A survey on transfer learning for evolving domains

Anonymous authors
Paper under double-blind review

Abstract

Transfer learning explores how to leverage knowledge from various tasks or domains (sources) to enhance predictive performance in related tasks or domains (targets). Typically, transfer learning research is segmented into several isolated sub-areas (such as domain generalisation, domain adaptation or multi-domain learning), each making distinct assumptions about target data availability, such as the quantity of data and labels available at training time. However, there are several real-world applications where these problems occur as a continuum, evolving from one stage to another as more data and labels are progressively collected from each domain. In those cases, a robust transfer learning solution should seamlessly integrate an expanding dataset and progressively improve its performance over time. In this survey, we review the state of the art in transfer learning from the perspective of this continuum, focusing on the data requirements of each method. We find that most methods are tailored to specific settings, and no current work considers an integrated view over the whole spectrum of data availability. We refer to this new perspective on transfer learning as *Transfer Learning for Evolving Domains (TrED)* and argue that it is an important and challenging direction for future research.

1 Introduction

Machine Learning (ML) models have become the state-of-the-art approach for various predictive tasks, such as classification and regression. However, the performance of these models relies heavily on the availability of data, which can be difficult and costly to obtain. For example, labelling an event for money laundering detection requires manual work from a domain expert, making the compilation of a large annotated dataset time-consuming and expensive (Barata et al., 2021).

One approach to address these challenges is Transfer Learning (TL), which aims to leverage knowledge from one or more tasks or domains (Source) to enhance performance in another task or domain (Target).[1] This covers a range of scenarios, each making different assumptions about the volume and type of data available from the source and target domains at training time. For example, some approaches may use a small amount of labelled data from the target domain, while others rely solely on labelled data from the source domain.

Traditionally, different TL scenarios are treated as distinct problems with specific solutions. However, in real-world applications, there are many cases where these scenarios exist along a continuum, evolving as more data and labels are collected. Consider global services such as a streaming platform (e.g., Netflix), an e-commerce site (e.g., Amazon), or a financial service (e.g., PayPal) expanding into a new country or region. Initially, there might be no user data from this new market, but they can use historical data from other geographies to design a first solution. As the service gains users, data from the new market is continuously collected, enabling the improvement and adaptation of the deployed system. One real-world example showing the importance of this adaptation was studied at Spotify, where it was found that "users behave and interact differently in different markets" and as such it was more effective to switch to a localised model once enough data is available to train it (Roitero et al., 2020).

Another illustrative example of changes in the nature of data used by ML models are regulatory changes, which may not only change the data collection processes but also the behaviour of people and institutions

[1] In this survey, we focus on the TL setting of having a single task and various domains.

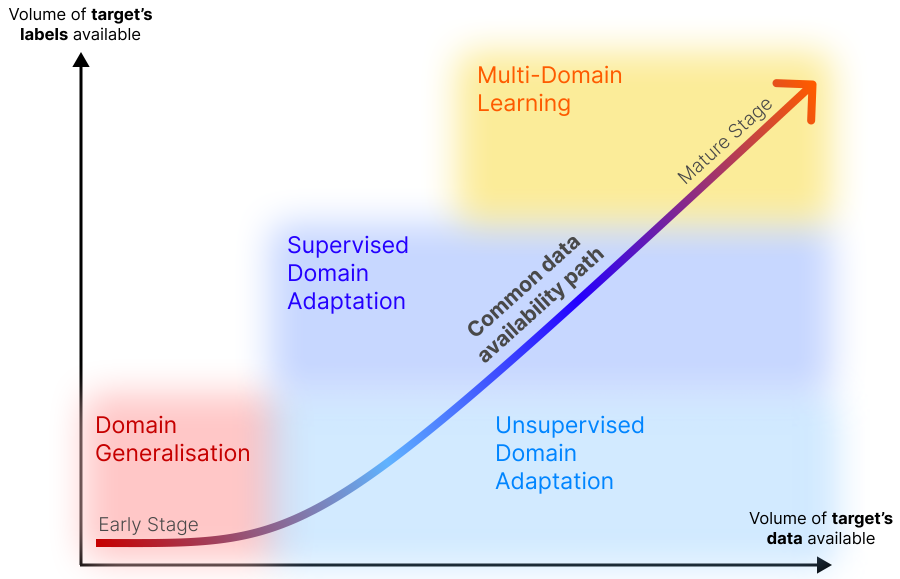

Figure 1: Transfer learning can be divided into different sub-problems that make different assumptions about target domain data and labels available. Once a new domain is included, the available data typically evolves over time according to the depicted arrow, starting from a Domain Generalisation setting (without target domain data), then moving towards Domain Adaptation (with target data but none or limited target labels), and later into Multi-Domain Learning (with considerable labelled target domain data).

governed by it. This can disrupt the distribution of features and/or labels for deployed ML models, leading to drops in predictive performance and the need to execute a model update. For example, due to the General Data Protection Regulation enforced by the European Union, companies may be required to restrict their data collection policies, which means that the deployed models would stop receiving several features that were used to train them (Sartor et al., 2020). However, as companies adapt to the new legal environment, they continue to collect and utilise data within the regulatory framework, leading to the development of new models that respect user privacy.

A similar progression can be seen during a new disease outbreak. In the initial absence of patient data, TL can leverage data from previous outbreaks or related diseases to develop early diagnostic models. As specific data for the new disease (such as symptoms and outcomes) accumulate over time, these models can be continuously updated and refined to improve their predictive accuracy. This strategy was recently employed during the COVID-19 pandemic. Early diagnostic models were developed using pre-trained computer vision models and data from similar infections. These models were then fine-tuned using the limited COVID-19 data available at the time (Altaf et al., 2021; Narin et al., 2021).

We consider scenarios where multiple source domains provide a large pool of labelled data. When a new target domain is introduced, it might initially lack any data. As data collection begins, obtaining labels may be delayed. Over time, the quantity of both instances and labels from the target domain increases. Figure 1 illustrates how this common data availability path overlaps with typical settings of some TL sub-problems.

This survey presents a novel perspective on TL, viewing these previously isolated problems as a continuum. We discuss the evolution of data availability and examine TL methods designed for various stages of this process. As no single method currently addresses the entire spectrum, we highlight promising techniques that could potentially span a broader range of scenarios. By formalising this new perspective as a new task, *Transfer Learning for Evolving Domains (TrED)*, this survey aims to establish a framework for research that will support the development of new TL methods that deal with the data availability continuum. From

an application point of view, we anticipate that the adoption of unified solutions, capable of effectively incorporating new target domain data, will speed up improvements in model performance and facilitate a more streamlined ML pipeline, reducing the need to frequently switch methodologies.

In Section 2 we introduce some notation and provide a formal definition of each sub-problem, as well as the newly proposed TrED problem. In Section 3 we present a literature review of TL, organised according to the data availability flow. We also discuss some foundation model approaches that can have some potential applications for TL and TrED. In Section 4 we present our conclusions and directions for future work.

2 Definitions and Notation

Transfer Learning (TL) is usually described as a set of methods to reuse knowledge from one task or domain to improve the performance on a different but related task or domain (Weiss et al., 2016). To formally ground the problem of learning across evolving domains, we first define the core concepts of "domains" and "tasks", then introduce a temporal framework to model the progressive availability of data, and later describe various problem settings related to TL.

2.1 Domains, Tasks and Transfer Learning

Following the definitions from Pan & Yang (2009), a **domain** $\mathcal{D} = \{\mathcal{X}, P(X)\}$ consists of a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X \in \mathcal{X}$ is a random variable representing the observed instances. As such, two domains $\mathcal{D}_1 = \{\mathcal{X}_1, P(X_1)\}$ and $\mathcal{D}_2 = \{\mathcal{X}_2, P(X_2)\}$ differ if they either have different feature spaces ($\mathcal{X}_1 \neq \mathcal{X}_2$) or different marginal probability distributions ($P(X_1) \neq P(X_2)$).

TL literature (Pan & Yang, 2009; Zhang et al., 2019; Zhuang et al., 2020) often presents two equivalent interpretations for the concept of task: as a deterministic predictive function $f : \mathcal{X} \to \mathcal{Y}$, or as a conditional probability distribution $P(Y|X)$. We adopt the probabilistic interpretation to better account for label uncertainty, noise, and potential concept drift over time. Given a specific domain $\mathcal{D}$, a **task** $\mathcal{T} = \{\mathcal{Y}, P(Y|X)\}$ consists of a label space $\mathcal{Y}$ and a conditional probability distribution $P(Y|X)$, which represents the predictive relationship between features and labels.

One domain can encompass multiple tasks (e.g. using the same set of images for an image classification task or an object detection task), and different domains can share the same task (e.g. having a spam detection task on texts from different languages). The formal definition of task would require that tasks in different domains should have the same label space and conditional probability distribution (and thus the same feature space) in order to be considered equal. We relax this requirement by considering tasks in different domains to be equivalent ($\mathcal{T}_1 \sim \mathcal{T}_2$) if they involve solving the same underlying problem. For example, we can have a common task of image classification on two image domains, one in colour and one in greyscale, even though the domains have different feature spaces (3 colour channels versus 1).

Transfer Learning is the sub-field of ML that studies how to leverage datasets from various domains and/or from different tasks to learn a better performing model. More formally, given a source domain $\mathcal{D}_S$ with a learning task $\mathcal{T}_S = \{\mathcal{Y}_S, P_S(Y|X)\}$ and a target domain $\mathcal{D}_T$ with a learning task $\mathcal{T}_T = \{\mathcal{Y}_T, P_T(Y|X)\}$, where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$, TL aims to use the knowledge of $\mathcal{D}_S$ and $\mathcal{T}_S$ to learn a predictive function $f : \mathcal{X} \to \mathcal{Y}$ that is a better approximation of $P_T(Y|X)$ than what could be learned only from $\mathcal{D}_T$. It is possible to have various source domains and various target domains, and these two sets can be disjoint, have some overlap, or be the same.

In this survey, we focus on the TL setting with different domains ($\mathcal{D}_S \neq \mathcal{D}_T$) but similar tasks ($\mathcal{T}_S \sim \mathcal{T}_T$).

2.2 Progressive data availability

In the standard ML setting, datasets are static collections of observations used to estimate $P(X)$ and $P(Y|X)$. However, in real-world pipelines, data is collected from multiple domains over time, often with labels arriving after a certain delay that depends on the annotation process.

