt itself is not transferable across different PFMs, thus leading to the limitations of scalability and reusability of prompt designs.
- (vi) Human involvement in prompting, e.g., crafting prompts or selecting appropriate templates, is time-consuming and bias-inducing.

**2.3.3. Unified Generalist Models.** The ambitious aspiration, shared by both research communities and industries, has always been to transition from specialization to unification, thereby constructing an ideal generalist model capable of addressing a diverse set of tasks with varying modalities. The advent of large language models (LLMs)

The blueprint of designing general-purpose multimodal foundation models aligns with the recent unified models such as Gato (Reed et al., 2022), Unified-IO (J. Lu et al., 2022), and OFA (P. Wang et al., 2022), Uni-Perceiver (H. Li et al., 2023; X. Zhu et al., 2022b), etc. These methods can perform a variety of tasks spanning from CV to NLP, without modality limitations. Please see Fig. 19 as an illustration.

To pretrain via a Transformer backbone for the general MTL usage, we need to tokenize the input multi-modal data. For images, the common practice should obey the sequencing of non-overlapping  $16 \times 16$  patches in raster order in ViT (Dosovitskiy et al., 2020), with the size of 256/16 for each patch. Typically, the bounding boxes of objects in region-based tasks are represented by the quantization scheme of Pix2Seq (T. Chen, Saxena, Li, Fleet, & Hinton, 2022). In the text preprocessing, the OFA framework adopts the exact same BPE Tokenizer (Sennrich et al., 2015) used in BART (Lewis et al., 2020), and its tokens are originally ordered along with the raw input text. Based on this prepossessing, it is possible to build a unified vocabulary for all visual, linguistic, and multi-modal tokens. After that, suppose we are given a sequence of tokens  $\mathbf{x}_{i,b}$  as input, where



**Figure 19.** The Framework of unified generalist model, which can unify architectures, tasks, and modalities through a simple seq2seq learning architecture.

$i = 1, \dots, I$  indexes the tokens in a data sample and  $b = 1, \dots, B$  indexes a sample in a training batch. The architecture for a unified model is parametrized by  $\theta$ . Then we are able to autoregressively train the model via the chain rule as follows:

$$(2.120) \quad \mathcal{L}_\theta(\mathbf{x}_{1,1}, \dots, \mathbf{x}_{i,b}) = \sum_{b=1}^B \log \prod_{i=1}^I p_\theta(\mathbf{x}_{i,b} | \mathbf{x}_{1,b}, \dots, \mathbf{x}_{i-1,b}) = \sum_{b=1}^B \sum_{i=1}^I \log p_\theta(\mathbf{x}_{i,b} | \mathbf{x}_{<i,b})$$

### Remarks

- (i) The unified generalist model allows for modality-agnostic and task-agnostic learning, overcoming the limitations inherent to specific tasks. This implies that any task can be modeled into an omnivorous model.
- (ii) The unified generalist model achieves parameter efficiency and saves storage space in terms of many tasks.
- (iii) The unified generalist model is pre-trained using multimodal data all at once but possesses enduring utility.

The concept of a unified architecture for multi-modal MTL can be traced back to OmniNet (Pramanik et al., 2019), taking insights from the potentials of Transformers such as, Pramanik et al. (2019) propose a single model in their work to support tasks with multiple input modalities as well as asynchronous MTL. J. Lu et al. (2020) investigates the relationships between vision-language (VL) tasks, and proposes a single model targeting 12 datasets simultaneously. Q. Li et al. (2021) introduces the concept of unified foundation models by jointly pre-training Transformers on unpaired images and text data. Unified Transformer (UniT) model (R. Hu & Singh, 2021) is a realization of this concept. It first features separate encoders for different input modalities and a shared decoder over the encoded input representations. Each task is associated with specific heads in the shared decoder. Unified Foundation Model Bai et al. (2022) and P. Wang et al. (2022) proposes One-for-All (OFA) as a task-agnostic and modality-agnostic framework. OFA aims to unify task-specific layers for downstream tasks, providing a versatile solution. However, it is important to note that OFA currently lacks support for video data and necessitates fine-tuning for downstream tasks. Uni-Perceiver (X. Zhu et al., 2022b) is a unified architecture for generic perception for zero-shot

and few-shot tasks, which includes a video tokenizer with temporal positional embeddings. UniPerceiver v2 (H. Li et al., 2023) further introduces task-balanced gradient normalization to ensure stable MTL, which enables larger batch-size training for various tasks. More importantly, unlike OFA (P. Wang et al., 2022), Uni-Perceiver v2 requires no task-specific adaptation. Mask DETR with Improved deNoising anchOr boxes (Mask DINO) (F. Li et al., 2023) is a unified framework designed for object detection and segmentation. Mask DINO uses an additional mask prediction branch to unify the query selection for masks. All-in-one Transformer (J. Wang et al., 2023) unifies video and text encoders via introducing a token rolling operation to encode temporal representations from videos. Omnivorous Masked Auto-Encoder(OmniMAE) (Girdhar et al., 2023) shows that MAE can be used to pretrain a ViT on images and videos without any human labels. OmniVec (Srivastava & Sharma, 2024) also pretrains a unified architecture from self-supervised masked data, including visual, audio, text, and 3D, which realizes the cross-modal task generalization.

### 3. MISCELLANEOUS

#### 3.1. Fairness and Bias in MTL.

While most of the existing research about bias and fairness implications primarily focuses on STL (Mehrabi et al., 2021), Y. Wang et al. (2021) pioneer the exploration of the fairness-accuracy trade-off within the MTL setting. The challenge of unaligned fairness goals arises in MTL models that optimize accuracy for all tasks. The introduction of novel multi-task fairness metrics, such as average relative fairness gap and average relative error, aids in quantifying this trade-off in MTL applications. Y. Li and Oymak (2023) emphasize that misspecification of majority and minority groups in involved tasks disproportionately affects minority tasks, and they propose over-parameterization as a viable solution to achieve fairness by covering all tasks. F. Hu et al. (2023) extend the definition of Strong Demographic Parity (Agarwal et al., 2019; Jiang et al., 2020) to MTL using multi-marginal Wasserstein barycenters (Chzhen et al., 2020), providing an optimal fair multi-task solution to the fairness-accuracy trade-off. Additionally, Roy and Ntoutsi (2022) further demonstrates that improving fairness can positively impact accuracy performance. Learning to Teach Fair Multi-Tasking (L2T-FMT) (Roy & Ntoutsi, 2022) introduces a teacher-student network to address fair MTL problems. In this framework, the teacher guides the student in selecting fairness or accuracy objectives during training, offering a dynamic approach to balancing these objectives. Drawing an analogy, Roy et al. (2023) liken the negative impact of task-specific fairness to negative transfer and introduces FairBranch, a method that groups related tasks to mitigate this negative transfer through fairness loss gradient conflict correction. In recent years, prioritizing fair MTL to mitigate biases arising from negative transfer has emerged as a promising direction. This approach can ensure that models treat all tasks fairly, avoiding disproportionate impacts on specific groups or tasks. By preventing biased outcomes, fair MTL contributes to averting potential societal harm.

#### 3.2. Security and Privacy in MTL.

*Attack and Defense.* MTL is an impactful technique employed to bolster attacks in diverse sectors. It notably expedites the creation of adversarial examples for numerous tasks simultaneously through the exploitation of task-shared knowledge (P. Guo, Xu, et al., 2020). In the field of automatic speaker verification, multi-task learning strategies have been utilized to identify replay attack spoofing and to classify different types of replay noise (Shim et al., 2018). With regard to reinforcement learning, the vulnerability of multi-task federated reinforcement learning algorithms to adversarial attacks has been examined, resulting in the development of an adaptable attack method

and a refined federated reinforcement learning algorithm (Anwar & Raychowdhury, 2021). Additionally, within the realm of deep reinforcement learning, a multi-objective strategy for developing attack policies has been suggested, considering both the performance degradation and the cost related to the attack (García et al., 2020). Conversely, MTL can also serve as a means to heighten the model's resilience, leading to an improved defense against a wide array of malicious attacks. For instance, the robustness of models to adversarial attacks on individual tasks has been shown to increase when models are trained on multiple tasks concurrently (P. Guo, Xu, et al., 2020; C. Mao et al., 2020). Likewise, multi-task learning has been employed for adversarial defense (Naseer et al., 2022), using supplementary data from the feature space to design more formidable adversaries and boost the model's resilience. Through the utilization of multi-task objectives, such as cross-entropy loss, feature-scattering, and margin losses, more powerful perturbations can be devised for adversarial training. This technique has been used in several domains, such as computer vision and speech recognition, and has demonstrated enhanced adversarial accuracy and resilience (Chan et al., 2021; Pal et al., 2021).

*Privacy-preserving.* Privacy-preserving multi-task learning (PP-MTL) (K. Liu et al., 2018) aims to ensure the confidentiality of sensitive data and boost learning outcomes by facilitating knowledge transfer across related tasks. PP-MTL algorithms employ cryptographic mechanisms to safeguard data residing across various locations or nodes, using these to relay cumulative data - for instance, gradients or supports - to a centralized server where the aggregated data is processed to create the desired models. Existing strategies cannot deliver a demonstrable or verifiable security assurance for the transferred cumulative data. To tackle this shortcoming, various innovative PP-MTL protocols have been suggested, leveraging cutting-edge cryptographic methods to deliver the strongest possible security assurance (K. Liu et al., 2018). Furthermore, differential private stochastic gradient descent algorithms have been employed to optimize the comprehensive multi-task model and safeguard the privacy of training data by introducing appropriately calibrated noise to the gradient of loss functions (C. Zhang et al., 2020). To maintain the privacy of distributed data, privacy-preserving distributed MTL frameworks have been introduced, incorporating a privacy-preserving proximal gradient algorithm. This algorithm updates models asynchronously and offers guaranteed differential privacy (L. Xie et al., 2017).

*Federated Learning.* Federated Multi-task Learning (FMTL) (Smith et al., 2017) represents a platform for training machine learning models over distributed device networks. By personalizing models for individual clients, it successfully navigates the statistical complexities posed by federated learning, given the heterogeneity of local data distributions (Smith et al., 2017). It effectively manages high communication overhead, lags, and reliability in distributed multi-task learning (Marfoq et al., 2021). The efficacy of FMTL has been demonstrated on real-world federated datasets, even with non-convex models (SarcheshmehPour et al., 2021). It can be utilized in both a central server-client and a fully decentralized structure and provides the capacity to serve personalized models to clients unseen during training (Corinzia et al., 2019). Furthermore, the over-the-air computation can be integrated within FMTL to enhance system efficiency, reducing channel usage without a substantial drop in learning performance (H. Ma et al., 2022).

### 3.3. Distribution Shifts in MTL.

While Multi-Task Learning (MTL) excels at leveraging shared information to boost individual task performance (1.3), its real-world applicability often hinges on its ability to adapt to unforeseen data distributions. Distribution shifts, where the data encountered during deployment deviates from the training distribution, are omnipresent challenges that can significantly degrade MTL performance, especially on new tasks or domains. Recognizing and mitigating these shifts is crucial not just

for maintaining the generalizability and resilience of MTL models but also for unlocking their full potential in real-world applications.

Recent research offers a diverse arsenal of approaches to tackle distribution shifts in MTL. Vision Transformer Adapters (ViTA) (Bhattacharjee et al., 2023) introduce dedicated modules within the model architecture that enhance adaptability to diverse tasks and data distributions. Techniques like regularizing spurious correlations (Z. Hu et al., 2022) target misleading associations between tasks, reducing their influence on the overall model performance. Scalarization methods provide a scalable framework for handling the complexities of multi-task and multi-domain learning while facing distribution shifts (Royer et al., 2023). Multi-objective learning strategies, exemplified by approaches addressing catastrophic forgetting in time-series applications (Mahmoud & Hajj, 2022), strive to mitigate the issue of forgetting previously learned skills when encountering new data. Finally, techniques like reward modeling (Faal et al., 2023) demonstrate their versatility in addressing distribution shifts, as seen in mitigating toxicity issues in transformer-based language models. This array of advancements underscores the ongoing efforts to equip MTL models with enhanced adaptability and resilience to varying task distributions, ultimately paving the way for their reliable and widespread real-world application.

Looking ahead, the evolving landscape of MTL research envisions models that not only react to distribution shifts but proactively anticipate and address them. As highlighted in a recent comprehensive study (Adhikarla et al., 2023), understanding and mitigating distribution shifts are becoming paramount for MTL’s success. The ability to navigate diverse and dynamic data distributions is crucial for the broader deployment of MTL in complex, real-world scenarios. By advancing techniques that enhance adaptability and robustness, researchers are striving to empower MTL models to excel in the face of evolving task and domain landscapes, unlocking their potential to revolutionize a wide array of applications.

### 3.4. Non-supervised MTL.

*semi-supervised learning.* Supervised learning has been a fundamental technique in machine learning in recent years. However, it faces the limitation of requiring a substantial amount of labeled data to yield promising results, a process that is both time-consuming and costly. To mitigate this, semi-supervised learning has been introduced, leveraging the diverse array of unlabeled datasets to reduce the dependence on labeled data. Previous existing semi-supervised algorithms are not often amenable to MTL, for instance, (Q. Liu et al., 2007) introduces a semi-supervised multitask learning (MTL) framework, featuring  $M$  parameterized classifiers. Each classifier is associated with a partially labeled data manifold and is jointly learned under a soft-sharing prior that influences their parameters. This approach effectively utilizes unlabeled data by basing the learning of classifiers on neighborhood structures. Besides, (Augenstein et al., 2018) presents a method that models the relationship between labels by inducing a joint label embedding space for multi-task learning and proposes a *TransferNetwork* which learns to transfer labels between tasks and uses semi-supervised learning to leverage them for training. In real-world applications, multi-task regression is a prevalent challenge. (Y. Zhang & Yeung, 2009) proposes the SMTR method, which is grounded in Gaussian Processes (GP). This method operates under the assumption that the kernel parameters for all tasks share a common prior. To enhance SMTR, the approach incorporates unlabeled data by modifying the GP prior’s kernel function into a data-dependent one. This modification leads to a semi-supervised extension of the original SMTR method, aptly named SSMTR. Additionally, (Z. Chen, Zhu, et al., 2020) introduces a multi-task mean teacher model for semi-supervised shadow detection, effectively utilizing unlabeled data and simultaneously learning multiple aspects of shadows. Specifically, they construct a multi-task baseline model designed to detect shadow regions,

edges, and count, leveraging the complementary information of these elements. This baseline model is then implemented in both student and teacher networks. The approach further involves aligning the predictions from the three tasks across these networks, using this alignment to compute a consistency loss on unlabeled data. This loss is combined with the supervised loss from labeled data based on the predictions of the multi-task baseline model, thereby enhancing the model's learning effectiveness. (Nguyen et al., 2019) proposed a network employing a multi-task learning approach to detect manipulated images and videos and to identify the manipulated regions within each query. To enhance the network's generalizability, a semi-supervised learning approach is integrated in which the architecture comprises an encoder and a Y-shaped decoder. The activation of encoded features facilitates binary classification. Meanwhile, the outputs of the decoder's branches serve distinct purposes: one for segmenting the manipulated regions and the other for reconstructing the input. This dual functionality significantly contributes to the improvement of the overall performance of the network. Semi-supervised multitask learning (MTL) has emerged as a popular field, with various preceding studies, as mentioned above, that propose different mechanisms that integrate semi-supervised concepts. These studies have demonstrated their effectiveness through numerous experimental results. Despite these advancements, there remains a substantial scope for further research in this subfield. Continued exploration in semi-supervised MTL promises to yield many more valuable insights and findings.

*unsupervised learning.* Moving beyond the realm of semi-supervised learning, the real-world often presents scenarios where obtaining labeled data of all tasks in MTL learning is not feasible, underscoring the significance of unsupervised learning in the field of multitask learning (MTL). OpenAI, in their groundbreaking study by (Radford et al., 2019), introduced the widely acclaimed GPT model, demonstrating a significant advancement in multitask learning (MTL) within the field of natural language processing. Their research showed that language models begin to autonomously learn a variety of MTL tasks - including question answering, machine translation, reading comprehension, and summarization - without the need for explicit supervision. This capability was notably observed when the GPT model was trained on *WebText*, a vast new dataset comprising millions of webpages. This development highlights a major stride in the field, showcasing the potential of large language models to adapt to a wide array of tasks through extensive unsupervised learning. Besides, to alleviate the limitation of existing clustering approaches that neglect the underlying relationship and treat these clustering tasks either individually or simply together, the study by (Q. Gu & Zhou, 2009a) introduces an innovative clustering approach called *Multi-taskclustering*, which conducts several related clustering tasks concurrently and leverages the relationships between these tasks to improve clustering performance. This approach comprises two key components: (1) Within-task clustering, which involves clustering the data for each task individually within its own input space, and (2) Cross-task clustering, where the shared subspace is learned simultaneously, and the data from all tasks are clustered together. This dual-faceted strategy optimizes the clustering results by combining individual task insights with cross-task synergies. Another notable example is in the context of point cloud tasks, where (Hassani & Haley, 2019) introduces an unsupervised multi-task model. This model is designed to concurrently learn point and shape features. It incorporates three unsupervised tasks: clustering, reconstruction, and self-supervised classification. These tasks are used to train a multi-scale graph-based encoder. Beyond, (Argyriou et al., 2006) introduces a method for learning a low-dimensional representation shared across multiple related tasks. This method extends the well-known 1-norm regularization problem by incorporating a novel regularizer that controls the number of features common to all tasks. The authors demonstrate that this approach can be formulated as a convex optimization problem and develop an iterative algorithm

to solve it. The algorithm operates in a dual-step manner: it alternates between a supervised step and an unsupervised step. In the unsupervised step, it learns representations common across tasks, while in the supervised step, it utilizes these common representations to learn task-specific functions. This approach effectively combines supervised and unsupervised learning techniques to enhance multi-task learning.

