University of the Peloponnese
Πανεπιστήμιο Πελοποννήσου

Ph.D. Thesis

Evidence Transfer: A Versatile Deep Representation Learning Method for Information Fusion

---

Author:
Athanasios Davvetas

Supervisor:
Professor Spiros Skiadopoulos

November 29, 2021

Evidence Transfer: Μια Ευέλικτη Μέθοδος Εκμάθησης Αναπαραστάσεων για τη Διαδικασία Σύντηξης Πληροφορίας

Διδακτορική Διατριβή
του
Αθανάσιου Δαββέτα

Διπλωματούχου Μηχανικών Πληροφορικής του Τ.Ε.Ι Αθήνας (νυν Πανεπιστήμιο Δυτικής Αττικής) (2016)

Συμβουλευτική Επιτροπή:	Σπυρίδων Σκιαδόπουλος	Επιβλέπων καθηγητής
	Χρήστος Τρυφωνόπουλος	
	Ευάγγελος Καρκαλέτσης	

Εγκρίθηκε από την επταμελή εξεταστική επιτροπή την 6/12/21

Όνομα	Βαθμίδα	Ίδρυμα
Κωνσταντίνος Βασιλάκης	Καθηγητής	Πανεπιστήμιο Πελοποννήσου
Θεόδωρος Γιαννακόπουλος	Ερευνητής Β΄	ΕΚΕΦΕ «Δημόκριτος»
Ευάγγελος Καρκαλέτσης	Ερευνητής Α΄	ΕΚΕΦΕ «Δημόκριτος»
Εμμανουήλ Κουμπαράκης	Καθηγητής	Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών
Ιωάννης Κωτίδης	Καθηγητής	Οικονομικό Πανεπιστήμιο Αθηνών
Σπυρίδων Σκιαδόπουλος	Καθηγητής	Πανεπιστήμιο Πελοποννήσου
Χρήστος Τρυφωνόπουλος	Αναπληρωτής Καθηγητής	Πανεπιστήμιο Πελοποννήσου

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Δρ. Σπυρίδωνα Σκιαδόπουλο, καθώς και τα υπόλοιπα μέλη της τριμελούς μου επιτροπής: Δρ. Ευάγγελο Καρκαλέτση και Δρ. Χρήστο Τρυφωνόπουλο, για την επίβλεψη και στήριξη τους, καθ' όλη την διάρκεια της διδακτορικής μου διατριβής. Ιδιαίτερα, θα ήθελα να ευχαριστήσω τον Δρ. Ηρακλή Α. Κλαμπάνο από το Ινστιτούτο Πληροφορικής και Τηλεπικοινωνιών του Εθνικού Κέντρου Έρευνας Φυσικών Επιστημών (Ε.Κ.Ε.Φ.Ε) «Δημόκριτος», για την πάσης φύσεως υποστήριξη και βοήθεια σε όλη την πορεία της διατριβής μου.

Επίσης, θα ήθελα να εκφράσω την ευγνωμοσύνη μου σε όλους τους οργανισμούς-μέλη που συμμετέχουν στο Πρόγραμμα Βιομηχανικών Υποτροφιών και ιδιαίτερα στο Ίδρυμα Σταύρος Νιάρχος, για την πολύτιμη υποτροφία που μου χορηγήθηκε. Ομοίως, ευχαριστώ το Ε.Κ.Ε.Φ.Ε «Δημόκριτος» για την αποτελεσματική διαχείριση, υποστήριξη και υλοποίηση του παραπάνω προγράμματος. Επίσης, θα ήθελα να ευχαριστήσω την Danaos Management Consultants για την ωφέλιμη συνεισφορά της, καθώς και για την απρόσκοπτη και επωφελή συνεργασία.

Τέλος, θα ήθελα να ευχαριστήσω θερμά την οικογένεια μου, τους φίλους μου και την σύντροφο μου για την ανιδιοτελή υποστήριξη τους.

Abstract

In the recent years, the collection of more and more data instances, has increasingly led to an abundance. After the investigation of efficient ways to deal with storing, managing and collecting of large-scale or diverse data, the research interest of the scientific community has shifted into the extraction of meaningful information from such collections. Deep learning lends itself particularly well to the process of extracting valuable information. Deep learning methods thrive with large-scale datasets. Due to their ability to learn alternative representations from raw observations, the abundance of data instances allows for generalised representations. In turn, generalised representations allow for effective learning of complex tasks. Despite valuable efforts in extraction of information from single data sources or data types, dealing with multiple diverse data sources remains an open question in the scientific community.

Representation learning enables combination and juxtaposition of multiple diverse data sources in a meaningful, common and lower-dimensional space. However, typical learning frameworks for joint representation learning, face a plethora of challenges. Initially, architectural decisions of the involved neural networks is often a product of manual work or application specific decisions that rarely generalise to multiple domains or tasks. At the same time, directly tying data sources in the input layers of the neural network introduces an expectation of constant availability. However, expecting all data sources to be constantly available, is not realistic in real world applications. In addition, the involvment of redudant or non-complementary data sources may lead to detoriating performance. However, dealing with such sources requires manual effort. Such effort, is put into creating explicit assumptions or rules that will ensure stability, or to understand the intricate relations between data sources, in order to avoid non-complementary ones.

This thesis includes the formulation and investigation of the hypothesis that external data evidence improves deep representation learning. The above investigation results in the proposal of a deep representation learning method, called Evidence Transfer (EviTraN). EviTraN is a versatile and automated fusion scheme based on deep representation learning, transfer learning and hybrid modelling between gener-

ative and discriminative views. In addition, it leads to the proposal of a set of evaluation criteria for deep representation learning for the purposes of information fusion. Furthermore, this thesis includes a theoretical interpretation of the above method, based on comparison with the well-received Information Bottleneck method, that acts as a stepping stone towards explainable modelling and open science. The evaluation process of EviTraN also includes a realistic scenario of detecting severe weather in an unsupervised manner. Thus, demonstrating its impact and potential use in additional real-world applications.

Experimental evaluation with artificially generated, as well as, realistic evidence sources suggest that EviTraN is a robust and effective method. In addition, it is versatile, as it allows the introduction of a variety of relations, including non-complementary ones. Furthermore, due to its learning process based on the transfer learning setting, it is a modular fusion scheme that does not require all data sources to be present during inference (only the primary data instances).

Περίληψη

Τα τελευταία χρόνια, η διαδικασία συλλογής ολοένα και περισσότερων δεδομένων έχει ως αποτέλεσμα την ύπαρξη πληθώρας δεδομένων. Μετά τη διερεύνηση αποτελεσματικών τρόπων αποθήκευσης, διαχείρισης και συλλογής δεδομένων μεγάλης κλίμακας ή ποικίλων τύπων, το ερευνητικό ενδιαφέρον της επιστημονικής κοινότητας μετατοπίστηκε στην εξαγωγή πληροφορίας από τέτοιου είδους συλλογές. Η βαθιά μάθηση (deep learning) χρησιμοποιείται συχνά για τη διαδικασία εξαγωγής πολύτιμης πληροφορίας. Οι μέθοδοι βαθιάς μάθησης ευδοκιμούν με σύνολα δεδομένων μεγάλης κλίμακας, λόγω της ικανότητάς τους να μαθαίνουν εναλλακτικές αναπαραστάσεις από ακατέργαστες παρατηρήσεις. Η διαθέσιμη πληθώρα δεδομένων επιτρέπει την εκμάθηση γενικευμένων αναπαραστάσεων. Με τη σειρά τους, οι γενικευμένες αναπαραστάσεις επιτρέπουν την αποτελεσματική εκμάθηση πολύπλοκων εργασιών. Παρά τις επιτυχείς προσπάθειες για την εξαγωγή πληροφοριών από μεμονωμένες πηγές δεδομένων ή τύπους δεδομένων, η αντιμετώπιση πολλαπλών διαφορετικών πηγών δεδομένων παραμένει ένα ανοιχτό ερώτημα στην επιστημονική κοινότητα.

Η εκμάθηση αναπαραστάσεων (representation learning) επιτρέπει τον συνδυασμό και την αντιπαράθεση πολλαπλών διαφορετικών πηγών δεδομένων σε έναν χώρο κοινό, ουσιαστικό και χαμηλότερων διαστάσεων. Ωστόσο, τα τυπικά πλαίσια μάθησης για κοινή εκμάθηση αναπαραστάσεων (joint representation learning) πρέπει να αντιμετωπίσουν μια πληθώρα προκλήσεων. Αρχικά, οι αρχιτεκτονικές αποφάσεις των εμπλεκόμενων νευρωνικών δικτύων είναι συχνά προϊόντα προερχόμενα από διαδικασίες ή αποφάσεις που εμπλέκουν ανθρώπινη παρέμβαση (μη αυτόματες). Οι συγκεκριμένες διαδικασίες ή αποφάσεις συνήθως αφορούν συγκεκριμένες εφαρμογές και σπάνια γενικεύονται σε πολλαπλούς τομείς ή εργασίες. Ταυτόχρονα, η απευθείας σύνδεση πηγών δεδομένων στα επίπεδα εισόδου του νευρωνικού δικτύου εισάγει μια προσδοκία σταθερής διαθεσιμότητας. Ωστόσο, σε πραγματικές εφαρμογές, η προσδοκία διαθεσιμότητας όλων των πηγών δεδομένων δεν είναι ρεαλιστική. Επιπλέον, η επίδοση των τυπικών πλαισίων μάθησης μπορεί να μειωθεί κατά τη χρήση περιττών ή μη συμπληρωματικών πηγών δεδομένων. Η αντιμετώπιση μια τέτοιας συμπεριφοράς, επίσης απαιτεί τη χρήση μη-αυτόματων διαδικασιών. Η χειρωνακτική εργασία που καταβάλλεται, σκοπεύει στη δημιουργία συγκεκριμένων υποθέσεων ή κανόνων που θα διασφαλίζουν τη σταθερότητα ή στην κατανόηση

των περίπλοκων σχέσεων μεταξύ των πηγών δεδομένων, προκειμένου να αποφευχθούν οι μη συμπληρωματικές σχέσεις.

Σε αυτή τη διατριβή, διερευνάται η υπόθεση ότι η χρήση εξωτερικών δεδομένων βελτιώνει την εκμάθηση αναπαραστάσεων. Η παραπάνω έρευνα καταλήγει στην πρόταση μιας μεθόδου εκμάθησης αναπαραστάσεων, που ονομάζεται Evidence Transfer (EviTraN). Η EviTraN είναι ένα ευέλικτο και αυτοματοποιημένο σχήμα σύντηξης πληροφορίας (information fusion) που βασίζεται στην εκμάθηση αναπαραστάσεων, τη μεταφορά μάθησης (transfer learning) και την υβριδική μοντελοποίηση (hybrid modelling). Επιπλέον, προτείνεται μια σειρά κριτηρίων αξιολόγησης για την εκμάθηση αναπαραστάσεων για τους σκοπούς της σύντηξης πληροφοριών. Ακόμα, η διατριβή περιλαμβάνει μια θεωρητική ερμηνεία της παραπάνω μεθόδου, βασισμένη στη σύγκριση με τη μέθοδο Information Bottleneck, η οποία αποτελεί θεμέλιο λίθο για επεξηγηματική μοντελοποίηση και ανοιχτή επιστήμη. Η διαδικασία αξιολόγησης της EviTraN περιλαμβάνει επίσης ένα ρεαλιστικό σενάριο ανίχνευσης έντονων καιρικών συνθηκών χωρίς επίβλεψη, αποδεικνύοντας έτσι τον αντίκτυπό της, καθώς και την πιθανή χρήση της σε πρόσθετες πραγματικές εφαρμογές.

Η πειραματική αξιολόγηση με τεχνητά παραγόμενες, καθώς και ρεαλιστικές πηγές πληροφορίας υποδηλώνει ότι η EviTraN είναι μια σταθερή και αποτελεσματική μέθοδος. Επιπλέον, είναι ευέλικτη, καθώς επιτρέπει την εισαγωγή ποικίλων σχέσεων, συμπεριλαμβανομένων των μη συμπληρωματικών. Ακόμα, λόγω της διαδικασίας εκμάθησής της που βασίζεται στη μεταφορά εκμάθησης (transfer learning), είναι ένα αρθρωτό σχήμα σύντηξης που δεν απαιτεί να υπάρχουν όλες οι πηγές δεδομένων κατά την εξαγωγή συμπερασμάτων (μόνο δεδομένα που ανήκουν στην κύρια συλλογή δεδομένων).

Contents

List of Figures

List of Abbreviations

EviTraN	Evidence Transfer
KLD	Kullback-Leibler Divergence
IB	Information Bottleneck
CNN	Convolutional Neural Networks
RGB	Red, Green and Blue
ITL	Inductive Transfer Learning
TTL	Transductive Transfer Learning
UTL	Unsupervised Transfer Learning

Chapter 1

Introduction

The rise of big data in previous years, has affected multiple aspects of technology, communications, operations and even various tasks in our everyday lives [29]. Although big data are characterised by multiple properties, its most common aspect is its large volume. Despite its effects on multiple domains and operations, the rise of big data has also driven the research interest of the scientific community into dealing with challenges that arise from their large volume. Dealing with such challenges answers scientific questions such as how to manage data from multiple sources?, how to process large-scale data? or how to make sense or extract meaningful and valuable information from large-scale data? Sharma [30] defined the above challenges as "*Representation of Unstructured Data*" and "*Analysis of Big Data*". The rise of big data lead to the emerging of multiple technologies, e.g., big data file systems or big data processing tools, that now allow effective operations over voluminous data such as managing, storing, retrieving or processing.

In this day and age, data not only can be found in an abundance, but we are also able to efficiently perform multiple operation on them, thanks to research done in the previous years. However, the research interest has shifted more into the analysis of data and specifically on making data-driven decisions in an automatic or semi-automatic manner. What described as living in the *data age* or *big data age*, is the empowerment of multiple services and technologies, such as machine learning (including artificial intelligence and deep learning), 5G networking or blockchain, from voluminous datasets [31]. The ability to extract automated decisions based on observation of data has affected multiple domains including data-driven decision support for management of flood risks [32]. Yet, the research question of "how to deal with heterogeneous or diverse data sources?" has become more and more relevant since many areas, such as the flood risk management, involve data from heterogeneous data sources.

Dealing with multiple heterogeneous, diverse or homogenous data sources is by no

means a new problem. Data fusion or information fusion is an age-old quest of combining multiple data sources or data products in order to achieve increased performance. The criteria for the quality of fusion process may vary depending on the application or task at hand. It frequently aims to increase the effectiveness of a task-specific metric or reduce the redundancy yielded from the combination of the individual data sources. In this booming of available data instances, data fusion has become increasingly relevant. Dealing with multiple data sources is not straightforward. Extracting decisions from multiple data sources, requires a fusion scheme capable of combining, associating, correlating or aligning the data sources. Despite, dealing with multiple data sources can be hard to manage and thus, dealing with a combined data product (as a result of some data aligning method) is more preferred in most applications.

In spite of data fusion being a scientific direction that has been deeply researched, fusion schemes that sprout from application or data type specific decisions are frequent. Typically, the combination or fusion of heterogeneous data sources involves the extraction or generation of data type specific transition rules or tables. This process requires a plethora of manual work and is typically hard to manage. In addition, such data type specific fusion schemes rarely generalise to other data types. At the same time, it may also involve trial and error procedures which are perplexing, rarely explainable and also require a plethora of resources. Deep learning lends itself to data fusion by allowing the automatic combination, association, correlation or alignment of data sources in lower dimensional and meaningful latent space.

The abundance of data instances has enabled the effectiveness of machine learning and especially deep learning methods. Deep learning is thriving in multiple applications, such as computer vision [33], prediction of DNA-RNA sequences [34] and food-related applications [35]. Meaningful representation learning is an attribute that is generally accepted as one of the key aspects that make deep learning particularly effective. Sufficient amounts of learning instances allow the learning of generalised representations, which in turn lead to learning of complex tasks. In addition, involving multiple data sources in deep representation learning provides a multi-view perspective of the problem or additional training instances and features. However, generally applicable deep representation learning frameworks that deal with heterogeneous, diverse or non-complementary data types are lacking.

Despite a plethora of data instances being available, labelled data instances are scarce. The scarcity of labelled data instances is an outcome of costly or timely annotation processes. Complex tasks require manual work for annotation. Despite automatic annotation processes existing, they are either application or data type specific, e.g., image-to-text or text-to-image only, or not as accurate as manually annotated instances. To alleviate costs or due to time restrictions, weak labels are oftenly utilised instead. Weak labels are labels with limitations such as being incom-

plete, noisy, containing errors or representing simpler tasks from that of the original task. Labelled data instances are vital to the training of supervised methods which are typically preferred by practitioners, due to being reliable and effective. However, limited amounts of labelled data affects both of these properties. To this end, methods that can learn both from labelled and unlabelled instances are often wanted.

This thesis includes the investigation of hypothesis: "external data evidence improves deep representation learning". Evaluating whether external data evidence improves the learning of deep representations or not, requires these key components: at least two data sources (one primary and multiple auxiliary/external), a representation learning mechanism and a way of tinkering with the initial set of representations in order to extract an improved set of representations. This thesis focuses on the investigation of a tinkering mechanism, which is the proposed method, called *Evidence Transfer* (EviTraN).

EviTraN is a representation learning method that aims to fuse multiple data sources by learning joint representations. It can be considered as a hybrid method of using auxiliary supervision in an unsupervised learning process. External evidence, which is a task outcome extracted from an external dataset, is introduced in the learning process of intermediate representations. This allows EviTraN, to learn combined representations of a primary dataset (with no labelled instances) along with external views from auxiliary datasets. As it involves task outcomes, it does not deal with challenges yielded from the involvement of heterogeneous datasets, such as requiring alignment or data type specific rules.

At the same time, the end-goal of the representation learning is the combination of diverse data sources (which may be potentially heterogeneous). It aims to fuse data sources in an intermediate-level (within the hidden layers of a neural network). The extraction of weak or strong task outcomes is preceding the fusion in order to deal with data heterogeneity. In this way, it allows the fusion of data sources by reducing manual work or trial and error decisions, as well as, being a generally applicable method, as it does not involve any data type specific processes.

1.1 Motivation

The goal of the study is to investigate the intelligent combination of diverse or heterogeneous types of data in an automated manner. Combining knowledge from multiple data sources allows for informed decisions and a multi-view perspective. In turn this allows for increased generalisation or effectiveness on an application or task, that repurposes the combined product of the fusion process.