Table 1: Data availability assumptions at training time t_{tr} from different TL settings: Domain Generalisation (DG), Unsupervised/Supervised Domains Adaptation (UDA / SDA), Multi-Domain Learning (MDL). In SDA, there may or may not be large volume of unlabelled target domain data. In MDL, there is no distinction between source and target domains, but there is a large volume of labelled data from all domains.

TL Setting	**Labelled Source Data** $\lvert D^L_{S_d}(t_{tr})\rvert$	**Unlabelled Target Data** $\lvert D^U_T(t_{tr})\rvert$	**Labelled Target Data** $\lvert D^L_T(t_{tr})\rvert$
DG	$\gg 0$	$= 0$	$= 0$
UDA	$\gg 0$	> 0	$= 0$
SDA	$\gg 0$	$\gg 0$ or NA	> 0
MDL	$\gg 0$	NA	$\gg 0$

To formalize this evolving availability, we refine the definition of dataset to include the temporal information. This way, each domain $\mathcal{D}_d$ is associated with a dynamically growing dataset $D_d = \{(x_i, y_i, t_i^x, t_i^y) \mid i = 1, \ldots, n_d\}$, where $x_i \in \mathcal{X}_d$ is a feature vector, $y_i \in \mathcal{Y}_d$ is the label, t_i^x is the timestamp when x_i is collected, and $t_i^y \geq t_i^x$ is the timestamp when y_i becomes available.

At any given time t, D_d can be decomposed into two disjoint subsets: a labelled dataset $D_d^L(t) = \{(x_i, y_i) \mid t_i^y \leq t\}$ with instances that have received their labels by time t; and an unlabelled dataset $D_d^U(t) = \{x_i \mid t_i^x \leq t < t_i^y\}$ with instances that have been observed, but whose labels are still unavailable at time t.

2.3 Standard TL settings

Within the topic of TL, different settings have been described and addressed, making different assumptions about the datasets used to train the ML models. In this survey, we mainly focus on four of these settings, which we identify as those that align with real-world data availability conditions (Figure 1): Domain Generalisation, Unsupervised Domain Adaptation, Supervised Domain Adaptation, and Multi-Domain Learning. They all assume that a large pool of labelled source domain data is available at training time t_{tr}, i.e., $\forall d \in \{1, \ldots, m\}, |D^L_{S_d}(t_{tr})| \gg 0$. Their main differences relate to whether target domain data is available and whether it is labelled or unlabelled. We summarize their assumptions in Table 1.

In **Domain Generalisation (DG)** (Zhou et al., 2022; Wang et al., 2022a), there is no target domain data available at training time ($|D^U_T(t_{tr})| = 0$ and $|D^L_T(t_{tr})| = 0$). The goal is to use $D^L_{S_1}(t_{tr}), \ldots, D^L_{S_m}(t_{tr})$ to learn a predictive function f that is a good approximation of $P_T(Y|X)$ for any new target domain $\mathcal{D}_T$, unknown at training time.

In **Unsupervised Domain Adaptation (UDA)** (Wilson & Cook, 2020), there is only unlabelled target domain data available at training time ($|D^U_T(t_{tr})| > 0$ but $|D^L_T(t_{tr})| = 0$). The goal is to use $D^L_{S_1}(t_{tr}), \ldots, D^L_{S_m}(t_{tr})$ and $D^U_T(t_{tr})$ to learn a predictive function f that approximates $P_T(Y|X)$ on $\mathcal{D}_T$.

In **Supervised Domain Adaptation (SDA)** (Wang & Deng, 2018), there is a small volume of labelled target domain data at training time, and there may be a much larger volume of unlabelled target domain data as well ($|D^U_T(t_{tr})| \gg |D^L_T(t_{tr})| > 0$). The goal is to use $D^L_{S_1}(t_{tr}), \ldots, D^L_{S_m}(t_{tr})$ and $D^L_T(t_{tr})$ (and $D^U_T(t_{tr})$ if available) to learn a predictive function f that is a good approximation of $P_T(Y|X)$ on $\mathcal{D}_T$.

In **Multi-Domain Learning (MDL)** (Yang & Hospedales, 2014), there are various labelled datasets from different domains available at training time, with no distinction between source and target domains ($\forall d \in \{1, \ldots, m\}, |D^L_d(t_{tr})| \gg 0$). The goal is to use $D^L_1(t_{tr}), \ldots, D^L_m(t_{tr})$ to learn one or multiple predictive functions such that the learned model(s) can perform well on each individual domain $\mathcal{D}_1, \ldots, \mathcal{D}_m$. Unlike DG, MDL has access to datasets from all domains (including the target) during training. Unlike Domain Adaptation, MDL aims to optimise performance across multiple domains simultaneously, rather than focusing on transferring knowledge to a single target domain.

2.4 Transfer Learning for Evolving Domains (TrED)

The four TL settings described in Section 2.3 are typically treated as distinct, isolated problems, each optimizing a model for a fixed snapshot of data availability at a specific training time t_{tr}. However, in real-world applications where data and labels are often collected gradually, the availability conditions change over time, and the different TL settings are realized as sequential stages in the lifecycle of every new domain, as depicted in Figure 1. The challenge is how to continuously integrate the new data from each domain in a way that allows the transfer learning model to seamlessly adapt and improve over time.

We refer to this scenario as Transfer Learning for Evolving Domains (TrED). By modelling datasets as a function of time t, we can map the traditionally isolated TL settings to stages of a newly introduced target domain. Suppose a new target domain $\mathcal{D}_T$ is introduced at time t_{start}, initially without any data ($D_T^L(t_{start}) = D_T^U(t_{start}) = \emptyset$), while there are large volumes of labelled data from all source domains ($|D_{S_d}^L(t_{start})| \gg 0, d \in \{1, \ldots, m\}$). The evolution of the problem can be described as follows:

- **Stage 1 (DG):** For $t \in [t_{start}, t_a[$, there is insufficient target domain data ($|D_T^U(t)| \approx 0$) to reliably estimate $P_T(X)$. The model must rely on source domains to generalize to the unseen $\mathcal{D}_T$.
- **Stage 2 (UDA):** For $t \in [t_a, t_b[$, some instances have been collected ($|D_T^U(t)| > 0$), but most of their labels are not available yet ($|D_T^L(t)| \approx 0$). The model must adapt using the target's marginal distribution $P_T(X)$.
- **Stage 3 (SDA):** For $t \in [t_b, t_c[$, the target domain contains a mix of labelled and unlabelled data ($|D_T^U(t)| > 0$ and $|D_T^L(t)| > 0$). The model can be fine-tuned to approximate $P_T(Y|X)$ directly.
- **Stage 4 (MDL):** For $t \geq t_c$, the volume of labelled target data becomes substantial ($|D_T^L(t)| \gg 0$). The target domain effectively matures into a new source domain, and the problem shifts to Multi-Domain Learning.

The transition points t_a, t_b, and t_c are not fixed constants but rather depend on environmental constraints, specifically the velocity of data collection and the distribution of label delay. Furthermore, while we categorize these stages into discrete intervals for clarity, in practice, the evolution of data availability is a continuous spectrum with diffuse boundaries. As such, a solution to the TrED problem should operate continuously across this spectrum, minimizing the cumulative error of approximating $P_T(Y|X)$ over time.

The TrED problem shares similarities with other dynamic learning paradigms, but there are differences that set it apart. Continual Learning (also known as Lifelong Learning or Incremental Learning) focuses on learning from a stream of data, gradually extending its knowledge base while retaining and using past experiences to aid future learning (De Lange et al., 2021; Van de Ven et al., 2022). Similar to TrED, Continual Learning involves a flow of new data, to which the model must adapt while leveraging previous knowledge. However, Continual Learning focuses on retaining knowledge over time and mitigating catastrophic forgetting, as tasks are typically presented sequentially. In contrast, TrED emphasises the transfer of knowledge across domains, utilising both historical and newly available data without the risk of forgetting since historical data can be revisited. Online Learning is a family of machine learning methods where data arrives sequentially and the model is updated continuously with each new data point to improve future predictions (Hoi et al., 2021). This type of techniques can be particularly useful in TrED as a way to integrate more recent data into the knowledge base of the model. However, the standard Online Learning paradigm usually deals with a single stream of data, whereas TrED operates in a multi-domain setting, with the additional focus on transferring knowledge across these multiple evolving domains. Online Transfer Learning (Zhao et al., 2014) describes a setting where target domain data arrives sequentially and some models had been learnt from source domains. While there is some overlap with TrED, in Online Transfer Learning, the learning happens in an online fashion with streaming data from the target domain, and only pre-trained models from source domains are used. TrED, on the other hand, assumes that source domain data itself is available and can be used alongside historical target domain data, making it unnecessary to learn strictly in an online manner.

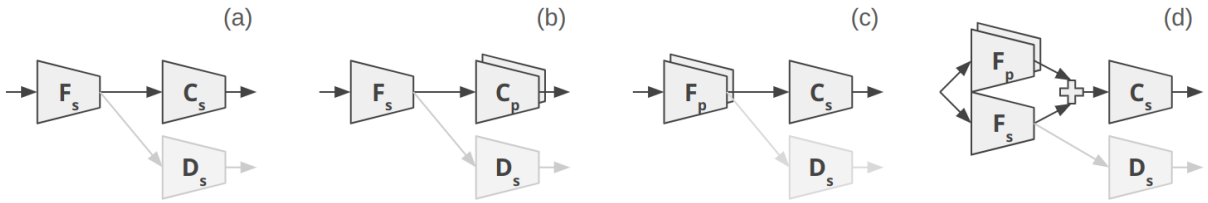

Figure 2: Diagrams of four different types of modular deep learning architectures used in transfer learning methods. The F components are feature extractors, C are task classifiers, D are optional domain discriminators. The subscripts s and p indicates whether the component is shared or private across different domains. Panel a) depicts a solution composed of a feature extractor and a classifier that are shared across all domains. Panel b) depicts a solution composed of a shared feature extractor and domain-specific classifiers. Panel c) depicts a solution composed of domain-specific feature extractors and a shared classifier. Panel d) depicts a solution composed of a shared feature extractor and domain-specific feature extractors, whose outputs are concatenated before being passed to a shared classifier. All panels can optionally include a discriminator.