### 3.5. Others.

**3.5.1. Applications of MTL.** In the deep learning era, the advancement of multimodal analysis and MTL paradigms has brought challenges and also opened up fantastic probabilities to the realm of MTL. In addition to the applications investigated in the paper, MTL plays an important role in many different fields such as visual assessment (J. Yu et al., 2019; W. Zhang et al., 2023), healthcare(K. Zhang et al., 2023; Y. Zhang, Wu, et al., 2023; Y. Zhao et al., 2023), transportation(Feng et al., 2023; H. Wang et al., 2023), language models (R. Hu & Singh, 2021; F. Liu et al., 2020) and recommender systems(Y. Deng et al., 2023; M. Zhang et al., 2023). Briefly,W. Zhang et al. (2023) develop a general and automated multitask learning scheme for image quality assessment by blind individuals. N. Zeng et al. (2023) combine MTL algorithms with a deep belief network for the diagnosis of Alzheimer’s disease. Wang *et al.* (H. Wang et al., 2023) propose a multi-task Weakly supervised learning framework to infer transition probability between road segments. M. Gao et al. (2023) utilize the relation-aware GCNs to fully capture the multi-relation neighborhood features.

Despite the achievements in recent years, many outstanding MTL approaches still suffer from limitations that restrict their application to certain real-world scenarios. For example, it is difficult to capture the complex inter-scenario correlations with multiple tasks. Besides, in large-scale tasks, it remains a challenge to design scalable models and deal with the parameter explosion issue. Therefore, the scalability of MTL models is still a direction worth exploring (M. Zhang et al., 2023).

**3.5.2. MTL+X. MTL + Continual Learning.** Biased forgetting of previous knowledge caused by new tasks remains challenging in continual learning. Lyu et al. (2021) propose Multi-Domain Multi-Task (MDMT) rehearsal to train the old tasks and new tasks together while keeping tasks from isolation. X. He et al. (2019) utilize meta-learning to achieve task-agnostic continual learning. MTL is a promising technique to mitigate catastrophic forgetting via learning task-relatedness.

**Multi-Task Reinforcement Learning (MTRL).** MTRL (Vithayathil Varghese & Mahmoud, 2020) holds promise in the context of Reinforcement Learning (RL), given the natural presence of diverse tasks like reach, push and pick in robotic manipulation. In the early stage, Wilson et al. (2007) approaches it as the solution to a sequence of Markov Decision Processes (MDPs) and employs a hierarchical Bayesian framework to infer the characteristics of new environments based on knowledge gained from previous environments. Hessel et al. (2019) introduce a method to automatically adjust the contribution of each task to the updates of a single agent. This ensures that all tasks exert a similar impact on the learning dynamics. Taiga et al. (2022) investigates multi-task pretraining and generalization in RL. Cheng et al. (2023) propose an attention-based multi-task reinforcement learning approach to learn a compositional policy for each task.

## 4. RESOURCES

In this section, we offer useful tools and resources that can help researchers and practitioners implement MTL models.

**Table 8.** Summary of common datasets used in MTL.

| Dataset                  | Source                          | Year     | Modality    | Task             | Synopsis  | #Task | #Sample     | Availability |
|--------------------------|---------------------------------|----------|-------------|------------------|---|-------|-------------|--------------|
| School Data              | ILEA                            | 1988     | Table       | Regression       | Predicting student exam scores based on 27 school features.           | 139   | 15,362      | Official     |
| SARCOS Data              | Humanoid Robotics               | 2000     | Table       | Regression       | Estimate inverse dynamics model.                                      | 7     | 44,484/4449 | Official     |
| Computer Survey Data     | Survey                          | 1996     | Table       | Regression       | Likelihood of purchasing personal computers.                          | 179   | -           | -            |
| Climate Dataset          | Sensor network                  | 2017-now | Table       | Regression       | Real-time climate data collected from four climate stations.          | 7     | -           | Official     |
| 20 Newsgroups            | Netnews articles                | 1995     | Text        | Classification   | Hierarchical text classification.                                     | 20    | 19,000      | Official     |
| Reuters-21578 Collection | Reuters                         | 1996     | Text        | Classification   | Reuters news documents with hierarchical categories.                  | 90    | 21,578      | Official     |
| MultiMNIST Dataset       | MNIST                           | 2017     | Image       | Classification   | Classify the digits on the different positions.                       | 2     | -           | Official     |
| ImageCLEF-2014           | Caltech, ImageNet, Pascal, Bing | 2014     | Image       | Classification   | Benchmark dataset for domain adaptation.                              | 4     | 2,400       | Official     |
| Office-Caltech Dataset   | Office, Caltech                 | 2012     | Image       | Classification   | Benchmark dataset for the annotation and retrieval of images.         | 4     | 2,533       | Official     |
| Office-31 Dataset        | Amazon, DSLR, Webcam            | 2010     | Image       | Classification   | Objects commonly encountered in office settings.                      | 3     | 4,110       | Official     |
| Office-Home Dataset      | Office                          | 2017     | Image       | Classification   | Object recognition and domain adaptation in the era of deep learning. | 4     | 15,588      | Official     |
| DomainNet Dataset        | UDA                             | 2019     | Image       | Classification   | Multi-source unsupervised domain adaptation research                  | 6     | 600,000     | Official     |
| EMMAs Dataset            | Amazon                          | 2023     | Image, Text | Classification   | Amazon product listings for category prediction                       | -     | 2,800,000   | Official     |
| SYNTHIA Dataset          | European Union                  | 2016     | Image       | Classification   | A synthetic dataset for semantic segmentation.                        | -     | 13,400      | Official     |
| SVHN Dataset             | Stanford                        | 2021     | Image       | Classification   | A digit classification benchmark dataset.                             | -     | 600,000     | Official     |
| CelebA Dataset           | MMLAB                           | 2018     | Image       | Classification   | A large-scale face attributes dataset.                                | 40    | 200,000     | Official     |
| CityScapes Dataset       | Daimler AG                      | 2016     | Image       | Dense prediction | Semantic urban scene understanding                                    | -     | 5,000       | Official     |
| NYU-Depth Dataset V2     | New York University             | 2012     | Image       | Dense prediction | Indoor scene understanding with per-pixel labels                      | 3     | 35,064      | Official     |
| PASCAL VOC Project       | University of Oxford            | 2010     | Image       | Dense prediction | Object recognition with multiple tasks                                | -     | -           | Official     |
| Taskonomy Dataset        | Standard                        | 2018     | Image       | Dense prediction | Diverse dataset with 26 tasks for task transfer learning              | 26    | 4,000,000   | Official     |
| STREET                   | Amazon                          | 2023     | Text        | Reasoning        | The multi-task structured reasoning and explanation benchmark         | -     | -           | -            |
| VKITTI2 Dataset          | Naver                           | 2020     | Video       | Segmentation     | A video dataset which is automatically labeled with ground truth      | 5     | -           | Official     |
| XTREME                   | Carnegie Mellon                 | 2020     | Text        | Translation, QA  | A multilingual benchmark for evaluating cross-lingual generalisation  | 9     | 400,000     | -            |
| Deepfashion Dataset      | Shopping Websites               | 2016     | Image       | Classification   | A large-scale clothes dataset with comprehensive annotations          | 2     | 800,000     | Official     |
| ACE05 Dataset            | News                            | 2005     | Text        | Classification   | A large corpus with annotated entities, relations and events          | 3     | 52,615      | Official     |
| ATIS Dataset             | Airline                         | 1990     | Text        | Classification   | A dataset with 17 unique intent categories.                           | 3     | 5,871       | Official     |

#### 4.1. Dataset.

In this section, we introduce benchmark datasets for MTL from a taxonomic perspective. Specifically, based on the different datasets spawning a series of typical data-driven models, we classify many MTL datasets into three categories: regression task, classification task, and dense prediction task.

**4.1.1. Regression task. Synthetic Data.** This dataset is often artificially defined by researchers, thus different from one another, e.g. Argyriou et al. (2008), Bakker and Heskes (2003), R. Caruana (1997), Evgeniou and Pontil (2004), Han and Zhang (2016), Jalali et al. (2010), J. Ma et al. (2018), Maurer et al. (2013), Nie et al. (2018), Parra and Tobar (2017), Titsias and Lázaro-Gredilla (2011), Y. Zhang and Yeung (2012a), and J. Zhou, Chen, and Ye (2011), to name a few. The features are often generated via drawing random variables from a shared distribution and adding irrelevant variants from other distributions, and the corresponding responses are produced by a specific computational method. In such a manner, data in different tasks would contain both the task-specific and -shared features that contribute to the learning for estimation.

**School Data.** Mortimore et al. (1988) comes from the Inner London Education Authority (ILEA) and contains 15,362 records of student examination, which are described by 27 student- and school-specific features from 139 secondary schools. The goal is to predict exam scores from 27 features, and the prediction in 139 schools would be generally handled as 139 tasks.

**SARCOS Data.**<sup>14</sup> This dataset is in humanoid robotics consists of 44,484 training examples and 4,449 test examples. The goal of learning is to estimate the inverse dynamics model of a 7 degrees-of-freedom (DOF) SARCOS anthropomorphic robot arm, each of which corresponds to a task and contains 21 features—7 joint positions, 7 joint velocities, and 7 joint accelerations.

**Computer Survey Data.** Lenk et al. (1996) is from a survey on the likelihood (11-point scale from 0 to 10) of purchasing personal computers. There are 20 computer models as examples, each of which contains 13 computer descriptions (e.g., price, CPU speed, and screen size) and 6 subject-level covariates (e.g., gender, computer knowledge, and work experience) as features and ratings of 179 subjects as targets, i.e., tasks. **Climate Dataset.**<sup>15</sup> This real-time dataset is collected from a

<sup>14</sup>2000. SARCOS Data. <http://gaussianprocess.org/gpml/data>

<sup>15</sup>2017-now. Climate Dataset. <https://www.cambermet.co.uk>

sensor network (e.g., anemometer, thermistor, and pressure transducer) of four climate stations—Cambermet, Chimet, Sotonmet and Bramblemet—in the south of England, which can represent 4 tasks as needed. The archived data are reported in 5-minute intervals, including  $\sim 10$  climate signals (e.g., wind speed, wave period, barometric pressure, and water temperature). Generally, air temperature is considered as the dependent variable and others as independent (Parra & Tobar, 2017; J. Zhao et al., 2019).

**4.1.2. Classification task. 20 Newsgroups.** Lang (1995) is a collection of approximately 19,000 net-news articles, organized into 20 hierarchical newsgroups according to the topic, such as root categories (e.g., comp, rec, sci, and talk) and sub-categories (e.g., comp.graphics, sci.electronics, and talk.politics.guns). Users can design different combinations as multiple text classifications tasks (J. He & Lawrence, 2011; Y. Mao et al., 2020; B. Tan et al., 2015; Xiao et al., 2020; X. Zhang et al., 2018).

**Reuters-21578 Collection.**<sup>16</sup> This text collection contains 21578 documents from Reuters newswire dating back to 1987. These documents were assembled and indexed with more than 90 correlated categories—5 top categories (i.e., exchanges, orgs, people, place, topic), and each of them includes variable sub-categories. Users can independently define the related multiple tasks by choosing different combinations of categories, e.g., Xiao et al. (2021) and X. Zheng et al. (2020) provide more detailed descriptions.

**CelebA Dataset.** CelebFaces Attributes Dataset (CelebA) (Z. Liu et al., 2018) is a large-scale face attributes dataset with more than 200K celebrity images, each with 40 attribute annotations. The images in this dataset cover large pose variations and background clutter. CelebA has large diversities, large quantities, and rich annotations, including 10,177 identities, 202,599 face images, and 5 landmark locations, 40 binary attribute annotations per image. The dataset can be employed as the training and test sets for the following computer vision tasks: face attribute recognition, face recognition, face detection, landmark (or facial part) localization, and face editing & synthesis.

**MultiMNIST Dataset.** This dataset originated from validating a capsule system (Sabour et al., 2017), but it is also a MTL version of MNIST dataset (LeCun et al., 1998). By overlaying multiple images together, traditional digit classification is converted to an MTL problem, where classifying the digits in different positions is considered as distinctive task. Sener and Koltun (2018) contributes a standard construction for the research community. **ImageCLEF-2014 Dataset.**<sup>17</sup> This dataset is a benchmark for domain adaptation challenge, which contains 2,400 images of 12 common categories selected from 4 domains: Caltech 256, ImageNet 2012, Pascal VOC 2012, and Bing. These 4 domains are commonly considered as different tasks in MTL.

**Office-Caltech Dataset.** B. Gong et al. (2012) is a standard benchmark for domain adaption in computer vision, consisting of real-world images of 10 common categories from the Office dataset and Caltech-256 dataset. There are 2,533 images from 4 distinct domains/tasks: Amazon, DSLR, Webcam, and Caltech.

**Office-31 Dataset.** Saenko et al. (2010) consists of 4,110 images from 31 object categories across 3 domains/tasks: Amazon, DSLR, and Webcam.

**Office-Home Dataset.** Venkateswara et al. (2017) is collected for object recognition to validate domain adaptation models in the era of deep learning, which includes 15,588 images in office and home settings (e.g., alarm clock, chair, eraser, keyboard, telephone, etc.) organized into 4 domains/tasks: Art (paintings, sketches and artistic depictions), Clipart (clipart images), Product

---

<sup>16</sup>1996. Reuters-21578 Collection. <http://www.daviddlewis.com/resources/testcollections/reuters21578/>

<sup>17</sup>2014. ImageCLEF-2014. <https://www.imageclef.org/2014/adaptation>

(product images from [www.amazon.com](http://www.amazon.com)), and Real-World (real-world objects captured with a regular camera).

*DomainNet Dataset.* X. Peng et al. (2019) is annotated for the purpose of multi-source unsupervised domain adaptation (UDA) research. It contains  $\sim 0.6$  million images from 345 categories across 6 distinct domains, e.g., sketch, infograph, quickdraw, real, etc.

*SYNTHIA Dataset.* Ros et al. (2016) is a synthetic dataset created to address the need for a large and diverse collection of images with pixel-level annotations for vision-based semantic segmentation in urban scenarios, particularly for autonomous driving applications. It consists of precise pixel-level semantic annotations for 13 classes, including sky, building, road, sidewalk, fence, vegetation, lane-marking, pole, car, traffic signs, pedestrians, cyclists, and miscellaneous objects.

*SVHN Dataset.* Street View House Numbers (SVHN) (R. Yang et al., 2021) is a digit classification benchmark dataset that contains 600,000  $32 \times 32$  RGB images of printed digits (from 0 to 9) cropped from pictures of house number plates. The cropped images are centered in the digit of interest, but nearby digits and other distractors are kept in the image. SVHN has three sets: training, testing sets and an extra set with 530,000 images that are less difficult and can be used for helping with the training process.

*Deepfashion Dataset.* DeepFashion (Z. Liu et al., 2016) is a large-scale clothes dataset with comprehensive annotations. It contains over 800,000 images, which are richly annotated with massive attributes, clothing landmarks, and correspondence of images taken under different scenarios including store, street snapshot, and consumer.

*ACE05 Dataset.*<sup>18</sup> The ACE 2005 Multilingual Training Corpus comprises the comprehensive collection of training data in English, Arabic, and Chinese for the 2005 Automatic Content Extraction (ACE) technology evaluation. The corpus includes diverse data types that have been annotated for entities, relations, and events. The Linguistic Data Consortium (LDC), with support from the ACE Program and additional assistance from LDC, carried out the annotation of this dataset.

*ATIS Dataset.* The ATIS (Airline Travel Information Systems) dataset (Hemphill et al., 1990) comprises audio recordings along with corresponding manual transcripts of human interactions with automated airline travel inquiry systems. These interactions involve individuals seeking flight-related information. The dataset includes 17 distinct intent categories representing different user intents. In the original data split, the training set contains 4,478 intent-labeled reference utterances, the development set contains 500 utterances, and the test set contains 893 utterances.

**4.1.3. Dense prediction task.** *CityScapes Dataset.* Cordts et al. (2016) consists of 5,000 images with high-quality annotations and 20,000 images with coarse annotations from 50 different cities, which contains 19 classes for semantic urban scene understanding. Specifically, pixel-wise semantic and instance segmentation together with ground truth inverse depth labels are often used as three different tasks (Kendall et al., 2018; S. Liu, Johns, & Davison, 2019) in MTL. *NYU-Depth Dataset V2.* Silberman et al. (2012) is comprised of 1,449 images from 464 indoor scenes across 3 cities, which contains 35,064 distinct objects of 894 different classes. The dense per-pixel labels of class, instance, and depth are used in many computer vision tasks, e.g., semantic segmentation, depth prediction, and surface normal estimation (Eigen & Fergus, 2015). *PASCAL VOC Project.*<sup>19</sup> This project (Everingham et al., 2010) provides standardized image datasets for object class recognition and also has run challenges evaluating performance on object class recognition from 2005 to 2012,

---

<sup>18</sup>2005. ACE05 Dataset. <https://catalog.ldc.upenn.edu/LDC2006T06>

<sup>19</sup>2005. Pascal VOC Project. <http://host.robots.ox.ac.uk/pascal/VOC>

**Table 9.** Summary of library for MTL.

| Library | Language | Supported Methods  |
|---------|----------|--|
| RMTL    | R        | Sparse structure learning (Tibshirani, 1996), multi-task feature selection (Obozinski et al., 2006), low rank MTL (Ji & Ye, 2009; Pong et al., 2010), graph-based regularised MTL (Widmer et al., 2010), multi-task clustering (Q. Gu & Zhou, 2009b)   |
| MALSAR  | Matlab   | Sparse structure learning (Tibshirani, 1996), regularized MTL (Evgeniou & Pontil, 2004), multi-task feature selection (Obozinski et al., 2006), dirty block-sparse model (Jalali et al., 2010), low rank MTL (Ji & Ye, 2009; Pong et al., 2010), convex ASO (J. Chen et al., 2009), sparse & low rank MTL (J. Chen et al., 2012), clustered MTL (J. Zhou, Chen, & Ye, 2011), robust MTL (J. Chen et al., 2011), robust multi-task feature learning (P. Gong, Ye, & Zhang, 2012), Temporal group Lasso (J. Zhou, Yuan, et al., 2011), convex fused sparse group Lasso (J. Zhou et al., 2012), incomplete multi-source feature learning (Yuan et al., 2012), multi-stage multi-task feature learning (P. Gong, Ye, & Zhang, 2012), multi-task clustering (Q. Gu & Zhou, 2009b) |
| LibMTL  | Python   | Cross-stitch (Misra et al., 2016), GradNorm (Z. Chen et al., 2018), Uncertainty Weighting (Kendall et al., 2018), MGDA-MTL (Sener & Koltun, 2018), MMOE (J. Ma et al., 2018), MultiNet++ (Chennupati et al., 2019), LTB (P. Guo, Lee, & Ulbricht, 2020), MTAN & DWA (S. Liu, Johns, & Davison, 2019), PCGrad (T. Yu et al., 2020), GradDrop (Z. Chen, Ngiam, et al., 2020), CGC & PLE (Tang et al., 2020), IMTL (L. Liu et al., 2021), GradVar (Z. Wang et al., 2021), CAGrad (B. Liu et al., 2021), DSelect-k (Hazimeh et al., 2021), RLW & RGW (B. Lin et al., 2022), Nash-MTL (Navon et al., 2022)  |

where VOC07<sup>20</sup>, VOC08<sup>21</sup>, and VOC12<sup>22</sup> are commonly used for MTL research. The multiple tasks cover classification, detection (e.g., body part, saliency, semantic edge), segmentation, attribute prediction (Farhadi et al., 2009), surface normals prediction (Maninis et al., 2019), etc. Many of the annotations are labeled or distilled by the followers (X. Chen et al., 2014; Maninis et al., 2019).