Domain experts are able to manually combine data in order to assess situations,

draw informed conclusions or make educated decisions. However, the data products of such combination processes are typically results of trial and error methods. Fusion processes that involve manual work are error-prone and may lead to suboptimal solutions. In addition, it may lead to less generally applicable methods as they may depend on the data types, e.g., specifically tied to images and text, such as image-to-text or text-to-image methods. At the same time, the combination of data through semantic information and metadata yields collection of schemas, transition tables and rules which are hard to manage. Semantic-based combination may also lead to tying between data and application, similarly to manual combination methods. Despite, metadata or other semantic information do not accommodate frequently used data types such as arrays, numbers, strings or encodings. On the other hand, analytics and data-science despite being used to discover structure in data for decision-making, they also often rely on trial and error processes. During such process the connection between the semantics of the data and the task at hand is not clear. This may be problematic when diverse data types are involved in analysis that involves deep learning.

Consider an example of combining textual and weather information with deep learning for classification. Weather consists of multiple physical variables which are typically studied over a grid of cells that represent geospatial areas. For the task at hand, some physical variables will be highly relevant for the task at hand, while others will be irrelevant. At the same time, text comes in many forms, e.g., bag of words, TF-IDF features, n-grams, etc. Deep learning classification fusion schemes often treat these fundamentally different features equally, by merging them within their layers. This leads the causal relationship between input types and the task at hand (output) to be unclear. On the other hand, including high-level information regarding the semantics of the different input types and aligning them with the semantics of the task at hand, would lead to traceability and argumentation over the analysis of results.

Using feed-forward neural networks which are designed for explicit representation learning, such as autoencoders, leads to automatic feature engineering. Explicit representation learning is the process of learning alternative representations of raw observations, typically without the use of supervision. Learned representations are more robust and consists of meaningful features that allow the discovery of semantic value, capable of enabling the association or alignment across data sources. In addition, the manifestation of such representations through unsupervised learning, allows their generalisation to a multitude of tasks. This will enable the extraction of reproducible data derivatives such as representations, clusters, etc. This procedure reduces the effort put into understanding the relations between data types in order to propose a proper combination, which in most alternative cases is application or data type specific. In addition, it enables the investigation of causal relations between

input and task.

However, the selection of explicit or implicit representation learning depends on the availability of labelled data samples or a set of properties expected from the learned set of representations. Implicit representation typically involves a supervised learning mechanism, while explicit representation learning is typically an unsupervised process. Consequently, the learned representations from implicit representation learning are typically task-specific, while learned representations from explicit representation learning are generally applicable. As explicit representation learning is an unsupervised learning process, the learning of representations is a process that involves low or intermediate level of information such as raw observations or intermediate/latent features.

Yet, for the extraction of some task outcomes only the observation of data features is not capable of transmitting the necessary amount of information for effective decision-making, such as severe weather detection (more in Chapter 6). Consider the example of a simple computer vision task of identifying whether an image depicts a horse or a chair. Humans are able to quickly decide whether a horse exists within an image, since we have a clear understanding of what a horse is. In the other words, we have learned the concept of horse through observation, which in practice is a representation produced in our brain. In future encounters with horses, we compare the visual signal with our representation in order to decide whether the visual signal depicts a horse. Now consider the case that we have no representation of the concepts of horse or chairs. At first sight both the horse and the chair may have similar characteristics such as the same amount of legs (four legs), colour (a range of dark colours) or long bodies. Therefore, trying to learn representations (concepts) from observations with similar characteristics could lead to misinterpretation or errors. As raw observations do not indicate higher level characteristics, such as the fact that horse are animals found in farms and chairs are inanimate objects found within homes or other social areas (offices, etc.).

Being able to introduce higher level concepts in the learning process of unsupervised representation learning methods would bear both advantages of implicit and explicit representation learning. In other words, the training objective would not require labelled instances, and therefore it will learn task-agnostic representations that would be able to be repurposed in a variety of applications. At the same time, involving weak or strong supervision in cases where such information is available, without explicitly requiring labelled instances in the training objective, would provide necessary information to distinguish between concepts with similar characteristics. Therefore, reducing misinterpretation or errors in learning the underlying explanatory factors of the involved concepts.

1.1.1 Towards Artificial Intelligence

From traditional machine learning methods to deep learning, unsupervised learning can be considered as the closest form of machine learning to artificial intelligence (AI). The model learns relevant features by itself without the use of supervision. A big debate on the goals of artificial intelligence has been raised. Pioneers of the unsupervised representation learning have highlighted the significance of unsupervised learning and especially unsupervised/explicit representation learning towards artificial intelligence as follows: "*Although specific domain knowledge can be used to help design representations, learning with generic priors can also be used, and the quest for AI is motivating the design of more powerful representation-learning algorithms implementing such priors*" [36].

Yet, Pennachin and Goertzel [37] debate that recent advancements in statistical AI (e.g., deep learning) have drifted from the original focus of the AI field, by focusing on creating models that demonstrate intelligence selectively on specialised applications or domains. The authors characterise this approach as being the "narrow" counterpart of AI. They suggested that Artificial General Intelligence (AGI) was the original focus of AI, with AGI being the counterpart to the "*collection of dumb specialists in small domains*" [38][1].

Despite, it is evident from this debate: that learning is not what makes a model intelligent, but rather its understanding of the subject's fundamental concepts. Even though unsupervised learning models are able to extract knowledge with no supervision, they are lacking in understanding the semantic value of the input or relations between essential parts. To this end, guiding the unsupervised learning process of such models with high-level information extracted from related data sources can be considered as a gateway to understanding the semantic value of the input. As in order to associate or align data sources, knowledge of the intricate mechanisms and interactions between the sources is required. Such knowledge which is introduced either implicitly – learning relations from observation of data instances or explicitly – learning a tangible version of their relation may allow a generalised perspective or even knowledge of the problem.

1.2 Contributions

This section includes the contributions of this thesis extracted from the investigation of the research hypothesis.

[1]This quote from Stork [38] appears in the preface of the work of Pennachin and Goertzel [37]

1.2.1 Evaluation Criteria of Deep Representation Learning for Fusion

To deal with challenges that arise during data fusion, the proposed method should satisfy a set of criteria capable of enabling an effective and robust data fusion. The evaluation criteria of EviTraN are: effectiveness, robustness and modularity. The effectiveness criterion is considered by most previous work. The combination of multiple data sources aims to produce alternative data source(s), feature set(s) or decision(s) of higher quality than their individual counterparts. However, one may define effectiveness in multiple ways. The definition of effectiveness may depend on the application, domain or data type specific properties. Therefore, the criterion of effectiveness varies depending on the task at hand. The objective of EviTraN is to produce meaningful representations of high semantic value that will represent the combined knowledge coming from both primary and auxiliary datasets. Therefore, its effectiveness relies on achieving the above objective.

Robustness represents an often neglected property of fusion schemes. A fusion scheme designed for generalised application should not make assumptions regarding the properties or types of the involved data sources. However, involving arbitrary relations may result in unwanted effects, such as reduced performance due to unrelated relations or malicious disruption of the training process. EviTraN should resist disturbance from noisy, non-complementary, uncorrelated or malicious evidence. It should preserve initial performance, as indicated before the introduction of evidence in the method.

Modularity is also a vital criterion that diverges EviTraN from previous work on using representation learning for fusion. In most cases, the auxiliary data or decision sources are explicitly involved as inputs of the deep neural network architecture. By explicitly involving all data sources as input, it conditions the training and as an extent the inference of the network, to require all data sources to be available at the time of training or testing. Having all data sources available at the time of testing is not realistic. As mentioned before, the procedure of annotation is often costly or timely and therefore, during later iteration of performing inference, auxiliary data sources may not be available. For that reason, the fusion step of EviTraN should be iterative, and not explicitly involve the auxiliary data sources. Relevant publications: *P2*, *P5*.

1.2.2 Versatile and Automatic Fusion Scheme

EviTraN combines data sources by involving task outcomes extracted from auxiliary data sources. It is able to involve any categorical variable. This enables Evi-

TraN to be versatile by involving any relations between auxiliary datasets and primary dataset. At the same time, ensuring resistance against disturbance of non-complementary or malicious relations, allows EviTraN to be widely applied to any applications without detoriating the initial performance. In practice, this means that EviTraN can be deployed in multiple applications as it bears no downsides in regard to performance. At the same time, EviTraN performs automatic correlation/association/alignment of data sources in a latent space through training with a composite training objective. The training objective and a pre-processing step, intermediate to initialisation and transfer, is able to ensure the effectiveness during meaningful relations, while also the robustness during low quality of evidence. EviTraN is a deep representation learning method that involves auxiliary task outcomes in the unsupervised training process of a primary dataset. In practice, this means that EviTraN learns from unlabelled data (primary dataset) and weak or strong labelled instances (auxiliary task outcomes – external evidence).

Furthermore, it is tested in three different learning settings which cover a broad spectrum of scenarios that involve auxiliary data sources. The three learning settings are: *hybrid*, *inaccurate* and *incomplete* learning. Hybrid learning involves meaningful categorical variables that indicate the outcome of a task. They are characterised by consistency and often portray a semantically related task to the primary dataset. Inaccurate learning involves noisy, non-corresponding or ill-intended categorical variables. Inaccurate learning is vital to the evaluation of the robustness criterion as these types of variables may be detrimental to the learning process. Incomplete learning involves auxiliary categorical variables of incomplete correspondence to the primary dataset, such as uniformly missing samples or missing samples of specific classes. Depending on the amount of missing samples, incomplete categorical variables may affect the learning process similar to inaccurate variables, which makes them highly volatile. Relevant publications: *P2* and *P5*.

1.2.3 Theoretical Interpretation of Evidence Transfer

Deep learning is notoriously connected with unexplainable or difficult to understand effectiveness. Most practitioners treat deep learning models as black boxes that perform complex tasks through training with large scale datasets. Researchers have frequently highlighted the advantages of being able to interpret the effects of a method. Being able to interpret the inner workings of a model allows for insightful communication of the results, potential increase in effectiveness by targeted adjustment of the moving parts of the architecture and being an important stepping stone towards open science. A deep representation learning fusion scheme which aims to be generally applicable such as EviTraN, should be accommodated by a theoretical

interpretation of its effects.

The comparison between the proposed method and the Information Bottleneck method [39] enables the interpretation of EviTraN. Information bottleneck has been previously used as a way of explaining the effectiveness of deep learning. EviTraN has very similar effects to that of infomration bottleneck, first it restricts an autoencoder into compressing the information of primary dataset into a dimensionally smaller space. At the same time, it increases the relevance of the latent features compared to the auxiliary introduced task outcomes. Such interpretation allows the use of EviTraN, as a fusion scheme, in multiple realistic scenarios, e.g., data with high practical impact. The learning process, as well as, topology and evaluation of the involved neural network architectures can be described with the use of ANNETT-O (P1). Relevant publications: *P1* and *P3*.

1.2.4 Evaluation in Realistic Scenario

One of these cases mentioned in the previous subsection is the realistic evaluation scenario which is included in this study. The experimental evaluation of EviTraN includes a use case of detecting severe weather in an unsupervised manner. The investigation of EviTraN in a realistic scenario revolves around improving deep representation learning for a primary dataset that consists of weather data, by introducing binary classification task outcomes of individual severe weather events, such as windstorms, floods and tornadoes. Weather data are very impactful, as they affect a multitude of aspects in our everyday lives. EviTraN is able to increase the effectiveness of individual severe weather detection and thus, indicating the real-world impact of being able to improve the learning of deep representations. Relevant publication: *P4*.

Publications. The investigation of the thesis hypothesis led in the following papers:

P1 Iraklis A Klampanos, Athanasios Davvetas, Antonis Koukourikos, and Vangelis Karkaletsis. Annett-o: an ontology for describing artificial neural network evaluation, topology and training. *International Journal of Metadata, Semantics and Ontologies*, 13(3):179–190, 2019.

P2 Athanasios Davvetas, Iraklis A Klampanos, and Vangelis Karkaletsis. Evidence transfer for improving clustering tasks using external categorical evidence. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

P3 Athanasios Davvetas, Iraklis A Klampanos, Spiros Skiadopoulos, and Vangelis Karkaletsis. The effect of evidence transfer on latent feature relevance for clus-

tering. In *Informatics*, volume 6, page 17. Multidisciplinary Digital Publishing Institute, 2019.

P4 Athanasios Davvetas and Iraklis A. Klampanos. Unsupervised severe weather detection via joint representation learning over textual and weather data. In *CEUR Workshop Proceedings*, volume 2844, pages 83–87, 2020. URL http://ceur-ws.org/Vol-2844/ainst7.pdf

P5 Athanasios Davvetas, Iraklis A Klampanos, Spiros Skiadopoulos, and Vangelis Karkaletsis. Evidence transfer: learning improved representations according to external heterogeneous task outcomes. Accepted for publication in *ACM Transactions on Knowledge Discovery from Data*.

P6 Athanasios Davvetas, Iraklis A Klampanos, Spiros Skiadopoulos, and Vangelis Karkaletsis. Deep representation learning for information fusion and its applications. Currently under review in *ACM Computing Surveys*.

Resources. The investigation of the thesis hypothesis led in the following code and dataset resources:

- EviTraN method implementation and evaluation code:
  - https://github.com/davidath/evitrac
  - https://github.com/davidath/incomplete-evidence-transfer
- Code for experiments of EviTraN's theoretical interpretation:
  - https://github.com/davidath/evidence-transfer-interpret
- Severe weather evaluation code:
  - https://github.com/davidath/severe-weather-detect
- Severe weather dataset:
  - https://github.com/davidath/severe-weather-dataset

1.3 Organisation

The rest of the thesis is organised as follows:

Chapter 2 contains background and related concepts which are relevant throughout the paper. It also contains related work in deep representation learning for fusion, as well as, a comparison of evidence transfer to previous work.

Chapter 3 includes the introduction of the task at hand, deep learning frameworks, training objective and learning settings of evidence transfer

Chapter 4 contains the theoretical interpretation of the effects of evidence transfer in the latent space of an autoencoder

Chapter 5 contains experimental evaluation in artificial evaluation scenarios

Chapter 6 includes experimental evaluation in the realistic use case scenario of unsupervised severe weather detection.

Chapter 2

Related work

This chapter contains previous work related to EviTraN and the task at hand. It also includes background and related concepts, to aid reading comprehension of the thesis.

2.1 Background and Related Concepts

This section provides the necessary information required for the purposes of understanding the related work, as well as the majority of the thesis.

2.1.1 Deep Learning

Deep learning is a special case of machine learning, that utilises deep neural networks. The term *deep* refers to the depth of the involved neural networks. Neural networks consist of layers. The most frequent type of layers are fully-connected layers, also known as dense or hidden. A minimal version of a neural network is one consisting of an input layer, a fully-connected layer and an output layer. This configuration is a shallow neural network, as it only involves a single hidden layer. Input and output layers do not exist in implementation level, they are constructs that highlight the expectations of input and output, such as shape, data type, etc.

Let X be the input dataset of the shallow configuration and Y be its output. The output of the fully-connected layer $FC1$ (which is also the output of the neural network) is $y = f(Wx + b)$. Where $x \in X$ and $y \in Y$ are instances from input and output datasets respectively, W is the *weight matrix*, b is the *bias vector* and f is a non-linear function. The objective of training is to adjust W and b from random initialisations into appropriate values that will result in correctly predicting y instances, through observation of x. The dimensions of W and b are shaped by the number of nodes in the layer. A cost function guides the training process (or learning process)

in order to learn appropriate trainable parameters. In this example an appropriate cost function would be to minimise the misclassification rate of y. Weight matrix and bias vector are the trainable parameters of the neural network (which multiply with additional layers, e.g., $W1, \dots, WN$ and $b1, \dots, bN$). Most frequently activation functions f are non-linear functions, such as Rectified Linear Unit (ReLU) [40].

The amount of layers define the depth of the model. While the width of model, is the number of total nodes within the model (aggregation of each individual layer width). The selection of the model's hyperparameters, such as depth, width, activation functions, define the *capacity* or *complexity* of the model.

Feed forward neural networks is the most common archetype of deep neural networks, with the only other alternative being recurrent neural networks (RNN). The characteristic of feedforward neural networks is that the information passes from the input(s) to the output(s). In other words, the input data pass through the hidden layers to the output layer. RNN is a type of neural network which are typically connected with time-series data. This is due to RNNs involving connections along parts of a sequence, e.g., words of a sentence, before propagating information to successive layers [41]. Such connections, allow learning of temporal associations within the data. An example of recurrent neural networks is neural networks that make use of Long Short-Term Memory (LSTM) layers [42].

2.1.2 Convolutional Neural Networks

Convolutional neural networks are deep neural networks that make use of convolutional layers. They are specialised layers, that perform a different operation than that of the fully-connected layers. As the name suggests, they perform the *convolution* operation over their input. They utilise lower-dimensional matrices (compared to the original image, the size of the matrices is a hyperparameter) called *kernels* (filters is also an alternative name). Kernels traverse through the image to produce lower-dimensional feature maps from segments of the input. The objective of the above procedure is to acquire a more accurate representation of the input, by combining its most relevant parts. Figure 2.1 depicts the traversal process of convolutional layers.

Convolutional neural networks (CNNs) are particularly effective with datasets that display temporal or spatial associations. Through convolution of smaller parts of the image, convolutional layers highlight correlations between neighbouring parts, e.g., pixels of an image. Unlike fully-connected layers, where each of the input features are involved individually. Furthermore, kernels can traverse through multiple channels at once. For that reason, CNNs are very active in various computer vision applications, since these applications involve coloured images that consist of RGB channels.

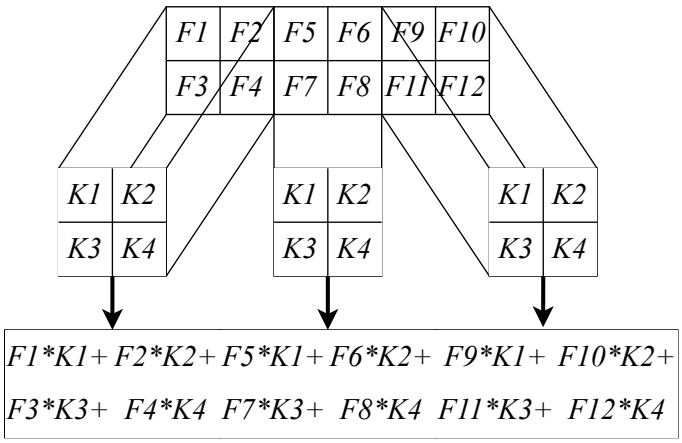


Figure 2.1: Example of a kernel traversing through a gridded input. The figure is inspired from a similar example in deep learning book [1].

One may classify the different CNN architectures based on the amount of dimensions that convolution operates on (Figure 2.4 depicts a generic CNN variation with 2D convolutional layers). To exploit datasets with temporal correlations, such as time-series data, the use of Conv1D layers (convolution over 1 dimension) is preferred. While to exploit datasets with spatial correlations, such as images or other gridded data, the use of Conv2D layers (convolution over 2 dimensions) is preferred. Exploitation of spatio-temporal correlations is feasible through an aggregation of Conv1D and Conv2D feature maps. However, implementations of Conv3D layers also exist.