3 Literature Review on TL: the TrED Perspective

In this section, we provide a comprehensive comparison of various Transfer Learning (TL) methods, each developed under different assumptions regarding data availability.

The beginning of this section mirrors the data availability stages discussed previously, with each subsection dedicated to a specific TL sub-problem: Domain Generalisation (Section 3.1), Unsupervised Domain Adaptation (Section 3.2), Supervised Domain Adaptation (Section 3.3), and Multi-Domain Learning (Section 3.4).

Each subsection begins with the framing of the corresponding TL sub-problem in TrED. We then review some methods proposed to tackle that sub-problem and their relevance to TrED.

The methods are discussed according to the type of approach:

- **Data Transformation:** Techniques that manipulate the data prior to model input, such as instance weighting, feature augmentation, feature mapping, and pseudo-labelling.
- **Model Adaptation:** Architectures and structural strategies designed to improve the transfer of knowledge. For example, many of the methods discussed incorporate modular deep learning architectures consisting of linked components. We illustrate four common designs in Figure 2.
- **Training Objective:** Methods that leverage specific learning algorithms or loss functions during training, such as adversarial losses, clustering, contrastive learning, and statistical distances.

Finally, we describe some methods that can be used in more than one stage of TrED (Section 3.5) and highlight foundation models as a new type of solution that is very relevant in the context of TL and TrED (Section 3.6).

A summary of the methods analysed, organized according to these two dimensions (TL sub-problem and type of approach) is given in Table 2.

3.1 Domain Generalisation

Domain Generalisation (DG) describes the problem of learning from one or multiple source domains, such that the solution has good performance on any unseen target domain.

Table 2: Table summarising the reviewed TL methods. Each row represents a type of technique that methods can use, with columns indicating the TL sub-problem each method addresses. The techniques are grouped into three major categories: data transformation, which involves techniques that manipulate the data prior to model input; model adaptation, which includes architectures and structural strategies designed to improve the transfer of knowledge; training objective, for techniques that leverage a specific learning algorithm or loss during training.

		DG	UDA	SDA	MDL
Data Transform.	Instance Weighting		(103; 119; 149)	(16; 103; 137)	
	Feature Augmentation	(147)	(7; 123)	(7; 18; 26; 47; 54; 99)	(136)
	Feature Mapping	(67)	(74; 102)	(74; 26; 85)	
	Pseudo-labelling	(59)	(7; 13; 57; 70; 74; 77; 87; 97; 104; 123; 124; 121; 140)	(7; 54; 74; 99)	
Model Adaptation	Modular DL (a)	(23; 53; 58)	(27; 32; 35; 65; 70; 87; 88; 89; 90; 92; 105; 104; 108; 124; 139)	(54; 91)	(60)
	Modular DL (b)		(57; 62; 75; 97)	(5; 62; 99)	(68)
	Modular DL (c)	(146)	(114; 146)		
	Modular DL (d)		(11)		(14; 37; 131; 132)
	Ensemble	(117; 146)	(75; 86; 146; 149)	(133)	(24)
	Other	(51; 55; 117)	(56; 63; 88; 123; 135)	(56; 133; 138)	(30; 66; 71; 76)
Training Objective	Adversarial Loss	(53; 147)	(11; 27; 39; 57; 65; 70; 88; 89; 90; 92; 104; 108; 114; 139; 140)	(54; 91)	(14; 37; 131)
	Clustering	(23)	(93; 97; 104; 135)	(54; 99)	
	Contrastive Loss		(43; 104; 124; 140)	(99)	(37)
	Meta Learning	(23; 28; 52; 58; 146)	(50; 61; 77; 146)	(50)	
	Reconstruction Loss	(53)	(11; 32; 35; 39)		(60)
	Statistical Distance	(23; 53)	(11; 35; 62; 63; 64; 70; 75; 93; 101; 113)	(62)	(30; 131)
	Other		(39; 86; 105)		(60)

3.1.1 DG as a stage of TrED

DG can be seen as the first stage of the TrED problem. In this setting, there are some labelled datasets $D^L_{S_1} \dots D^L_{S_n}$ from source domains and the goal is to develop a solution that performs well on a target domain $\mathcal{D}_T$ from which no data is available.

This is a particularly challenging problem due to the lack of knowledge about the target domain. Even if all domains share the same feature space $\mathcal{X}$, the marginal probability distribution of the target domain $P(X_T)$ can differ from those of the source domains, a problem closely related to covariate shift (Shimodaira, 2000). Similarly, the target conditional probability $P_T(Y|X)$ can differ from the source conditional probability $P_X(Y|X)$, a problem related to concept drift (Widmer & Kubat, 1996). Lastly, the label space of the target domain $\mathcal{Y}_T$ can differ from those of the source domains, leading to a scenario known as open set recognition (Scheirer et al., 2012). These potential differences between source and target domains cannot be directly diagnosed due to the lack of target domain data.

Since DG does not require prior access to target domain data, a solution for the DG problem can be directly applied to any stage of TrED problem. However, it is expected that as data and labels from the target domain are gathered over time, other types of solutions that leverage this information will surpass DG methods in predictive performance. Nonetheless, by studying the methods explored in this section, we can design TrED solutions that have good predictive performance during the early stages of new domains.

3.1.2 Current literature on DG

Data Transformation. Some methods propose to transform the source domain data in such a way to promote the generalisation capabilities of a model trained with it.

The Domain Invariant Component Analysis (DICA) (Muandet et al., 2013) is a feature mapping technique that first maps input data into a higher-dimensional space using a kernel function, and then learns a transformation matrix to map it to a lower-dimensional space. The first step is done to capture non-linear relationships in the data, while the second step should retain the necessary information to predict the output accurately and make the features invariant to the domain differences. The model is trained on the transformed data, which is expected to generalise better to new domains.

The Deep Domain Adversarial Image Generation (DDAIG) (Zhou et al., 2020) is a feature augmentation technique in which a Domain Transformation Network (DoTNet) is used to create perturbations on the original instances. These perturbed examples are used to augment the training of the classifier, while a domain discriminator controls how much the perturbations can deviate from the real data. This approach helps the model generalise better to the new target domain by exposing it to a wider variety of data during training, thereby improving its robustness to unseen scenarios.

Another method (Lin et al., 2024) incorporates unlabelled source domain data into the training via an iterative pseudo-labelling framework: a model is trained on labelled data, pseudo-labels are generated for unlabelled samples, instances with low label noise are selected based on agreement between two surrogate networks, and these are progressively added back into the training set.

Model Adaptation. Some methods rely on modular deep learning architectures to promote regularisation. One possibility (Dou et al., 2019; Li et al., 2019) is to separate the network into a feature extractor and a classifier that are shared for all domains (Figure 2(a), but without the discriminator D). The Maximum Mean Discrepancy-based Adversarial AutoEncoder (MMD-AAE) (Li et al., 2018b) uses the same design, but also include an adversarial network (Figure 2(a), with discriminator D).

Other methods propose different adaptations to deep learning architectures to improve the generalisation capabilities of the models trained. The Stochastic Feature Augmentation method (Li et al., 2021b) suggests to add normally distributed noise to the activations of the networks' layers, with the option of also considering the label when generating this noise. Another method (Li et al., 2017) proposes that the network layers should have one extra dimension with the size of the number of domains (each slice being domain specific) plus one that is shared across all domains. The authors suggest singular value decomposition as a way to decompose the potentially large weight tensors. The Leave-One-Domain-Out (LODO) method (Vu et al.,

2022) employs an architecture where we first learn an adaptor-head for each source domain, and then use attention to combine the outputs of all adaptors, effectively creating an ensemble of models that leverages the strengths of each domain-specific adaptor.

Training Objective. A possible strategy for domain generalisation is to use a loss function that incentivises generalisation capabilities. DDAIG (Zhou et al., 2020) uses a domain discriminator as the source of an adversarial loss, providing feedback about how to perturb the data in order to improve generalisation. MMD-AAE (Li et al., 2018b) also leverages an adversarial network, but as a regularisation technique for learning the encodings of an autoencoder. Beside the adversarial loss from the discriminator and the reconstruction loss from the autoencoder, this method also includes a maximum mean discrepancy term in the loss function, to minimize the difference between encodings across different domains.

Other approaches leverage meta-learning to adapt the loss function itself, enabling the model to optimise for better performance in unseen domains. Meta-learning can be applied either as an augmentation to the standard cross-entropy loss (Dou et al., 2019; Li et al., 2018a; 2019), or as its replacement (Gao et al., 2022).

The Model-Agnostic Learning of Semantic Features (Dou et al., 2019) computes two terms when computing its meta-loss. The first measures the KL divergence between the confusion matrix of predicted classes from different domains, in order to promote the alignment of class relationships across domains. The second promotes the clustering of the representations according to class, regardless of the domain.

3.1.3 Adapting DG methods towards TrED

Some of these methods (Li et al., 2021b; Zhou et al., 2020) are optimised for improving model robustness, meaning that the model's performance won't degrade significantly when encountering data that deviates slightly from the training distribution. Even though this is a desirable property for a TrED solution, the new target domain can differ much more than what those perturbations account for. Therefore, a solution solely based on perturbing the data or the model during training is probably not general enough.

The methods that use meta-learning (Dou et al., 2019; Gao et al., 2022; Li et al., 2018a; 2019) also aim to improve robustness by using hold-out domains during training to directly optimise for potentially out-of-distribution but realistic data. Since these methods only adapt the loss function, they are flexible to different model architectures and can easily be combined with other techniques. However, while these methods provide a solid starting point for generalising to new domains, they lack built-in mechanisms to incorporate newly collected data from the target domain as it becomes available.

Methods such as LODO (Vu et al., 2022) are promising for adapting to the TrED setting. New adapters can be added when more domains are introduced, and the fusion layer can be continuously fine-tuned. However, constantly retraining a large deep learning model can introduce instability if not properly regularized, and it can also be computationally expensive. To maintain efficiency and stability, a robust TrED solution should avoid increasing the number of tunable parameters in proportion to the number of domains and minimize the need for frequent retraining.

3.2 Unsupervised Domain Adaptation

Unsupervised Domain Adaptation (UDA) describes the problem of sharing knowledge from one or multiple source domains to a specific target domain, from which a pool of unlabelled data is available.