*Taskonomy Dataset.* Zamir et al. (2018) is currently the most diverse product for computer vision in MTL, consisting of 4 million samples from 3D scans of ~ 600 buildings. This product is a dictionary of 26 tasks (e.g., 2D, 2.5D, 3D, semantics, etc.) as a computational taxonomic map for task transfer learning. Accordingly, Tiny-Tasknomy (Standley et al., 2020) with 5 sampled dense prediction tasks, e.g., semantic segmentation, surface normal prediction, depth prediction, keypoint detection, and edge detection is considered a commonly used benchmark in MTL.

**4.1.4. Others. EMMA Dataset.** EMMA Dataset (Standley et al., 2023) comprises more than 2.8 million objects from Amazon product listings, each annotated with images, listing text, mass, price, product ratings, and its position in Amazon’s product-category taxonomy. It includes a comprehensive taxonomy of 182 physical materials, and objects are annotated with one or more materials from this taxonomy. EMMA offers a new benchmark for multi-task learning in computer vision and NLP, allowing for the addition of new tasks and object attributes at scale.

**STREET.** STREET (Ribeiro et al., 2023) is a multi-task benchmark for structured reasoning and explanations in NLP. It consists of five existing datasets (ARC, SCONE, GSM8K, AQUA-RAT, and AR-LSAT) and introduces a unified reasoning formulation with textual logical units and reasoning graphs. Evaluation metrics and empirical performance analysis using T5-large and GPT-3 models are provided, along with error explanations on a per-dataset basis.

**VKITTI2 Dataset.** Virtual KITTI (Gaidon et al., 2016) is a new video dataset, automatically labeled with accurate ground truth for object detection, tracking, scene and instance segmentation, depth, and optical flow. Virtual KITTI 2 (Cabon et al., 2020) is a more photo-realistic and better-featured version of the original virtual KITTI dataset. It exploits recent improvements of the Unity game engine and provides new data such as stereo images or scene flow.

**XTREME.** The XTREME (Cross-lingual Transfer Evaluation of Multilingual Encoders) (J. Hu et al., 2020) benchmark is a multi-task evaluation framework to assess the cross-lingual generalization capabilities of multilingual representations across 40 languages and 9 tasks. It highlights the performance disparity between models tested on English, which achieve human-level performance on numerous tasks, and cross-lingually transferred models, which exhibit a significant performance gap, particularly in syntactic and sentence retrieval tasks.

<sup>20</sup>2007. Pascal VOC Challenge 2007. <http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html>

<sup>21</sup>2008. Pascal VOC Challenge 2008. <http://host.robots.ox.ac.uk/pascal/VOC/voc2008/index.html>

<sup>22</sup>2012. Pascal VOC Challenge 2012. <http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html>

#### 4.2. Software Resources.

To provide playgrounds for researchers to fairly compare different state-of-the-art algorithms in a unified environment, open-source platforms for MTL merge out. Herein we introduce three popular software resources that aim at variant populations in terms of the implementation languages, algorithm comprehensiveness, downstream task realms, and modularization focuses.

*Regularized Multi-Task Learning (RMTL)*.<sup>23</sup> It is a relatively small yet practical R library for MTL, especially for the ones on biological-related tasks. It includes ten algorithms applicable for regression, classification, joint predictor selection, task clustering, low-rank learning and incorporation of biological networks.

*Multi-tAsk Learning via StructurAl Regularization (MALSAR)*.<sup>24</sup> It is a MTL package implemented with Matlab. Compared to RMTL, it does not particularly focus on a certain field yet includes more algorithms. In MALSAR, it implements 14 models with 26 of their variations to test their effectiveness.

*Library for Multi-Task Learning (LibMTL)*.<sup>25</sup> It is a comprehensive open-source Python library built on PyTorch for MTL. There are 104 MTL models combined by 8 architectures and 13 loss weighting strategies in LibMTL. Moreover, it guarantees unified and consistent evaluations among models on three computer vision datasets. Different from the above packages, LibMTL is well-modularized and supports customization over different components such as loss weighting strategies or architectures.

#### 4.3. Evaluation Metric.

4.3.1. *Single-task Metric*. In this section, we will introduce some single-task metrics that can be used to evaluate the performance of individual tasks in a multi-task learning (MTL) setup.

Regression Task Metric.

*Root Mean Squared Error (RMSE)*: RMSE is a commonly used metric to measure the average prediction error in regression tasks. It calculates the square root of the average of squared differences between predicted and true values. RMSE gives higher weights to larger errors, making it sensitive to outliers. It is calculated as:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\tilde{y}_i - y_i)^2}$$

where  $y_i$  represents the true value,  $\tilde{y}_i$  denotes the predicted value, and  $n$  stands for the total number of samples.

*Mean Absolute Percentage Error (MAPE)*: MAPE is a metric used to evaluate the accuracy of predictions in percentage terms. It measures the average percentage difference between predicted and true values. This metric is commonly used in business forecasting tasks. It is calculated as:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{\tilde{y}_i - y_i}{y_i} \right| \times 100$$

*Symmetric Mean Absolute Percentage Error (SMAPE)*: SMAPE is similar to MAPE but has the advantage of being symmetric, meaning it treats overestimations and underestimations equally. It calculates the average percentage difference between predicted and true values, considering the

---

<sup>23</sup><https://cran.r-project.org/web/packages/RMTL/index.html>

<sup>24</sup><https://github.com/jiayuzhou/MALSAR>

<sup>25</sup><https://github.com/median-research-group/LibMTL>

absolute sum of both. It is calculated as:

$$SMAPE = \frac{100}{n} \sum_{i=1}^n \frac{|\tilde{y}_i - y_i|}{(|\tilde{y}_i| + |y_i|)/2}$$

*Coefficient of Determination  $R^2$  (R-squared):*  $R^2$  is a statistical metric that represents the proportion of variance in the dependent variable (the target) that is predictable from the independent variable (the prediction). It indicates how well the predicted values fit the actual data. It is calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^n (\tilde{y}_i - y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

where  $\bar{y}$  is the mean of the true values  $y_i$ .

Classification Task Metric.

*Confusion Matrix:* A confusion matrix is a table that allows visualization of the performance of a classification model. It presents the number of true positive (TP), false positive (FP), true negative (TN), and false negative (FN) predictions. The confusion matrix is usually represented as follows:

|                 | Predicted Positive | Predicted Negative |
|-----------------|--------------------|--------------------|
| Actual Positive | TP                 | FN                 |
| Actual Negative | FP                 | TN                 |

*Accuracy:* Accuracy is one of the most straightforward classification metrics, representing the proportion of correctly classified instances over the total number of instances in the dataset. It is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

*Precision:* Precision is a metric that measures the proportion of true positive predictions (correctly predicted positive instances) over the total number of positive predictions made by the model. It is calculated as:

$$Precision = \frac{TP}{TP + FP}$$

*Recall (Sensitivity or True Positive Rate - TPR):* Recall calculates the proportion of true positive predictions (correctly predicted positive instances) over the total number of actual positive instances in the dataset. It is calculated as:

$$Recall = \frac{TP}{TP + FN}$$

*F1 Score:* The F1 score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is especially useful when there is an uneven class distribution. It is calculated as:

$$F1\_Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

*Specificity (True Negative Rate):* Specificity measures the proportion of true negative predictions (correctly predicted negative instances) over the total number of actual negative instances in the dataset. It is calculated as:

$$Specificity = \frac{TN}{TN + FP}$$

*Precision-Recall Curve:* The precision-recall curve is a graphical representation of the tradeoff between precision and recall for different classification thresholds. It plots the precision on the y-axis against the recall on the x-axis as the threshold varies.

*Area Under the Receiver Operating Characteristic Curve (AUC-ROC):* AUC-ROC is a metric that evaluates the performance of a binary classification model across various discrimination thresholds. It represents the area under the ROC curve, where ROC stands for the Receiver Operating Characteristic.

*Formula:* The AUC-ROC is typically computed using various threshold values to calculate the True Positive Rate (TPR) and False Positive Rate (FPR) at each threshold. The AUC-ROC is then obtained by plotting TPR against FPR and calculating the area under the curve.

Object Detection Task Metric.

*Bounding Box:* In object detection, algorithms typically predict bounding boxes and class labels for objects in an image. A bounding box is represented by a set of four coordinates:  $(x_{min}, y_{min}, x_{max}, y_{max})$ , which define the top-left and bottom-right corners of the box.

*Intersection Over Union (IoU):* The IoU measures the overlap between the predicted bounding box  $P$  and the ground truth bounding box  $G$ . It is defined as:

$$IoU(P, G) = \frac{Area(P \cap G)}{Area(P \cup G)}$$

*True Positive (TP), False Positive (FP), and False Negative (FN):* - A detection is considered a TP if the IoU with the ground truth exceeds a given threshold (typically 0.5) and the class label matches. - A detection is an FP if the IoU is below this threshold, or if there is no corresponding ground truth. - An FN represents a ground truth box which had no detected box surpassing the IoU threshold.

*Precision:*

$$Precision = \frac{TP}{TP + FP}$$

*Recall:*

$$Recall = \frac{TP}{TP + FN}$$

*Mean Average Precision (mAP):* The mAP is a widely-used metric in object detection, averaging the precision values at different recall levels across all classes.

*Precision-Recall Curve for Object Detection:* This curve plots precision against recall values for different IoU thresholds, offering insights into a detection model's performance.

*Average Recall (AR):* AR averages the recall values obtained at various IoU thresholds.

Image Segmentation Metrics.

*Pixel Accuracy:* Pixel accuracy is a simple metric that measures the proportion of pixels that are correctly classified. For a given image or set of images, it is defined as the ratio of correctly classified pixels to the total number of pixels.

$$Pixel\ Accuracy = \frac{\text{Number of correctly classified pixels}}{\text{Total number of pixels}}$$

*Boundary F1 Score (BF):* The Boundary F1 Score evaluates the accuracy of the boundaries in a segmentation task. Given predicted boundaries  $P$  and ground truth boundaries  $G$ , the BF score is the F1 score (harmonic mean of precision and recall) calculated based on the detected boundary pixels.

$$Precision = \frac{\text{Number of true positive boundary pixels}}{\text{Number of true positive boundary pixels} + \text{Number of false positive boundary pixels}}$$

$$Recall = \frac{\text{Number of true positive boundary pixels}}{\text{Number of true positive boundary pixels} + \text{Number of false negative boundary pixels}}$$

$$BF = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

*Panoptic Quality (PQ):* The Panoptic Quality metric combines segmentation (things and stuff) and detection (things only) into a single score. It is defined as:

$$PQ = \frac{\sum(p_i \times r_i)}{N_{\text{matched regions}} + \frac{1}{2} \times N_{\text{false positive regions}} + \frac{1}{2} \times N_{\text{false negative regions}}}$$

Where  $p_i$  is the precision and  $r_i$  is the recall for each matched region  $i$ .  $N_{\text{matched regions}}$  is number of matched regions.  $N_{\text{false positive regions}}$  is number of false positive regions.  $N_{\text{false negative regions}}$  is number of false negative regions.

#### Image Generation Metrics.

*Peak Signal-to-Noise Ratio (PSNR):* PSNR is a traditional quality metric used to measure the quality of a reconstructed image compared to an original image. Higher values of PSNR indicate better quality. It is defined as:

$$PSNR = 10 \times \log_{10} \left( \frac{MAX_I^2}{MSE} \right)$$

Where  $MAX_I$  is the maximum possible pixel value of the image (often 255 for an 8-bit image), and  $MSE$  is the Mean Squared Error between the original and the reconstructed image.

*Structural Similarity Index Measure (SSIM):* SSIM measures the structural similarity between two images. It provides a more perceptual-based assessment of image quality than PSNR. A value of 1 indicates the images are identical in terms of structural information.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Where  $x$  and  $y$  are two images,  $\mu$  represents the mean,  $\sigma$  represents the variance,  $\sigma_{xy}$  is the covariance of  $x$  and  $y$ , and  $C_1$  and  $C_2$  are constants to avoid instability when the denominator is close to zero.

*Inception Score (IS):* The Inception Score is used to evaluate the quality and diversity of generated images in GANs. A higher IS indicates both better image quality and greater diversity. It's calculated using a pre-trained Inception model.

$$IS = \exp(E_x[\text{KL}(p(y|x)||p(y))])$$

Where  $x$  is an image,  $y$  is the label predicted by the Inception model, and  $KL$  is the Kullback-Leibler divergence.

*Fréchet Inception Distance (FID):* FID measures the similarity between the generated images and real images. It computes the Fréchet distance between two Gaussians fitted to the feature representations of the Inception network for both sets of images. Lower FID scores indicate that the two sets of images are more similar, implying better generation quality.

$$FID = ||\mu_1 - \mu_2||^2 + \text{Tr}(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{0.5})$$

Where  $\mu_1, \Sigma_1$  are the mean and covariance of the feature representations for real images and  $\mu_2, \Sigma_2$  are those for generated images.

Text Generation Metrics.

*BLEU (Bilingual Evaluation Understudy):* BLEU is a metric originally designed for machine translation but is also used in text generation. It measures how many n-grams in the generated text match the n-grams in the reference text(s). The score ranges between 0 and 1, with 1 being a perfect match.

$$BLEU = \min \left( 1, \frac{\text{length of generated text}}{\text{length of reference text}} \right) \times \exp \left( \sum_{n=1}^N w_n \log p_n \right)$$

Where  $w_n$  are the weights for each n-gram (typically  $w_n = \frac{1}{N}$ ),  $p_n$  is the precision of n-grams, and  $N$  is the maximum n-gram order.

*ROUGE (Recall-Oriented Understudy for Gisting Evaluation):* Primarily used for evaluating summary generation, ROUGE measures the overlap between the n-grams in the generated text and the reference text(s).

$$ROUGE - N = \frac{\sum_{s \in \text{reference summaries}} \sum_{\text{n-gram} \in s} \text{Count}_{\text{match}}(\text{n-gram})}{\sum_{s \in \text{reference summaries}} \sum_{\text{n-gram} \in s} \text{Count}(\text{n-gram})}$$

Where  $\text{Count}_{\text{match}}$  is the number of matching n-grams between the generated text and reference summary, and  $\text{Count}$  is the number of n-grams in the reference summary.

*Perplexity:* Used for evaluating language models, perplexity measures how well the probability distribution predicted by the model aligns with the true distribution of the words in the text. Lower perplexity values indicate better model performance.

$$\text{Perplexity} = \exp \left( -\frac{1}{N} \sum_{i=1}^N \log p(w_i) \right)$$

Where  $N$  is the total number of words, and  $p(w_i)$  is the model's predicted probability for word  $w_i$ .

*Self-BLEU:* A metric that evaluates the diversity of generated texts. It computes the BLEU score between each generated text and all other generated texts. Lower Self-BLEU scores indicate higher diversity.

*Distinct-N:* Measures the diversity of generated content by computing the ratio of unique n-grams to the total number of generated n-grams. Higher values of Distinct-N indicate greater diversity.

$$Distinct - N = \frac{\text{Number of unique n-grams}}{\text{Total number of generated n-grams}}$$

**4.3.2. Multi-task Metric.** In this section, we denote by  $M_{MTL}^t$  and  $M_{STL}^t$  the STL measurements of MTL method and STL baseline for the  $t$ -th task, respectively.  $M_{STL}^t \downarrow$  indicates that a lower value has better performance for the measurement  $M_{STL}^t$ , and vice versa.

Delta (D. Dong et al., 2015). The performance of MTL method can be simply defined as the difference of the STL measurement between the STL baseline and MTL method:

$$(4.2) \quad \text{Delta} = M_{MTL}^t - M_{STL}^t, t = 1, \dots, T,$$

where  $M$  was set to be BLEU-4 (Papineni et al., 2002) in D. Dong et al. (2015),.

MTL gain (Tang et al., 2020). To evaluate the benefit of MTL method over the STL baseline on the  $t$ -th task, MTL gain is computed as below:

$$(4.3) \quad \text{MTL gain} = (-1)^{\mathbb{1}\{M_{STL}^t \downarrow\}} (M_{MTL}^t - M_{STL}^t), t = 1, \dots, T,$$

which is consistent with any positive or negative measurements (c.f. Delta (D. Dong et al., 2015)).