2.1.3 Autoencoders

From a high-level perspective autoencoders can be seen as composition of two procedures, the *Encoding* and *Decoding* procedures. Subsequently, two components named: Encoder and Decoder perform the aforementioned procedures. The encoder transforms (encodes) the input into representations squeezed through a bottleneck, while the decoder transforms (decodes) a representation back to its original form. One may perceive the autoencoder as an end-to-end neural network or as a composition of two individual sub-networks (during this case, it is more frequent to perceive encoder and decoder as stochastic mappings instead of deterministic functions [1]). An overview of a generic autoencoder is depicted in Figure 2.2.

In general the depth of each subnetwork is arbitrary. However, it is generally accepted that deep encoders and decoders bear more advantages than their shallow counterparts [1]. Furthermore, Hinton and Salakhutdinov [43] showed experimentally that deep autoencoders lead to greater compression than their shallower counterparts. Despite the depth, autoencoders require a plethora of choices regarding the hyper-

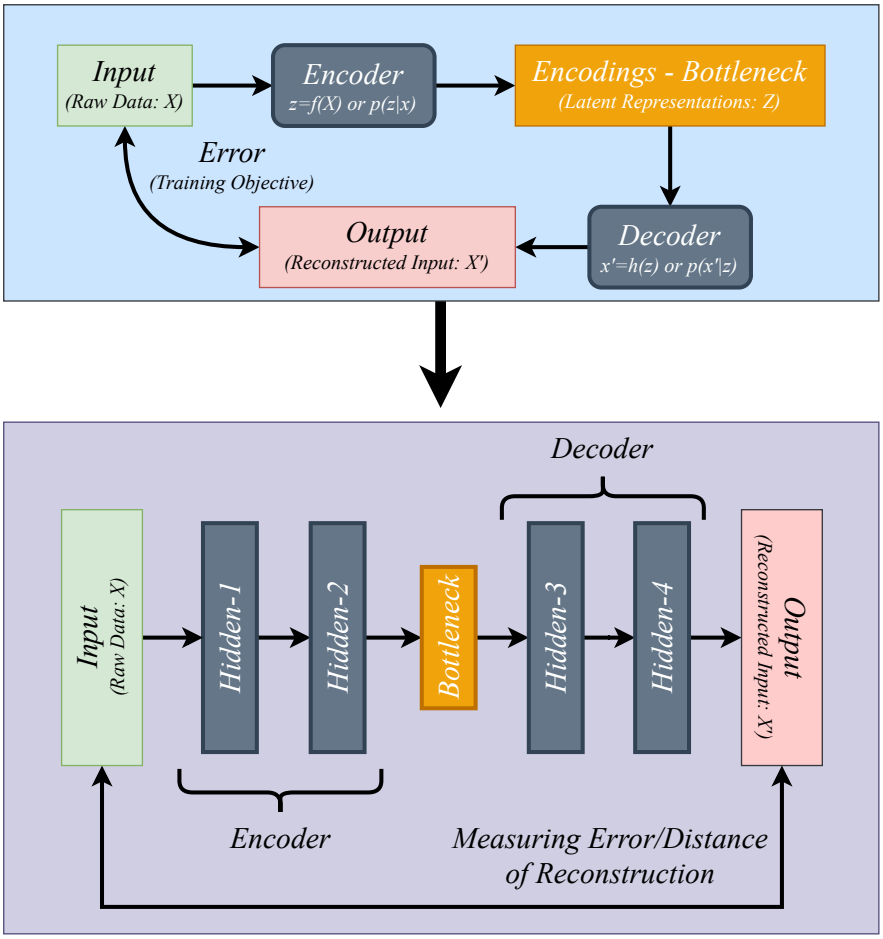


Figure 2.2: Overview of autoencoders and a corresponding generic neural network architecture.

parameters such as activation functions, layer width, training objective, layer choice, etc. From such choices, variations of autoencoders arise such as Convolutional or Recurrent Autoencoders [44, 45].

One of the common misconceptions around the autoencoders is that they act as identity functions. Meaning that the autoencoder tries to perfectly reconstruct the input in its output. However, autoencoders that act as identity functions suggest that the bottleneck layer does not introduce required restrictions for the encoder to learn meaningful representations. Although there is a large discussion regarding the properties of "good" representations (more in Chapter 5), one of the properties that suggest meaningful representation learning is the ability of the encoder to generalise, such as the example shown in Figure 2.3. To this end, to avoid lazy training of a large capacity autoencoder to act as an identity function additional restrictions in the form of regularisation are introduced. Well known regularisations are: Denoising Autoencoders [46], Sparse Autoencoders [47], Contractive Autoencoders [48], etc.

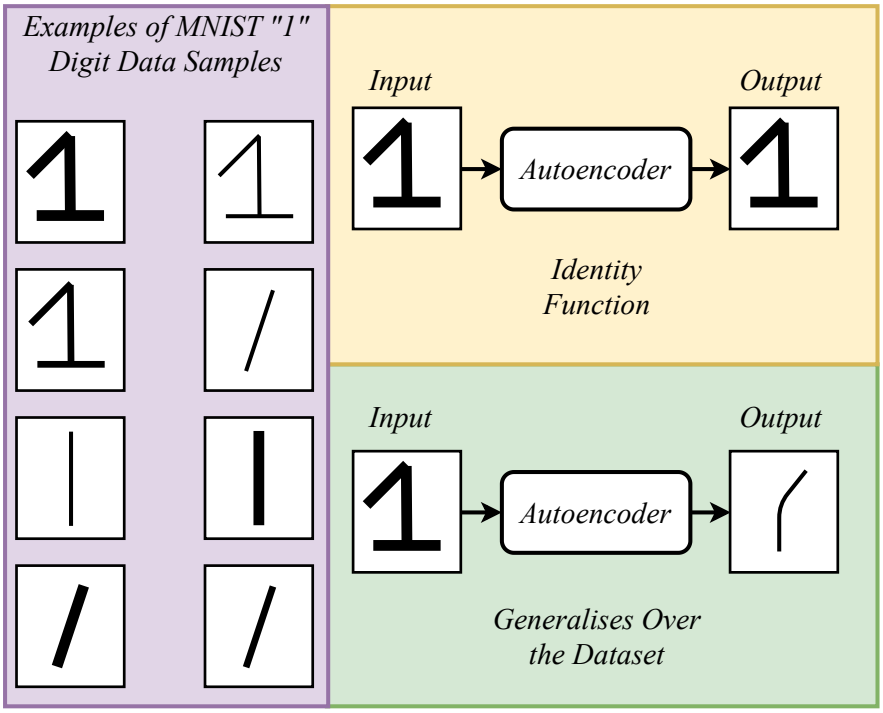


Figure 2.3: Example of good reconstruction with recreated data samples from MNIST [2]. The indication of learning generalised representations does not come from perfect reconstruction but rather than being able to capture multiple variations of the same concept, such as tilted one digits and regular one digits.

Another misconception is that they are the deep learning equivalent of PCA. Principal Component Analysis [49] is a traditional machine learning method that learns the principal components of a data collection through linear operations. Unlike PCA, autoencoders are able to learn representations from non-linear data, due to the non-linearities involved in the activation functions of their neural network architecture.

Denoising autoencoders are based on a very simple but rather effective idea. They introduce noise in the input data. The decoder then, must learn to reconstruct the original input without the introduced noise. Therefore, the decoder should learn to "remove" the noise found in the input, i.e., to denoise the input. The introduced noise may vary from dropout[1], random values drawn from a well-known distribution, e.g., normal or uniform, salt and pepper[2], etc. Bengio et al. [51] have proven that denoising autoencoders are generative models, meaning that they learn latent variables involved

[1]Dropout is a stochastic procedure that transforms a random portion of the features within each sample in a batch into zero value [50].

[2]Salt and pepper noise for deep neural networks is a stochastic procedure that transforms a random portion of the features within each sample in a batch into either zero or one value. The name refers to the end product bearing similarities to image that has salt and pepper all over it.

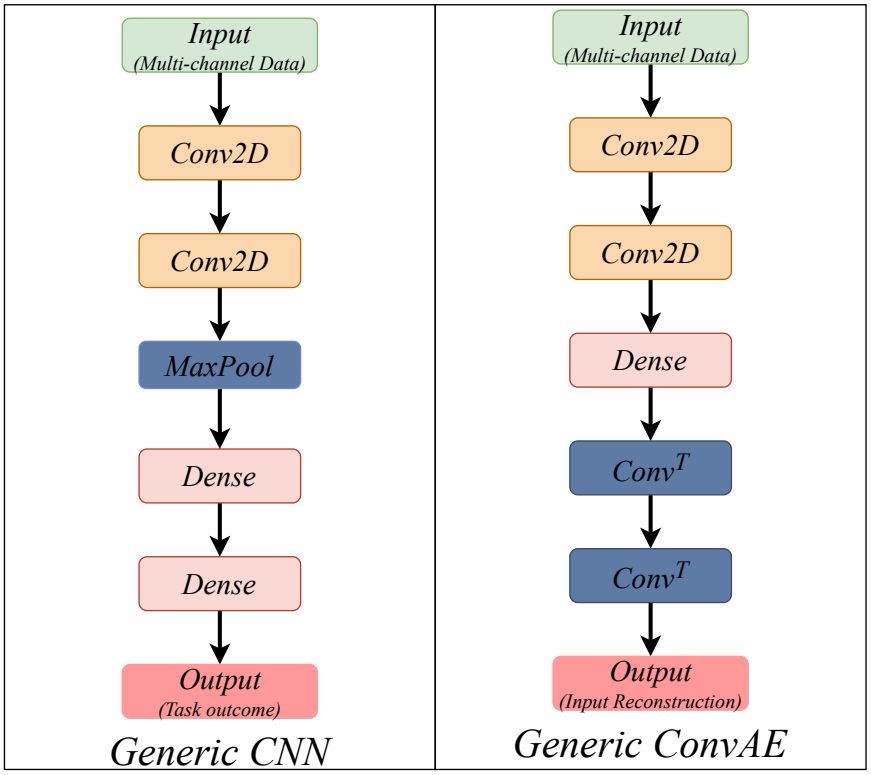


Figure 2.4: Neural network architecture of a generic CNN and a generic convolutional autoencoder.

in the generative process, that can be repurposed in order to manifest synthetic data instances.

Convolutional autoencoders (depicted in Figure 2.4) are a variation of the original autoencoder framework, which involve convolutional layers. The encoder utilises convolutional layers in order to exploit the correlations of the dataset. A fully-connected layer aggregated all feature maps of the last convolutional layer, which acts as a bottleneck. On the other hand, the decoder utilises the inverse operation, performed by transpose convolutional layers also known as deconvolutional layers. Convolutional autoencoders increase the complexity of the model. In turn, it allows for learning of meaningful representations from gridded or time-series data.

2.1.4 Batch normalisation

Batch normalisation is a normalisation method developed by Ioffe and Szegedy [52]. The concept of batch normalisation is to adjust the centre and scale of each batch. The motivation of batch normalisation was to deal with internal covariate shift, which is a phenomenon of internal layers shifting their means and variance, during training. Batch normalisation has been implemented into a layer structure.

Batch normalisation layers with trainable parameters β and γ produce output y based on $x = (x^{(1)}, \ldots, x^{(d)})$, where d is the number of dimensions. Each dimension is normalised separately. The output y is shown in Equation 2.1. The result of batch normalisation is an output of zero mean and unit variance. A stability term ϵ may be added to the variance in computation of $\hat{x}^{(k)}$. Scaling and shifiting performed by y allows for restoring of representation power of the network. Batch normalisation allows for faster and more stable optimisation of the neural network. It is also possible to increase the effectiveness of the model.

$$\begin{aligned} E[x] &= \frac{1}{N}\sum_{i=1}^{N} x_i \\ Var[x] &= \frac{1}{N}\sum_{i=1}^{N}(x_i - E[x]) \\ \hat{x}_i^{(k)} &= \frac{x_i^{(k)} - E[x]^{(k)}}{\sqrt{Var[x^{(k)}]}} \text{ with } k \in [1, d] \\ y_i^{(k)} &= \gamma^{(k)}\hat{x}_i^{(k)} + \beta^{(k)} \end{aligned} \tag{2.1}$$

The above notation and terminology is consistent to that found in the work of Ioffe and Szegedy [52].

2.1.5 Generative Vs. Discriminative Models

A high-level definition of generative models, as the name suggests, can be: probabilistic models that learn to generate samples from an observable data distribution. In the context of probabilistic classification, Ng and Jordan [53] mention that: "*Generative classifiers learn a model of the joint probability $p(x, y)$, of the inputs x and label y*". In a more general context, one may consider a generative model to be a probabilistic model that learns a joint probability $p_\theta(x, y)$ of observed random variable x and target random variable y. Generative models that involve deep neural networks in the process of learning said joint probability, are characterised as *deep*. In that case, parameters θ represent the trainable parameter of the neural network.

In the context of representation learning, it is often assumed that the data generation process involves some unobserved latent variables, where the generated data instances are conditional distribution that involve these latent variables [15]. Generative models are most frequently trained without supervision. Since, they aim to learn a joint probability between the input and a target variable.

A high-level definition of discriminative models, as the name suggests, can be: probabilistic models that learn to "discriminate" data samples into various sets that often represent semantic groups. Ng and Jordan [53] in the context of probabilistic

classification mention that: "*Discriminative classifiers model the posterior $p(y|x)$ directly or learn a direct map from inputs x to the class labels*". In a more general context, one may consider a discriminative model to be a probabilistic model that learns posterior $p(y|x)$ of target random variable y given observable random variable x. Discriminative models utilise labelled examples in order to learn a posterior between input and target variable (i.e., most often are trained with supervision).

2.1.6 Representation Learning

Representation learning is at the same time, an explicit machine learning task and an implicit, inherent process of deep neural networks. Deep neural networks learn complex tasks through learning of simpler representations in each intermediate layer. As layers sequentially pass information to the next, all intermediate representations are aggregated in the final output layer, which is typically the prediction layer. The objective of representation learning is to learn alternative representations of raw observations. In other words, it aims to transform the raw observations into an alternative space. The learning of latent representations is motivated by their repurpose towards an ultimate objective. Examples of such objectives are: learning of a more complex task based on less complex versions of the raw observations or extracting insight regarding characteristics or data distribution of the raw observations.

The inherent learning of representations in deep neural networks is an implicit procedure. Deep neural networks do not explicitly strive to learn alternative representations. The process of learning simpler representations is a side effect from the process of learning a down-stream task. On the other hand, the need for explicit representation may rise. Explicit representation learning aims at various objectives such as compression, dimensionality reduction, generation, clustering or studying the characteristics of the dataset in a more compact feature space. Implicit or explicit representation learning are distinguishable based on their training objective. Conventionally, implicit representation learning is part of supervised learning, with the training objective aiming to reduce the misclassification rate of the classifier. Explicit representation learning is part of unsupervised learning, with the training objective aiming to learn unobserved variables that are involved in the generative process of the dataset.

For that reason, implicit representation learning produces task-specific representations. Its learning process requires an adequate amount of labelled instances. On the other hand, explicit representation learning does not require labelled instances. In addition, it produces representations that are transferable to other tasks. However, evaluating the performance of explicit representation learning is more complex than implicit representation learning (more regarding the evaluation in Chapter 5).

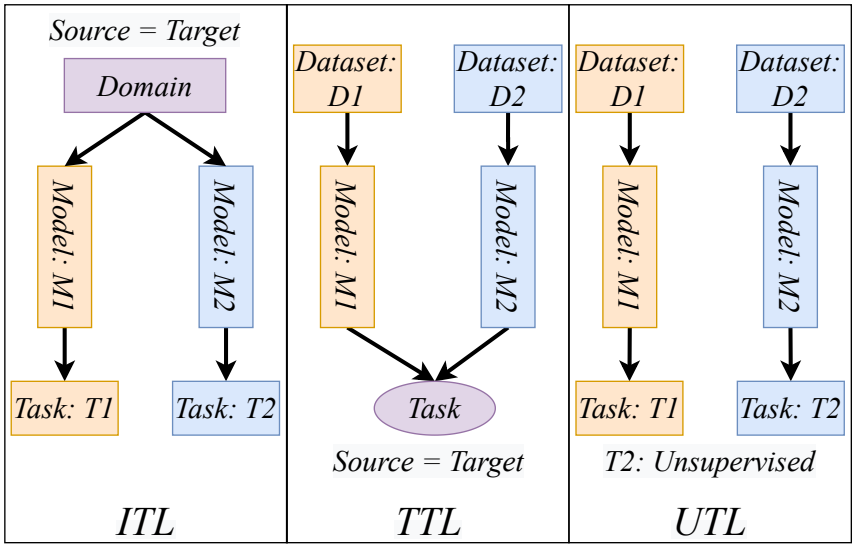


Figure 2.5: Types of transfer learning according to Pan and Yang [3].

2.1.7 Transfer Learning

Transfer learning is a learning archetype of machine learning. It is based on the idea that the products of training, such as trainable parameters or effective hyperparameters, should be transferable to other models that perform similar tasks. In other words, transfer learning encourages the repurposing and adjustment of already trained models for new tasks. This property of transfer learning allows training and testing data to be drawn from different data distributions [3]. For example, the training could involve images from cars, while the task at hand could be to identify trucks within images. Adjusting of pre-trained models reduces optimisation errors from random initialisations. For instance, minimising a cost from random initialisation, depending on the seed, may lead to a local minimum. However, initialisation from a good performing pre-trained model would avoid such errors. In addition, it reduces the use of resources and effort put into adjustment of hyperparameters.

According to Pan and Yang [3], transfer learning can be classified into three categories: (i) Inductive Transfer Learning (ITL), (ii) Transductive Transfer Learning (TTL) and (iii) Unsupervised Transfer Learning (UTL) (also shown in Figure 2.5). ITL involves the transfer of knowledge from dissimilar tasks, which are outcomes based on the observation of a common feature space. TTL involves the transfer of knowledge from similar tasks, which are outcomes based on the observation of dissimilar feature spaces. UTL involves the transfer of knowledge from dissimilar tasks, which are outcomes based on the observation of dissimilar feature spaces. In addition, most often UTL consists of transferring knowledge from a source task to an unsupervised target task.