3.2.1 UDA as a stage of TrED

UDA can be seen as the second stage of the TrED problem. In this setting, there are some labelled datasets $D^L_{S_1}, \ldots, D^L_{S_n}$ from source domains and an unlabelled dataset D^U_T from the target domain, and the goal is to develop a solution that performs well on that target domain $\mathcal{D}_T$.

Unlike the DG setting, we can use D^U_T to estimate $P_T(X)$ during the UDA stage of TrED. By comparing each $P_{S_i}(X)$ estimated from $D^L_{S_i}$ against $P_T(X)$, it is possible to measure how much the data distribution changes from each source to the target domain. This knowledge can then be used to manipulate the source domain

data to better approximate the target domain's feature distribution. For example, this can be achieved by re-weighting the source domain samples or by transforming the source feature values through techniques such as feature normalisation. By doing so, we can reduce distributional differences and improve performance on the target domain. Additionally, the divergence of $P(X_{S_i})$ and $P(X_T)$ can indicate the similarity of the domains, allowing solutions to weight the contributions of each source domains differently.

However, similarly to the previous stage of the TrED problem, the labels from the target domain are still unavailable. This limitation means that we cannot directly estimate the target domain's conditional probability $P_T(Y|X)$. Additionally, even if a model successfully approximates conditional probability on the source domains, $P_T(Y|X)$ may be different due to potential concept drift between the source and target domains. Without target domain labels, there is no direct way to measure the model's performance in the target domain, further complicating the adaptation process.

Given that a typical UDA solution requires access to target domain data, it cannot be deployed during the first stage of the TrED problem. However, since this type of solution does not require target domain labels, it can still be used quite early, without being affected by potential labelling delays or costs. Furthermore, it is expected that solutions designed with access to $P_T(X)$ will have better predictive performance than its DG counterparts. Additionally, some methods were proposed that bridge the gap between UDA and SDA, potentially making the transition from the UDA stage to future stages of the TrED problem easier.

3.2.2 Current literature on UDA

Data Transformation. Several works propose to solve the unsupervised part of UDA through the use of pseudo-labels. The Co-Training for Domain Adaptation method (Chen et al., 2011) maintains two data pools: a labelled pool, initially composed of source domain data, and an unlabelled pool, initially composed of target domain data. Using a co-training approach, the authors simultaneously train two models on the labelled pool, which then iteratively annotate and integrate data from the unlabelled pool into the training process. Another method was proposed (Sener et al., 2016) to iteratively infer labels for the unlabelled target domain data based on each instance's nearest neighbours on the source domain. Similar to this strategy, the Cross-domain Contrastive Learning method (Wang et al., 2022c) produces pseudo-labels based on the k-means clustering method, using class prototypes from the source domain as initialisation for the clusters. Another method was proposed (Saito et al., 2017) to train a common feature extractor with three different classifiers on top using the labelled source domain data. Two of them produce the pseudo-labels for the target data, on which the third classifier is trained. The pseudo-labels are only included in the training if the predicted class of the two classifiers match and at least one of them predicts the class with sufficiently high probability. In the Multi-source Domain Adaptation with Weak Supervision method (Li et al., 2021c), the authors train multiple source models and then use pseudo-labels on the target to tune the weights of the weighted average that combines their scores. The Meta Self-Learning for Multi-source Domain Adaptation method (Qiu et al., 2021) starts by pre-training a model on source domain data, then using it to provide pseudo-labels on the target. Then the model is fine tuned using meta-learning using source and target data combined, updating the pseudo-labels each round. Another method was proposed (Wang, 2023) to split the source data into two sub-sets, one to train a set of candidate models and the other to train one single imputation model. The imputation model creates pseudo-labels for the target domain, and the best among the candidate models is selected by evaluating on the pseudo-labelled target data. Other methods (Nguyen et al., 2025; Sun et al., 2025; Yuan et al., 2023) use pseudo-labels to compute and align the class conditional distributions of the embeddings from the feature extractor.

There are other data transformation methods that are not based on using pseudo-labels. The Balanced Distribution Adaptation method Wang et al. (2017) is an instance weighting technique that aligns the marginal and conditional distributions of the source and target domains, which is especially useful to solve class imbalance problems. The Fully Test-time Adaptation for Tabular data method (Zhou et al., 2025) also uses instance weighting to filter low-quality predictions from the test-time adaptation. The instance weight depends on the uncertainty of the prediction and whether it aligns with the predictions of neighbouring instances. The Correlation Alignment (CORAL) method (Sun et al., 2016) is a feature transformation technique where the source data is first "whitened" by removing the features correlations of the source

domain, and then "re-coloured" by adding the correlation of the target domain to the source features. After this transformation, one can train a model using labelled source data.

The Continual Test-Time Adaptation method (Wang et al., 2022b) combines pseudo-labelling with data augmentation. It keeps two copies of the adapted model: the teacher creates the pseudo-labels for augmented target data, which are then used to fine-tune the parameters of the student. The parameters of teacher are the result of the exponential moving average of the parameters of the student over the rounds of fine-tuning.

Model Adaptation Several authors (Saito et al., 2018a;c; Tang & Jia, 2020; Wang et al., 2022c) split the networks into two components (Figure 2(a), but without the discriminator D): the first layers of the network are the feature extractor, which is responsible for computing a useful and general representation for the examples; and the last layers of the network are the task classifier, which computes the label prediction given the features from the feature extractor. Some authors train more than one classifier (Saito et al., 2018b; 2017), but they are still used both for source and target data. Besides these two types of components, other methods (Ghifary et al., 2016; He et al., 2023a) add a new component to learn how to reconstruct target examples, given the features from the feature extractor (similar to an auto-encoder) Another method (Sun et al., 2019) suggests to use several self-supervised heads on top of the feature extractor.

Another common design in terms of modular deep learning is to use a feature extractor, a task classifier and a domain discriminator (Figure 2(a), with discriminator D). The domain discriminator provides a signal to the features extractor, incentivising it to generate more domain independent features. The Domain-Adversarial Neural Networks method (Ganin et al., 2016) was the first to propose this idea for the domain adaptation setting. Since then other methods were proposed that also follow this strategy (e.g. (Sun et al., 2025)). The Conditional Domain Adversarial Networks method (Long et al., 2018) complements this idea by conditioning the domain discriminator, passing it the concatenation of the feature extractor's and task classifier's outputs, with the goal of improving discriminability. The Universal Adaptation Network method (You et al., 2019) applies a similar technique to solve the universal domain adaptation problem, where the test time classes may or may not have been seen during training. In their work, the authors train two discriminators: one to provide signal to the feature extractor and the other to be used at test time to detect unknown classes. Another method (Saito et al., 2019b) uses two discriminators as well, but at different depths along the feature extractor, thus aligning representations with distinct levels of abstraction. The Conditional Adversarial Support Alignment method (Nguyen et al., 2025) uses the domain discriminator to align the embeddings from the feature extractor, conditioned on the predicted class from the pseudo-labels.

All methods mentioned in the two previous paragraph have a feature extractor and a task classifier that are shared for all domains. Other methods have a shared feature extractor with a domain specific classifier (Figure 2(b)). One method that follows this setup (Sener et al., 2016) trains a source domain classifier using source domain labels and a target domain classifier using pseudo-labels from a k-NN method. In Deep Adaptation Networks (Long et al., 2015), the shared feature extractor is composed of some pre-trained layers that are frozen during training and some fine-tuned layers. The different domain-specific classifiers are aligned across domains using a maximum mean discrepancy loss. In the Moment Matching for Multi-Source Domain Adaptation method (Peng et al., 2019), the authors train a single feature extractor and one classifier per source domain. To make predictions on the target domain, the scores of all classifiers are combined using an ensemble strategy. The Multi-source Domain Adaptation with Weak Supervision method (Li et al., 2021c) also trains a shared feature extractor and private classifier per source domain, plus a domain discriminator to incentivise the feature extractor to be more general.

Another option is to have domain specific features extractors, with a shared classifier on top (Figure 2(c)). The Adversarial Discriminative Domain Adaptation method (Tzeng et al., 2017) starts by training source-specific feature extractor and a shared classifier, using labelled source data. Then, the source feature extractor is fixed, while training a target-specific feature extractor adversarially with a domain discriminator. At test time, the target feature extractor and the shared classifier are used to make predictions.

The Domain Separation Networks method (Bousmalis et al., 2016) trains one feature extractor that is shared for all domains and one private feature extractor per domain, with a shared classifier on top (Figure 2(d),

with discriminator D). For each domain, the outputs of the its private feature extractor and of the shared feature extractor are concatenated, before being passed to the shared classifier.

For the case of having multiple source domains, some authors propose to build an ensemble type of solution, where multiple models from the source domains are trained and then combined to make predictions on the target. The Moment Matching for Multi-Source Domain Adaptation method (Peng et al., 2019) trains one task classifier per source domain and combines their scores on the target domain either with a simple average, or a weighted average following some heuristic. The OVANet method (Saito & Saenko, 2021) trains multiple source models, but instead of combining the scores at inference, the most appropriate source model is selected and used to make a prediction. The Fully Test-time Adaptation for Tabular data method (Zhou et al., 2025) adapts multiple models to the target domain with different learning rates and computes the weighted average of the predictions according to their loss values on target batched data.

Training Objective Several methods include unsupervised losses due to the lack of target domain labels, such as clustering (Saito et al., 2020; Sun et al., 2025; Sener et al., 2016; Yang et al., 2021), contrastive (Huang et al., 2022; Sun et al., 2025; Wang et al., 2022c; Yuan et al., 2023), and reconstruction (Bousmalis et al., 2016; Ghifary et al., 2016; He et al., 2023a; Hoffman et al., 2018) or others (Sun et al., 2019).

Another common option is to include a divergence metric in the loss that measures the distance between the features from source and target domain. These metrics can be the MMD (Long et al., 2015; 2016; 2017; Tzeng et al., 2014), the difference of covariances (Sun & Saenko, 2016), the KL divergence (Nguyen et al., 2025), the Sinkhorn divergence (He et al., 2023a), or other moment distance functions (Peng et al., 2019).