$\Delta_m$  (Maninis et al., 2019). The performance of MTL method can be quantified by calculating the average per-task drop with respect to the single-task baseline using STL measurements:

$$(4.4) \quad \Delta_m = \frac{1}{T} \sum_{t=1}^T (-1)^{\mathbb{1}\{M_{Baseline}^t \downarrow\}} (M_{MTL}^t - M_{Baseline}^t) / M_{Baseline}^t,$$

$\Delta_p$  (B. Lin et al., 2022). Given that many single tasks can be measured by several metrics, e.g. semantic segmentation measured by mIoU and pixacc, by following  $\Delta_m$  (Maninis et al., 2019), the average of the relative improvement over the MTL method on each metric of each task could be formulated as the MTL performance measurement:

$$(4.5) \quad \Delta_p = \frac{1}{T} \sum_{t=1}^T \frac{1}{M_t} \sum_{m=1}^{M_t} (-1)^{\mathbb{1}\{M_{Baseline}^{t,m} \uparrow\}} (M_{MTL}^{t,m} - M_{Baseline}^{t,m}) / M_{Baseline}^{t,m},$$

where  $M_t$  is the number of metrics used for the  $t$ -th task.  $M_{Baseline}^{t,m}$  denotes the  $m$ -th performance measurement of the baseline method, e.g. the STL or vanilla MTL method, for the  $t$ -th task.

## 5. DISCUSSION

In this section, we will discuss several key questions and explore future directions concerning the theories and applications of MTL.

*Multi-Task Pretraining.* While MTL has demonstrated its remarkable success in real-world scenarios, delving into its underlying mechanisms becomes even more imperative in the era of PFMs. When data in the wild are pre-trained using scalable foundation models to exhibit modality- and task-agnostic characteristics (§ 2.3), an essential question arises: What proportions of different tasks in the pretraining phase can yield best task-generalizable performance?

*Competitive or Collaborative?* While many proposed MTL methods offer benefits to each task under their specific settings, competitive tasks continue to exist in real-world scenarios. Distinguishing between them without human priors before employing MTL remains a challenge. Task prior sharing (§ 2.1.5) and task clustering methods (§ 2.1.6) can play a crucial role, as they can help to know task relations and do not conflict with other multi-task representation learning methods.

*Blessed or Cursed by Large Number of Tasks?* While MTL with a small number of tasks has been proven to outperform STL, and MTL with a large number of tasks has been demonstrated to be learnable, the underlying relationships between these models and the number of tasks raise intriguing questions. The introduction of a new task typically introduces both knowledge and noise to existing tasks. If all tasks are trained equally, (e.g., LLMs), without any selective mechanisms, what are the outcomes for the final learned model concerning each individual task?

*MTL for Other Things.* The pursuit of performance through MTL has been shown to have potential drawbacks in terms of fairness (§ 3.1), security and privacy (§ 3.2). However, MTL can also contribute to learning fairness or enhancing security and privacy for involved tasks by incorporating novel metrics. In certain situations, a favorable trade-off between these considerations may exist.

*Illuminating the Unseen with MTL:* To underscore the impactful insights provided by MTL, consider a compelling example where MTL results significantly advanced our understanding of a complex problem. In a medical imaging scenario, MTL was applied to simultaneously predict multiple health-related outcomes, such as disease progression, severity, and patient response to treatment. Unlike STL approaches, MTL unveiled intricate dependencies and interactions between these outcomes, showcasing that certain imaging features played dual roles in influencing multiple health aspects. This holistic perspective allowed researchers to identify subtle correlations and nuanced patterns that were previously obscured by individual task-centric analyses. MTL, in this case, not only improved predictive accuracy but also unraveled hidden intricacies within the data, providing a richer and more comprehensive understanding of the medical conditions under investigation.

This example exemplifies how MTL can reveal intricate relationships and enhance interpretability beyond the capabilities of traditional STL methods.

## 6. CONCLUSION

In this survey, we introduce the MTL from rough to precise and review methodologies covering traditional ML, DL, and PFMs era. First, we present the background of MTL, covering the scope, formal definition, comparisons with other paradigms, and motivations behind MTL. After that, we explore how MTL works well and provide the reasons to explain its intrinsic mechanisms. We formalize and illustrate MTL in a framework and further expand the methodology overview based on this MTL framework. Specifically, we summarize the sparse structure learning, feature learning, low-rank learning, and decomposition methods in the traditional learning era. We categorize MTL in DL into feature sharing, task balancing, and neural architecture search methods; recent task- and modality-agnostic foundation models are also discussed as they can learn universal comprehensiveness across tasks with different data modalities.

To sum it up, MTL methods in the traditional learning era prefer to "drop" distinctive (task-specific) features to seek consensus. For instance, the classical  $\ell_{2,1}$  norm can realize grouped feature selection across tasks to exploit common features that are effective and efficient for joint performance enhancement. Another example is the low-rank learning methods that try to explore common underlying representations via imposing low-dimensional properties for essential factors, where a small set of factors is supposed to govern multiple tasks. However, when it comes to deep learning models, powerful computational resources make it possible to handle all the features from different tasks, and its hierarchical structure with multiple layers can learn feature interaction across tasks at various levels of abstraction. Accordingly, MTL has been dominated by feature fusing and task-balancing techniques via introducing learnable parameters in the past decade. These learnable parameters play a crucial role in cross-task communication and eavesdropping during the combined training. However, the explanations and mechanisms of these complicated interactions inside the networks still remain poorly understood. More recently, unified foundation models have shown promising results for MTL in real-world scenarios, as data with versatile modalities can be trained simultaneously to learn universal and effective comprehensiveness.

Overall, we hope this paper provides an extensive review of the research community for a comprehensive understanding of research advances, current and future challenges, and opportunities or prospects for the MTL.

**Disclosure Statement.** The authors have no conflicts of interest to declare.

**Acknowledgments.** This paper is the result of a collaborative effort, with each author contributing significantly to various aspects:

- Yutong Dai orchestrated two critical optimizations in MTL, detailed in § 2.2.5 and § 2.2.6.
- Xiaokang Liu contributed by writing and organizing the section on MTL via low-rank factorization (§ 2.1.4).
- Jin Huang was responsible for the figure and layout designs, ensuring visual clarity and coherence.
- Ke Zhang was instrumental in writing and structuring the Graph-based MTL section (§ 2.2.9).
- Rong Zhou authored the STL metrics section and played a key role in organizing parts of the datasets.
- Yishan Shen focused on developing the MTL through prior sharing, as outlined in § 2.1.5.

- Eashan Aahikarla delved deeply into the distribution shifts that occur in MTL (§ 3.3).
- Wenxuan Ye took charge of organizing the GitHub website for this project, facilitating broader access and collaboration.
- Yixin Liu was pivotal in developing the security and privacy section for the MTL framework, as detailed in § 3.2.
- Zhaoming Kong and Kai Zhang were actively involved in discussions about the scope and structure of this survey.
- Jun Yu initiated this project in 2021 and managed the contents not specifically mentioned above, providing overall leadership and direction.
- Prof. Moore, Prof. Davison, Prof. Namboodiri and Prof. Yin contributed significantly by offering feedback and suggestions during the paper’s development.
- Prof. Chen finalizes the paper structure, edited different versions of the manuscript, and tailored the materials towards the audiences of the research community.

All authors above actively participated in the proofreading and discussion stages of this paper. We extend our sincere gratitude to all for their valuable contributions and collective effort in bringing this research to this final version.

## REFERENCES

- Adhikarla, E., Luo, D., & Davison, B. D. (2022). Memory defense: More robust classification via a memory-masking autoencoder.
- Adhikarla, E., Zhang, K., Yu, J., Sun, L., Nicholson, J., & Davison, B. D. (2023). Robust computer vision in an ever-changing world: A survey of techniques for tackling distribution shifts.
- Agarwal, A., Dudík, M., & Wu, Z. S. (2019). Fair regression: Quantitative definitions and reduction-based algorithms. *International Conference on Machine Learning*, 120–129.
- Alexey, D., Fischer, P., Tobias, J., Springenberg, M. R., & Brox, T. (2016). Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE TPAMI*, 38(9), 1734–1747.
- Al-Halah, Z., Ramakrishnan, S. K., & Grauman, K. (2022). Zero experience required: Plug & play modular transfer learning for semantic visual navigation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17031–17041.
- Allenby, G. M., & Rossi, P. E. (1998). Marketing models of consumer heterogeneity. *Journal of econometrics*, 89(1-2), 57–78.
- Alon, N., Reichman, D., Shinkar, I., Wagner, T., Musslick, S., Cohen, J. D., Griffiths, T., Ozcimder, K., et al. (2017). A graph-theoretic approach to multitasking. *Advances in neural information processing systems*, 30.
- Ando, R. K., Zhang, T., & Bartlett, P. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(11).
- Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. (2023). Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.
- Anwar, A., & Raychowdhury, A. (2021). Multi-task federated reinforcement learning with adversaries. *arXiv preprint arXiv:2103.06473*.
- Argote, L., Ingram, P., Levine, J. M., & Moreland, R. L. (2000). Knowledge transfer in organizations: Learning from the experience of others. *Organizational behavior and human decision processes*, 82(1), 1–8.
- Argyriou, A., Evgeniou, T., & Pontil, M. (2006). Multi-task feature learning. *Advances in neural information processing systems*, 19.
- Argyriou, A., Evgeniou, T., & Pontil, M. (2008). Convex multi-task feature learning. *Machine learning*, 73(3), 243–272.

- Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. *International conference on machine learning*, 214–223.
- Arora, N., Allenby, G. M., & Ginter, J. L. (1998). A hierarchical bayes model of primary and secondary demand. *Marketing Science*, 17(1), 29–44.
- Asai, A., Salehi, M., Peters, M. E., & Hajishirzi, H. (2022). Attempt: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 6655–6672.
- Augenstein, I., Ruder, S., & Søgaard, A. (2018). Multi-task learning of pairwise sequence classification tasks over disparate label spaces. *arXiv preprint arXiv:1802.09913*.
- Bachman, P., Hjelm, R. D., & Buchwalter, W. (2019). Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32.
- Bai, J., Men, R., Yang, H., Ren, X., Dang, K., Zhang, Y., Zhou, X., Wang, P., Tan, S., Yang, A., et al. (2022). Ofasys: A multi-modal multi-task learning system for building generalist models. *arXiv preprint arXiv:2212.04408*.
- Bakker, B., & Heskes, T. (2003). Task clustering and gating for bayesian multitask learning.
- Bao, H., Dong, L., Piao, S., & Wei, F. (2021). Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*.
- Bengio, Y., Léonard, N., & Courville, A. (2013). Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.
- Bhattacharjee, D., Süsstrunk, S., & Salzmann, M. (2023). Vision transformer adapters for generalizable multitask learning. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19015–19026.
- Bhattacharjee, D., Zhang, T., Süsstrunk, S., & Salzmann, M. (2022). Mult: An end-to-end multitask learning transformer. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12031–12041.
- Bohdal, O., Yang, Y., & Hospedales, T. (2021). Evograd: Efficient gradient-based meta-learning and hyperparameter optimization. *Advances in Neural Information Processing Systems*, 34, 22234–22246.
- Bojanowski, P., & Joulin, A. (2017). Unsupervised learning by predicting noise. *International Conference on Machine Learning*, 517–526.
- Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
- Bonab, H., Aliannejadi, M., Vardasbi, A., Kanoulas, E., & Allan, J. (2021). Cross-market product recommendation. *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 110–119.
- Bondhugula, V., Govindaraju, N., & Manocha, D. (2006). *Fast svd on graphics processors* (tech. rep.). Tech. rep., UNC Chapel Hill.
- Bonilla, E. V., Chai, K., & Williams, C. (2007). Multi-task gaussian process prediction. *Advances in neural information processing systems*, 20.
- Boyd, S., Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.
- Brauwers, G., & Frasincar, F. (2021). A general survey on attention mechanisms in deep learning. *IEEE Transactions on Knowledge and Data Engineering*.
- Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1993). Signature verification using a "siamese" time delay neural network. *Advances in neural information processing systems*, 6.
- Brookes, A. J. (1999). The essence of snps. *Gene*, 234(2), 177–186.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877–1901.

- Brüggemann, D., Kanakis, M., Obukhov, A., Georgoulis, S., & Van Gool, L. (2021). Exploring relational context for multi-task dense prediction. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15869–15878.
- Bunea, F., She, Y., & Wegkamp, M. H. (2012). Joint variable and rank selection for parsimonious estimation of high-dimensional matrices. *The Annals of Statistics*, 40(5).
- Burgos-Artizzu, X. P., Perona, P., & Dollár, P. (2013). Robust face landmark estimation under occlusion. *Proceedings of the IEEE international conference on computer vision*, 1513–1520.
- Cabon, Y., Murray, N., & Humenberger, M. (2020). Virtual kitti 2. *arXiv preprint arXiv:2001.10773*.
- Cao, K., You, J., & Leskovec, J. (2022). Relational multi-task learning: Modeling relations between data and tasks. *International Conference on Representation Learning (ICLR)*.
- Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. *Proceedings of the European conference on computer vision (ECCV)*, 132–149.
- Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33, 9912–9924.
- Caruana, R. (1997). Multitask learning. *Machine learning*, 28(1), 41–75.
- Caruana, R. A. (1993). Multitask learning: A knowledge-based source of inductive bias. *The Proceedings of the 10th International Conference on Machine Learning*.
- Caruna, R. (1993). Multitask learning: A knowledge-based source of inductive bias. *Machine learning: Proceedings of the tenth international conference*, 41–48.
- Chai, H., Wei, X., Ma, H., & Jiang, X. (2022). Knowledge-enhanced graph transformer network for multi-behavior and item-knowledge session-based recommendation. *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 3421–3426.
- Chan, P. P., Wang, Y., Kees, N., & Yeung, D. S. (2021). Multiple-model based defense for deep reinforcement learning against adversarial attack. *International Conference on Artificial Neural Networks*, 42–53.
- Chen, H., Li, W., Sheng, X., Ye, Q., Zhao, H., Xu, Y., & Bai, F. (2022). Machine learning based on the multimodal connectome can predict the preclinical stage of alzheimer’s disease: A preliminary study. *European Radiology*, 32(1), 448–459.
- Chen, J., Liu, J., & Ye, J. (2012). Learning incoherent sparse and low-rank patterns from multiple tasks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(4), 1–31.
- Chen, J., Tang, L., Liu, J., & Ye, J. (2009). A convex formulation for learning shared structures from multiple tasks. *ICML*, 137–144.
- Chen, J., Zhou, J., & Ye, J. (2011). Integrating low-rank and group-sparse structures for robust multi-task learning. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 42–50.
- Chen, K., Chan, K.-S., & Stenseth, N. C. (2012). Reduced rank stochastic regression with a sparse singular value decomposition. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 74(2), 203–221.
- Chen, K., Dong, H., & Chan, K.-S. (2013). Reduced rank regression via adaptive nuclear norm penalization. *Biometrika*, 100(4), 901–920.
- Chen, L., & Huang, J. Z. (2012). Sparse reduced-rank regression for simultaneous dimension reduction and variable selection. *Journal of the American Statistical Association*, 107(500), 1533–1545.
- Chen, M., Du, J., Pasunuru, R., Mihaylov, T., Iyer, S., Stoyanov, V., & Kozareva, Z. (2022, July). Improving in-context few-shot learning via self-supervised training. In M. Carpuat, M.-C. de Marneffe, & I. V. Meza Ruiz (Eds.), *Proceedings of the 2022 conference of the north*

- american chapter of the association for computational linguistics: Human language technologies (pp. 3558–3573). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2022.naacl-main.260>
- Chen, T., Chen, X., Du, X., Rashwan, A., Yang, F., Chen, H., Wang, Z., & Li, Y. (2023). Adamv-moe: Adaptive multi-task vision mixture-of-experts. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 17346–17357.
- Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. *International conference on machine learning*, 1597–1607.
- Chen, T., Kornblith, S., Swersky, K., Norouzi, M., & Hinton, G. E. (2020). Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33, 22243–22255.
- Chen, T., Saxena, S., Li, L., Fleet, D. J., & Hinton, G. (2022). Pix2seq: A language modeling framework for object detection. *International Conference on Learning Representations*. <https://openreview.net/forum?id=e42KbIw6Wb>
- Chen, T., Saxena, S., Li, L., Lin, T.-Y., Fleet, D. J., & Hinton, G. E. (2022). A unified sequence interface for vision tasks. *Advances in Neural Information Processing Systems*, 35, 31333–31346.
- Chen, W., Ma, X., Wang, X., & Cohen, W. W. (2022). Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.
- Chen, X., He, J., Lawrence, R., & Carbonell, J. G. (2012). Adaptive multi-task sparse learning with an application to fmri study. *Proceedings of the 2012 SIAM International Conference on Data Mining*, 212–223.
- Chen, X., Li, L., Fei, Q., Zhang, N., Tan, C., Jiang, Y., Huang, F., & Chen, H. (2023). One model for all domains: Collaborative domain-prefix tuning for cross-domain ner. *arXiv preprint arXiv:2301.10410*.
- Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., & Yuille, A. (2014). Detect what you can: Detecting and representing objects using holistic models and body parts. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1971–1978.
- Chen, X., Ding, M., Wang, X., Xin, Y., Mo, S., Wang, Y., Han, S., Luo, P., Zeng, G., & Wang, J. (2023). Context autoencoder for self-supervised representation learning. *International Journal of Computer Vision*, 1–16.
- Chen, X., Fan, H., Girshick, R., & He, K. (2020). Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*.
- Chen, X., Xie, S., & He, K. (2021). An empirical study of training self-supervised vision transformers.
- Chen, Y.-C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., & Liu, J. (2020). Uniter: Universal image-text representation learning. *European conference on computer vision*, 104–120.
- Chen, Y., Liu, Y., Xu, R., Yang, Z., Zhu, C., Zeng, M., & Zhang, Y. (2022). Unisumm: Unified few-shot summarization with multi-task pre-training and prefix-tuning. *arXiv preprint arXiv:2211.09783*.
- Chen, Z., Badrinarayanan, V., Lee, C.-Y., & Rabinovich, A. (2018). Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. *International conference on machine learning*, 794–803.
- Chen, Z., Ngiam, J., Huang, Y., Luong, T., Kretzschmar, H., Chai, Y., & Anguelov, D. (2020). Just pick a sign: Optimizing deep multitask models with gradient sign dropout. *Advances in Neural Information Processing Systems*, 33, 2039–2050.