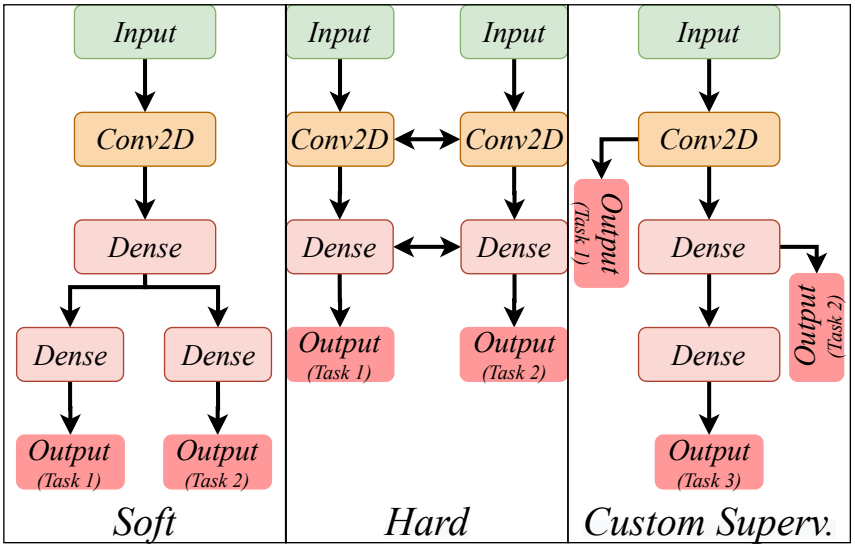


Figure 2.6: Categories of multi-task learning according to Elliot Meyerson [4]. Visualisation of universal representations approach is omitted, since it involves model adjustment in a lower-level (layer level), rather than specific architectural decisions.

2.1.8 Multi-task Learning

Multi-task learning (MTL) is essentially inductive transfer learning. However, in MTL the objective is often to learn all tasks at the same time, rather than transferring knowledge across one or multiple tasks. In practice, MTL aims to combine relevant knowledge found across similar tasks, in order to increase the overall perception of the model or to produce a more relevant feature space. For that reason, it frequently leads to increased performance in all tasks.

According to Elliot Meyerson [4], MTL can be classified into four categories: (i) classical, (ii) column-based, (iii) supervision at custom depths and (iv) universal representation (also shown in Figure 2.6). Classical approach consists of a single encoder model with task specific decoder streams. The decoders produce task outcomes based on task-invariant features produced by the encoder model (also known as *hard parameter sharing*). Column-based consists of individual task specific streams which are typically trained by sharing their parameters (each streams can be seen as a column, also known as *soft parameter sharing*). Supervision at custom depths involves of a single end-to-end model. However, it consists of multiple outputs which are outcomes based on different depth model levels. Universal representation consists of adapting the layers with task-specific parameters. Since the name universal representations can be confusing in the context of studying the related work of representation learning for information fusion, from now on this approach will be referred to as *domain specific adaptation of layers*.

2.1.9 Weak Supervision

A frequent classification of machine learning and as an extent, deep learning methods, is according to their types of learning. Most famous types are: *Supervised Learning* and *Unsupervised Learning*. According to Goodfellow et al. [1]: "*Supervised learning algorithms experience a dataset containing features, but each example is also associated with a label or target*", while "*Unsupervised learning algorithms experience a dataset containing many features, then learn useful properties of the structure of this dataset*".

Weak supervision can be seen as a middle ground between full supervision and no supervision (unsupervised). Weak supervision involves cases with limited supervision. Limitations may include noisy labels, incomplete correspondence or no correspondence at all. According to Zhou [54], weak supervision can be classified as: (i) Incomplete, (ii) Inexact and (iii) Inaccurate. Incomplete supervision involves datasets that consists mostly of unlabelled data and a small portion of labelled data. Inexact supervision involves bags of instances instead of individually labelled instances. Inaccurate supervision involves datasets with inaccuracies within its associated labelset. Examples of inaccuracies are noisy, non-corresponding or irrelevant labels.

2.1.10 Information Theory

Information theory includes fundamental concepts around the act of transmitting information from a source to a receiver. According to lexico.com [55], the definition of information theory is: "*The mathematical study of the coding of information in the form of sequences of symbols, impulses, etc. and of how rapidly such information can be transmitted, for example through computer circuits or telecommunications channels*". While according to Cover and Thomas [56]: "*Information theory answers two fundamental questions in communication theory: What is the ultimate data compression (answer: the entropy H), and what is the ultimate transmission rate of communication (answer: the channel capacity C)*".

These fundamental concepts are presented as follows. Entropy quantifies the uncertainty of a random variable [56] (as shown in Equation 2.2). Random variables that follow a uniform distribution have high entropy. Consider the example of rolling a fair dice. Trying to predict the outcome of the dice is completely random, since all outcomes are equivalently probable.

$$H(X) = -\sum_{x \in X} p(x) \log p(x) \tag{2.2}$$

One may desire to measure the entropy of a random variable based on observation

of another random variable. This is known as measuring the conditional entropy of two random variables (as shown in Equation 2.4).

$$\begin{aligned} P(x|y) &= \frac{P(x)P(y|x)}{P(y)} = \frac{P(x,y)}{P(y)} \rightarrow P(x,y) \quad = P(x|y)P(y) \\ P(y|x) &= \frac{P(y)P(x|y)}{P(x)} = \frac{P(x,y)}{P(x)} \rightarrow P(x,y) \quad = P(y|x)P(x) \end{aligned} \tag{2.3}$$

$$\begin{aligned} H(Y|X) &= -\sum_{(x,y)} p(x,y) \log p(y|x) \\ \xrightarrow{(2.3)} H(Y|X) &= -\sum_{(x,y)} p(x,y) \log \frac{p(x,y)}{p(x)} \end{aligned} \tag{2.4}$$

Equation 2.3 involves a transformation based on Bayes' rule. Bayes' rule is a formula from the domain of probability which in combination with logarithmic properties are frequently utilised to simplify concepts in information theory. Logarithmic properties that are frequently used are show in Equation 2.5.

$$\begin{aligned} \log_a(x * y) &= log_a x + log_a y \\ \log_a \left(\frac{x}{y}\right) &= log_a x - log_a y \end{aligned} \tag{2.5}$$

Mutual information (as shown in Equation 2.6) quantifies common information found in two random variables. Cover and Thomas [56] defines it as: "*The mutual information I (X; Y) is a measure of the dependence between the two random variables. It is symmetric in X and Y and always non-negative and is equal to zero if and only if X and Y are independent*".

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ I(Y;X) &= H(Y) - H(Y|X) \end{aligned} \tag{2.6}$$

The above notation and terminology is consistent with that found in the work of Cover and Thomas [56].

2.1.11 Data Fusion

The combination of various data sources for the purposes of producing more efficient results in an appropriate goal, is the concept that most frequently describes *data fusion*. White [57] defined data fusion as "*a process dealing with the association, correlation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete and timely assessments of situations and threats as well as their significance*". In the definition

of White [57], not only data, but also information comes up as a component involved in the combination. This idea popularized the term *information fusion*.

The combination of data products derived from raw observations is the concept of information fusion. Data products may involve alternative features or decisions, and therefore imply a higher semantic level. As indicated by Foo and Ng [58], the focus of research shifted from semantically low-level information fusion to high-level information fusion. According to Boström et al. [59]: "*Information fusion is the study of efficient methods for automatically or semi-automatically transforming information from different sources and different points in time into a representation that provides effective support for human or automated decision making*". Comparison of these two definitions implies that they involve similar concepts.

To define information fusion as a research field, Boström et al. [59] in addition to information fusion definitions also considered and discussed definitions of data fusion. Furthermore, Boström et al. [59] explicitly mention that in practice these two terms are sometimes synonyms. At the same time, Steinberg et al. [60] mentions that due to lack of a general term, the term *data* also covers its subsets, e.g., information, knowledge, etc.

Another term that is closely related to data fusion is *data alignment*. Data alignment is the transformation of diverse types of data into a common frame. A well-known case of data alignment is adjustment of sensors into a common coordinate system [61]. Despite, data alignment may also involve non-diverse data. Data alignment aims to find means of transition, such as rules or tables, between data sources. The objective of the transition is to create a unified space based on which all data sources can be referenced.

Another concept that frequently appears in data fusion is redundancy. Redundancy is the propagation of irrelevant information that may create noise or errors. The redundancy may increment with the introduction of additional data sources. In telecommunications incremental redundancy has been defined as the transmission of "*increments of redundant bits after errors are observed*" [62]. The downsides of involving redundant data sources are: that it may require additional computational resources to compensate to the large number of features [63] or it might disrupt the performance of various methods (if not explicitly dealt with) [64].

In machine learning the most frequent cases of incremental redundancy are: involving data features already present in other sources, which are irrelevant for the task at hand or involving non-complementary data sources. For a better understanding of the difference between the two cases, consider the following example from the domain of computer vision. Figure 2.7, depicts five versions of the same image. The coloured version, each individual red, green and blue versions and grey scale version. All versions depict an object with white background. Let the task at hand be iden-

Figure 2.7: Example of redundant images. The first three images depict the red, green and blue version of the fourth image that is the coloured version. All images depict an object (a folder) over a white background. The red, green and blue versions of the coloured image propagate the redundant information of background, which is not relevant to the object detection task. The greyscale version depicted in the fifth image is also redundant to the coloured version, since the coloured version also depicts texture.

tifying the object within the image. Fusing individual red, green and blue channels introduces redundant features, i.e., background features. Background features are redundant, since they do not provide any insight for the task at hand. At the same time, greyscale version is redundant to the coloured counterpart, since the coloured image involves additional information by the combination of individual channels, which is texture of the object.

2.1.12 Evaluation Criteria of Fusion Methods

Meng et al. [5] proposed a list of criteria based on which the performance of fusion methods is evaluated[3]:

Efficiency.

> "*Efficiency is used to evaluate if a data fusion model makes use of resources economically ... The efficiency reflected by execution time should be evaluated to demonstrate model advance through comparison with other models*".
>
> An alternative to dealing with multiple data sources individually, is to utilise the data products of their combination in order to reduce the required processing power. Treating each data source individually, depending on the volume of available data features or available data samples may require excessive amounts of available processing power. Such resources may either not be available (e.g., insufficient memory) or they are possibly reduced through sharing of multiple individuals (e.g., memory is sufficient but per user restrictions are insufficient for the task at hand). Therefore, the economical use of resources is frequently a desirable criterion during information fusion.

Quality.

[3] Text in italics and quotes indicates definitions from Meng et al. [5]

"*What is the direct impact on a fusion algorithm? To which degree does the model improve the information accuracy? Quality is the core of data fusion*".

The process of fusing a variety of data sources directs towards an end-goal. As a result, quantitative evaluation metrics represent the quality of the fusion process. The evaluation metrics vary, depending on the end-goal. For example, consider the process of fusing features from multiple data sources in order to achieve better identification of objects. A straightforward evaluation metric of the fusion quality could be the accuracy of identified objects. Meaning that, increased performance in the process of quality should lead to increased performance in correctly identifying objects.

Stability.

"*Stability is used to evaluate a fusion model's ability to keep working well in a stable manner in different situations*".

Depending on the end-goal of the fusion process, the stability of the method may be critical. For example, an unstable information fusion process in medical applications may be costly and therefore, considered critical (resources: time, computational time, materials, etc. – direct impact, such as physical implications).

Robustness.

"*Robustness evaluates the strength of a fusion model to resist disturbance. When an underlying environment is changed, fusion quality should be ensured*".

Combining multiple information sources may yield challenges. Various information sources may often contain noise or lead to aggregated noise through their fusion. The fusion process should be able to deal with such noise, either inherent or aggregated, in order to ensure its operability.

Extensibility.

"*Extensibility means that a data fusion model can be easily further improved and widely used in many situations ... Extensibility is a valuable feature for wide adoption of the data fusion model in practice*".

Mechanisms within applications are often overlapping. For example in computer vision, object detection mechanism is reused by many applications such as semantic segmentation or pose estimation. The process of fusing information sources is no exception. Modular mechanisms and methods found in information fusion should be generalised in order to be adopted in other domains or applications.

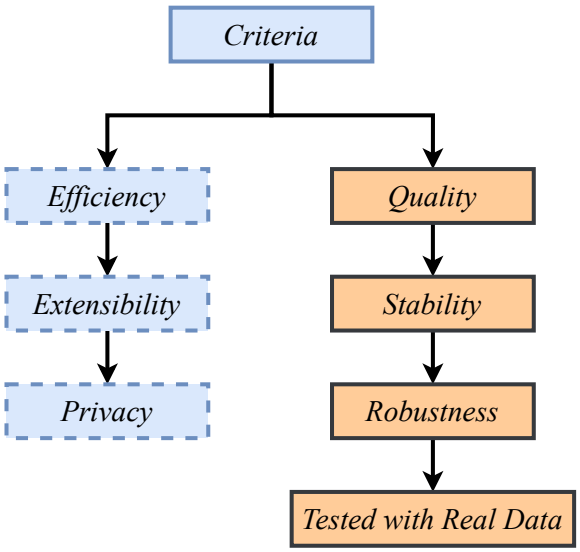


Figure 2.8: Overview of criteria proposed by Meng et al. [5]. Boxes on the left side depicted with dashed lines represent criteria which are less generally applicable.

Privacy.

"*In some application scenarios, data used for fusion may be sensitive and private, which induces security requirements on the fusion model*".

Information fusion in certain applications may require ensurance of privacy. Fusion of personal or sensitive data should be handled with caution within the fusion process.

Tested with real world data sets.

"*In a solid research, experiments are dispensable to testify the performance of a model prove its effectiveness and show its advantages*".

The inspiration to perform fusion of multiple data sources usually comes from real world experience. Artificial or toy[4] datasets which are used for educational or other purposes are not as complex as real world datasets. Evaluation with real world datasets is a critical evaluation criterion, since it relays the real world impact of the fusion process.

Figure 2.8 visualises the criteria proposed by Meng et al. [5]. In this day and age resource limitation are generally not so frequent due to cloud computing. Cloud

[4]It is common to refer to datasets that are simple or easy to solve as toy. An example of such dataset is the Iris Plant Dataset [65, 66]. Iris plant dataset is one of the most famous datasets in pattern recognition. However, modern methods and algorithms achieve great performance with low amounts of effort.

services provide access to resources that fit the needs of the problem. Due to that, *Efficiency* criterion is often neglected during the fusion process. However, relying on such cloud services may be costly. Therefore, being not a sustainable solution in the long term. To this end, using the available resources efficiently is still a relevant issue to consider. Yet, compared to other criteria, considering efficiency of resources during the fusion process is less relevant than in previous years.

Incompliance to *Privacy* criterion, despite being unethical and potentially illegal (e.g., medical applications), is also critical to the development of information fusion methods. Preserving the privacy of the users involved in the data acquisition (or other parts of fusion process) is a major stepping stone towards inspiring trust. A relation of trust between involved users and scientists not only allows for major advancements, but also inspires the sharing of such information. For example, data fusion applications that detect lung diseases are not feasible without the acquisition of relevant data. An environment of distrust towards the fusion method may encourage users to not share their sensitive information and therefore restrict advancements in this area, which has a direct societal impact. Yet, dealing with sensitive or private data is not the norm. Most domains deal with publicly available or data that do not expose information or activity of users. Thus, concluding that privacy is not a generalisable criterion for evaluating the fusion process.

Aiming to develop extensible data fusion methods promotes reuse of effective and stable modules or methods in multiple domains or applications. However, the implementation or design of method typically aims to a specific application, task or domain and thus, its extensibility to other applications, task or domains may not be practical. This could be either due to underlying assumptions not being applicable to the new domain or either due to the two domains being unrelated and thus lacking similar concepts. Therefore, *Extensibility* criterion is not a general evaluation criterion. The remaining criteria such as *Quality, Stability, Robustness* and *Tested with Real Data* are more broadly applicable.

2.1.13 Classification of Information Fusion Methods

To study the relevance of EviTraN compared to previous work in learning representations for the purposes of fusing data sources, a classification of methods should be selected. In this study, the classification of Dai and Khorram [6] and Luo et al. [7] which is based on the abstraction level of the data, is adopted. The abstraction levels, also shown in Figure 2.9, are:

Raw-data/signal level. As the name suggests, signal-level fusion involves raw signals or observations. The fusion at this level aims to produce a combined signal

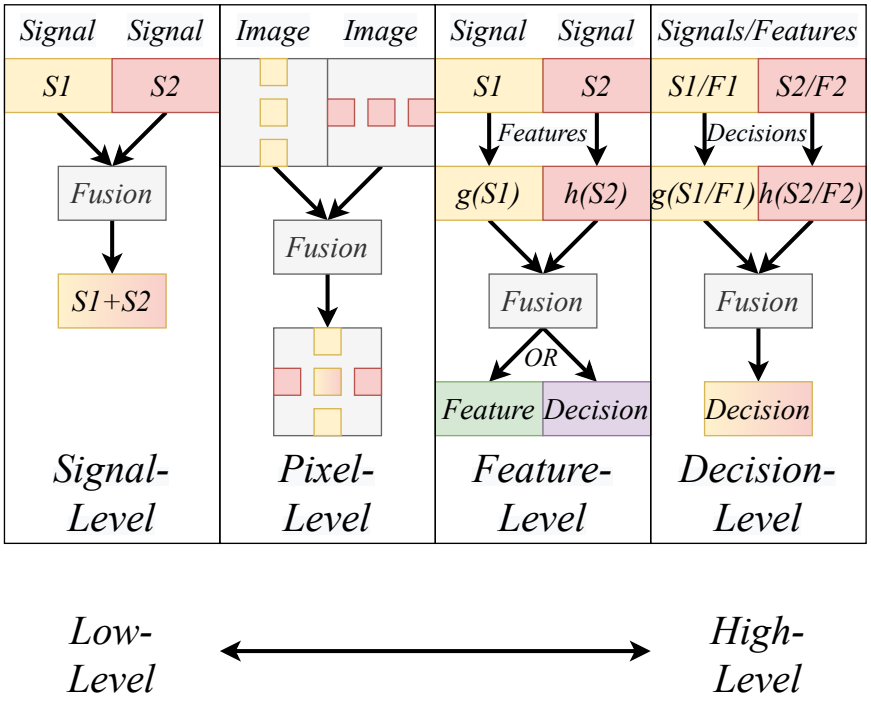


Figure 2.9: Abstraction level of data, according to Dai and Khorram [6] and Luo et al. [7].

of increased quality or of reduced redundancy. The combination operation, depending on the interpretation can be a linear, probabilistic or other appropriate operations.

Pixel level. This level involves combination of pixels, it consists of computer vision applications, e.g., object detection.

Feature/intermediate level. This level involves characteristics extracted from raw observations. The fusion at this level aims to produce a new set of refined features or decision level information such as task outcomes. Feature-level fusion is popular in deep learning.

Decision/high level. This level involves decision level information, such as task outcomes. The fusion at this level aims to produce new refined decisions.

2.2 Representation Learning for Signal-Level Fusion

Compared to the other two levels of fusion, signal-level fusion can be considered as the less sophisticated one. As it directly involves raw observations, which are un-

processed, this may result in the introduction of a plethora of features which may not be relevant. A comparative study conducted by Cui et al. [67], involved fusion of two modalities: colour and depth images from an RGB-D sensor. The evaluation process of the fusion involved the task of face recognition. During their study, signal-level fusion yielded the worse performance than a respective single source solution that involved only RGB images for one of the datasets. At the same time, it barely performed better in the other dataset. The above study is merely an indication of signal-level not only being less sophisticated but also being less efficient than the alternative levels. However, to draw definitive conclusions more investigation is required.