Several authors use an adversarial loss to align the learnt representations from multiple domains. Usually this is achieved by training a domain discriminator to distinguish the domains given the representations from the feature extractor (Bousmalis et al., 2016; Ganin et al., 2016; Long et al., 2018; Nguyen et al., 2025; Saito et al., 2019b; Sun et al., 2025; Tzeng et al., 2017; You et al., 2019; Yuan et al., 2023). Instead of having a discriminator, another option is to train the feature extractor and the classifier cooperatively on source domain data, but adversarially from each other on the target domain data (Saito et al., 2018c; Tang & Jia, 2020). Two other methods (Saito et al., 2018b;a) also train the feature extractor adversarially with the task classifiers, but using two task classifiers which are trained to maximise their discrepancy, while the generator wants to minimise it. Their difference is that one of them trains two parallel classifiers from scratch (Saito et al., 2018b), while the other uses dropout to create two versions of a single classifier (Saito et al., 2018a).

The CyCADA method (Hoffman et al., 2018) adapts the CycleGAN (Zhu et al., 2017) architecture to the domain adaptation setting by learning mappings from source to target domain and vice-versa. CyCADA combines multiple of the aforementioned types of losses: a supervised loss using source labelled data, a reconstruction loss between the input and the output of the two mappings in sequence, and an adversarial loss between the mappings and a discriminator. The authors also include a semantic consistency loss, to incentivise the labels from before and after the mapping to match, mitigating the problem of label flipping.

3.2.3 Adapting UDA methods towards TrED

The use of pseudo-labels is a common technique in UDA to address the lack of labels. Effective pseudo-labelling strategies can mitigate the problem of label delay that affects some of the TrED settings. Furthermore, methods that utilise pseudo-labels to adapt solutions to a new domain can seamlessly incorporate real labels once they become available, without requiring any modifications. However, the success of this approach heavily depends on the quality and fidelity of the pseudo-labels: if they do not approximate the true labels well, the adaptation step can significantly degrade predictive performance. Additionally, generating pseudo-labels requires the presence of target domain data, which excludes these techniques during the first stage of the TrED continuum (the typical setting of DG).

Similarly, methods based on feature transformation techniques or that use unsupervised losses for model training are not applicable in the DG setting, as they also require target domain data. However, some of these techniques can still be employed as regularisation during the development of the initial solution to enhance its generalisation capabilities. For example, incorporating the feedback from a domain discriminator

when training a feature extractor can incentivise it to learn domain agnostic features, which are useful for any stage of the TrED problem.

Lastly, certain modular DL architectures described in this section are more adaptable to settings without target domain data, particularly those where all domains share the same feature extractor and classifier (Figure 2(a)). This is because domain-specific components typically require domain-specific data for training.

3.3 Supervised Domain Adaptation

Supervised Domain Adaptation (SDA) describes the problem of sharing knowledge from one or multiple source domains to a specific target domain, from which a pool of labelled data is available.

3.3.1 SDA as a stage of TrED

Supervised Domain Adaptation can be seen as the third stage of the TrED problem, in which some labelled target domain data has been collected. More formally, there are some labelled datasets $D^L_{S_1}, \ldots, D^L_{S_n}$ from source domains and a labelled dataset D^L_T from the target domain, and the goal is to develop a solution that performs well on that target domain $\mathcal{D}_T$.

Similar to the UDA setting, we can use D^L_T to estimate $P_T(X)$ during the SDA stage of the TrED problem. However, now that we have access to labelled target domain data, we can also estimate the target domain's conditional probability $P_T(Y|X)$ and sample the label space $\mathcal{Y}_T$ and compare with the source domains ones.

By comparing each source domain model with a target domain model, either directly in the case of interpretable models or indirectly through cross-domain evaluations (testing source domain models on D^L_T and/or target domain model on $D^L_{S_i}$), we can estimate the similarity between the source and target domains. A higher similarity suggests that knowledge from the source domains will transfer more effectively, allowing some solutions to weight the contributions of each source domain differently, potentially leading to better overall predictive performance.

Also, during this stage we can use (at least part of) D^L_T as a validation set for our TrED solution. This facilitates the design or selection of solutions that not only perform well in theory but also demonstrate good predictive performance on actual target domain data.

However, having access to a reasonably large labelled dataset from the target domain is a requirement that can take a significant amount of calendar time to meet. This means that, between the moment when a new target domain is added and the point when we can use a SDA method, there is the need for a more unrestricted TrED solution. On the other hand, once a SDA type of solution is deployed in a TrED setting, it can potentially be used indefinitely since the available data from any domain usually increases over time.

3.3.2 Current literature on SDA

Data Transformation Some SDA methods rely on instance weighting techniques for boosting-based learning algorithms. The TrAdaBoost method (Dai et al., 2007) uses boosting iterations where the weight of source instances decreases when the error for those is large, because those instances would probably have a negative impact on performance on the target. The TaskTrAdaBoost method (Yao & Doretto, 2010) generalises this idea for the setting where there are more than one source domain. In each boosting iteration, it selects the weak learner from the domain that has the smallest error on the target.

Other methods use feature augmentation techniques. One solution (Daumé III, 2009) involves repeating the space of features three times: one copy is populated by source domain data, other by target domain data, and the third is populated by both. This way, the model can learn domain specific patterns from the domain specific entries, or general patterns from the shared entries. The EasyAdapt method (Kumar et al., 2010) extends the idea to also make use of a possible set of unlabelled target data. The Heterogeneous Feature Augmentation (Duan et al., 2012) uses the same tripartite representation to tackle the problem of heterogeneous domains. If the source and target domains have different features, we learn mappings to project them to a common sub-space on the shared entries. In the context of feature mapping approaches, there is

another method (Saenko et al., 2010) where the authors propose to learn a simple linear transformation to map from one domain to the other.

Similar to UDA, some SDA methods also use pseudo-labels when some target domain labels are missing (sometimes called semi-supervised domain adaptation). The Contrastive Learning for Domain Adaptation method (Singh, 2021) uses the prediction of the model during its training as pseudo-labels to cluster target domain data and align those clusters with source domain data clusters. The Cross-Domain Adaptive Clustering method (Li et al., 2021a) has a similar setup, but includes a feature augmentation technique to increase the input variety of the model.

Model Adaptation Some methods (Li et al., 2021a; Saito et al., 2019a) follow the modular DL strategy composed of a shared feature extractor and a shared classifier (Figure 2(a), but without the discriminator D). Other methods (Bao et al., 2019; Singh, 2021) have a shared feature extractor, but a domain specific classifier instead (Figure 2(b), but without the discriminator D). The H-score (Bao et al., 2019) was proposed to help select the optimal point where the network should be split between feature extractor and classifier.

A thorough experimentation using modular deep learning strategies in the context of SDA was conducted (Yosinski et al., 2014), with different choices of frozen, fine-tuned or re-initialised parameters when mixing components from different domains.

The OMTL-PTP method (Yang et al., 2024) method trains one base learner per source domain and stores their knowledge in an external memory with attention. An online target domain learner is then continuously updated by integrating the knowledge from source learners and its previous state, with weight based on the performance on the current target domain batch.

Training Objective Some modular deep learning methods include an adversarial loss, where the feature extractor and the classifier are optimised to min-max some metric. In the Minimax Entropy method (Saito et al., 2019a), besides the cross-entropy loss measured on labelled source and target data, the authors include an entropy-based adversarial loss calculated on unlabelled target data. The feature extractor is trained to minimise it, while the classifier is trained to maximise it. In the Cross-Domain Adaptive Clustering method (Li et al., 2021a), there is an adversarial adaptive clustering metric that the feature extractor is minimising while the classifier is maximising.

Some methods (Li et al., 2021a; Singh, 2021) include a clustering loss to align instances that have the same class but belong to different domains. The Contrastive Learning for Domain Adaptation method (Singh, 2021) also includes a contrastive learning loss to help stabilise these clusters on the target domain.

3.3.3 Adapting SDA methods towards TrED

Even though the TrAdaBoost (Dai et al., 2007) and TaskTrAdaBoost (Yao & Doretto, 2010) methods are not directly applicable to DG problem, the core idea of reducing the weight of instances that likely have a negative impact on target domain performance can be adapted to the TrED setting. Initially, when target domain data is scarce, weights can focus on generalising to out-of-distribution data. As more target domain data becomes available, the weighting policy can shift to more targeted adaptation.

The tripartite representation used in some SDA methods (Daumé III, 2009; Duan et al., 2012; Kumar et al., 2010) enable models to learn both domain-specific and general patterns, facilitating smoother transitions as new target data arrives. However, it has the disadvantage of scaling the input size linearly with the number of domains, which likely limits its applicability to settings with only a few distinct domains.

Some SDA methods (Li et al., 2021a; Singh, 2021) use pseudo-labels to address the issue of missing labels on the target domain. This is an indication that this strategy is not only useful for UDA but can also bring value for later stages of the TrED problem. Also, in settings where label delay is present, this strategy can be used to provide an estimate of the label for the most recent data, while the true label is not available.

The CLDA method (Singh, 2021) demonstrates how unsupervised techniques like clustering can be adapted to utilise target domain labels, illustrating the flexibility of such approaches for various stages of TrED.

3.4 Multi-Domain Learning

Multi-Domain Learning (MDL) describes the problem of sharing knowledge from multiple domains to develop a solution that has good performance in all of those domains.

3.4.1 MDL as a stage of TrED

MDL represents a mature stage within the continuum of the TrED problem, in which a sufficiently large volume of labelled data is available from all domains. More formally, there are some labelled datasets $D_1^L, \ldots, D_n^L$ from a set of domains and the goal is to leverage the shared knowledge of all domains to develop a solution that performs well across these domains.

This is a different problem from those discussed in previous sections, as the optimisation is not focused on a specific target domain $\mathcal{D}_T$, but rather the solution should perform well in all "source" domains $\mathcal{D}_i$. Nevertheless, MDL can be viewed as natural progression of the SDA, where the size of D_T^L has grown sufficiently large such that it is equally valid to transfer knowledge from $\mathcal{D}_{S_i}$ to $\mathcal{D}_T$ or vice-versa. Therefore, solving the problem of MDL can be framed as solving all combinations of SDA, where each domain $\mathcal{D}_i$ alternates as the target domain, and the remaining domains $\mathcal{D}_1, \ldots, \mathcal{D}_n \setminus \mathcal{D}_i$ act as source domains.