- Chen, Z.-M., Wei, X.-S., Wang, P., & Guo, Y. (2019). Multi-label image recognition with graph convolutional networks. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5177–5186.
- Chen, Z., Zhu, L., Wan, L., Wang, S., Feng, W., & Heng, P.-A. (2020). A multi-task mean teacher for semi-supervised shadow detection. *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 5611–5620.
- Chen, Z., Wang, X., Xie, X., Parsana, M., Soni, A., Ao, X., & Chen, E. (2021). Towards explainable conversational recommendation. *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 2994–3000.
- Chen, Z., Shen, Y., Ding, M., Chen, Z., Zhao, H., Learned-Miller, E. G., & Gan, C. (2023). Mod-squad: Designing mixtures of experts as modular multi-task learners. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11828–11837.
- Cheng, G., Dong, L., Cai, W., & Sun, C. (2023). Multi-task reinforcement learning with attention-based mixture of experts. *IEEE Robotics and Automation Letters*.
- Chennupati, S., Sistu, G., Yogamani, S., & A Rawashdeh, S. (2019). Multinet++: Multi-stream feature aggregation and geometric loss strategy for multi-task learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 0–0.
- Chong, R., Kong, C., Wu, L., Liu, Z., Jin, Z., Yang, L., Fan, Y., Fan, H., & Yang, E. (2023). Leveraging prefix transfer for multi-intent text revision. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 1219–1228.
- Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. (2022). Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.
- Chowdhuri, S., Pankaj, T., & Zipser, K. (2019). Multinet: Multi-modal multi-task learning for autonomous driving. *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1496–1504.
- Chu, Z., Chen, J., Chen, Q., Yu, W., He, T., Wang, H., Peng, W., Liu, M., Qin, B., & Liu, T. (2023). A survey of chain of thought reasoning: Advances, frontiers and future. *arXiv preprint arXiv:2309.15402*.
- Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. (2022). Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.
- Chzhen, E., Denis, C., Hebiri, M., Oneto, L., & Pontil, M. (2020). Fair regression with wasserstein barycenters. *Advances in Neural Information Processing Systems*, 33, 7321–7331.
- Ciliberto, C., Rossasco, L., & Villa, S. (2015). Learning multiple visual tasks while discovering their structure. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 131–139.
- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.
- Corinzia, L., Beuret, A., & Buhmann, J. M. (2019). Variational federated multi-task learning. *arXiv preprint arXiv:1906.06268*.
- Council, N. R., et al. (2000). *How people learn: Brain, mind, experience, and school: Expanded edition*. National Academies Press.
- Crawshaw, M. (2020). Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*.
- Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning. *Advances in neural information processing systems*, 28.

- Dai, J., He, K., & Sun, J. (2016). Instance-aware semantic segmentation via multi-task network cascades. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3150–3158.
- Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable convolutional networks. *Proceedings of the IEEE international conference on computer vision*, 764–773.
- Dantone, M., Gall, J., Fanelli, G., & Van Gool, L. (2012). Real-time facial feature detection using conditional regression forests. *2012 IEEE conference on computer vision and pattern recognition*, 2578–2585.
- De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., & Tuytelaars, T. (2021). A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7), 3366–3385.
- Deist, T. M., Grewal, M., Dankers, F. J., Alderliesten, T., & Bosman, P. A. (2021). Multi-objective learning to predict pareto fronts using hypervolume maximization. *arXiv preprint arXiv:2102.04523*.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, 248–255.
- Deng, Y., Zhang, W., Xu, W., Lei, W., Chua, T.-S., & Lam, W. (2023). A unified multi-task learning framework for multi-goal conversational recommender systems. *ACM Transactions on Information Systems*, 41(3), 1–25.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Ding, Y., Wu, Y., Huang, C., Tang, S., Yang, Y., Wei, L., Zhuang, Y., & Tian, Q. (2022). Learning to learn by jointly optimizing neural architecture and weights. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 129–138.
- Donahue, J., Krähenbühl, P., & Darrell, T. (2016). Adversarial feature learning. *arXiv preprint arXiv:1605.09782*.
- Donahue, J., & Simonyan, K. (2019). Large scale adversarial representation learning. *Advances in neural information processing systems*, 32.
- Dong, D., Wu, H., He, W., Yu, D., & Wang, H. (2015). Multi-task learning for multiple language translation. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1723–1732.
- Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., Sun, X., Xu, J., & Sui, Z. (2022). A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.
- Doshi, K., & Yilmaz, Y. (2022). Rethinking video anomaly detection-a continual learning approach. *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 3961–3970.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*. <https://openreview.net/forum?id=YicbFdNTTy>
- Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., & Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. *Advances in neural information processing systems*, 27.
- Du, N., Huang, Y., Dai, A. M., Tong, S., Lepikhin, D., Xu, Y., Krikun, M., Zhou, Y., Yu, A. W., Firat, O., et al. (2022). Glam: Efficient scaling of language models with mixture-of-experts. *International Conference on Machine Learning*, 5547–5569.

- Dwivedi, K., & Roig, G. (2019). Representation similarity analysis for efficient task taxonomy & transfer learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12387–12396.
- Efrat, A., & Levy, O. (2020). The turking test: Can language models understand instructions? *arXiv preprint arXiv:2010.11982*.
- Ehrgott, M. (2005). *Multicriteria optimization* (Vol. 491). Springer Science & Business Media.
- Eigen, D., & Fergus, R. (2015). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *Proceedings of the IEEE international conference on computer vision*, 2650–2658.
- Eigen, D., Puhrsch, C., & Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27.
- Eigen, D., Ranzato, M., & Sutskever, I. (2013). Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314*.
- Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2), 303–338.
- Evgeniou, T., Micchelli, C. A., Pontil, M., & Shawe-Taylor, J. (2005). Learning multiple tasks with kernel methods. *Journal of machine learning research*, 6(4).
- Evgeniou, T., & Pontil, M. (2004). Regularized multi-task learning. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 109–117.
- Faal, F., Schmitt, K., & Yu, J. Y. (2023). Reward modeling for mitigating toxicity in transformer-based language models. *Applied Intelligence*, 53(7), 8421–8435.
- Fan, Z., Sarkar, R., Jiang, Z., Chen, T., Zou, K., Cheng, Y., Hao, C., Wang, Z., et al. (2022). M<sup>3</sup>vit: Mixture-of-experts vision transformer for efficient multi-task learning with model-accelerator co-design. *Advances in Neural Information Processing Systems*, 35, 28441–28457.
- Farhadi, A., Endres, I., Hoiem, D., & Forsyth, D. (2009). Describing objects by their attributes. *2009 IEEE conference on computer vision and pattern recognition*, 1778–1785.
- Fazel, M., Hindi, H., & Boyd, S. P. (2001). A rank minimization heuristic with application to minimum order system approximation. *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*, 6, 4734–4739.
- Fedus, W., Zoph, B., & Shazeer, N. (2022). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1), 5232–5270.
- Fei-Fei, L., Fergus, R., & Perona, P. (2006). One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4), 594–611.
- Feng, F., Zou, Z., Liu, C., Zhou, Q., & Liu, C. (2023). Forecast of short-term passenger flow in multi-level rail transit network based on a multi-task learning model. *Sustainability*, 15(4), 3296.
- Fernando, H. D., Shen, H., Liu, M., Chaudhury, S., Murugesan, K., & Chen, T. (2023). Mitigating gradient bias in multi-objective learning: A provably convergent approach. *The Eleventh International Conference on Learning Representations*. <https://openreview.net/forum?id=dLAYGdKt2>
- Fink, M. (2004). Object classification from a single example utilizing class relevance metrics. *Advances in neural information processing systems*, 17.
- Fliege, J., & Svaiter, B. F. (2000). Steepest descent methods for multicriteria optimization. *Mathematical methods of operations research*, 51, 479–494.
- Friedman, J., Hastie, T., Höfling, H., & Tibshirani, R. (2007). Pathwise coordinate optimization. *The annals of applied statistics*, 1(2), 302–332.
- Fu, K.-S., & Mui, J. (1981). A survey on image segmentation. *Pattern recognition*, 13(1), 3–16.

- Fu, Y., Peng, H., Sabharwal, A., Clark, P., & Khot, T. (2022). Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*.
- Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4), 193–202.
- Gaidon, A., Wang, Q., Cabon, Y., & Vig, E. (2016). Virtual worlds as proxy for multi-object tracking analysis. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4340–4349.
- Ganin, Y., & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. *International conference on machine learning*, 1180–1189.
- Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., & Lempitsky, V. (2016). Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1), 2096–2030.
- Gao, M., Li, J.-Y., Chen, C.-H., Li, Y., Zhang, J., & Zhan, Z.-H. (2023). Enhanced multi-task learning and knowledge graph-based recommender system. *IEEE Transactions on Knowledge and Data Engineering*.
- Gao, Y., Bai, H., Jie, Z., Ma, J., Jia, K., & Liu, W. (2020). Mtl-nas: Task-agnostic neural architecture search towards general-purpose multi-task learning. *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 11543–11552.
- Gao, Y., Ma, J., Zhao, M., Liu, W., & Yuille, A. L. (2019). Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3205–3214.
- García, J., Majadas, R., & Fernández, F. (2020). Learning adversarial attack policies through multi-objective reinforcement learning. *Engineering Applications of Artificial Intelligence*, 96, 104021.
- Ghiasi, G., Zoph, B., Cubuk, E. D., Le, Q. V., & Lin, T.-Y. (2021). Multi-task self-training for learning general representations. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8856–8865.
- Gidaris, S., Singh, P., & Komodakis, N. (2018). Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.
- Girdhar, R., El-Nouby, A., Singh, M., Alwala, K. V., Joulin, A., & Misra, I. (2023). Omnimae: Single model masked pretraining on images and videos. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10406–10417.
- Girshick, R. (2015). Fast r-cnn. *Proceedings of the IEEE international conference on computer vision*, 1440–1448.
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587.
- Gómez, E., Gomez-Villegas, M., & Marín, J. M. (1998). A multivariate generalization of the power exponential family of distributions. *Communications in Statistics-Theory and Methods*, 27(3), 589–600.
- Gonen, H., Iyer, S., Blevins, T., Smith, N. A., & Zettlemoyer, L. (2022). Demystifying prompts in language models via perplexity estimation. *arXiv preprint arXiv:2212.04037*.
- Gong, B., Shi, Y., Sha, F., & Grauman, K. (2012). Geodesic flow kernel for unsupervised domain adaptation. *2012 IEEE conference on computer vision and pattern recognition*, 2066–2073.
- Gong, P., Ye, J., & Zhang, C.-s. (2012). Multi-stage multi-task feature learning. *Advances in neural information processing systems*, 25.
- Gong, P., Ye, J., & Zhang, C. (2012). Robust multi-task feature learning. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 895–903.

- Gong, Y., Luo, X., Zhu, Y., Ou, W., Li, Z., Zhu, M., Zhu, K. Q., Duan, L., & Chen, X. (2019). Deep cascade multi-task learning for slot filling in online shopping assistant. *Proceedings of the AAAI conference on artificial intelligence*, 33(01), 6465–6472.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139–144.
- Gori, M., Monfardini, G., & Scarselli, F. (2005). A new model for learning in graph domains. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, 2, 729–734.
- Görnitz, N., Widmer, C., Zeller, G., Kahles, A., Rätsch, G., & Sonnenburg, S. (2011). Hierarchical multitask structured output learning for large-scale sequence segmentation. *Advances in Neural Information Processing Systems*, 24.
- Goyal, P., Caron, M., Lefauveux, B., Xu, M., Wang, P., Pai, V., Singh, M., Liptchinsky, V., Misra, I., Joulin, A., et al. (2021). Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*.
- Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., & He, K. (2017). Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*.
- Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33, 21271–21284.
- Grossberg, S. T. (2012). *Studies of mind and brain: Neural principles of learning, perception, development, cognition, and motor control* (Vol. 70). Springer Science & Business Media.
- Gu, J., Han, Z., Chen, S., Beirami, A., He, B., Zhang, G., Liao, R., Qin, Y., Tresp, V., & Torr, P. (2023). A systematic survey of prompt engineering on vision-language foundation models. *arXiv preprint arXiv:2307.12980*.
- Gu, Q., & Zhou, J. (2009a). Learning the shared subspace for multi-task clustering and transductive transfer classification. *2009 Ninth IEEE International Conference on Data Mining*, 159–168. <https://doi.org/10.1109/ICDM.2009.32>
- Gu, Q., & Zhou, J. (2009b). Learning the shared subspace for multi-task clustering and transductive transfer classification. *2009 Ninth IEEE International Conference on Data Mining*, 159–168.
- Gu, Y., Dong, L., Wei, F., & Huang, M. (2023). Pre-training to learn in context. *arXiv preprint arXiv:2305.09137*.
- Guizilini, V., Li, J., Ambrus, R., & Gaidon, A. (2021). Geometric unsupervised domain adaptation for semantic segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8537–8547.
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. *Advances in neural information processing systems*, 30.
- Guo, M.-H., Xu, T.-X., Liu, J.-J., Liu, Z.-N., Jiang, P.-T., Mu, T.-J., Zhang, S.-H., Martin, R. R., Cheng, M.-M., & Hu, S.-M. (2022). Attention mechanisms in computer vision: A survey. *Computational Visual Media*, 8(3), 331–368.
- Guo, P., Lee, C.-Y., & Ulbricht, D. (2020). Learning to branch for multi-task learning. *International Conference on Machine Learning*, 3854–3863.
- Guo, P., Xu, Y., Lin, B., & Zhang, Y. (2020). Multi-task adversarial attack. *arXiv preprint arXiv:2011.09824*.
- Gupta, S., Mukherjee, S., Subudhi, K., Gonzalez, E., Jose, D., Awadallah, A. H., & Gao, J. (2022). Sparsely activated mixture-of-experts are robust multi-task learners. *arXiv preprint arXiv:2204.07689*.

- Gutmann, M., & Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 297–304.
- Han, L., & Zhang, Y. (2015). Learning tree structure in multi-task learning. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 397–406.
- Han, L., & Zhang, Y. (2016). Multi-stage multi-task learning with reduced rank. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).
- Hand, D. J., & Till, R. J. (2001). A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine learning*, 45, 171–186.
- Hao, Y., Sun, Y., Dong, L., Han, Z., Gu, Y., & Wei, F. (2022). Structured prompting: Scaling in-context learning to 1,000 examples. *arXiv preprint arXiv:2212.06713*.
- Hashimoto, K., Xiong, C., Tsuruoka, Y., & Socher, R. (2017). A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks [To appear]. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. <http://arxiv.org/abs/1611.01587>
- Hassani, K., & Haley, M. (2019). Unsupervised multi-task feature learning on point clouds. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8160–8171.
- Hazimeh, H., Zhao, Z., Chowdhery, A., Sathiamoorthy, M., Chen, Y., Mazumder, R., Hong, L., & Chi, E. (2021). Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. *Advances in Neural Information Processing Systems*, 34, 29335–29347.
- He, J., & Lawrence, R. (2011). A graphbased framework for multi-task multi-view learning. *ICML*.
- He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked autoencoders are scalable vision learners. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16000–16009.
- He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- He, P., Liu, X., Gao, J., & Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.
- He, X., Sygnowski, J., Galashov, A., Rusu, A. A., Teh, Y. W., & Pascanu, R. (2019). Task agnostic continual learning via meta learning. *arXiv preprint arXiv:1906.05201*.
- Hemphill, C. T., Godfrey, J. J., & Doddington, G. R. (1990). The ATIS spoken language systems pilot corpus. *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*. <https://aclanthology.org/H90-1021>
- Henaff, O. (2020). Data-efficient image recognition with contrastive predictive coding. *International conference on machine learning*, 4182–4192.
- Herzig, R., Abramovich, O., Ben Avraham, E., Arbelle, A., Karlinsky, L., Shamir, A., Darrell, T., & Globerson, A. (2024). Promptonomyvit: Multi-task prompt learning improves video transformers using synthetic scene data. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 6803–6815.
- Heskes, T. (2000). Empirical bayes for learning to learn.
- Hessel, M., Soyer, H., Espeholt, L., Czarnecki, W., Schmitt, S., & Van Hasselt, H. (2019). Multi-task deep reinforcement learning with popart. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), 3796–3803.
- Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *NIPS Deep Learning and Representation Learning Workshop*. <http://arxiv.org/abs/1503.02531>

- Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., & Bengio, Y. (2018). Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*.
- Ho, N., Schmid, L., & Yun, S.-Y. (2022). Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.
- Honovich, O., Shaham, U., Bowman, S. R., & Levy, O. (2022). Instruction induction: From few examples to natural language task descriptions. *arXiv preprint arXiv:2205.10782*.
- Horn, R. A., & Johnson, C. R. (2012). *Matrix analysis*. Cambridge university press.
- Hospedales, T., Antoniou, A., Micaelli, P., & Storkey, A. (2021). Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9), 5149–5169.
- Hsu, D., Kakade, S. M., & Zhang, T. (2010). Robust matrix decomposition with outliers. *arXiv preprint arXiv:1011.1518*.
- Hu, F., Ratz, P., & Charpentier, A. (2023). Fairness in multi-task learning via wasserstein barycenters. *arXiv preprint arXiv:2306.10155*.
- Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.
- Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., & Johnson, M. (2020). Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. *International Conference on Machine Learning*, 4411–4421.
- Hu, R., & Singh, A. (2021). Unit: Multimodal multitask learning with a unified transformer. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1439–1449.
- Hu, Z., Zhao, Z., Yi, X., Yao, T., Hong, L., Sun, Y., & Chi, E. (2022). Improving multi-task generalization via regularizing spurious correlation. *Advances in Neural Information Processing Systems*, 35, 11450–11466.
- Huang, P.-Y., Chang, X., Hauptmann, A., & Hovy, E. (2020). Forward and backward multimodal nmt for improved monolingual and multilingual cross-modal retrieval. *Proceedings of the 2020 International Conference on Multimedia Retrieval*, 53–62.
- Huang, Z., Lin, S., Liu, G., Luo, M., Ye, C., Xu, H., Chang, X., & Liang, X. (2023). Fuller: Unified multi-modality multi-task 3d perception via multi-level gradient calibration. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3502–3511.
- Huo, Y., Zhang, M., Liu, G., Lu, H., Gao, Y., Yang, G., Wen, J., Zhang, H., Xu, B., Zheng, W., et al. (2021). Wenlan: Bridging vision and language by large-scale multi-modal pre-training. *arXiv preprint arXiv:2103.06561*.
- Iyer, S., Lin, X. V., Pasunuru, R., Mihaylov, T., Simig, D., Yu, P., Shuster, K., Wang, T., Liu, Q., Koura, P. S., et al. (2022). Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*.
- Jacob, G. M., Agarwal, V., & Stenger, B. (2023). Online knowledge distillation for multi-task learning. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2359–2368.
- Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural computation*, 3(1), 79–87.
- Jaggi, M. (2013). Revisiting frank-wolfe: Projection-free sparse convex optimization. *International conference on machine learning*, 427–435.
- Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., & Makedon, F. (2020). A survey on contrastive self-supervised learning. *Technologies*, 9(1), 2.
- Jalali, A., Sanghavi, S., Ruan, C., & Ravikumar, P. (2010). A dirty model for multi-task learning. *Advances in neural information processing systems*, 23.