CNNs for signal-level fusion is the most representative implicit representation learning fusion scheme, while joint latent space is the most representative explicit representation learning fusion scheme for signal-level fusion. Data alignment at signal-level can be utilised both for implicit and explicit representation learning.

2.2.1 Data Alignment for Signal-Level Fusion

Signal-level fusion involves unprocessed signals. To deal with any heterogeneities, it is frequent to perform a data alignment procedure as a pre-processing method. The approach of fusing signals by aligning them in a common space, is common in applications like database schema matching [68] or semantic-based fusion [69]. The alignment process typically does not involve deep learning. Representation learning of aligned sources involves only a single combined source. It can be either implicit or explicit.

EviNets [70] is an implicit representation learning method for question answering, which receives an aligned data source as input. The involved data types are text and specialised knowledge base triples from open domain question answering. To align the data, they first automatically retrieve the entities from relevant text. Then, they transform both entities from the text and the knowledge base triples into embeddings through Bag Of Words [71]. EviNets select the most relevant embeddings based on a task appropriate scoring function.

Amorim et al. [72] proposed an alignment method to deal with heterogeneity of data extracted from social media. They perform an automatic alignment by involving a pre-trained Mask-RCNN model [73]. The authors extract predicted labels by feeding images to the pre-trained model. Predictions over a certain threshold, which are considered as successful, are concatenated at the end of the tweet. An autoencoder produces latent representations based on the aligned embeddings from image and text data. The authors repurpose the learned representations for the task of novelty detection. The above process is presented in Algorithm 1.

Algorithm 1: Automatic data alignment of image and text signals for novelty detection in social media, proposed by Amorim et al. [72].

Data: I: images extracted from tweets, T: extracted text from tweets.
Result: Alignment of image and text signals from social media.

```
forall images i in I do
    /* Extract labels from pre-trained Mask-RCNN */
    C = Mask-RCNN.predict(i);
    forall c in C do
        if class assignment probability of c over threshold then
            keep c in C collection;
        else
            remove c from C;
        end
    end
    Concatenate C at the end of respective t ∈ T text;
end
Extract latent features Z, from autoencoder trained on augmented T text
 collection;
Find novel instances from novelty detection algorithm with input Z;
```

2.2.2 CNNs for Signal-Level Fusion.

CNNs serve as a connection between the discrepancy created by involving low-level information, such as signals, into high-level implicit representation learning objectives. CNNs are able to produce refined features which are able to deal with that semantic discrepancy. The input of these CNNs, consists of individual data sources, introduced as channels. The combination takes place in the initial convolutional layer, which is the one closest to the input. Using CNNs for signal-level fusion,typically involves implicit representation learning.

Full Patch Labelling by Learned Upsampling (CNN-FPL) [74] is a CNN architecture that produces class labelled maps for the task of land cover classification. To predict the class labelled maps, the proposed architecture receives a multi-channel input that consists of near infrared, green, red and normalised digital surface channels. The main concept of the architecture is similar to that of convolutional autoencoder. The input is downscaled with the use of sequential convolutional layers and is then upscaled to the original shape through deconvolutional layers.

Xu et al. [75] proposed a 3D CNN for the extraction of features from three-dimensional grids extracted from LiDAR data. The fusion of three-dimensional grids aims to perform the task of land cover classification which guides the training of the extractor. A 3D CNN is also propose by Yun-Mei et al. [76], in order to deal with heterogeneous annotation of electroencephalogram (EEG) data, as a result of varying

medical equipment. The proposed 3D CNN architecture is able to extract spatio-temporal features from heterogeneously annotated EEG data, in order to predict EEG abnormalities.

2.2.3 Joint Latent Space Frameworks for Signal-Level Fusion

Joint latent space frameworks perform automatic alignment and extraction of correlations between data sources by enforcing a joint latent space. The enforced joint latent space, acts as a bottleneck that compresses the most relevant information from all involved data sources. Joint latent space for signal-level fusion, typically involves explicit representation learning and autoencoding frameworks. Also, they typically involve multiple input and output streams. Since explicit representation learning does not involve decisions (high-level information), the aggregation operations are usually less sophisticated, e.g., concatenation. Although, more sophisticated operations such as convolution may be utilised, if higher model complexity is required for a certain application. In addition, convolutional layers is an efficient way of reducing the input and output streams into single streams. However, it differs from previous CNN approach, since the representation learning objective is explicit (e.g., reconstruction error).

Correlation Neural Networks (CorrNet) [8] (as shown in Figure 2.10[5]) is a joint latent space framework that mostly involves dual modalities. CorrNet is a dual stream autoencoder with an enforced bottleneck, that involves the inner-most layers of both streams. It involves a composite training objective of three terms. Reconstruction error of input and output pairs, reconstruction error of input and output pairs with inverse encodings (encode with one modal – use decoder of the other modal) and correlation between encoding pairs.

Multimodal Autoencoder (MMAE) [9] (as shown in Figure 2.11 (a)) is an autoencoder that involves multiple modalities as a concatenated single stream. The motivation behind development of MMAE, is to fill missing data from modalities by utilising the ability of autoencoders to reconstruct input. To do so, MMAE first is trained with complete modalities. In a later step, the authors remove values from modalities by using value -1 to represent a missing value. The error of reconstruction is measured with cross entropy. An aggregation of all reconstruction errors for each modality is considered as the final composite objective.

In previous work, we proposed a convolutional autoencoder (as shown in Figure 2.11 (b)) that involves multiple pressure levels of a weather variable (500, 700 and

[5]This figure and the following figures in Chapter 3 that depict neural network architectures from related work, depict simpler architectures than the ones found in the original work. The simpler architectures aim to represent the main concept of the framework, without unnecessary repetition of layers. The actual number of layers may vary from the original work.

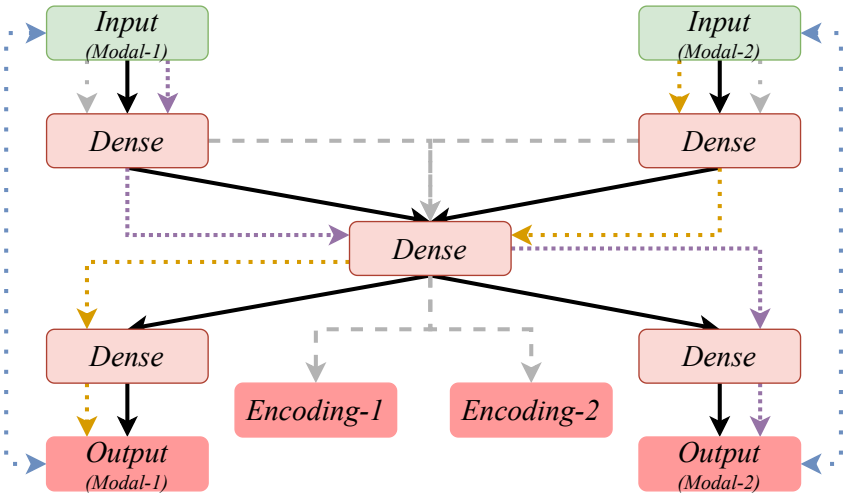


Figure 2.10: Neural network architecture of Correlation Neural Networks (CorrNet) [8]. The different arrows represent the information flow of the three different training objectives.

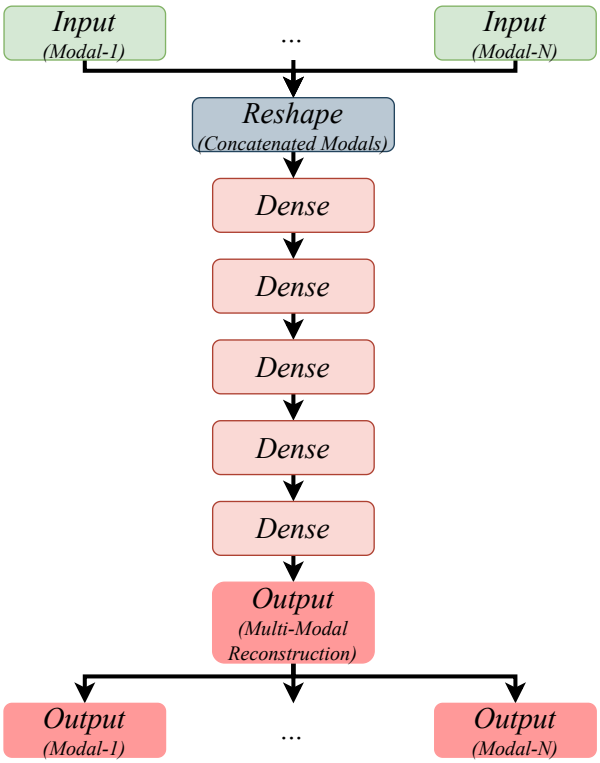
900 hPa of GHT, more regarding aspects of weather data in Chapter 6) [10]. The proposed convolutional autoencoder, fused each pressure level into a single stream, through their introduction as channel of the initial convolutional layer. The explicit representation learning of the above autoencoder, aims to extract weather patterns for source estimation of nuclear events.

2.3 Representation Learning for Intermediate-Level Fusion

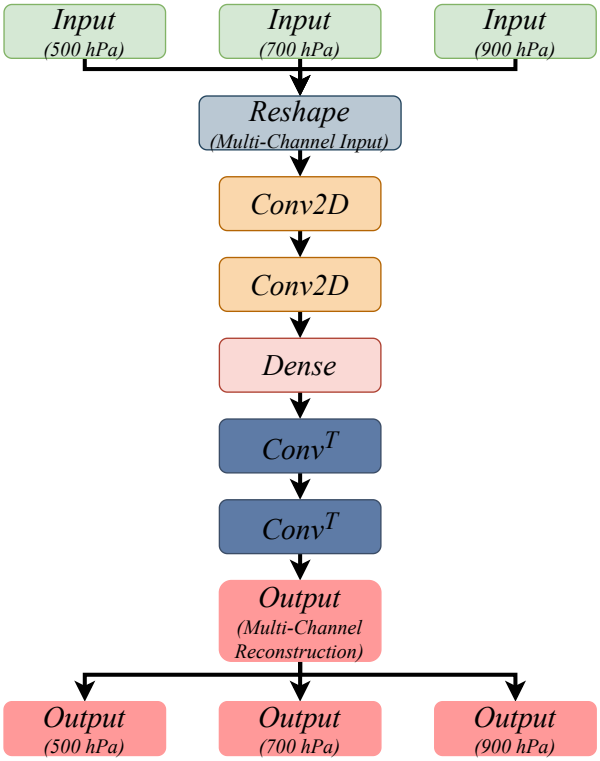
Unlike signal-level, intermediate-level involves refined features from pre-processing raw observations or unprocessed signals. Deep neural networks inherently learn intermediate features through their hidden layers, therefore deep learning frequently lends itself in this level. Implicit representation learning in this level involves two strategies: *intermediate merging* and *attention-based alignment*. Explicit representation learning in this level involves two strategies: *generative and discriminative hybrids* and *self-fusion*.

2.3.1 Intermediate Merging

Intermediate merging is well-practised within deep learning. It involves the design and training of individual neural network streams (sub-networks), which are aggregated in an appropriate depth. After aggregation, typically a single stream (prediction layer) is involved in an implicit representation learning objective. Each individual



(a) MMAE [9]



(b) Weather ConvAE [10]

Figure 2.11: Neural network architectures of MMAE [9] and Weather ConvAE [10].

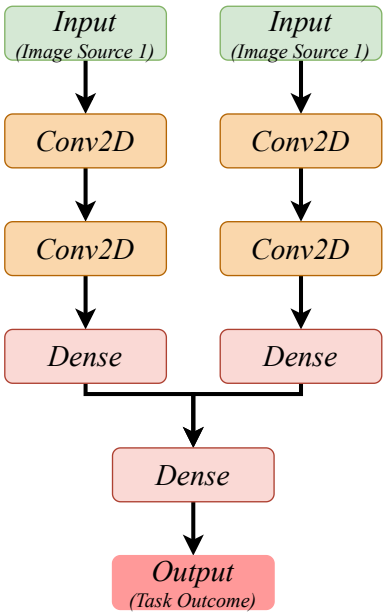
sub-network receives as input a single data source. The sub-networks aim to learn meaningful latent features from the individual sources. Therefore, their design is tailored to the characteristics of the dataset, e.g., Conv2D for gridded data. Repurposing of sub-networks for other tasks is common after training. Prediction layer allows back-propagation to all sub-networks, which in turn leads to extraction of most relevant aspects of each individual data source.

Zhou et al. [11] proposed an intermediate merging framework that consists of fully-connected layers. The learning objective of this framework is to jointly learn representations from multiple dialogue utterances, for the task of dialogue act recognition. The sub-networks do not share their weights. A single joint fully-connected layer acts as the combination operation before introducing joint features into the prediction layer. The architecture is shown in Figure 2.12 (b).

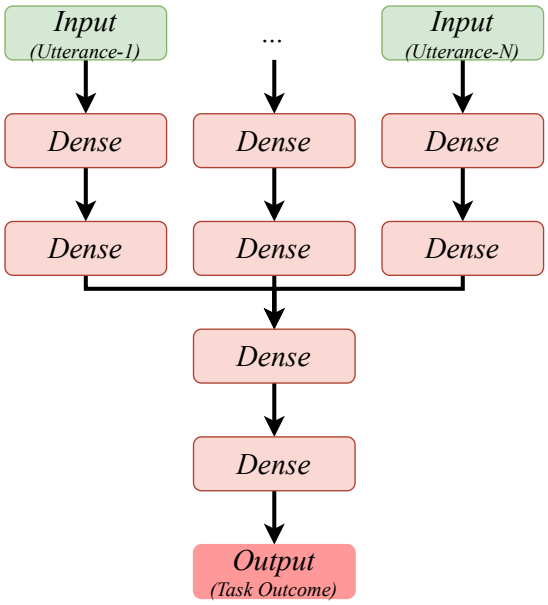
X-CNN is a unique intermediate merging strategy proposed by Velickovic et al. [13]. X-CNN consists of individual sequences of convolutional and pooling layers (one for each channel of the input), which are followed by more sequences of convolutional and pooling pairs. Each stage of convolutional layers that follows the initial pair involves connection from all modalities. The across connections are merged with concatenation. Thus, creating sub-networks that include features from all modalities. The architecture is show in Figure 2.13 (b).

Simonyan and Zisserman [12] proposed an intermediate merging approach based on aggregation of scores. The two-stream proposed architecture, aims to learn representations for the task of video classification. It consists of a spatial 2D CNN, that receives video frames and a temporal 2D CNN that receives optical displacement flows. Optical flow displacement is a data product that depicts motion between frames. To produce said product, consecutive video frames are utilised. The fusion is auxiliary to the learning process. After training, the decisions produced by each sub-network, is aggregated either through averaging or SVM training through stacking the scores after L_2 normalisation. The aggregation of scores is the final task outcome of the framework. The architecture is show in Figure 2.13 (a).

Even though intermediate merging is frequently practised, the depth of the merging, as well as, the merging operation, requires significant effort in experimentation. The merging depth and operation are vital to the effectiveness of the fusion. Park et al. [77] investigated various intermediate merging strategies for the task of action recognition. Similar to the previous framework it also involved two sub-networks: a spatial and a temporal stream, that receive as input frames and optical flow displacement, respectively. The proposed strategies were: concatenation of two streams, concatenation before aggregation in a joint layer, merging by element-wise product (referred to as multiplicative fusion) of feature maps extracted from convolutional layers or fully-connected layer outputs. In their study, multiplicative fusion performed

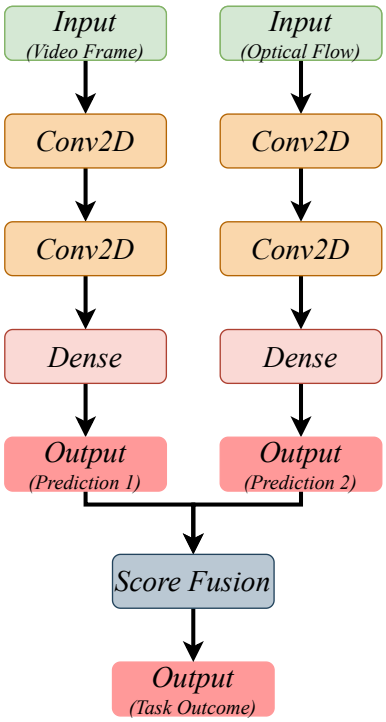


(a) Generic Intermediate Merging Architecture

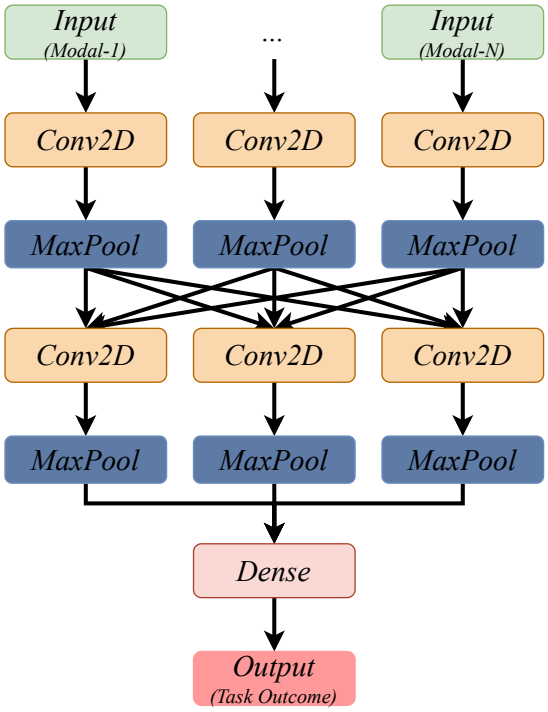


(b) Fully-connected merging approach [11]

Figure 2.12: Neural network architectures of a generic and fully-connected intermediate merging [11].



(a) Score merging approach [12]



(b) X-CNN approach [13]

Figure 2.13: Neural network architectures of score [12] and X-CNN [13] approaches.

better than concatenation.

Intermediate merging differs from joint latent space frameworks in thee ways. First, it involves implicit representation learning, unlike joint latent space that involves explicit representation learning. Second, joint latent space directly involves unprocessed signals, while in intermediate merging the extraction of features is part of learning process (inherently performed by each sub-network). Lastly, joint latent space frameworks directly involve input and output pairs in the training objective.

2.3.2 Attention-Based Alignment

This special case of data alignment exploits the attention mechanism in order to align data sources in an intermediate level. Attention mechanism was introduced after the initial success of Sequence-to-Sequence model, also known as Seq2Seq, in machine translation. Seq2Seq framework is very similar to autoencoder framework. An encoder network receives a data sequence, that transforms into a single representation. Then a decoder network transforms the representation back into a sequence. In machine translation, the input and output sequences are two different languages. Attention dealt with certain shortcomings of that model, such as dealing with a fixed-length encoder vector. Attention [14] (as shown in Figure 2.14) was introduced as an alignment model that was able to weight the importance of position in input sequence, with respect to the output positions. A feed-forward neural network calculates the correlation between input and output positions. The correlation,is called *alignment score*.Through calculation of the alignment scores, attention is able to automatically align the input and output sequences in an intermediate-level. Therefore, learning a deep learning intuitive method of transitioning between input and output domains. To this end, attention lend itself to additional applications, such as image annotation Xu et al. [78] and hierarchical attention [79].