The central challenge in the MDL setting is to effectively leverage the shared knowledge from all domains to achieve a better performance compared to solutions trained on individual domains. At this stage, data availability is no longer a concern since a large volume of labelled data from each domain is already accessible. This means it is possible to train domain-specific models using only $D_{S_i}^L$, but this approach can be sub-optimal for several reasons. Firstly, maintaining a separate model for each domain increases the complexity of deployment and maintenance linearly with the number of domains. Secondly, a model trained on data from multiple domains may achieve better predictive performance by incorporating diverse knowledge, thereby surpassing domain-specific models. Moreover, MDL solutions can enhance robustness to data drifts, as they are trained on a variety of data distributions sampled from the different domains.

3.4.2 Current literature on MDL

Data Transformation A feature augmentation method was proposed (Yang & Hospedales, 2014) where a "semantic descriptor" is passed to the model as a separate channel (besides the input features) containing meta-information about the domain, which provides extra context to the model. By including meta-information such as domain-specific characteristics, origin, or contextual background, the model can better understand and adapt to the variations across domains. This allows the model to learn more nuanced and robust representations, ultimately improving its generalisation capabilities across multiple domains.

Model Adaptation Some methods proposed modular deep learning strategies to optimise the training of multi-domain learning models. One option (Lin & Xue, 2025) is to train a feature extractor and a classifier using data from all domains (Figure 2(a), but without the discriminator D). Other methods (Nam & Han, 2016) keep the shared feature extractor but train a domain-specific classifier (Figure 2(b), but without the discriminator D). Another method (Xi et al., 2024) trains a private feature extractor per domain and a shared feature extractor that learns from all domains, concatenates their outputs and computes the prediction with a shared classifier (Figure 2(d), but without the discriminator D). Other methods (Chen & Cardie, 2018; He et al., 2023b; Wu & Guo, 2020) follow the same strategy, but include a domain discriminator that trains adversarially with the shared feature extractor.

A related approach (Garg et al., 2022) addresses what the authors refer to as "multi-domain incremental learning". In this method, a model is sequentially trained on different domains while maintaining performance on previously learned domains. The architecture is similar to the shared-private feature extractor, but instead of having separate parallel networks, it integrates shared and domain-specific layers within a single feature extractor. As new domains are introduced, additional domain-specific layers are added within the existing feature extractor. Instead of using a shared classifier, the model employs domain-specific classifiers, allowing it to handle each domain's unique characteristics effectively.

Another method (Omi et al., 2022) proposes to alternate between shared and private layers within the model. It starts by training a domain-independent backbone model, and then add a domain-specific layer between each layer of this backbone. Both types of layers are then trained and fine-tuned together, with the model's weights being updated only after processing a batch from each domain.

The previous two methods proposed adding domain-specific layers, either in sequence or in parallel. One method (Mancini et al., 2020) was proposed to instead just learn a binary mask on top of the shared layers. This binary masked is used to create affine transformations of the network parameters to effectively adapt a pre-trained model to multiple domains. This approach strikes a balance between expressivity and efficiency, allowing for significant domain-specific customisation while maintaining a low parameter overhead.

The AdapterFusion method (Pfeiffer et al., 2021) proposes a learning algorithm that alternates between knowledge extraction and knowledge composition stages, to mitigate issues like catastrophic forgetting and dataset balancing. Initially, task-specific adaptors are trained independently, encapsulating task-specific information while keeping the underlying pre-trained model fixed. Then, a fusion mechanism combines these adaptors' representations, leveraging knowledge from multiple tasks in a non-destructive manner.

A different method was proposed (Dredze et al., 2010) following an ensemble-based approach. It starts by training one confidence-weighted linear classifiers per domain and then combines the parameters of the linear models, taking into account their confidence estimate.

Training Objective Several methods incorporate adversarial loss via a domain discriminator, encouraging the shared feature extractor to learn domain-agnostic representations (Chen & Cardie, 2018; He et al., 2023b; Wu & Guo, 2020).

Other approaches introduce statistical distances into the training loss with varying objectives. When employing both shared and private feature extractors, a distance metric can be used to ensure these extractors learn distinct representations (Wu & Guo, 2020). Divergence metrics can also be utilised to align the representations of current and past versions of the model, aiding in the retention of knowledge from previous domains in an incremental learning context (Garg et al., 2022).

The Multi-Domain Contrastive Learning method (He et al., 2023b) incorporates two contrastive losses: a supervised inter-domain loss that brings instances with the same label closer together regardless of domain, and an unsupervised intra-domain loss that clusters data within each domain.

Another method (Lin & Xue, 2025) combines a multi-task classification loss (that leverages different data types, in this case text and tabular data) with a reconstruction loss (to ensure that the embeddings from the feature extractor retain the key information from each domain).

3.4.3 Adapting MDL methods towards TrED

The semantic descriptor method (Yang & Hospedales, 2014) demonstrates how additional contextual information about domains can be used to improve model performance. For TrED, incorporating evolving semantic descriptors that capture the progression of domains over time could be used to continuously adapt the model, even without target domain data. However, having readily available semantic descriptors that accurately represent domain differences can be challenging for unseen or unknown domains.

Similar to the other stages covered in previous sections, some MDL methods are based on modular deep learning strategies, composed of feature extractors and classifiers. These methods allow the model to retain domain-specific knowledge while also generalising across domains, which aligns well with the needs of TrED. However, this type of models can have increased complexity and higher computational costs when handling a large number of domains. Furthermore, balancing the shared and private components effectively to avoid negative transfer remains a challenge.

The ensemble-based strategy (Dredze et al., 2010) of combining domain-specific classifiers weighted by their confidence is particularly promising for handling the incremental nature of data in TrED. As new data from different domains become available, an ensemble approach can dynamically integrate new models, while

Table 3: Table summarising the reviewed TL methods that address more than one stage of the TrED problem. Similar to Table 2, each row represents a type of technique that methods can use, with columns indicating the TL sub-problem each method addresses.

		DG	UDA	SDA	MDL
Data Transform.	Instance Weighting		(103)	(103)	
	Feature Augmentation		(7)	(7)	
	Feature Mapping		(74)	(74)	
	Pseudo-labelling		(7; 74)	(7; 74)	
Model Adaptation	Modular DL (a)				
	Modular DL (b)		(62)	(62)	
	Modular DL (c)	(146)	(146)		
	Modular DL (d)				
	Ensemble	(146)	(146)		
	Other		(56)	(56)	
Training Objective	Adversarial Loss				
	Clustering				
	Contrastive Loss				
	Meta Learning	(146)	(50; 146)	(50)	
	Reconstruction Loss				
	Statistical Distance		(62)	(62)	
	Other				

leveraging the strength of existing ones. However, ensemble methods can suffer from high computational and memory costs, especially as the number of domains and models grows.

Lastly, the use of adversarial and/or contrastive losses in MDL methods (Chen & Cardie, 2018; He et al., 2023b) can promote domain generalisation and lead to learning more robust representations. However, adversarial training can be unstable and challenging to tune, while contrastive losses depend heavily on the quality of the negative samples, which may not always be representative or easy to obtain in a TrED setting.

In conclusion, MDL methods are designed to handle multiple domains simultaneously, so their principles and techniques are highly applicable to TrED. By adapting these methods, we can develop models that are capable of continuous adaptation, ensuring consistent performance across a wide range of data availability scenarios. However, to fully realise their potential in TrED, it is crucial to address the challenges related model complexity, computational costs, and the stability of training.

3.5 Methods that span multiple stages

Only a small part of the reviewed methods have been proposed to directly address more than one of the stages discussed in the previous sections. We list these methods in Table 3.

The most common multi-stage type of method addresses both unsupervised and supervised domain adaptation. One of those methods (Li & Hospedales, 2020) alternates between using meta-learning to optimise the initial condition of the model, and doing unsupervised or supervised adaptation to the target, depending on the availability of labels. The Assign-and-Transform-Iteratively method (Panareda Busto & Gall, 2017) starts by assigning pseudo-labels to the unlabelled part of target domain data. Then, learn a mapping on the source to approximate classes from source to classes from target. Lastly, train a classifier on the mapped source labels. The AdaMatch method (Berthelot et al., 2021) also uses a pseudo-labelling strategy. It starts by training a model on labelled source domain data, which is then used to create the pseudo-labels on the unlabelled target instances. Then, the logits of these pseudo-labels are normalised, such that the class distribution of the pseudo-labels matches the class distribution of ground-truth labels (ideally from the target domain, if available, otherwise from the source domain). The final adaptation model is trained using the resulting labelled target domain data. Another method (Sun et al., 2011) proposes to use instance re-weighting on the source domain twice: the first time to align marginal distributions, and the second time

to align the conditional distributions. The model is then trained on the re-weighted source domain labelled data, and optionally on target domain labelled data as well. The AdaBN method (Li et al., 2018c) updates the batch normalization parameters according to the statistics of the target domain. This technique can also be used in SDA when fine-tuning the model.

The Meta-DMoE method (Zhong et al., 2022) was proposed to do unsupervised domain adaptation, but it can be viewed as a domain generalisation method, followed by a domain adaptation step. It starts by training one feature extractor per source domain, whose outputs are combined by an aggregator, which in turn feeds into a classifier trained in a supervised way with source domain data. Then, a distillation model is trained to approximate the outputs of the aggregator on source domain data. The weights of the distillation model are optimised using meta-learning to adapt in one batch to a new target domain. The distillation model and the classifier are used at inference time to make predictions. If the one batch adaptation to the target domain is skipped, the distillation model and the classifier can be used in the DG setting.

One method (Yang & Hospedales, 2014) that was proposed for MDL can also be used for DG. In this setup, the model receives two vectors as input: one contains the input features; the other contains a "semantic descriptor" which encodes information about the domain to which the features belong to. As long as the semantic descriptor can be built at inference time for new domains, this method can be used even when no target data is available, like in the DG setting.

3.6 Foundation Models

One promising technique for tackling the TrED problem that has gained popularity recently is the use of foundation models (FM). Foundation models are large-scale pre-trained models that serve as a general-purpose framework for various downstream tasks (Bommasani et al., 2021).

3.6.1 Current literature on FMs

Large language models (LLMs) are a type of FM for natural language processing (NLP), usually based on the transformer architecture (Vaswani et al., 2017) and trained with a self-supervised learning task. Examples of this type of model include the GPT family (Radford et al., 2018; 2019; Brown et al., 2020), BERT (Devlin et al., 2018), LaMDA (Thoppilan et al., 2022), OPT (Zhang et al., 2022) and PaLM (Chowdhery et al., 2023). These models are capable of capturing nuanced language patterns and contextual information, making them highly effective for tasks such as text classification, summarisation, translation, and conversational AI.