- Jang, E., Gu, S., & Poole, B. (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.
- Javaloy, A., & Valera, I. (2022). Rotograd: Gradient homogenization in multitask learning. *International Conference on Learning Representations*. <https://openreview.net/forum?id=T8wHz4rnuGL>
- Jawanpuria, P. K., Lapin, M., Hein, M., & Schiele, B. (2015). Efficient output kernel learning for multiple tasks. *Advances in neural information processing systems*, 28.
- Ji, S., & Ye, J. (2009). An accelerated gradient method for trace norm minimization. *Proceedings of the 26th annual international conference on machine learning*, 457–464.
- Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., & Duerig, T. (2021). Scaling up visual and vision-language representation learning with noisy text supervision. *International Conference on Machine Learning*, 4904–4916.
- Jia, X., Xiong, Y., Zhang, J., Zhang, Y., & Zhu, Y. (2020). Few-shot radiology report generation for rare diseases. *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 601–608.
- Jiang, R., Pacchiano, A., Stepleton, T., Jiang, H., & Chiappa, S. (2020). Wasserstein fair classification. *Uncertainty in artificial intelligence*, 862–872.
- Jie, B., Zhang, D., Cheng, B., Shen, D., & Initiative, A. D. N. (2015). Manifold regularized multitask feature learning for multimodality disease classification. *Human brain mapping*, 36(2), 489–507.
- Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., & Levy, O. (2020). Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the association for computational linguistics*, 8, 64–77.
- Kao, P.-Y., Kao, S.-M., Huang, N.-L., & Lin, Y.-C. (2021). Toward drug-target interaction prediction via ensemble modeling and transfer learning. *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2384–2391.
- Kato, T., Kashima, H., Sugiyama, M., & Asai, K. (2007). Multi-task learning via conic programming. *Advances in Neural Information Processing Systems*, 20.
- Kendall, A., Gal, Y., & Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7482–7491.
- Kim, B., Kim, H., Lee, S.-W., Lee, G., Kwak, D., Jeon, D. H., Park, S., Kim, S., Kim, S., Seo, D., et al. (2021). What changes can large-scale language models bring? intensive study on hyperclova: Billions-scale korean generative pretrained transformers. *arXiv preprint arXiv:2109.04650*.
- Kim, D., Cho, D., Yoo, D., & Kweon, I. S. (2018). Learning image representations by completing damaged jigsaw puzzles. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 793–802.
- Kim, W., Son, B., & Kim, I. (2021). Vilt: Vision-and-language transformer without convolution or region supervision. *International Conference on Machine Learning*, 5583–5594.
- Kirillov, A., He, K., Girshick, R., Rother, C., & Dollár, P. (2019). Panoptic segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9404–9413.
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. (2023). Segment anything. *arXiv preprint arXiv:2304.02643*.
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35, 22199–22213.
- Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review*, 51(3), 455–500.

- Kwak, K., Yun, H. J., Park, G., Lee, J.-M., Initiative, A. D. N., et al. (2018). Multi-modality sparse representation for alzheimer's disease classification. *Journal of Alzheimer's disease*, 65(3), 807–817.
- Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., & Navab, N. (2016). Deeper depth prediction with fully convolutional residual networks. *2016 Fourth international conference on 3D vision (3DV)*, 239–248.
- Lang, K. (1995). Newsweeder: Learning to filter netnews. *Proceedings of the Twelfth International Conference on Machine Learning*, 331–339.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
- LeCun, Y., Bottou, L., Orr, G. B., & Müller, K.-R. (2002). Efficient backprop. In *Neural networks: Tricks of the trade* (pp. 9–50). Springer.
- Lee, H., Lee, S., Chong, S., & Hwang, S. J. (2021). Hardware-adaptive efficient latency prediction for nas via meta-learning. *Advances in Neural Information Processing Systems*, 34, 27016–27028.
- Lee, S., Zhu, J., & Xing, E. (2010). Adaptive multi-task lasso: With application to eqtl detection. *Advances in neural information processing systems*, 23.
- Lenk, P. J., DeSarbo, W. S., Green, P. E., & Young, M. R. (1996). Hierarchical bayes conjoint analysis: Recovery of partworth heterogeneity from reduced experimental designs. *Marketing Science*, 15(2), 173–191.
- Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. <https://doi.org/10.18653/v1/2020.acl-main.703>
- Li, F., Zhang, H., Xu, H., Liu, S., Zhang, L., Ni, L. M., & Shum, H.-Y. (2023). Mask dino: Towards a unified transformer-based framework for object detection and segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3041–3050.
- Li, H., Zhu, J., Jiang, X., Zhu, X., Li, H., Yuan, C., Wang, X., Qiao, Y., Wang, X., Wang, W., et al. (2023). Uni-perceiver v2: A generalist model for large-scale vision and vision-language tasks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2691–2700.
- Li, J., Li, D., Savarese, S., & Hoi, S. (2023). Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.
- Li, J., Li, D., Xiong, C., & Hoi, S. (2022). Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *International Conference on Machine Learning*, 12888–12900.
- Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., & Hoi, S. C. H. (2021). Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34, 9694–9705.
- Li, J., Zhou, P., Xiong, C., & Hoi, S. C. (2020). Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*.
- Li, L. H., Yatskar, M., Yin, D., Hsieh, C.-J., & Chang, K.-W. (2019). Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.
- Li, Q., Gong, B., Cui, Y., Kondratyuk, D., Du, X., Yang, M.-H., & Brown, M. (2021). Towards a unified foundation model: Jointly pre-training transformers on unpaired images and text. *arXiv preprint arXiv:2112.07074*.

- Li, W.-H., & Bilen, H. (2020). Knowledge distillation for multi-task learning. *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, 163–176.
- Li, X. L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.
- Li, X., Lv, K., Yan, H., Lin, T., Zhu, W., Ni, Y., Xie, G., Wang, X., & Qiu, X. (2023). Unified demonstration retriever for in-context learning. *arXiv preprint arXiv:2305.04320*.
- Li, X., & Qiu, X. (2023). Finding supporting examples for in-context learning. *arXiv preprint arXiv:2302.13539*.
- Li, Y., & Oymak, S. (2023). On the fairness of multitask representation learning. *ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5.
- Li, Y., Yang, M., & Zhang, Z. (2018). A survey of multi-view representation learning. *IEEE transactions on knowledge and data engineering*, 31(10), 1863–1883.
- Lin, B., YE, F., Zhang, Y., & Tsang, I. (2022). Reasonable effectiveness of random weighting: A litmus test for multi-task learning. *Transactions on Machine Learning Research*. <https://openreview.net/forum?id=jjtFD8A1Wx>
- Lin, X., Yang, Z., Zhang, Q., & Kwong, S. (2020). Controllable pareto multi-task learning. *arXiv preprint arXiv:2010.06313*.
- Lin, X., Zhen, H.-L., Li, Z., Zhang, Q.-F., & Kwong, S. (2019). Pareto multi-task learning. *Advances in neural information processing systems*, 32.
- Liu, B., Liu, X., Jin, X., Stone, P., & qiang liu. (2021). Conflict-averse gradient descent for multi-task learning. In A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems*. [https://openreview.net/forum?id=\\_61Qh8tULj](https://openreview.net/forum?id=_61Qh8tULj)
- Liu, C., Cao, W., Wu, S., Shen, W., Jiang, D., Yu, Z., & Wong, H.-S. (2020). Asymmetric graph-guided multitask survival analysis with self-paced learning. *IEEE Transactions on Neural Networks and Learning Systems*, 33(2), 654–666.
- Liu, F., Li, G., Zhao, Y., & Jin, Z. (2020). Multi-task learning based pre-trained language model for code completion. *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*, 473–485.
- Liu, H., Cui, L., Liu, J., & Zhang, Y. (2021). Natural language inference in context-investigating contextual reasoning over long texts. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15), 13388–13396.
- Liu, H., Simonyan, K., & Yang, Y. (2018). Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*.
- Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., & Chen, W. (2021). What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.
- Liu, J., Chen, T., Liang, Z., Jiang, H., Xiao, Y., Wei, F., Qian, Y., Hao, Z., & Han, B. (2023a). Hierarchical prompt tuning for few-shot multi-task learning. *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 1556–1565. <https://doi.org/10.1145/3583780.3614913>
- Liu, J., Chen, T., Liang, Z., Jiang, H., Xiao, Y., Wei, F., Qian, Y., Hao, Z., & Han, B. (2023b). Hierarchical prompt tuning for few-shot multi-task learning. *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 1556–1565.
- Liu, J., Ji, S., Ye, J., et al. (2009). Slep: Sparse learning with efficient projections. *Arizona State University*, 6(491), 7.
- Liu, J., Ji, S., & Ye, J. (2012). Multi-task feature learning via efficient l2, 1-norm minimization. *arXiv preprint arXiv:1205.2631*.
- Liu, K., Uplavikar, N., Jiang, W., & Fu, Y. (2018). Privacy-preserving multi-task learning. *2018 IEEE International Conference on Data Mining (ICDM)*, 1128–1133.

- Liu, L., Li, Y., Kuang, Z., Xue, J.-H., Chen, Y., Yang, W., Liao, Q., & Zhang, W. (2021). Towards impartial multi-task learning. *International Conference on Learning Representations*. <https://openreview.net/forum?id=IMPnRXEWpvr>
- Liu, P., Qiu, X., & Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.
- Liu, P., Qiu, X., & Huang, X. (2017). Adversarial multi-task learning for text classification. *arXiv preprint arXiv:1704.05742*.
- Liu, Q., Liao, X., & Carin, L. (2007). Semi-supervised multitask learning. *Advances in Neural Information Processing Systems*, 20.
- Liu, S. (2018). *Exploration on deep drug discovery: Representation and learning* (tech. rep.).
- Liu, S., Liang, Y., & Gitter, A. (2019). Loss-balanced task weighting to reduce negative transfer in multi-task learning. *Proceedings of the AAAI conference on artificial intelligence*, 33(01), 9977–9978.
- Liu, S., Qu, M., Zhang, Z., Cai, H., & Tang, J. (2022). Structured multi-task learning for molecular property prediction. *International Conference on Artificial Intelligence and Statistics*, 8906–8920.
- Liu, S., Johns, E., & Davison, A. J. (2019). End-to-end multi-task learning with attention. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1871–1880.
- Liu, S., & Vicente, L. N. (2021). The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning. *Annals of Operations Research*, 1–30.
- Liu, S., & Vicente, L. N. (2020). A review of multi-objective optimization: Theory and algorithms.
- Liu, W., Su, J., Chen, C., & Zheng, X. (2021). Leveraging distribution alignment via stein path for cross-domain cold-start recommendation. *Advances in Neural Information Processing Systems*, 34, 19223–19234.
- Liu, X., Ji, K., Fu, Y., Tam, W., Du, Z., Yang, Z., & Tang, J. (2022). P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 61–68.
- Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., & Tang, J. (2021). Self-supervised learning: Generative or contrastive. *IEEE transactions on knowledge and data engineering*, 35(1), 857–876.
- Liu, X., Tong, X., & Liu, Q. (2021). Profiling pareto front with multi-objective stein variational gradient descent. *Advances in Neural Information Processing Systems*, 34, 14721–14733.
- Liu, Y., Wang, Z., Jin, H., & Wassell, I. (2018). Multi-task adversarial network for disentangled feature learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3743–3751.
- Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A convnet for the 2020s. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11976–11986.
- Liu, Z., Luo, P., Qiu, S., Wang, X., & Tang, X. (2016). Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1096–1104.
- Liu, Z., Luo, P., Wang, X., & Tang, X. (2018). Large-scale celebfaces attributes (celeba) dataset. *Retrieved August, 15(2018)*, 11.
- Livni, R., Shalev-Shwartz, S., & Shamir, O. (2014). On the computational efficiency of training neural networks. *Advances in neural information processing systems*, 27.
- Long, M., Cao, Z., Wang, J., & Yu, P. S. (2017). Learning multiple tasks with multilinear relationship networks. *Advances in neural information processing systems*, 30.

- Lopes, I., Vu, T.-H., & de Charette, R. (2023). Cross-task attention mechanism for dense multi-task learning. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2329–2338.
- Lounici, K., Pontil, M., Tsybakov, A. B., & Van De Geer, S. (2009). Taking advantage of sparsity in multi-task learning. *arXiv preprint arXiv:0903.1468*.
- Lozano, A. C., & Swirszcz, G. (2012). Multi-level lasso for sparse multi-task regression. *Proceedings of the 29th International Conference on International Conference on Machine Learning*, 595–602.
- Lu, J., Clark, C., Zellers, R., Mottaghi, R., & Kembhavi, A. (2022). Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv:2206.08916*.
- Lu, J., Goswami, V., Rohrbach, M., Parikh, D., & Lee, S. (2020). 12-in-1: Multi-task vision and language representation learning. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10437–10446.
- Lu, Y., Bartolo, M., Moore, A., Riedel, S., & Stenetorp, P. (2022, May). Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 8086–8098). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2022.acl-long.556>
- Lu, Y., Kumar, A., Zhai, S., Cheng, Y., Javidi, T., & Feris, R. (2017). Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5334–5343.
- Lu, Y., Sirejiding, S., Ding, Y., Wang, C., & Lu, H. (2024). Prompt guided transformer for multi-task dense prediction. *IEEE Transactions on Multimedia*.
- Luo, Q., Sorokin, M., & Ha, S. (2021). A few shot adaptation of visual navigation skills to new observations using meta-learning. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 13231–13237.
- Lyu, F., Wang, S., Feng, W., Ye, Z., Hu, F., & Wang, S. (2021). Multi-domain multi-task rehearsal for lifelong learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10), 8819–8827.
- Ma, H., Yuan, X., Fan, D., Ding, Z., Wang, X., & Fang, J. (2022). Over-the-air federated multi-task learning. *ICC 2022-IEEE International Conference on Communications*, 5184–5189.
- Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., & Chi, E. H. (2018). Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1930–1939.
- Ma, P., Du, T., & Matusik, W. (2020). Efficient continuous pareto exploration in multi-task learning. *International Conference on Machine Learning*, 6522–6531.
- Mahapatra, D., & Rajan, V. (2020). Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. *International Conference on Machine Learning*, 6597–6607.
- Mahmoud, R. A., & Hajj, H. (2022). Multi-objective learning to overcome catastrophic forgetting in time-series applications. *ACM Trans. Knowl. Discov. Data*, 16(6). <https://doi.org/10.1145/3502728>
- Maninis, K.-K., Radosavovic, I., & Kokkinos, I. (2019). Attentive single-tasking of multiple tasks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1851–1860.
- Mao, C., Gupta, A., Nitin, V., Ray, B., Song, S., Yang, J., & Vondrick, C. (2020). Multitask learning strengthens adversarial robustness. *Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II* 16, 158–174.
- Mao, Y., Liu, W., & Lin, X. (2020). Adaptive adversarial multi-task representation learning. *International Conference on Machine Learning*, 6724–6733.