2.3.3 Generative and Discriminative Model Hybrids.

Generative and Discriminative model hybrids combine the views of the two respective models. Hybrid models are very popular in semi-supervised learning setting, as they are able to learn from both unlabelled and labelled data. In regard to fusion, hybrid models allow the learning of intermediate-level features from a joint learning process, that involves signal-level and decision-level features (input data and discriminative view respectively). Variational autoencoder (VAE) and Generative Adversarial Networks (GAN) are two very established generative models, which are frequently part of hybrid modelling.

Variational Autoencoder is a deep learning implementation of variational infer-

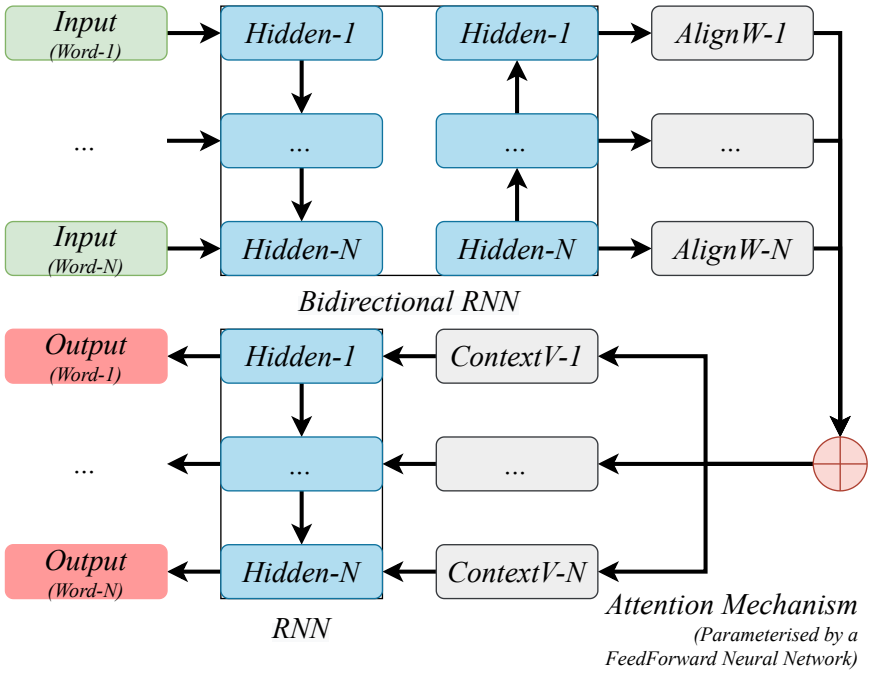
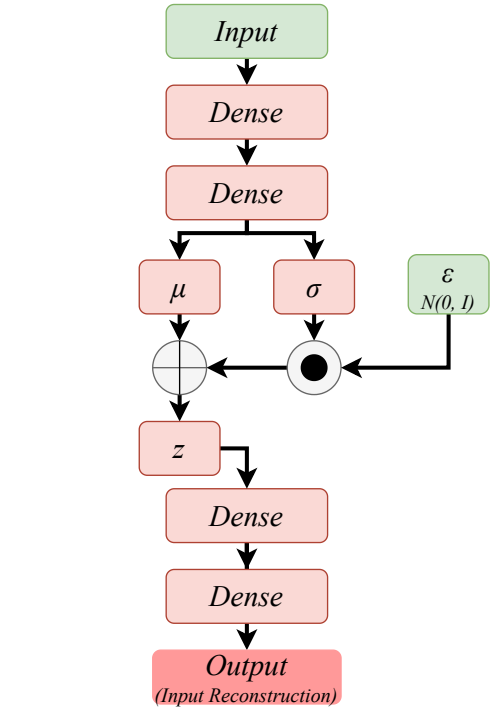


Figure 2.14: Neural network architecture of attention mechanism [14].

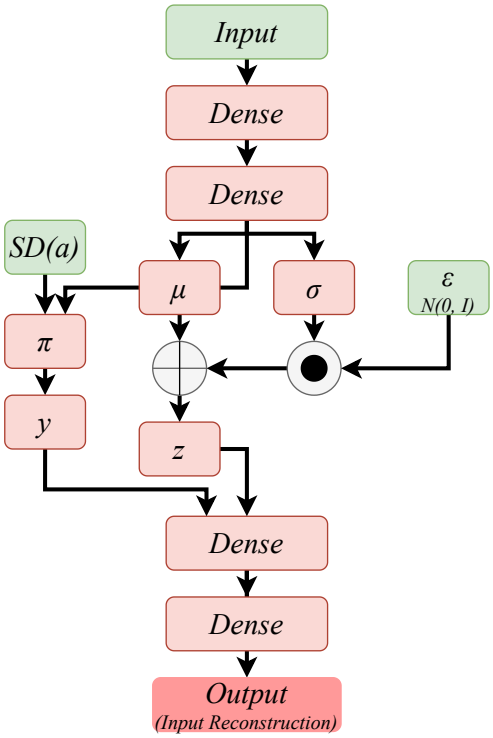
ence [15]. Its objective is to learn latent variables involved in the generative distribution, which are able to represent all variations of the data. The training process of VAE, involves an additional training objective. The additional objective, aims to constraint the distribution of latent variables into a multivariate Gaussian prior. The first unsupervised model (also known as M1) has been adapted into a hybrid version that includes a discriminative view. The proposed M2 model [16], consists of two encoder streams. The first stream predicts latent variables, similarly to M1 model. The second stream predicts class labels. During unlabelled data, the second stream is trained to output Symmetric Dirichlet distribution samples, similarly to Gaussian prior constraint of M1. During labelled data samples, the second stream learns to correctly classify the data instances. The decoder involves both streams. The M1 and M2 model variations are depicted in Figure 2.15.

Semi-Supervised Sequential Variational Autoencoder (SSVAE) [17], adapted the M2 model for text classification. SSVAE involves two LSTM encoders, with similar output to the M2 model. The classifier stream minimises the entropy of the output, instead of imposing a Symmetric Dirichlet distribution, during unlabelled data. Based on this adaptation, Zhang et al. [18] proposed a joint representation learning for multi-modal sentiment classification by stacking individual uni-modal SVAEs. The uni-modal SVAEs share common architecture characteristics with SSVAE. SSVAE and uni-modal SVAEs hybrid models are depicted in Figure 2.16.

The above models introduce discriminative views explicitly. An alternative way

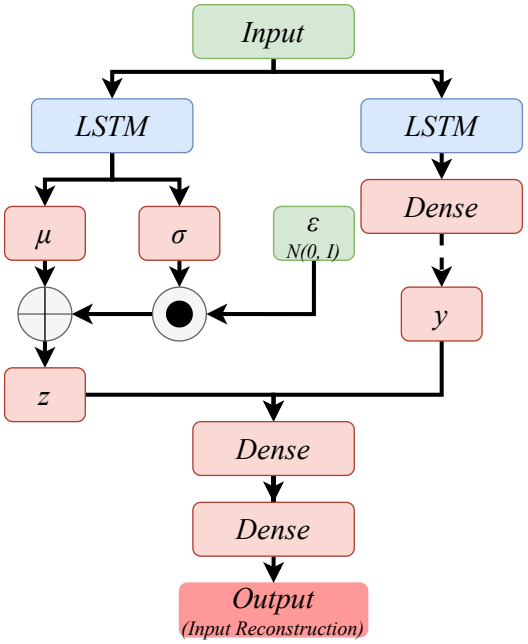


(a) M1 model [15]

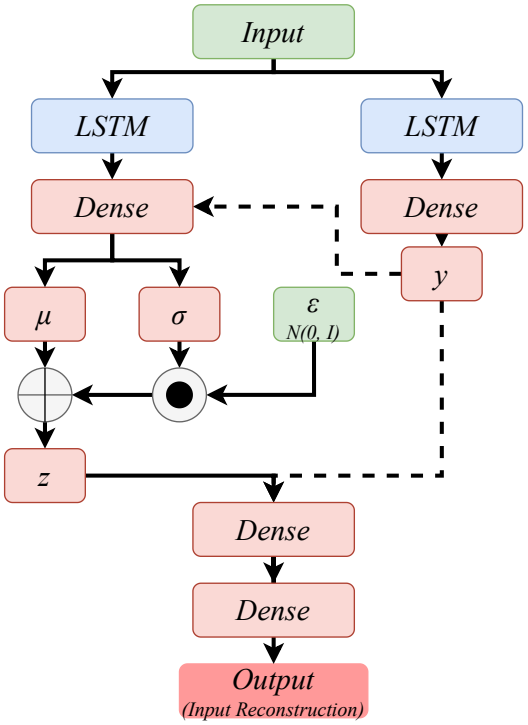


(b) M2 model [16]

Figure 2.15: Neural network architectures of M1 [15] and M2 [16] model variation of VAE.



(a) SSVAE [17]



(b) Uni-modal SVAE [18]

Figure 2.16: Neural network architectures of SSVAE [17] and Uni-modal SVAE [18] hybrid models.

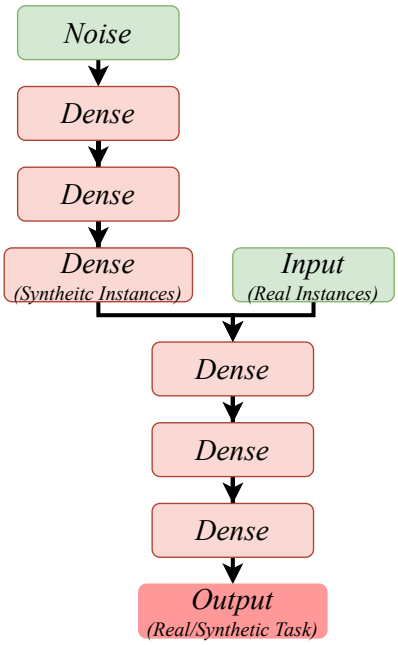
of introducing discriminative views, is implicitly introducing a discriminative view. ML-VAE [80] is a framework based on this idea. The motivation behind ML-VAE is that data samples within a certain group share common characteristics. For that reason, the data instances during training are introduced in groups. This is similar to separating the dataset into categories based on a discriminative view and then introducing each group, as a batch of training samples.

Goodfellow et al. [19] proposed the Generative Adversarial Network framework, that quickly became one of the most influential works in generative modelling. The authors proposed a unique training strategy that involves two adversarial sub-networks. The *Generator* and *Discriminator* sub-networks, which are adversaries during the duration of the training. The objective of the generator is to manifest realistic synthetic data instances that will be able to trick the discriminator into classifying them as real. On the other hand, the discriminator tries to distinguish between synthetic and real data instances. From this adversarial training, the generator learns to generate realistic data instances.

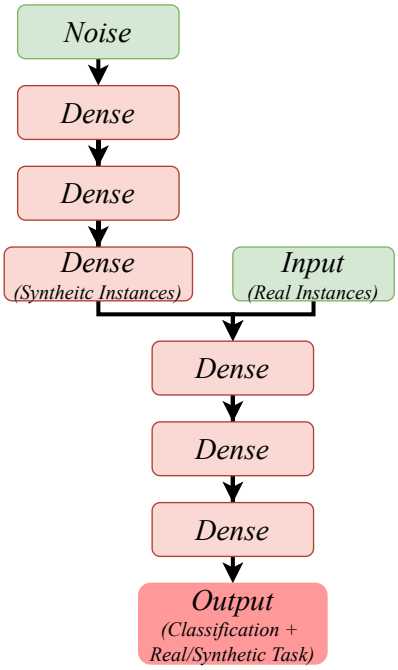
The first hybrid variation of GANs, is the Categorical Generative Adversarial Network (CatGAN) [20, 21]. The CatGAN framework introduces a specialised discriminator which in addition to classifying whether a data instance is real or synthetic, it also predicts the class of the instance. The class prediction is the product of a softmax activation that produces class probabilities. The objective of CatGAN is similar to that of GANs, however it involves an additional term that measures the classification error of the discriminator. CatGAN is also able to perform unsupervised clustering by involving the minimisation of entropy of the class probabilities. GAN and CatGAN hybrid variation are depicted in Figure 2.17.

Sricharan et al. [22] proposed that the discriminator should be divided. The proposed Semi-Supervised GAN (SS-GAN), consists of stacking two discriminators, as well as, a generator that also involves a class attribute vector. The first discriminator, is an unsupervised discriminator that classifies whether an instance is fake or real. During training with labelled data samples, the supervised discriminator (which is an intermediate extension of the unsupervised discriminator) is trained to correctly predict class attribute vector. Figure 2.18 depicts the SS-GAN model architecture.

The training strategy of GANs was adapted by other frameworks as well. Makhzani and Frey [81] proposed the use of adversarial training in autoencoders. The proposed Adversarial Autoencoder (AAE), involves a discriminator stream that imposes certain distributions over the latent space. The choice of the distribution causes different effects over the latent space. Unlike VAEs, the imposing of a prior distribution in AAE is more implicit, since AAE imposes the prior by drawing samples from the distribution and the adversarial training. AAE can be adapted to a plethora of tasks, such as supervised classification, semi-supervised classification, unsupervised clustering and

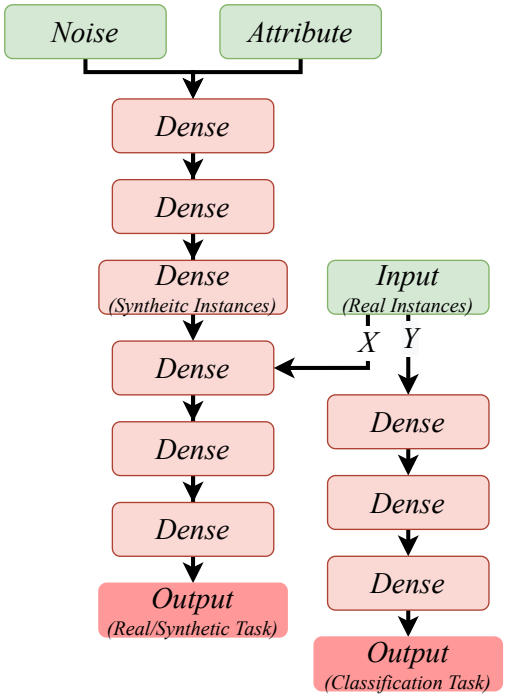


(a) GAN [19]



(b) CatGAN [20, 21]

Figure 2.17: Neural network architecture of GAN [19] and CatGAN [20, 21] hybrid model variation.



(a) SS-GAN [22]

Figure 2.18: Neural network architecture of SS-GAN [22].

dimensionality reduction.

2.3.4 Self-fusion

Self-fusion frameworks, bear some similarities with hybrid models. However, unlike hybrid models that involve supervised task outcomes or labelled data, self-fusion involves unsupervised task outcomes. Most often, they are alternative views of the original dataset (*clone fusion*) or unsupervised down-stream task outcomes, such as clustering (*clustering as auxiliary view*).

Clone fusion is a self-fusion scheme that involves alternative views, perspectives or features of a single dataset. In the context of deep representation learning, it typically involves sub-networks with similar architectures but varying hyperparameters, such as activation functions, convolution strides or filter sizes. However, other operations that extract alternative views can be used, e.g., traditional machine learning algorithms.

Ng et al. [23] proposed an autoencoding framework to deal with class imbalance. The proposed Dual Autoencoding Features (DAF) framework (depicted in Figure 2.19), aims to learn a meaningful latent space that will improve decision boundaries between the available classes. DAF consists of two autoencoders with the same architecture, but different activations (sigmoid and tanh), which capture different intermediate views of the dataset. The autoencoders are first trained individually. Then, a

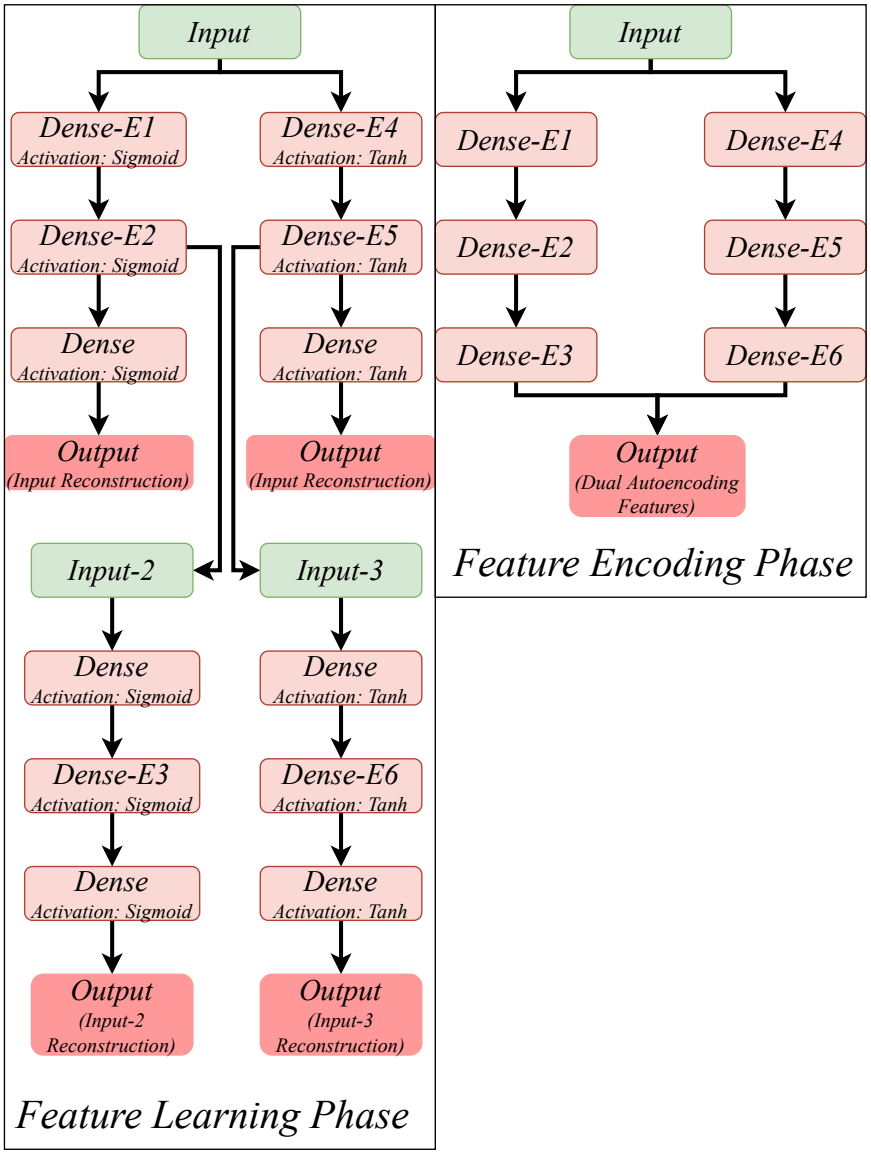


Figure 2.19: Neural network architecture of DAF [23].

combined latent representation is extracted from the two independent autoencoders.