In computer vision (CV), models pre-trained on large image datasets like ImageNet (Deng et al., 2009) have become the standard for feature extraction in tasks like image classification or object detection. Different architectures have been proposed, such as Inception Network (Szegedy et al., 2015), ResNet (He et al., 2016), DenseNet (Huang et al., 2017) or EfficientNet (Tan & Le, 2019). More recently, generative models more similar to LLMs have also been rising in popularity, such as the DALL-E family (Ramesh et al., 2021; 2022; Betker et al., 2023) and Stable Diffusion (Rombach et al., 2022). These models can generate detailed and complex images from text descriptions, demonstrating their comprehensive representations of real-world concepts. They exhibit the ability to generalise to unseen images, abstract real-world objects, and apply these abstractions in new and varied contexts.

Given their success in NLP and CV, other FMs have been developed for different use cases, including (but not limited to) tabular data (Guo et al., 2025; Hollmann et al., 2022; Kim et al., 2024; Thomas et al., 2024; Wang & Sun, 2022; Zhang et al., 2023a), time series (Das et al., 2023; Garza & Mergenthaler-Canseco, 2023), audio (Borsos et al., 2023; Vyas et al., 2023; Yang et al., 2023), medicine (Khare et al., 2021; Li et al., 2024; Tu et al., 2024; Zhang et al., 2023b) and finance (Skalski et al., 2023; Wu et al., 2023). These specialised foundation models are tailored for specific domains, providing strong baselines that can be incrementally updated as new data becomes available. Some methods (Gardner et al., 2024; Hegselmann et al., 2023) have also explored adapting NLP FMs to process tabular data, demonstrating how to leverage the rich context of LLMs to interpret the meaning of column names and cell values.

Some FMs have been proposed that are multi-modal, meaning that they can receive and process different types of data. Early examples include CLIP (Radford et al., 2021) and ALIGN (Jia et al., 2021), which use

separate encoders for image and text data, trained jointly using contrastive objectives to align representations in a shared embedding space. Later, other generative transformer-based architectures were proposed, such as Flamingo (Alayrac et al., 2022), PaLM-E (Driess et al., 2023) and GPT-4 (Achiam et al., 2023). More recently, native multi-modal models have emerged, trained end-to-end on diverse data types including text, images, audio, video, and code. Examples include GPT-4o (OpenAI et al., 2024), Gemini (Team et al., 2023), Claude 3 family (Anthropic, 2024), and Llama 3 family (Grattafiori et al., 2024). Combining modalities enables the model to learn richer representations, leading to a more robust understanding of underlying concepts, which ultimately enhances generalisation across tasks (Sarfraz et al., 2024).

3.6.2 Applying FMs on TrED

In the context of TrED, FMs offer significant potential, since they have vast knowledge about the world already encoded in their parameters, so they do not require extensive re-training for each new task. On the other hand, this raises important risks of leakage: it is possible that data from the current domain was used during training of the FM. This would mean that the evaluation results are meaningless.

They also present various options for optimization, some of which are more appropriate at different stages of TrED. In the DG stage, the zero-shot inference capabilities of FMs can often act as a robust baseline for the target domain, due to their massive and diverse pre-training datasets. Still, to perform a new task for which there is no data, one needs to "explain" it to the FMs, usually through natural language. As such, there has been a lot of work on prompt engineering to optimize the instructions of the model, in order to improve its predictive performance and robustness (Brown et al., 2020).

In the UDA stage, FMs can be used to generate high-quality pseudo-labels for smaller, specialized models. This can be an effective option to distil the knowledge of the FM into a smaller model that is cheaper to deploy and host, and that can guarantee lower latencies at inference time (two big limitations of FMs).

In the SDA stage, the few labelled target domain examples available can be used to do few-shot In-Context Learning (ICL). It allows these models to adapt to new tasks and domains by simply providing examples or instructions directly in the input prompt, without the need for any retraining of the model (Dong et al., 2022). The limitation of this solution is the size of the context. As the labelled data grows, it may not fit entirely inside the prompt of the FM.

The limitation of context size becomes even more important in the MDL stage, when the labelled target examples hit the model's maximum context window. The solution may be to fine-tune the FM with the new data or the integration of Retrieval-Augmented Generation (Lewis et al., 2020) to pull historical target domain data dynamically. However, FM can still suffer from overfitting or catastrophic forgetting when adapted to highly specialised tasks without careful management (Wang et al., 2023).

4 Conclusion

In this survey, we study the subject of transfer learning, exploring methods that share knowledge across different domains. We describe how this subject is traditionally segmented into distinct static sub-problems, such as Domain Generalisation, Unsupervised and Supervised Domain Adaptation, and Multi-Domain Learning, each making different assumptions regarding data availability. However, in real-world applications, data and labels from new domains are often collected gradually over time, presenting a more dynamic challenge.

We propose the new problem of Transfer Learning for Evolving Domains (TrED) to describe this situation, providing a new perspective on TL by viewing these previously isolated problems as stages within a continuum. In our literature review, we find that most methods are constrained to specific settings of this process, with no current work offering an integrated approach that spans the entire continuum of data availability. We identify promising techniques to generalise across the different stages of TrED, but further research is necessary to create models that can seamlessly transfer knowledge as new data becomes available.

By viewing transfer learning as an evolving, continuous process, this perspective paves the way for more adaptive, resilient models capable of improving over time. We believe this approach will contribute to creating robust systems suited to real-world scenarios, where data is dynamic and constantly evolving.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Fouzia Altaf, Syed MS Islam, and Naeem Khalid Janjua. A novel augmented deep transfer learning for classification of covid-19 and other thoracic diseases from x-rays. *Neural Computing and Applications*, 33 (20):14037–14048, 2021.

AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024.

Yajie Bao, Yang Li, Shao-Lun Huang, Lin Zhang, Lizhong Zheng, Amir Zamir, and Leonidas Guibas. An information-theoretic approach to transferability in task transfer learning. In *2019 IEEE international conference on image processing (ICIP)*, pp. 2309–2313. IEEE, 2019.

Ricardo Barata, Miguel Leite, Ricardo Pacheco, Marco OP Sampaio, João Tiago Ascensão, and Pedro Bizarro. Active learning for imbalanced data under cold start. In *Proceedings of the Second ACM International Conference on AI in Finance*, pp. 1–9, 2021.

David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alexey Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. In *International Conference on Learning Representations*, 2021.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

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

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. Audiolm: a language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. *Advances in neural information processing systems*, 29, 2016.

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

Minmin Chen, Kilian Q Weinberger, and John Blitzer. Co-training for domain adaptation. *Advances in neural information processing systems*, 24, 2011.

Xilun Chen and Claire Cardie. Multinomial adversarial networks for multi-domain text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, 2018.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pp. 193–200, 2007.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023.

Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

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

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in neural information processing systems*, 32, 2019.

Mark Dredze, Alex Kulesza, and Koby Crammer. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning*, 79:123–149, 2010.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Lixin Duan, Dong Xu, and Ivor W Tsang. Learning with augmented features for heterogeneous domain adaptation. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, 2012.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

Boyan Gao, Henry Gouk, Yongxin Yang, and Timothy Hospedales. Loss function learning for domain generalization by implicit gradient. In *International Conference on Machine Learning*, pp. 7002–7016. PMLR, 2022.

Josh Gardner, Juan C Perdomo, and Ludwig Schmidt. Large scale transfer learning for tabular data via language modeling. *Advances in Neural Information Processing Systems*, 37:45155–45205, 2024.

Prachi Garg, Rohit Saluja, Vineeth N Balasubramanian, Chetan Arora, Anbumani Subramanian, and CV Jawahar. Multi-domain incremental learning for semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 761–771, 2022.

Azul Garza and Max Mergenthaler-Canseco. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023.

Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 597–613. Springer, 2016.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yue Guo, Wentao Zhang, Xiaojun Zhang, Vincent W Zheng, and Yi Yang. Efficient multi-expert tabular language model for banking. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 2271–2281, 2025.

Huan He, Owen Queen, Teddy Koker, Consuelo Cuevas, Theodoros Tsiligkaridis, and Marinka Zitnik. Domain adaptation for time series under feature and label shifts. In *International conference on machine learning*, pp. 12746–12774. PMLR, 2023a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Rui He, Shengcai Liu, Jiahao Wu, Shan He, and Ke Tang. Multi-domain learning from insufficient annotations. *arXiv preprint arXiv:2305.02757*, 2023b.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: Few-shot classification of tabular data with large language models. In *International conference on artificial intelligence and statistics*, pp. 5549–5581. PMLR, 2023.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pp. 1989–1998. Pmlr, 2018.

Steven CH Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey. *Neurocomputing*, 459:249–289, 2021.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Jiaxing Huang, Dayan Guan, Aoran Xiao, Shijian Lu, and Ling Shao. Category contrast for unsupervised domain adaptation in visual tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1203–1214, 2022.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021.

Yash Khare, Viraj Bagal, Minesh Mathew, Adithi Devi, U Deva Priyakumar, and CV Jawahar. Mmbert: Multimodal bert pretraining for improved medical vqa. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 1033–1036. IEEE, 2021.

Myung Jun Kim, Leo Grinsztajn, and Gael Varoquaux. Carte: Pretraining and transfer for tabular learning. In *International Conference on Machine Learning*, pp. 23843–23866. PMLR, 2024.

Abhishek Kumar, Avishek Saha, and Hal Daume. Co-regularization based semi-supervised domain adaptation. *Advances in neural information processing systems*, 23, 2010.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36, 2024.

Da Li and Timothy Hospedales. Online meta-learning for multi-source and semi-supervised domain adaptation. In *European Conference on Computer Vision*, pp. 382–403. Springer, 2020.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.

Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018a.

Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5400–5409, 2018b.

Jichang Li, Guanbin Li, Yemin Shi, and Yizhou Yu. Cross-domain adaptive clustering for semi-supervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2505–2514, 2021a.

Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M Hospedales. A simple feature augmentation for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8886–8895, 2021b.

Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018c.

Yichuan Li, Kyumin Lee, Nima Kordzadeh, Brenton Faber, Cameron Fiddes, Elaine Chen, and Kai Shu. Multi-source domain adaptation with weak supervision for early fake news detection. In *2021 IEEE International Conference on Big Data (Big Data)*, pp. 668–676. IEEE, 2021c.