- Marfoq, O., Neglia, G., Bellet, A., Kameni, L., & Vidal, R. (2021). Federated multi-task learning under a mixture of distributions. *Advances in Neural Information Processing Systems*, 34, 15434–15447.
- Martin, D. R., Fowlkes, C. C., & Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, 26(5), 530–549.
- Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2), 442–451.
- Maurer, A., Pontil, M., & Romera-Paredes, B. (2013). Sparse coding for multitask and transfer learning. *International conference on machine learning*, 343–351.
- McCann, B., Bradbury, J., Xiong, C., & Socher, R. (2017). Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, 30.
- McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). 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.
- Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6), 1–35.
- Meng, Q., Pawlowski, N., Rueckert, D., & Kainz, B. (2019). Representation disentanglement for multi-task learning with application to fetal ultrasound. *arXiv preprint arXiv:1908.07885*.
- Mesbahi, M. (1999). A semi-definite programming solution of the least order dynamic output feedback synthesis problem.
- Meyerson, E., & Miikkulainen, R. (2018). Beyond shared hierarchies: Deep multitask learning through soft layer ordering. *International Conference on Learning Representations*. <https://openreview.net/forum?id=BkXmYfbAZ>
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.
- Min, S., Lewis, M., Zettlemoyer, L., & Hajishirzi, H. (2022, July). MetaICL: Learning to learn in context. In M. Carpuat, M.-C. de Marneffe, & I. V. Meza Ruiz (Eds.), *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 2791–2809). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2022.naacl-main.201>
- Misra, I., Shrivastava, A., Gupta, A., & Hebert, M. (2016). Cross-stitch networks for multi-task learning. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3994–4003.
- Mitchell, T. M. (1980). *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research ...
- Mitchell, T. M. (1997). *Machine learning* (Vol. 1). McGraw-hill New York.
- Momma, M., Dong, C., & Liu, J. (2022). A multi-objective/multi-task learning framework induced by pareto stationarity. *International Conference on Machine Learning*, 15895–15907.
- Moosavi, N. S., & Strube, M. (2016). Which coreference evaluation metric do you trust? a proposal for a link-based entity aware metric. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 632–642.
- Mortimore, P., Sammons, P., Stoll, L., & Ecob, R. (1988). *School matters*. Univ of California Press.
- Naseer, M., Khan, S., Hayat, M., Khan, F. S., & Porikli, F. (2022). Stylized adversarial defense. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5), 6403–6414.
- Navon, A., Achituv, I., Maron, H., Chechik, G., & Fetaya, E. (2021). Auxiliary learning by implicit differentiation. *International Conference on Learning Representations*. <https://openreview.net/forum?id=n7wIfYPdVet>

- Navon, A., Shamsian, A., Achituve, I., Maron, H., Kawaguchi, K., Chechik, G., & Fetaya, E. (2022). Multi-task learning as a bargaining game. *International Conference on Machine Learning*, 16428–16446.
- Navon, A., Shamsian, A., Chechik, G., & Fetaya, E. (2021). Learning the pareto front with hyper-networks. *International Conference on Learning Representations*. <https://openreview.net/forum?id=NjF772F4ZZR>
- Negahban, S., & Wainwright, M. J. (2008). Joint support recovery under high-dimensional scaling: Benefits and perils of  $\ell_{1,\infty}$ -regularization. *Advances in Neural Information Processing Systems*, 21, 1161–1168.
- Nemirovski, A. (1994). Efficient methods in convex programming. *Lecture notes*.
- Nesterov, Y. (1998). Introductory lectures on convex programming.
- Nesterov, Y. (2013). Gradient methods for minimizing composite functions. *Mathematical programming*, 140(1), 125–161.
- Nesterov, Y. E. (1983). A method for solving the convex programming problem with convergence rate  $o(1/k^2)$ . *Dokl. akad. nauk Sssr*, 269, 543–547.
- Nguyen, H. H., Fang, F., Yamagishi, J., & Echizen, I. (2019). Multi-task learning for detecting and segmenting manipulated facial images and videos. *2019 IEEE 10th international conference on biometrics theory, applications and systems (BTAS)*, 1–8.
- Nie, F., Hu, Z., & Li, X. (2018). Calibrated multi-task learning. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2012–2021.
- Niu, Z., Zhong, G., & Yu, H. (2021). A review on the attention mechanism of deep learning. *Neurocomputing*, 452, 48–62.
- Noroozi, M., & Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. *European conference on computer vision*, 69–84.
- Noroozi, M., Pirsiavash, H., & Favaro, P. (2017). Representation learning by learning to count. *Proceedings of the IEEE international conference on computer vision*, 5898–5906.
- Obozinski, G., Taskar, B., & Jordan, M. (2006). Multi-task feature selection. *Statistics Department, UC Berkeley, Tech. Rep*, 2(2.2), 2.
- Oord, A. v. d., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.
- OpenAI. (2023). Gpt-4 technical report.
- Oseledets, I. V. (2011). Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5), 2295–2317.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35, 27730–27744.
- Pal, M., Jati, A., Peri, R., Hsu, C.-C., AbdAlmageed, W., & Narayanan, S. (2021). Adversarial defense for deep speaker recognition using hybrid adversarial training. *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6164–6168.
- Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), 1345–1359.
- Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.
- Parameswaran, S., & Weinberger, K. Q. (2010). Large margin multi-task metric learning. *Advances in neural information processing systems*, 23.
- Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural networks*, 113, 54–71.

- Parra, G., & Tobar, F. (2017). Spectral mixture kernels for multi-output gaussian processes. *Advances in Neural Information Processing Systems*, 30.
- Patel, V. M., Gopalan, R., Li, R., & Chellappa, R. (2015). Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3), 53–69.
- Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2536–2544.
- Pearson, K. (1895). VII. note on regression and inheritance in the case of two parents. *proceedings of the royal society of London*, 58(347-352), 240–242.
- Peng, H., Yang, R., Wang, Z., Li, J., He, L., Philip, S. Y., Zomaya, A. Y., & Ranjan, R. (2021). Lime: Low-cost and incremental learning for dynamic heterogeneous information networks. *IEEE Transactions on Computers*, 71(3), 628–642.
- Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., & Wang, B. (2019). Moment matching for multi-source domain adaptation. *Proceedings of the IEEE/CVF international conference on computer vision*, 1406–1415.
- Peng, Z., Dong, L., Bao, H., Ye, Q., & Wei, F. (2022). Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*.
- Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.
- Pérez-García, F., Scott, C., Sparks, R., Diehl, B., & Ourselin, S. (2021). Transfer learning of deep spatiotemporal networks to model arbitrarily long videos of seizures. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 334–344.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. <https://doi.org/10.18653/v1/N18-1202>
- Plank, B., Søgaard, A., & Goldberg, Y. (2016). Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. *arXiv preprint arXiv:1604.05529*.
- Pong, T. K., Tseng, P., Ji, S., & Ye, J. (2010). Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization*, 20(6), 3465–3489.
- Pramanik, S., Agrawal, P., & Hussain, A. (2019). Omninet: A unified architecture for multi-modal multi-task learning. *arXiv preprint arXiv:1907.07804*.
- Quellec, G., Lamard, M., Conze, P.-H., Massin, P., & Cochener, B. (2020). Automatic detection of rare pathologies in fundus photographs using few-shot learning. *Medical image analysis*, 61, 101660.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. *International Conference on Machine Learning*, 8748–8763.
- Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 9.
- Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., Aslanides, J., Henderson, S., Ring, R., Young, S., et al. (2021). Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1), 5485–5551.

- Raghu, A., Lorraine, J., Kornblith, S., McDermott, M., & Duvenaud, D. K. (2021). Meta-learning to improve pre-training. *Advances in Neural Information Processing Systems*, 34, 23231–23244.
- Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., & Sutskever, I. (2021). Zero-shot text-to-image generation. *International Conference on Machine Learning*, 8821–8831.
- Ravaut, M., Joty, S., & Chen, N. F. (2022). Summareranker: A multi-task mixture-of-experts re-ranking framework for abstractive summarization. *arXiv preprint arXiv:2203.06569*.
- Redko, I., Morvant, E., Habrard, A., Sebban, M., & Bennani, Y. (2019). *Advances in domain adaptation theory*. Elsevier.
- Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J. T., et al. (2022). A generalist agent. *arXiv preprint arXiv:2205.06175*.
- Ren, X., Zhou, P., Meng, X., Huang, X., Wang, Y., Wang, W., Li, P., Zhang, X., Podolskiy, A., Arshinov, G., et al. (2023). Pangu- $\Sigma$ : Towards trillion parameter language model with sparse heterogeneous computing. *arXiv preprint arXiv:2303.10845*.
- Ribeiro, D., Wang, S., Ma, X., Zhu, H., Dong, R., Kong, D., Burger, J., Ramos, A., Wang, W., Huang, Z., et al. (2023). Street: A multi-task structured reasoning and explanation benchmark. *arXiv preprint arXiv:2302.06729*.
- Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, 22, 400–407.
- Rohde, A., & Tsybakov, A. B. (2011). Estimation of high-dimensional low-rank matrices1. *The Annals of Statistics*, 39(2), 887–930.
- Romera-Paredes, B., Aung, H., Bianchi-Berthouze, N., & Pontil, M. (2013). Multilinear multitask learning. *International Conference on Machine Learning*, 1444–1452.
- Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., & Bengio, Y. (2014). Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III* 18, 234–241.
- Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. M. (2016). The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3234–3243.
- Roy, A., Koutlis, C., Papadopoulos, S., & Ntoutsi, E. (2023). Fairbranch: Fairness conflict correction on task-group branches for fair multi-task learning. *arXiv preprint arXiv:2310.13746*.
- Roy, A., & Ntoutsi, E. (2022). Learning to teach fairness-aware deep multi-task learning. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 710–726.
- Royer, A., Blankevoort, T., & Bejnordi, B. E. (2023). Scalarization for multi-task and multi-domain learning at scale. *arXiv preprint arXiv:2310.08910*.
- Rubin, O., Herzig, J., & Berant, J. (2021). Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.
- Rubin, O., Herzig, J., & Berant, J. (2022, July). Learning to retrieve prompts for in-context learning. In M. Carpuat, M.-C. de Marneffe, & I. V. Meza Ruiz (Eds.), *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 2655–2671). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2022.naacl-main.191>
- Ruchte, M., & Grabocka, J. (2021). Scalable pareto front approximation for deep multi-objective learning. *2021 IEEE international conference on data mining (ICDM)*, 1306–1311.

- Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.
- Ruder, S., Bingel, J., Augenstein, I., & Søgaard, A. (2019). Latent multi-task architecture learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), 4822–4829.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088), 533–536.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115, 211–252.
- Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. *Advances in neural information processing systems*, 30.
- Saenko, K., Kulis, B., Fritz, M., & Darrell, T. (2010). Adapting visual category models to new domains. *European conference on computer vision*, 213–226.
- Sanh, V., Wolf, T., & Ruder, S. (2019). A hierarchical multi-task approach for learning embeddings from semantic tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01), 6949–6956.
- SarcheshmehPour, Y., Tian, Y., Zhang, L., & Jung, A. (2021). Networked federated multi-task learning.
- Sarkar, R., Liang, H., Fan, Z., Wang, Z., & Hao, C. (2023). Edge-moe: Memory-efficient multi-task vision transformer architecture with task-level sparsity via mixture-of-experts. *arXiv preprint arXiv:2305.18691*.
- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. *IEEE transactions on neural networks*, 20(1), 61–80.
- Sener, O., & Koltun, V. (2018). Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31.
- Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
- Senushkin, D., Patakin, N., Kuznetsov, A., & Konushin, A. (2023). Independent component alignment for multi-task learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20083–20093.
- Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., & Sun, J. (2019). Objects365: A large-scale, high-quality dataset for object detection. *Proceedings of the IEEE/CVF international conference on computer vision*, 8430–8439.
- Shazeer, N., Mirhoseini, \*, Maziarz, \*, Davis, A., Le, Q., Hinton, G., & Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *International Conference on Learning Representations*. <https://openreview.net/forum?id=B1ckMDqlg>
- She, Y. (2017). Selective factor extraction in high dimensions. *Biometrika*, 104(1), 97–110.
- Shen, S., Yang, S., Zhang, T., Zhai, B., Gonzalez, J. E., Keutzer, K., & Darrell, T. (2024). Multi-task vision-language prompt tuning. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 5656–5667.
- Shi, G., Li, Q., Zhang, W., Chen, J., & Wu, X.-M. (2023). Recon: Reducing conflicting gradients from the root for multi-task learning. *The Eleventh International Conference on Learning Representations*. [https://openreview.net/forum?id=ivwZO-HnzG\\_](https://openreview.net/forum?id=ivwZO-HnzG_)
- Shim, H., Jung, J., Heo, H., Yoon, S., & Yu, H. (2018). Replay attack spoofing detection system using replay noise by multi-task learning. *arXiv preprint arXiv: 1808.09638*.
- Shukor, M., Dancette, C., Rame, A., & Cord, M. (2023). Unified model for image, video, audio and language tasks. *arXiv preprint arXiv:2307.16184*.
- Signoretto, M., De Lathauwer, L., & Suykens, J. A. (2013). Learning tensors in reproducing kernel hilbert spaces with multilinear spectral penalties. *arXiv preprint arXiv:1310.4977*.

- Silberman, N., Hoiem, D., Kohli, P., & Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. *European conference on computer vision*, 746–760.
- Simchowitz, M., Tosh, C., Krishnamurthy, A., Hsu, D. J., Lykouris, T., Dudik, M., & Schapire, R. E. (2021). Bayesian decision-making under misspecified priors with applications to meta-learning. *Advances in Neural Information Processing Systems*, 34, 26382–26394.
- Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2013). A sparse-group lasso. *Journal of computational and graphical statistics*, 22(2), 231–245.
- Sinha, A., Chen, Z., Badrinarayanan, V., & Rabinovich, A. (2018). Gradient adversarial training of neural networks. *arXiv preprint arXiv:1806.08028*.
- Smith, V., Chiang, C.-K., Sanjabi, M., & Talwalkar, A. S. (2017). Federated multi-task learning. *Advances in neural information processing systems*, 30.
- Søgaard, A., & Goldberg, Y. (2016). Deep multi-task learning with low level tasks supervised at lower layers. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 231–235. <https://doi.org/10.18653/v1/P16-2038>
- Song, F., Liu, X., & Ma, X. (2021). Transfer learning for gene ranking across cancers. *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 1886–1891.
- Sorensen, T., Robinson, J., Rytting, C., Shaw, A., Rogers, K., Delorey, A., Khalil, M., Fulda, N., & Wingate, D. (2022, May). An information-theoretic approach to prompt engineering without ground truth labels. In S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 819–862). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2022.acl-long.60>
- Spearman, C. (1961). The proof and measurement of association between two things.
- Sperduti, A., & Starita, A. (1997). Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3), 714–735.
- Srivastava, S., & Sharma, G. (2024). Omnivec: Learning robust representations with cross modal sharing. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1236–1248.
- Standley, T., Gao, R., Chen, D., Wu, J., & Savarese, S. (2023). An extensible multimodal multi-task object dataset with materials. *arXiv preprint arXiv:2305.14352*.
- Standley, T., Zamir, A., Chen, D., Guibas, L., Malik, J., & Savarese, S. (2020). Which tasks should be learned together in multi-task learning? *International Conference on Machine Learning*, 9120–9132.
- Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., & Dai, J. (2019). Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*.
- Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. *Proceedings of the IEEE international conference on computer vision*, 843–852.
- Sun, G., Probst, T., Paudel, D. P., Popović, N., Kanakis, M., Patel, J., Dai, D., & Van Gool, L. (2021). Task switching network for multi-task learning. *Proceedings of the IEEE/CVF international conference on computer vision*, 8291–8300.
- Sun, H., Xu, J., Zheng, K., Zhao, P., Chao, P., & Zhou, X. (2021). Mfnp: A meta-optimized model for few-shot next poi recommendation. *IJCAI*, 3017–3023.
- Sun, Q., Xiang, S., & Ye, J. (2013). Robust principal component analysis via capped norms. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 311–319.
- Sun, T., He, Z., Zhu, Q., Qiu, X., & Huang, X. (2023, July). Multitask pre-training of modular prompt for Chinese few-shot learning. In A. Rogers, J. Boyd-Graber, & N. Okazaki

- (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 11156–11172). Association for Computational Linguistics. <https://doi.org/10.18653/v1/2023.acl-long.625>
- Sun, Y., Wang, S., Feng, S., Ding, S., Pang, C., Shang, J., Liu, J., Chen, X., Zhao, Y., Lu, Y., et al. (2021). Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. *arXiv preprint arXiv:2107.02137*.
- Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., & Wu, H. (2019). Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.
- Sun, Y., Wang, S., Li, Y., Feng, S., Tian, H., Wu, H., & Wang, H. (2020). Ernie 2.0: A continual pre-training framework for language understanding. *Proceedings of the AAAI conference on artificial intelligence*, 34(05), 8968–8975.
- Taiga, A. A., Agarwal, R., Farebrother, J., Courville, A., & Bellemare, M. G. (2022). Investigating multi-task pretraining and generalization in reinforcement learning. *The Eleventh International Conference on Learning Representations*.
- Tan, B., Song, Y., Zhong, E., & Yang, Q. (2015). Transitive transfer learning. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1155–1164.
- Tan, H., & Bansal, M. (2019). Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.
- Tang, H., Liu, J., Zhao, M., & Gong, X. (2020). Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations. *Proceedings of the 14th ACM Conference on Recommender Systems*, 269–278.
- Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., et al. (2022). Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.
- Thung, K.-H., & Wee, C.-Y. (2018). A brief review on multi-task learning. *Multimedia Tools and Applications*, 77(22), 29705–29725.
- Tian, Y., Krishnan, D., & Isola, P. (2020). Contrastive multiview coding. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI* 16, 776–794.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267–288.
- Titsias, M., & Lázaro-Gredilla, M. (2011). Spike and slab variational inference for multi-task and multiple kernel learning. *Advances in neural information processing systems*, 24.
- Tomioka, R., & Suzuki, T. (2013). Convex tensor decomposition via structured schatten norm regularization. *Advances in neural information processing systems*, 26.
- Tong, Z., Song, Y., Wang, J., & Wang, L. (2022). Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35, 10078–10093.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Trivedi, H., Balasubramanian, N., Khot, T., & Sabharwal, A. (2022). Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.
- Uematsu, Y., Fan, Y., Chen, K., Lv, J., & Lin, W. (2019). Sofar: Large-scale association network learning. *IEEE transactions on information theory*, 65(8), 4924–4939.
- Vafaeikia, P., Namdar, K., & Khalvati, F. (2020). A brief review of deep multi-task learning and auxiliary task learning. *arXiv preprint arXiv:2007.01126*.