Clone fusion can also be introduced through self-augmentation. Hu et al. [82] proposed the Information Maximisation Self Augmented Training (IMSAT) framework, which learns discrete representations by encouraging the representations to be invariant to various augmentations of the dataset. IMSAT learns to maximise the mutual information between discrete representations and augmented data. The augmentation procedure is part of the training process.

Peng et al. [83] proposed extracting representations from a random walk with restart over biological networks. Since random walk is a stochastic process, multiple iterations will yield different representations. An autoencoder receives all different

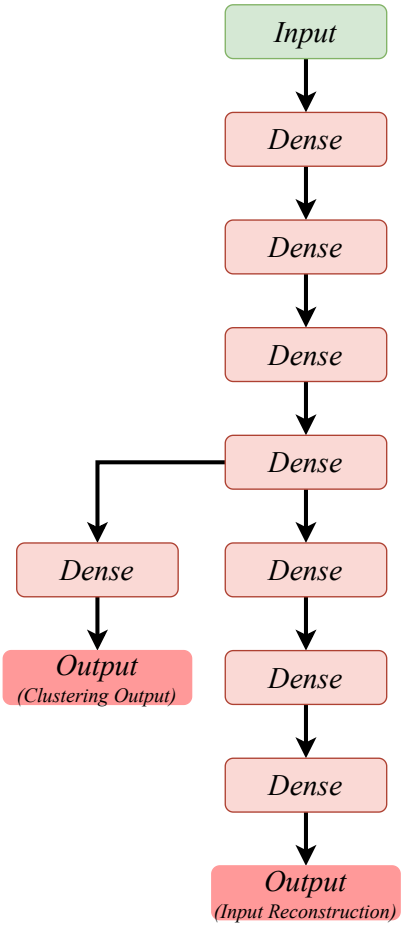


Figure 2.20: Neural network architecture of DEC [24].

representations produced by the random walk. The autoencoder aims to produce representations augmented with domain knowledge. The domain knowledge, is involved through a training objective that promotes pairwise similarity between gene nodes. The final outcome of the process, which is the outcome of gene function prediction, is performed by a CNN through observation of the representations extracted from the autoencoder.

Clustering as auxiliary view is a fusion scheme that involves the outcome of a clustering algorithm, as an external discriminative view. The outcome of a clustering algorithm, can be seen as a complementary discriminative view of the dataset. The task outcome of the clustering algorithm may be either used as input in the deep representation learning framework or as a training objective term.

Yang et al. [84] proposed an explicit representation learning framework based on expectation-minimisation type of traning. The proposed Joint Unsupervised Learning (JULE) framework, uses a CNN sub-network to extract meaningful features from images, as well as, exploiting the RNN mechanism to perform agglomerative clus-

tering. The training steps of JULE are: first update a clustering membership based on the current latent representations from the CNN. Then, it updates the latent representations based on the clustering membership extracted from the specialised RNN.

Xie et al. [24] proposed a clustering specific fine-tuning of the initial latent representations learned from the original autoenconding framework. The proposed Deep Embedded Clustering (DEC) framework (as depicted in Figure 2.20), involves the introduction of a clustering specific layer and training objective term. The fine-tune step, aims to minimise the Kullback-Liebler Divergence between the auxiliary layer and a self-referencing target t-distribution that performs soft cluster assignment.

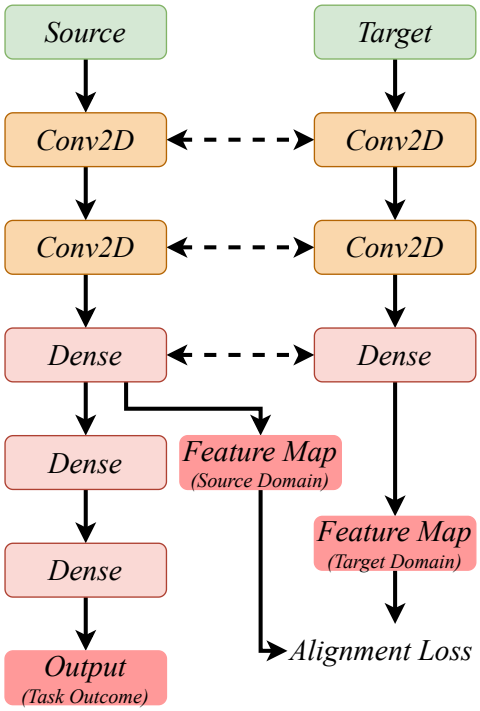
Chang et al. [85] proposed an end-to-end framework called Deep Adaptive Image Clustering (DAC). DAC is based on the binary pairwise-classification problem. Given a pair of images, a classifier should decide whether two images belong in the same cluster or not. The initial class probabilities are generated from a CNN, which DAC iteratively adapts based on the above objective with unsupervised training.

2.3.5 Domain Adaptation

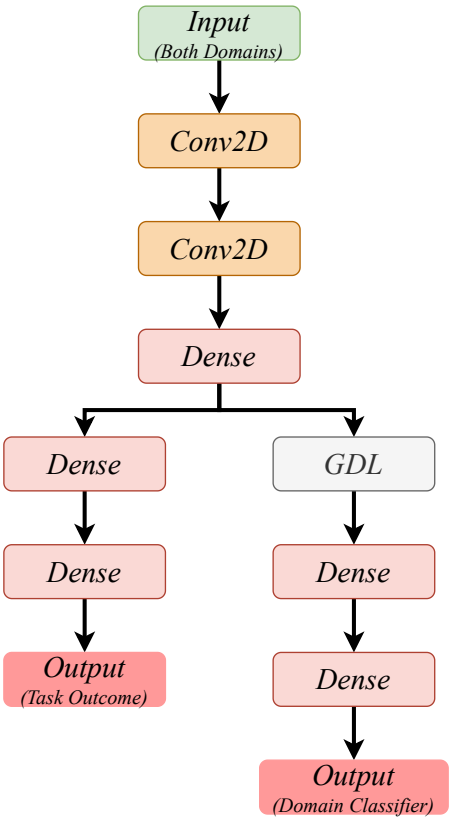
Domain adaptation is case of transductive transfer learning. Domain adaptation frameworks utilise a shared task, as a common point of reference between multiple domains. The objective is to transfer knowledge from a source domain(s) into a relevant target domain. It typically involves adaptation of pre-trained networks with frozen weights or multiple sub-networks with shared weights. Domain adaptation can be used for both implicit and explicit representation learning.

A frequent approach is using a Siamese network architecture. The Siamese architecture involves two neural network streams, with a single output that corresponds to the similarity between inputs [86]. Motiian et al. [87] utilised the Siamese architecture with shared weights for supervised domain adaptation (as depicted in Figure 2.21 (a)). The training involves a composite objective that consists of a classification loss, an alignment loss that encourages similarly labelled instances to produce closely related features and a suitable similarity loss that encourages dissimilarly labelled instances to be distanced.

Kushibar et al. [89] proposed a weight freezing strategy for supervised domain adaptation. The proposed architecture, involves a pre-trained CNN that receives magnetic resonance images (MRIs) from three different scanners. Each scanner depicts a different orientation. Dense layers that combine features from the different MRI streams are not frozen. The final prediction layer starts from random initialisation. The aim of the framework is to adapt the predictions to domain shifts between MRIs.



(a) Motiian et al. [87]



(b) Ganin and Lempitsky [88]

Figure 2.21: Neural network architecture of domain adaptation frameworks.

Ganin and Lempitsky [88] proposed an unsupervised domain adaptation framework (as depicted in Figure 2.21 (b)), that aims to adapt to domain shifts from unlabelled target tasks. The architecture involves a single stream encoder that receives both domains as input, to produce feature maps based on joint learning. Two task-specific streams extend the encoder into producing common task outcomes, while the second stream tries to distinguish feature maps between domains. The composite objective is regularised by using a special layer called *gradient reversal layer*. Gradient reversal is a special layer proposed by the same authors, that encourages domain invariant features through the process of reversing the gradient during backpropagation, i.e., multiplying its gradient with a negative hyperparameter. Gradient reversal layer is used in the domain classifier branch.

2.4 Representation Learning for Decision-Level Fusion

Decision-level fusion involves high-level information, such as decisions or task outcomes. Decision-level fusion is similar to the process of ensemble methods in traditional machine learning, which combine decisions using various operations, such as Averaging [90], Stacking [91], Bagging [92] and Boosting[93] to produce a final more effective decision. In deep learning, transfer learning frequently lends itself as a core model for the fusion process. Most frequent learning settings are unsupervised transfer learning and multi-task learning.

2.4.1 Unsupervised Transfer Learning

Unsupervised transfer learning transfers knowledge from dissimilar tasks, which are outcomes based on dissimilar domains. Since, the target task is typically an unsupervised task, the representation learning of these frameworks is typically explicit.

Adaptation and Re-Identification Network (ARN) [25] is an explicit representation learning framework, that deals with limited labelled instances, in the task of person re-identification. ARN is an autoencoder with multiple encoder branches. The encoder is a pre-trained ResNet-50, that receives images from both domains and produces feature maps for its respective domain. The feature maps are connected with one shared module and two private modules. The shared module receives both streams, while the private modules receive each individual feature map. The encoder outputs four different streams, two from the shared module and two from the private modules. The decoder receives a concatenation of summed latent vectors, consisting of shared and private feature maps, in order to reconstruct either the source or target domain.

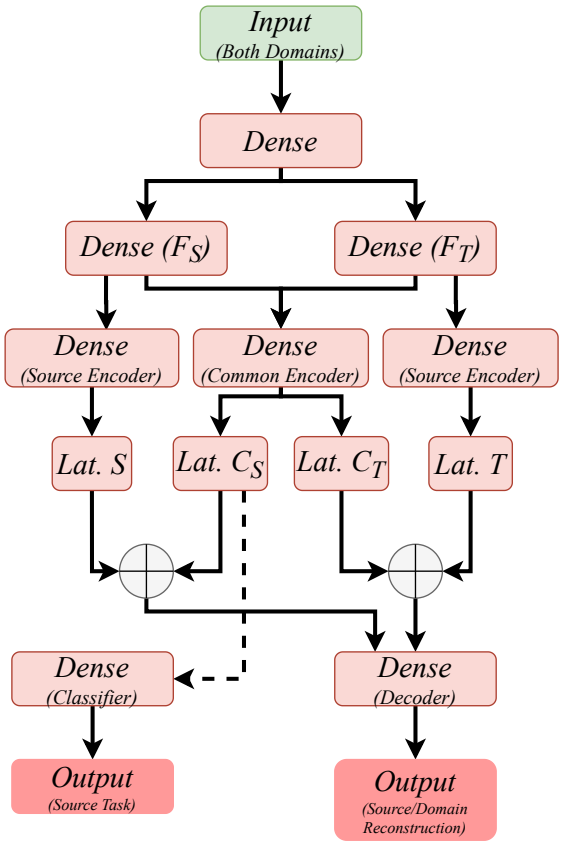


Figure 2.22: Neural network architecture of ARN framework [25].

In addition, a classifier is connected with the shared latent vector, in order to learn from labelled instances of the source domain. Figure 2.22 depicts neural network architecture of ARN

Wu et al. [94] proposed an unsupervised deep transfer learning framework, to perform fault diagnosis in fog radio access networks. The proposed framework, learns both from labelled and unlabelled fault instances in source and target domains. The framework consists of a single encoder network that receives images from both domains and performs three tasks: classification, domain discrimination (distinguish between domains based on their feature maps) and minimising the distribution discrepancy between the feature maps of source and target domains, in order to encourage the learning of domain-invariant features.

2.4.2 Multi-Task Learning

Multi-task learning is inherently an implicit representation learning fusion scheme, as it involves labelled data instances for multiple tasks. However, certain categories of multi-task learning are also repurposed for explicit representation learning. As men-

tioned previously in Section 2.1.7, multi-task learning typically aims to concurrently learn all tasks. It also consists of categories: *hard parameter sharing*, *soft parameter sharing*, *supervision at custom depths* and *domain specific adaptation of layers*.

Hard parameter sharing is a multi-task learning category, that involves a single task-invariant sub-network that is extended with multiple task-specific streams. An established hard parameter sharing architecture for explicit representation learning is the framework proposed by Srivastava et al. [45]. Srivastava et al. [45] extended the LSTM autoencoder framework by involving an additional stream with the objective to predict future sequences. The final training objective consists of two unsupervised objectives: reconstruction of the current sequence and prediction of the next sequence through observation of the current one.

An alternative version to concurrently learning each task, is to disjointly optimise each task. Multi-Domain Network (MDNet) [95] is a hard parameter sharing framework that learns domain invariant representations for visual tracking. MDNet receives input sequences of image frames from videos. Since each video is a separate domain for MDNet, each task-specific branch is trained individually depending on the current domain. Disjoint optimisation between video classification and annotation is also proposed by Kim et al. [96].

On the other hand, Cipolla et al. [97] proposed a composite training objective for joint optimisation of individual task scores. The composite objective involves homoscedastic uncertainty [98] as a principled alternative to manual or uniform linear fusion operations. The framework of Cipolla et al. [97] involves training with instance, semantic segmentation and depth regression tasks for visual understanding. Instance segmentation is the task of extracting vector masks of individual objects in an image. Semantic segmentation is the task of producing a meaningful segmentation of the original image to semantically meaningful groups of classes. Depth regression is the task of measuring the depth of each pixel/relative distance of each pixel from the sensor.

Soft parameter sharing is a multi-task learning category that involves individual sub-networks for each task, which is the opposite direction of that of hard parameter sharing. The sub-networks typically share their weights, in order for all sub-networks to be concurrently informed by all task outcomes.

Chen et al. [99] proposed a soft parameter sharing autoencoder, for explicit learning of phonetic-and-semantic embeddings for the task of spoken content retrieval. The autoencoder contains several branches, which are all involved in various training objectives. Phonetic embeddings with disentangled speaker characteristics are extracted by merging the two encoder branches: phonetic and speaker embeddings, into a single decoder that reconstructs its input. The speaker encoder branch measures the distance between utterances produced by the same speaker, in order to learn similar

embeddings for the same speaker. On the other hand, the phonetic encoder branch is involved in an adversarial training with a discriminator that learns to distinguish between whether two utterances come from the same speaker or not.

EdgeStereo [100] is a soft parameter sharing CNN for the tasks of stereo matching and edge detection. In computer vision, the task of stereo matching aims to find corresponding pixels from two viewpoints. The output of stereo matching is a disparity map, depicting the displacement between pairs of corresponding pixels. Edge detection aims to find object boundaries. EdgeStereo involves two sub-networks for each task that share weights.

Knowledge distillation is the task of transferring knowledge acquired from a deep network into a shallower counterpart. It was first proposed by Hinton et al. [101]. Instead of shared weights, the knowledge distillation framework follows a student-teacher type of training. A pre-trained deep neural network "teaches" a shallower student model counterpart to produce similar predictions. The comparison of predictions is back-propagated to the student model, in order to align its predictions to the teacher model. This idea was adapted by Li et al. [102], that proposed the Meta-learning based Noise-Tolerant Training (MLNT). MLNT trains a student network to create robust predictions, by training the student model with synthetic noisy labels. A teacher model is utilised to compare predictions using KLD as a distance measure.

Supervision at custom depths is similar to hard parameter sharing, however instead of extending a sub-network with additional task-specific branches, it introduces the prediction of tasks at appropriate depths. Søgaard and Goldberg [103] suggested, that the tasks categorised as semantically low-level should be predicted based on lower-level intermediate features. Learning more accurate features in shallow levels, through the use of "shallower" tasks, should also enable the performance of higher-level tasks. This idea is based on the property of deep neural networks to learn complex tasks from feature aggregations of previous layers. Hashimoto et al. [104] proposed the Joint Many-Task (JMT) model framework for NLP tasks. JMT learns concurrently five different NLP tasks, where the prediction of each outcome is hierarchically placed within a deep neural network from shallower to deeper layers.

Domain specific adaptation of layers stems from the investigation of Bilen and Vedaldi [105], regarding the training of a representation learning framework which will produce universal representations from multiple domains. The main focus of the investigation was computer vision. In their study, they used a single shared architecture across all domains. However, the batch normalisation layers were adapted with domain-specific scaling factors, which were able to deal with inter-domain statistical shifts. Furthermore, Rebuffi et al. [106] proposed adapter modules that were injected onto deep learning architectures, such as ResNet [107]. The proposed adapter modules tackle the learning of multiple domains, and are able to intervene in multiple

depths or connections, such as serial or parallel.

The next chapter introduces the core concepts and objectives of EviTraN, along with its evaluation criteria and deep learning framework.

Chapter 3

Evidence Transfer

Evidence transfer is a deep learning method that aims to improve initial learned representations, based on auxiliary tasks extracted from unobserved external datasets. This chapter presents the fundamental concepts of the method and introduction of the learning settings, that pertain to EviTraN based on the properties of introduced auxiliary tasks. Furthermore, it includes details regarding the deep learning implementation of the method, such as variations of its neural network architecture and training strategy.

3.1 Introduction

This section introduces the task at hand, fundamental concepts and objectives of EviTraN. The high-level description of the involved concepts, acts as a prelude to the presentation of deep learning implementation in Section 3.4. It also includes the evaluation criteria of the method, which can be generally applied during representation learning for information fusion.

3.1.1 Objective and Concepts

Let primary dataset $X = \{x^{(1)}, \ldots, x^{(N)}\}$ be the dataset of interest for the task of representation learning. In other words, let the primary task be the process of learning latent variables $Z = \{z^{(1)}, \ldots, z^{(N)}\}$ (also known as latent representations) in an unsupervised manner. Latent representations Z, are the outcome of a predictive function $Z = G(X, \theta)$, with input a primary dataset X and function parameters θ. In unsupervised representation learning the predictive function G is typically a generative model. Figure 3.1 depicts an overview of the objective of EviTraN.

Furthermore, let $\mathcal{V} = \{V_1, \ldots, V_L\}$ be a set of external categorical variables, where each of V_l with $l \in \{1, \ldots, L\}$ represents an auxiliary task acquired from

a set of external datasets $\epsilon X = \{\epsilon x_1, \ldots, \epsilon x_L\}$. $\mathcal{V}$ is termed as *external evidence*. Each external dataset ϵx_l is related to primary dataset X with a relation r, such that $\epsilon x_l = r_l(X)$ or $\epsilon x_l = r_l(X, \phi_l)$ during cases where relation r_l involves some model with parameters ϕ_l. Auxiliary task outcomes $\mathcal{V}$ are extracted from decision models $\mathcal{F} = \{f_1, \ldots, f_L\}$, with corresponding set of model parameters $\Psi = \{\psi_1, \ldots, \psi_L\}$. External datasets ϵX, relations r and decision models $\mathcal{F}$ are unobserved.