Yiying Li, Yongxin Yang, Wei Zhou, and Timothy Hospedales. Feature-critic networks for heterogeneous domain generalization. In *International Conference on Machine Learning*, pp. 3915–3924. PMLR, 2019.

Luojun Lin, Han Xie, Zhishu Sun, Weijie Chen, Wenxi Liu, Yuanlong Yu, and Lei Zhang. Semi-supervised domain generalization with evolving intermediate domain. *Pattern Recognition*, 149:110280, 2024.

Yuxiu Lin and Pei Xue. Multi-task learning for macroeconomic forecasting based on cross-domain data fusion. *Journal of Computer Technology and Software*, 4(6), 2025.

Hong Liu, Mingsheng Long, Jianmin Wang, and Yu Wang. Learning to adapt to evolving domains. *Advances in neural information processing systems*, 33:22338–22348, 2020.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. *Advances in neural information processing systems*, 29, 2016.

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

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in neural information processing systems*, 31, 2018.

Massimiliano Mancini, Elisa Ricci, Barbara Caputo, and Samuel Rota Bulò. Boosting binary masks for multi-domain learning through affine transformations. *Machine Vision and Applications*, 31(6):42, 2020.

Krikamol Muandet, David Balduzzi, and Bernhard Schoelkopf. Domain generalization via invariant feature representation. In *30th International Conference on Machine Learning (ICML)*, 2013.

Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4293–4302, 2016.

Ali Narin, Ceren Kaya, and Ziynet Pamuk. Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. *Pattern Analysis and Applications*, 24:1207–1220, 2021.

Anh T Nguyen, Lam Tran, Anh Tong, Tuan-Duy H Nguyen, and Toan Tran. Casual: Conditional support alignment for domain adaptation with label shift. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 19668–19676, 2025.

Kazuki Omi, Jun Kimata, and Toru Tamaki. Model-agnostic multi-domain learning with domain-specific adapters for action recognition. *IEICE TRANSACTIONS on Information and Systems*, 105(12):2119–2126, 2022.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter

Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. Gpt-4o system card, 2024. URL `https://arxiv.org/abs/2410.21276`.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *Proceedings of the IEEE international conference on computer vision*, pp. 754–763, 2017.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 487–503, 2021.

Shuhao Qiu, Chuang Zhu, and Wenli Zhou. Meta self-learning for multi-source domain adaptation: a benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1592–1601, 2021.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Kevin Roitero, Ben Carterette, Rishabh Mehrotra, and Mounia Lalmas. Leveraging behavioral heterogeneity across markets for cross-market training of recommender systems. In *Companion Proceedings of the Web Conference 2020*, pp. 694–702, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pp. 213–226. Springer, 2010.

Kuniaki Saito and Kate Saenko. Ovanet: One-vs-all network for universal domain adaptation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8980–8989. IEEE Computer Society, 2021.

Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*, pp. 2988–2997. PMLR, 2017.

Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Adversarial dropout regularization. In *International Conference on Learning Representations*, 2018a.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3723–3732. IEEE, 2018b.

Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 153–168, 2018c.

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8050–8058, 2019a.

Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6956–6965, 2019b.

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. *Advances in neural information processing systems*, 33:16282–16292, 2020.

Fahad Sarfraz, Bahram Zonooz, and Elahe Arani. Beyond unimodal learning: The importance of integrating multiple modalities for lifelong learning. *arXiv preprint arXiv:2405.02766*, 2024.

Giovanni Sartor, Francesca Lagioia, et al. The impact of the general data protection regulation (gdpr) on artificial intelligence, 2020.

Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772, 2012.

Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. *Advances in neural information processing systems*, 29, 2016.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Ankit Singh. Clda: Contrastive learning for semi-supervised domain adaptation. *Advances in Neural Information Processing Systems*, 34:5089–5101, 2021.

Piotr Skalski, David Sutton, Stuart Burrell, Iker Perez, and Jason Wong. Towards a foundation purchasing model: Pretrained generative autoregression on transaction sequences. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pp. 141–149, 2023.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pp. 443–450. Springer, 2016.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Qian Sun, Rita Chattopadhyay, Sethuraman Panchanathan, and Jieping Ye. A two-stage weighting framework for multi-source domain adaptation. *Advances in neural information processing systems*, 24, 2011.

Xiaona Sun, Zhenyu Wu, Zhiqiang Zhan, and Yang Ji. Contrastive conditional alignment based on label shift calibration for imbalanced domain adaptation. In *International Conference on Pattern Recognition*, pp. 13–28. Springer, 2025.

Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Hui Tang and Kui Jia. Discriminative adversarial domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5940–5947, 2020.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Valentin Thomas, Junwei Ma, Rasa Hosseinzadeh, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony Caterini. Retrieval & fine-tuning for in-context tabular models. *Advances in Neural Information Processing Systems*, 37:108439–108467, 2024.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

Tao Tu, Shekoofeh Azizi, Danny Driess, Mike Schaekermann, Mohamed Amin, Pi-Chuan Chang, Andrew Carroll, Charles Lau, Ryutaro Tanno, Ira Ktena, et al. Towards generalist biomedical ai. *NEJM AI*, 1(3): AIoa2300138, 2024.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176, 2017.

Gido M Van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Thuy Vu, Shahram Khadivi, Dinh Phung, and Gholamreza Haffari. Domain generalisation of nmt: Fusing adapters with leave-one-domain-out training. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 582–588, 2022.

Apoorv Vyas, Bowen Shi, Matthew Le, Andros Tjandra, Yi-Chiao Wu, Baishan Guo, Jiemin Zhang, Xinyue Zhang, Robert Adkins, William Ngan, et al. Audiobox: Unified audio generation with natural language prompts. *arXiv preprint arXiv:2312.15821*, 2023.

Jindong Wang, Yiqiang Chen, Shuji Hao, Wenjie Feng, and Zhiqi Shen. Balanced distribution adaptation for transfer learning. In *2017 IEEE international conference on data mining (ICDM)*, pp. 1129–1134. IEEE, 2017.

Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and S Yu Philip. Generalizing to unseen domains: A survey on domain generalization. *IEEE transactions on knowledge and data engineering*, 35(8):8052–8072, 2022a.

Kaizheng Wang. Pseudo-labeling for kernel ridge regression under covariate shift. *arXiv preprint arXiv:2302.10160*, 2023.

Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7201–7211, 2022b.

Rui Wang, Zuxuan Wu, Zejia Weng, Jingjing Chen, Guo-Jun Qi, and Yu-Gang Jiang. Cross-domain contrastive learning for unsupervised domain adaptation. *IEEE Transactions on Multimedia*, 2022c.

Zhenyi Wang, Enneng Yang, Li Shen, and Heng Huang. A comprehensive survey of forgetting in deep learning beyond continual learning. *arXiv preprint arXiv:2307.09218*, 2023.

Zifeng Wang and Jimeng Sun. Transtab: Learning transferable tabular transformers across tables. *Advances in Neural Information Processing Systems*, 35:2902–2915, 2022.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3:1–40, 2016.

Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23:69–101, 1996.

Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.

Yuan Wu and Yuhong Guo. Dual adversarial co-learning for multi-domain text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 6438–6445, 2020.

Dongbo Xi, Zhen Chen, Yuexian Wang, He Cui, Chong Peng, Fuzhen Zhuang, and Peng Yan. Large-scale multi-domain recommendation: an automatic domain feature extraction and personalized integration framework. *arXiv preprint arXiv:2404.08361*, 2024.

Biao Yang, Junrui Zhu, Zhitao Yu, Fucheng Fan, Xiaofeng Liu, and Rongrong Ni. Fast adaptation trajectory prediction method based on online multisource transfer learning. *IEEE Transactions on Automation Science and Engineering*, 2024.

Dongchao Yang, Jinchuan Tian, Xu Tan, Rongjie Huang, Songxiang Liu, Xuankai Chang, Jiatong Shi, Sheng Zhao, Jiang Bian, Xixin Wu, et al. Uniaudio: An audio foundation model toward universal audio generation. *arXiv preprint arXiv:2310.00704*, 2023.

Shiqi Yang, Yaxing Wang, Joost Van De Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8978–8987, 2021.

Yongxin Yang and Timothy M Hospedales. A unified perspective on multi-domain and multi-task learning. *arXiv preprint arXiv:1412.7489*, 2014.

Yi Yao and Gianfranco Doretto. Boosting for transfer learning with multiple sources. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 1855–1862. IEEE, 2010.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Universal domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2720–2729, 2019.

Jingyang Yuan, Xiao Luo, Yifang Qin, Zhengyang Mao, Wei Ju, and Ming Zhang. Alex: Towards effective graph transfer learning with noisy labels. In *Proceedings of the 31st ACM international conference on multimedia*, pp. 3647–3656, 2023.

Han Zhang, Xumeng Wen, Shun Zheng, Wei Xu, and Jiang Bian. Towards foundation models for learning on tabular data. *arXiv preprint arXiv:2310.07338*, 2023a.

Jing Zhang, Wanqing Li, Philip Ogunbona, and Dong Xu. Recent advances in transfer learning for cross-dataset visual recognition: A problem-oriented perspective. *ACM Comput. Surv.*, 52(1), feb 2019. ISSN 0360-0300.

Kai Zhang, Jun Yu, Zhiling Yan, Yixin Liu, Eashan Adhikarla, Sunyang Fu, Xun Chen, Chen Chen, Yuyin Zhou, Xiang Li, et al. Biomedgpt: A unified and generalist biomedical generative pre-trained transformer for vision, language, and multimodal tasks. *arXiv preprint arXiv:2305.17100*, 2023b.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Peilin Zhao, Steven CH Hoi, Jialei Wang, and Bin Li. Online transfer learning. *Artificial intelligence*, 216: 76–102, 2014.

Tao Zhong, Zhixiang Chi, Li Gu, Yang Wang, Yuanhao Yu, and Jin Tang. Meta-dmoe: Adapting to domain shift by meta-distillation from mixture-of-experts. *Advances in Neural Information Processing Systems*, 35:22243–22257, 2022.

Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 13025–13032, 2020.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4396–4415, 2022.

Zhi Zhou, Kun-Yang Yu, Lan-Zhe Guo, and Yu-Feng Li. Fully test-time adaptation for tabular data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23027–23035, 2025.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.