- Vandenberghe, L., & Boyd, S. (1996). Semidefinite programming. *SIAM review*, 38(1), 49–95.
- Vandenhende, S., Georgoulis, S., De Brabandere, B., & Van Gool, L. (2019). Branched multi-task networks: Deciding what layers to share. *arXiv preprint arXiv:1904.02920*.
- Vandenhende, S., Georgoulis, S., Gool, L. V., & Brabandere, B. D. (2020). Branched multi-task networks: Deciding what layers to share. *31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7-10, 2020*. <https://www.bmvc2020-conference.com/assets/papers/0213.pdf>
- Vandenhende, S., Georgoulis, S., Van Gansbeke, W., Proesmans, M., Dai, D., & Van Gool, L. (2021). Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence*.
- Vandenhende, S., Georgoulis, S., & Van Gool, L. (2020). Mti-net: Multi-scale task interaction networks for multi-task learning. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, 527–543.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- Veličković, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., & Hjelm, R. D. (2018). Deep graph infomax. *arXiv preprint arXiv:1809.10341*.
- Venkateswara, H., Eusebio, J., Chakraborty, S., & Panchanathan, S. (2017). Deep hashing network for unsupervised domain adaptation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5018–5027.
- Vithayathil Varghese, N., & Mahmoud, Q. H. (2020). A survey of multi-task deep reinforcement learning. *Electronics*, 9(9), 1363.
- Vu, N. T., Gupta, P., Adel, H., & Schütze, H. (2016). Bi-directional recurrent neural network with ranking loss for spoken language understanding. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6060–6064.
- Vu, T., Lester, B., Constant, N., Al-Rfou, R., & Cer, D. (2021). Spot: Better frozen model adaptation through soft prompt transfer. *arXiv preprint arXiv:2110.07904*.
- Wang, H., Li, J., Wu, H., Hovy, E., & Sun, Y. (2022). Pre-trained language models and their applications. *Engineering*.
- Wang, H., Zhang, Z., Fan, Z., Chen, J., Zhang, L., Shibasaki, R., & Song, X. (2023). Multi-task weakly supervised learning for origin-destination travel time estimation. *IEEE Transactions on Knowledge and Data Engineering*.
- Wang, J., Ge, Y., Yan, R., Ge, Y., Lin, K. Q., Tsutsui, S., Lin, X., Cai, G., Wu, J., Shan, Y., et al. (2023). All in one: Exploring unified video-language pre-training. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6598–6608.
- Wang, L., Huang, B., Zhao, Z., Tong, Z., He, Y., Wang, Y., Wang, Y., & Qiao, Y. (2023). Videomae v2: Scaling video masked autoencoders with dual masking. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14549–14560.
- Wang, P., Yang, A., Men, R., Lin, J., Bai, S., Li, Z., Ma, J., Zhou, C., Zhou, J., & Yang, H. (2022). Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. *International Conference on Machine Learning*, 23318–23340.
- Wang, R., Chen, D., Wu, Z., Chen, Y., Dai, X., Liu, M., Jiang, Y.-G., Zhou, L., & Yuan, L. (2022). Bevt: Bert pretraining of video transformers. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14733–14743.
- Wang, S., Li, Y., Li, H., Zhu, T., Li, Z., & Ou, W. (2022). Multi-task learning with calibrated mixture of insightful experts. *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 3307–3319.
- Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3), 1–34.

- Wang, Y., Dong, X., Li, G., Dong, J., & Yu, H. (2022). Cascade regression-based face frontalization for dynamic facial expression analysis. *Cognitive Computation*, 14(5), 1571–1584.
- Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Arunkumar, A., Ashok, A., Dhanasekaran, A. S., Naik, A., Stap, D., et al. (2022). Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*.
- Wang, Y., Wang, X., Beutel, A., Prost, F., Chen, J., & Chi, E. H. (2021). Understanding and improving fairness-accuracy trade-offs in multi-task learning. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1748–1757.
- Wang, Z., Tsvetkov, Y., Firat, O., & Cao, Y. (2021). Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. *International Conference on Learning Representations*. [https://openreview.net/forum?id=F1vEjWK-lH\\_](https://openreview.net/forum?id=F1vEjWK-lH_)
- Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., & Le, Q. V. (2021). Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.
- Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. (2022). Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35, 24824–24837.
- Wei, J., Hou, L., Lampinen, A., Chen, X., Huang, D., Tay, Y., Chen, X., Lu, Y., Zhou, D., Ma, T., et al. (2023). Symbol tuning improves in-context learning in language models. *arXiv preprint arXiv:2305.08298*.
- Wei, W., Huang, C., Xia, L., Xu, Y., Zhao, J., & Yin, D. (2022). Contrastive meta learning with behavior multiplicity for recommendation. *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 1120–1128.
- Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural networks*, 1(4), 339–356.
- Widmer, C., Leiva, J., Altun, Y., & Rätsch, G. (2010). Leveraging sequence classification by taxonomy-based multitask learning. *Research in Computational Molecular Biology: 14th Annual International Conference, RECOMB 2010, Lisbon, Portugal, April 25-28, 2010. Proceedings* 14, 522–534.
- Wilson, A., Fern, A., Ray, S., & Tadepalli, P. (2007). Multi-task reinforcement learning: A hierarchical bayesian approach. *Proceedings of the 24th international conference on Machine learning*, 1015–1022.
- Wimalawarne, K., Sugiyama, M., & Tomioka, R. (2014). Multitask learning meets tensor factorization: Task imputation via convex optimization. *Advances in neural information processing systems*, 27.
- Wu, Z., Xiong, Y., Yu, S. X., & Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3733–3742.
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1), 4–24.
- Xiao, Y., Chang, Z., & Liu, B. (2020). An efficient active learning method for multi-task learning. *Knowledge-Based Systems*, 190, 105137.
- Xiao, Y., Wen, J., & Liu, B. (2021). A new multi-task learning method with universum data. *Applied Intelligence*, 51(6), 3421–3434.
- Xie, L., Baytas, I. M., Lin, K., & Zhou, J. (2017). Privacy-preserving distributed multi-task learning with asynchronous updates. *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 1195–1204.

- Xie, S., Zheng, H., Liu, C., & Lin, L. (2018). Snas: Stochastic neural architecture search. *arXiv preprint arXiv:1812.09926*.
- Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., & Hu, H. (2022). Simmim: A simple framework for masked image modeling. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9653–9663.
- Xu, B., Wang, Q., Mao, Z., Lyu, Y., She, Q., & Zhang, Y. (2023).  $k$  Nn prompting: Beyond-context learning with calibration-free nearest neighbor inference. *arXiv preprint arXiv:2303.13824*.
- Xu, C., Xu, Y., Wang, S., Liu, Y., Zhu, C., & McAuley, J. (2023). Small models are valuable plug-ins for large language models. *arXiv preprint arXiv:2305.08848*.
- Xu, C., Tao, D., & Xu, C. (2013). A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*.
- Xu, D., Ouyang, W., Wang, X., & Sebe, N. (2018). Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 675–684.
- Xu, L., Ouyang, W., Bennamoun, M., Boussaid, F., Sohel, F., & Xu, D. (2021). Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6984–6993.
- Xu, X., Zhao, H., Vineet, V., Lim, S.-N., & Torralba, A. (2022). Mtformer: Multi-task learning via transformer and cross-task reasoning. *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, 304–321.
- Xu, Y., Li, X., Yuan, H., Yang, Y., Zhang, J., Tong, Y., Zhang, L., & Tao, D. (2022). Multi-task learning with multi-query transformer for dense prediction. *arXiv preprint arXiv:2205.14354*.
- Yang, C., Pan, J., Gao, X., Jiang, T., Liu, D., & Chen, G. (2022). Cross-task knowledge distillation in multi-task recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(4), 4318–4326.
- Yang, J., Duan, J., Tran, S., Xu, Y., Chanda, S., Chen, L., Zeng, B., Chilimbi, T., & Huang, J. (2022). Vision-language pre-training with triple contrastive learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15671–15680.
- Yang, R., Deng, Y., Zhu, A., Tong, X., & Chen, Z. (2021). Few shot learning based on the street view house numbers (svhn) dataset. *Edge Computing and IoT: Systems, Management and Security: First EAI International Conference, ICECI 2020, Virtual Event, November 6, 2020, Proceedings 1*, 86–102.
- Yang, Y., & Hospedales, T. (2016). Deep multi-task representation learning: A tensor factorisation approach. *arXiv preprint arXiv:1605.06391*.
- Yang, Z., Gan, Z., Wang, J., Hu, X., Ahmed, F., Liu, Z., Lu, Y., & Wang, L. (2022). Unitab: Unifying text and box outputs for grounded vision-language modeling. *European Conference on Computer Vision*, 521–539.
- Yang, Z., Zhang, Y., Yu, J., Cai, J., & Luo, J. (2018). End-to-end multi-modal multi-task vehicle control for self-driving cars with visual perceptions. *2018 24th International Conference on Pattern Recognition (ICPR)*, 2289–2294.
- Yao, J., Wang, F., Jia, K., Han, B., Zhou, J., & Yang, H. (2021). Device-cloud collaborative learning for recommendation. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 3865–3874.
- Ye, J., Wu, Z., Feng, J., Yu, T., & Kong, L. (2023). Compositional exemplars for in-context learning. *arXiv preprint arXiv:2302.05698*.
- Ye, M., & qiang liu. (2022). Optimization in pareto set: Searching special pareto model. *The 38th Conference on Uncertainty in Artificial Intelligence*. <https://openreview.net/forum?id=SZ4G8L8j5l5>
- Ye, Q., Zha, J., & Ren, X. (2022). Eliciting transferability in multi-task learning with task-level mixture-of-experts. *arXiv preprint arXiv:2205.12701*.

- Yi, K., Ge, Y., Li, X., Yang, S., Li, D., Wu, J., Shan, Y., & Qie, X. (2022). Masked image modeling with denoising contrast. *arXiv preprint arXiv:2205.09616*.
- Yu, J., Cui, C., Geng, L., Ma, Y., & Yin, Y. (2019). Towards unified aesthetics and emotion prediction in images. *2019 IEEE international conference on image processing (ICIP)*, 2526–2530.
- Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., & Finn, C. (2020). Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33, 5824–5836.
- Yu, Z., He, L., Wu, Z., Dai, X., & Chen, J. (2023). Towards better chain-of-thought prompting strategies: A survey. *arXiv preprint arXiv:2310.04959*.
- Yuan, L., Wang, Y., Thompson, P. M., Narayan, V. A., & Ye, J. (2012). Multi-source learning for joint analysis of incomplete multi-modality neuroimaging data. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1149–1157.
- Yun, H., & Cho, H. (2023). Achievement-based training progress balancing for multi-task learning. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16935–16944.
- Zamir, A. R., Sax, A., Shen, W., Guibas, L. J., Malik, J., & Savarese, S. (2018). Taskonomy: Disentangling task transfer learning. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3712–3722.
- Zellers, R., Lu, X., Hessel, J., Yu, Y., Park, J. S., Cao, J., Farhadi, A., & Choi, Y. (2021). Merlot: Multimodal neural script knowledge models. *Advances in Neural Information Processing Systems*, 34, 23634–23651.
- Zeng, N., Li, H., & Peng, Y. (2023). A new deep belief network-based multi-task learning for diagnosis of alzheimer’s disease. *Neural Computing and Applications*, 35(16), 11599–11610.
- Zeng, W., Ren, X., Su, T., Wang, H., Liao, Y., Wang, Z., Jiang, X., Yang, Z., Wang, K., Zhang, X., et al. (2021). Pangu- $\alpha$ : Large-scale autoregressive pretrained chinese language models with auto-parallel computation. *arXiv preprint arXiv:2104.12369*.
- Zhang, C., Hu, X., Xie, Y., Gong, M., & Yu, B. (2020). A privacy-preserving multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *Frontiers in neurorobotics*, 13, 112.
- Zhang, D., Hu, Y., Ye, J., Li, X., & He, X. (2012). Matrix completion by truncated nuclear norm regularization. *2012 IEEE Conference on computer vision and pattern recognition*, 2192–2199.
- Zhang, J. (2006). A probabilistic framework for multi-task learning. *Ph.D. Thesis, Carnegie Mellon University*.
- Zhang, K., Yu, J., Yan, Z., Liu, Y., Adhikarla, E., Fu, S., Chen, X., Chen, C., Zhou, Y., Li, X., et al. (2023). Biomedgpt: A unified and generalist biomedical generative pre-trained transformer for vision, language, and multimodal tasks. *arXiv preprint arXiv:2305.17100*.
- Zhang, L., et al. (2023). Demt: Deformable mixer transformer for multi-task learning of dense prediction. *arXiv preprint arXiv:2301.03461*.
- Zhang, M., Yin, R., Yang, Z., Wang, Y., & Li, K. (2023). Advances and challenges of multi-task learning method in recommender system: A survey. *arXiv preprint arXiv:2305.13843*.
- Zhang, R., Isola, P., & Efros, A. A. (2016). Colorful image colorization. *Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III* 14, 649–666.
- Zhang, R., Isola, P., & Efros, A. A. (2017). Split-brain autoencoders: Unsupervised learning by cross-channel prediction. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1058–1067.
- Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. (2022). Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

- Zhang, T. (2010). Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11(3).
- Zhang, T. (2013). Multi-stage convex relaxation for feature selection. *Bernoulli*, 19(5B), 2277–2293.
- Zhang, W., Zhai, G., Wei, Y., Yang, X., & Ma, K. (2023). Blind image quality assessment via vision-language correspondence: A multitask learning perspective. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14071–14081.
- Zhang, X., Zhang, X., Liu, H., & Luo, J. (2018). Multi-task clustering with model relation learning. *IJCAI*, 3132–3140.
- Zhang, Y., Zhang, Y., & Wang, W. (2022). Learning linear and nonlinear low-rank structure in multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*.
- Zhang, Y., Feng, S., & Tan, C. (2022). Active example selection for in-context learning. *arXiv preprint arXiv:2211.04486*.
- Zhang, Y., Cheng, D. Z., Yao, T., Yi, X., Hong, L., & Chi, E. H. (2021a). A model of two tales: Dual transfer learning framework for improved long-tail item recommendation. *Proceedings of the Web Conference 2021*, 2220–2231.
- Zhang, Y., Cheng, D. Z., Yao, T., Yi, X., Hong, L., & Chi, E. H. (2021b). A model of two tales: Dual transfer learning framework for improved long-tail item recommendation. *Proceedings of the web conference 2021*, 2220–2231.
- Zhang, Y., Wu, X., Fang, Q., Qian, S., & Xu, C. (2023). Knowledge-enhanced attributed multi-task learning for medicine recommendation. *ACM Transactions on Information Systems*, 41(1), 1–24.
- Zhang, Y., & Yang, Q. (2017). Learning sparse task relations in multi-task learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).
- Zhang, Y., & Yang, Q. (2021). A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*.
- Zhang, Y., & Yeung, D.-Y. (2009). Semi-supervised multi-task regression. *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2009, Bled, Slovenia, September 7-11, 2009, Proceedings, Part II* 20, 617–631.
- Zhang, Y., & Yeung, D.-Y. (2010). Multi-task learning using generalized t process. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 964–971.
- Zhang, Y., & Yeung, D.-Y. (2012a). A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536*.
- Zhang, Y., & Yeung, D.-Y. (2012b). Multi-task boosting by exploiting task relationships. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 697–710.
- Zhang, Y., & Yeung, D.-Y. (2013). Multilabel relationship learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 7(2), 1–30.
- Zhang, Y., & Yeung, D.-Y. (2014). A regularization approach to learning task relationships in multitask learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(3), 1–31.
- Zhang, Y., Zhou, K., & Liu, Z. (2023). What makes good examples for visual in-context learning? *arXiv preprint arXiv:2301.13670*.
- Zhang, Z., Luo, P., Loy, C. C., & Tang, X. (2014). Facial landmark detection by deep multi-task learning. *European conference on computer vision*, 94–108.
- Zhang, Z., Cui, Z., Xu, C., Jie, Z., Li, X., & Yang, J. (2018). Joint task-recursive learning for semantic segmentation and depth estimation. *Proceedings of the European Conference on Computer Vision (ECCV)*, 235–251.
- Zhang, Z., Cui, Z., Xu, C., Yan, Y., Sebe, N., & Yang, J. (2019). Pattern-affinitive propagation across depth, surface normal and semantic segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4106–4115.

- Zhang, Z., Zhang, A., Li, M., & Smola, A. (2022). Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.
- Zhao, J., Du, B., Sun, L., Zhuang, F., Lv, W., & Xiong, H. (2019). Multiple relational attention network for multi-task learning. *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & Data Mining*, 1123–1131.
- Zhao, J., Xie, X., Xu, X., & Sun, S. (2017). Multi-view learning overview: Recent progress and new challenges. *Information Fusion*, 38, 43–54.
- Zhao, Y., Wang, X., Che, T., Bao, G., & Li, S. (2023). Multi-task deep learning for medical image computing and analysis: A review. *Computers in Biology and Medicine*, 153, 106496.
- Zhen, L., Hu, P., Wang, X., & Peng, D. (2019). Deep supervised cross-modal retrieval. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10394–10403.
- Zheng, F., Deng, C., Sun, X., Jiang, X., Guo, X., Yu, Z., Huang, F., & Ji, R. (2019). Pyramidal person re-identification via multi-loss dynamic training. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8514–8522.
- Zheng, X., Lin, L., Liu, B., Xiao, Y., & Xiong, X. (2020). A multi-task transfer learning method with dictionary learning. *Knowledge-Based Systems*, 191, 105233.
- Zhou, C., Li, Q., Li, C., Yu, J., Liu, Y., Wang, G., Zhang, K., Ji, C., Yan, Q., He, L., et al. (2023). A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*.
- Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q., et al. (2022). Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.
- Zhou, J., Chen, J., & Ye, J. (2011). Clustered multi-task learning via alternating structure optimization. *Advances in neural information processing systems*, 24.
- Zhou, J., Liu, J., Narayan, V. A., & Ye, J. (2012). Modeling disease progression via fused sparse group lasso. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1095–1103.
- Zhou, J., Yuan, L., Liu, J., & Ye, J. (2011). A multi-task learning formulation for predicting disease progression. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 814–822.
- Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., & Kong, T. (2021). Ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*.
- Zhou, L., Cui, Z., Xu, C., Zhang, Z., Wang, C., Zhang, T., & Yang, J. (2020). Pattern-structure diffusion for multi-task learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4514–4523.
- Zhou, Z.-H., Zhang, M.-L., Huang, S.-J., & Li, Y.-F. (2012). Multi-instance multi-label learning. *Artificial Intelligence*, 176(1), 2291–2320.
- Zhu, J., Zhu, X., Wang, W., Wang, X., Li, H., Wang, X., & Dai, J. (2022). Uni-perceiver-moe: Learning sparse generalist models with conditional moes. *Advances in Neural Information Processing Systems*, 35, 2664–2678.
- Zhu, X., Hu, H., Lin, S., & Dai, J. (2019). Deformable convnets v2: More deformable, better results. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9308–9316.
- Zhu, X., Zhu, J., Li, H., Wu, X., Li, H., Wang, X., & Dai, J. (2022a). Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16804–16815.
- Zhu, X., Zhu, J., Li, H., Wu, X., Li, H., Wang, X., & Dai, J. (2022b). Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16804–16815.

- Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1), 43–76.
- Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2), 301–320.