For simplicity purposes assume that $L = 1$ and therefore, one single source of external evidence is available. V_1 is the outcome of an auxiliary task, acquired from a decision model f_1 with model parameters ψ_1 and input external dataset ϵx_1. Where external dataset ϵx_1, is related to primary dataset with relation $\epsilon x_1 = r_1(X, \phi_1)$. Therefore, external categorical variable V_1 contains the outcome of decision model f_1, i.e., auxiliary class memberships $\{V_1^{(1)}, \ldots, V_1^{(M)}\}$.

The objective of EviTraN is to improve the unsupervised process of learning latent representations Z, through observation of primary dataset X (primary task) by utilising auxiliary task outcomes $\mathcal{V}$, extracted from related but unobserved external datasets ϵX. The objective of EviTraN should be a new set of model parameters θ, that will allow for more optimal (based on predefined satisfaction criteria) predictive function G, such that $Z = G(X, \theta)$.

3.1.2 Evaluation Criteria

From the introduction and discussion of previously defined evaluation criteria for information fusion in Section 2.1.12, it is evident that only some of them are generally applicable. These criteria, namely: *quality*, *stability*, *robustness and tested with real data*, should also be considered during deep representation learning for information fusion. The proposed evaluation criteria for EviTraN method, compose a general set of criteria for deep representation learning for information fusion.

Any deep representation learning framework for the purpose of information fusion should at least satisfy the following criteria:

Effectiveness. The effectiveness criterion acts as a mean of measuring the satisfaction of the predefined objective, e.g., learning more optimal predictive function G through exploitation of external evidence. This is criterion is similar to *quality* and *tested with real data*. The satisfaction of this criterion depends on auxiliary information sources, representing meaningful and relevant relations to the primary dataset. In that case, the framework should discover these relations and utilise them towards learning of more accurate parameters θ of function G. Subsequently, the effectiveness of the method should scale with multiple sources. Meaning that, multiple relevant information sources should lead to greater results than a single relevant source.

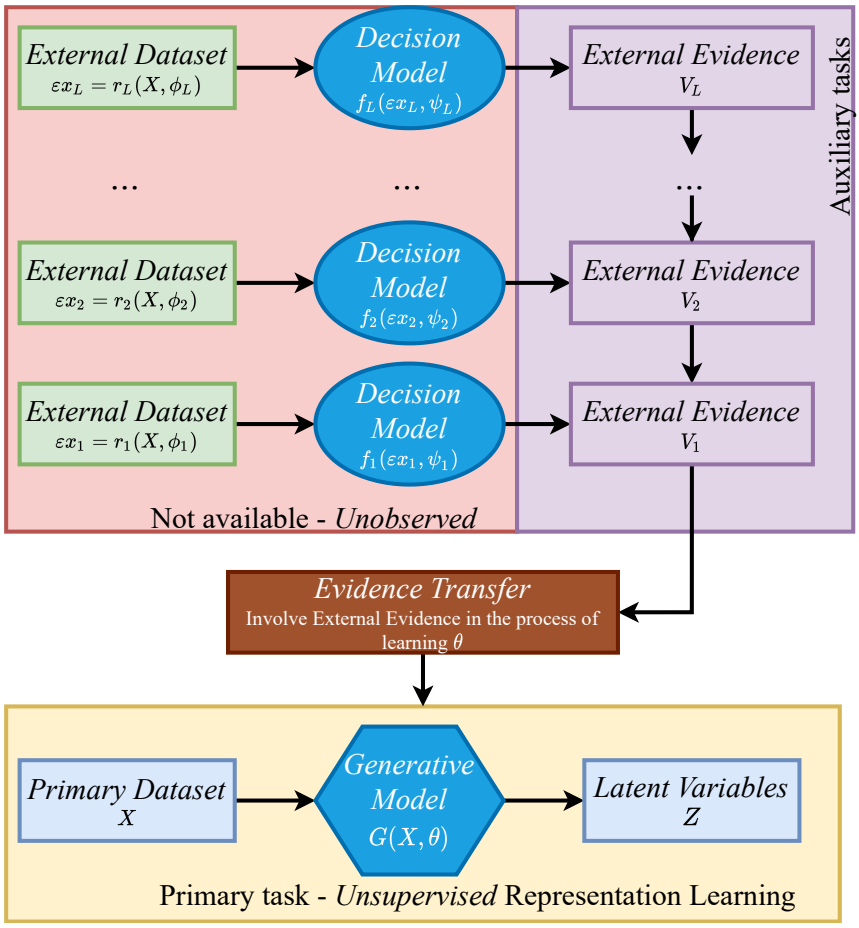


Figure 3.1: Overview of the objective of EviTraN. The primary task of the method is the process of learning underlying variables Z from observation of primary dataset X. The learning process is a generative model with trainable parameters θ. EviTraN utilises external categorical variables V_1, ..., V_L (external evidence) extracted from unperceived decision models f_1, ..., f_L with input unobserved external datasets ϵx_1, ..., ϵx_L. EviTraN aims to transfer relevant knowledge from external categorical variables to improve learning of trainable parameters θ.

Robustness. The robustness criterion acts as a mean of resisting disturbance introduced by auxiliary information sources. This criterion is similar to *stability* and *robustness*. In addition, it preserves the applicability of the framework in a variety of cases, as introducing additional assumptions may reduce the applicability of the framework. However, the involvement of information sources that represent arbitrary relations, may include cases where the introduced source does not contribute any insight towards the learning of the primary task. Any proposed framework, should aim to preserve the initial performance of the primary task. Initial performance refers to the effectiveness of the framework before the introducing of auxiliary information sources.

Modularity. Deep learning methods that directly involve multiple information sources may condition themselves to require all sources to be present during inference. Expecting auxiliary information sources to be available during inference is not realistic, due to limitation in labelling such as costly or timely procedures. At the same time, dedicating extra time or computational resources towards total re-training of the model, upon presence of each auxiliary information source, is also costly. Therefore, the framework should be deployed in a manner that allows for incremental and independent training during the presence of new sources.

3.2 Comparison of Evidence Transfer to Previous Work

This section presents a high-level comparison, between the proposed EviTraN method and previous work presented in Chapter 2.

3.2.1 Machine Learning Perspective

In regard to learning type, as well as, from a systemic view, EviTraN is a hybrid representation learning method. In deep learning literature, associating the term *hybrid* with a systemic view of the deep learning framework, is an often occurring phenomenon. Describing a deep learning model as hybrid refers to the model's ability to acquire joint properties from two sub-networks/sub-systems: a *Generative Model* and a *Discriminative Model*.

From a systemic perspective, EviTraN combines the generalisation properties of an unsupervised model, along with conditioning introduced from a supervised discriminative model. Furthermore, EviTraN introduces supervision of arbitrary semantic levels of supervision in an otherwise completely unsupervised learning process. In the context of EviTraN, any type of external **categorical** variable can be utilised and be considered as external evidence. The term "external" refers to evidence sources being of auxiliary nature to the primary task. In other words, external evidence represents categorical samples (with the most straightforward being task outcomes) with no explicit relation to the primary task of learning latent representations.

As the name of the method suggests, EviTraN can be further categorised as a Transfer Learning method. EviTraN utilises auxiliary sources of categorical variables in order to influence the process of learning latent representations. For consistency with the transfer learning notation introduced from Pan and Yang [3]: let $D_{Prime}\{\mathbf{x}, P(X)\}$ be our target domain with $\mathbf{x}$ being the feature space of pri-

mary dataset and $P(X)$ being the true generative distribution of data samples $X = \{x^{(1)}, \ldots, x^{(N)}\}$. Also, let $T_{Prime}\{\mathbf{z}, Z\}$ be the target task with $\mathbf{z}$ being a continuous latent feature space and $Z = G(X, \theta) = \{z^{(1)}, \ldots, z^{(N)}\}$ being the latent representations. Then, EviTraN aims to utilise source task $T_{Aux}\{\mathbf{v}, \mathcal{V}\}$, where $\mathbf{v} = \{\mathbf{v}_1, \ldots, \mathbf{v}_L\}$ is a set of categorical feature spaces and $\mathcal{V} = \{V_1, \ldots, V_L\}$ is a set of external evidence sources, to improve the learning of target task T_{Prime}. Furthermore, let source domain $D_{Aux}\{\boldsymbol{\epsilon}\mathbf{x}, \epsilon X\}$ with $\boldsymbol{\epsilon}\mathbf{x} = \{\boldsymbol{\epsilon}\mathbf{x}_1, \ldots, \boldsymbol{\epsilon}\mathbf{x}_L\}$ being a set of unobserved feature spaces and $\epsilon X = \{\epsilon x_1, \ldots, \epsilon x_L\}$ being a set of unobserved external datasets.

Depending on the relations between each external dataset and the primary dataset, EviTraN can be further classified into two more subgroups. Assuming a single source of external evidence $\mathcal{V} = \{V_1\}$, if $\epsilon x_1 = r_1(X, \phi_1) = X$, meaning that external evidence V_1 is acquired from decision model f_1 with input the primary dataset X, then EviTraN is related to *Inductive Transfer Learning*. Oppositely, if V_1 is the outcome of decision model f_1 with input $\epsilon x_1 = r_1(X, \phi_1)$, then EviTraN is related to *Unsupervised Transfer Learning*. Figure 3.2 depicts the machine learning overview of EviTraN.

Therefore, EviTraN is an unsupervised transfer learning method, that utilises a hybrid of a generative and discriminative model as the delivery mechanism, for the transfer of knowledge between tasks. To this end, the hybrid modelling can be considered as the fusion operation. EviTraN does not involve unprocessed signals like signal-level fusion, thus being able to bypass challenges such as dealing with voluminous data or requiring a large amount of resources for training. At the same time, it also bypasses manual work required for data alignment at signal-level, as finding the most relevant features for the external task is automatically performed (more in Chapter 4).

Furthermore, EviTraN deals with expectations that arise during intermediate-level fusion schemes such as intermediate merging. Intermediate merging involves features from multiple sub-networks. During training data instances from all data sources are available. However, the expectation of having all data sources available during prediction of new instances is not realistic. Yet, missing correspondence between sources may lead to inference problems in intermediate merging. Comparison to attention-based fusion schemes is impractical, since EviTraN does not involve the attention mechanism. In addition, comparison to self-fusion scheme is also not practical since the motivation behind EviTraN is to be used with task outcomes extracted from auxiliary datasets. As is, with domain adaptation, since EviTraN does not involve a single common task.

Being an unsupervised transfer learning, EviTraN is able to deal with challenges such as dealing with non-complementary data or expecting auxiliary data to be available during inference. In the domain of transfer learning, dealing with non-

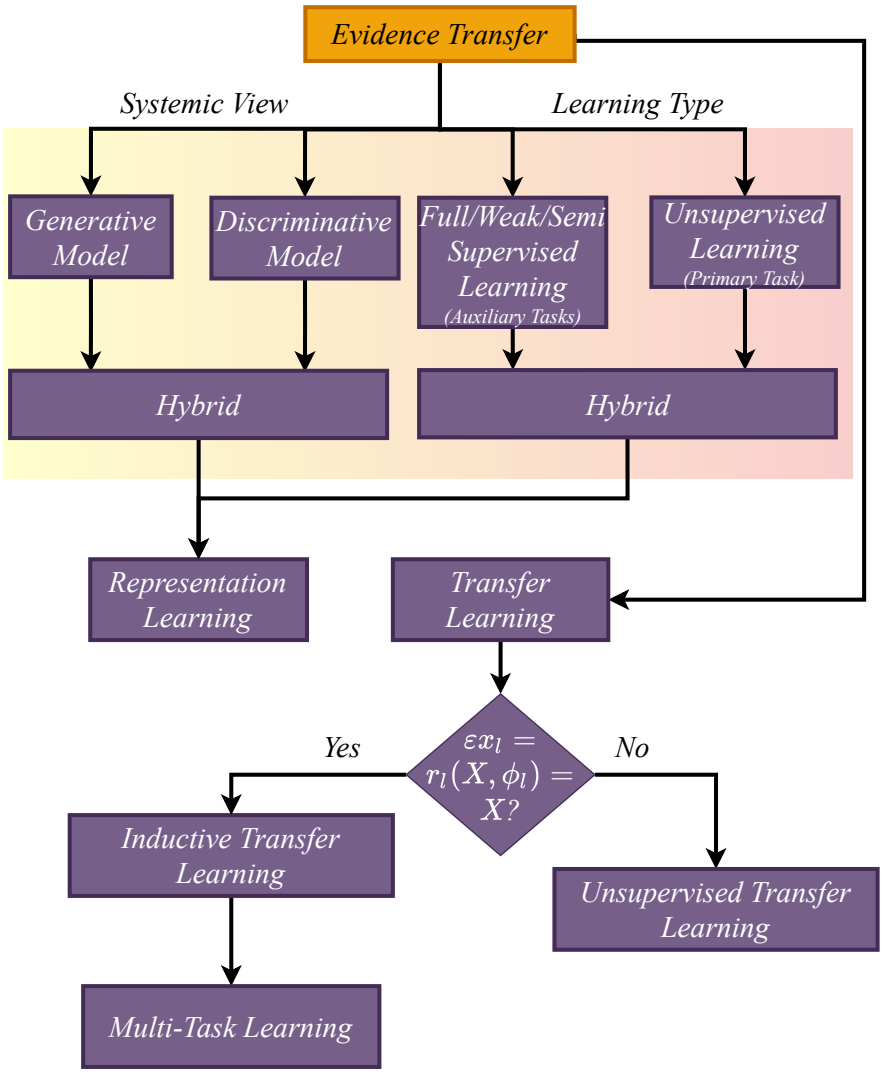


Figure 3.2: Machine Learning Overview of EviTraN.

complemetary data sources is known as negative transfer learning [108]. Furthermore, transfer learning aims to transfer knowledge from tasks, meaning that the source task is not required after training. At the same time, it also deals with data irregularities such as incomplete correspondence between data sources by considering the use of EviTraN in the incomplete learning setting (more in Chapter 3).

3.2.2 Information Fusion Overview

From a machine learning perspective, EviTraN aims to guide the unsupervised learning process of learning latent representations with arbitrary semantic levels of supervision. However, the main motivation of EviTraN is to provide an efficient,

effective and robust method for intelligent data fusion, utilising deep learning.

The objective of EviTraN can be reiterated from an information fusion view. Primary dataset X is information at signal/low-level. Auxiliary task outcomes $\mathcal{V}$ is information at decision/high-level that is extracted from signal/low-level information sources $\epsilon X = \{\epsilon x_1, \ ..., \ \epsilon x_L\}$ through the processes of $\mathcal{F} = \{f_1, \ ..., \ f_L\}$. Where each of external data sources $\{\epsilon x_1, \ ..., \ \epsilon x_L\}$ may be potentially heterogeneous. Latent representations Z is feature/intermediate-level information, which is the outcome of EviTraN.

From an information fusion perspective, EviTraN aims to fuse signal-level information from a primary data source X and external decision-level information sources $\mathcal{V}$, extracted from potentially heterogeneous and external data sources ϵX. The outcome of this fusion is a new set of feature-level information Z that are of improved quality. The quality of the new feature-set is evaluated based on the previous criteria.

3.3 Learning Settings

Involving some form of supervision from auxiliary tasks in the process of learning representations, may introduce bias. Such bias, may affect the end-goal of representation learning, such as repurposing learned representations in a downstream task or studying the properties of the primary data distribution in a less complex space. Locatello et al. [109] concluded in their study that the role of inductive biases and implicit or explicit supervision, is crucial in the unsupervised learning of disentangled representations. Therefore, from a benevolent perspective involving some form of supervision in an unsupervised process, highlights semantically high-level information. Such information, could not be observed during completely unsupervised learning, as it only involves data features. Converting decision-level information, such as outcomes of auxiliary tasks into latent features, can be broadly applied in multiple applications. Such conversion, can lead to increased performance towards an end-goal or increased knowledge regarding the data distribution.

On the other hand, the ability to guide a totally unsupervised learning process, with arbitrary quality of supervision can be maliciously exploited. Such exploitation, may include denial-of-service or other malicious activities. Since the design of EviTraN aims at generalised application in multiple use-cases and domains, dealing with such evidence sources is critical. Malicious attacks with the use of malevolent evidence sources, could lead to ill-intended final decisions from the repurposing of latent representations produced from EviTraN. To this end, the satisfaction of the predefined criteria should be evaluated in multiple learning settings. Such as hybrid, inaccurate and incomplete learning settings.

3.3.1 Hybrid Learning

Hybrid learning is the most straightforward case of utilising auxiliary data sources or task outcomes within an unsupervised learning framework. It involves evidence sources that represent meaningful relations, which convey relevant information to the learning of the primary task. This learning setting is characterised as such, due to EviTraN being able to learn both from unlabelled data instances (primary data samples) and weakly or strongly labelled instances (task outcomes of unobserved external datasets). Involving meaningful relations from auxiliary introduced evidence sources, does not require any measures to preserve initial performance (unlike the two following learning settings). Evidence sources involved in this setting are usually relevant task outcomes produced from weak or strong supervision.

3.3.2 Inaccurate Evidence Transfer

The act of introducing additional assumptions within a learning framework, aims to utilise certain inherent properties of the available data. For example, using convolutional layers to capture spatial correlations within image data. In this example, convolutional layers typically should be more effective than fully-connected layers which can be repurposed for a variety of data. However, such assumptions may reduce the applicability of the method. In the current example, the use of convolutional layers restricts the neural network architecture exclusively to gridded data types. Furthermore, consider PCA (presented in Section 2.1.3). PCA expects its input to be linear data. Although it excels with linear data, applying PCA on non-linear data is not practical.

EviTraN involves an unsupervised primary task or target task in the context of transfer learning. Unsupervised learning lacks semantically high-level information, such as class labels. Therefore, to introduce assumptions regarding the relation between primary data and external evidence is complex, as there is a semantic discrepancy between signal-level information (primary data) and decision-level information (external evidence – auxiliary task outcomes).

To this end, EviTraN does not involve assumptions regarding the relations between primary data and auxiliary task outcomes. In practice, EviTraN should be able to involve any external categorical variable. However, allowing such degrees of freedom during selection of evidence with arbitrary distribution, quality or relation may negatively impact the learning process.

Inaccurate evidence transfer is an analogous to inaccurate weak supervision. It refers to a use-case of EviTraN, where the introduced auxiliary task outcomes do not convey information that can improve the learning of the primary task. The inability

to contribute in the process of learning the primary task, can be attributed to the evidence source being either naturally uninformative or maliciously tampered with.

The description of these two categories is described, as follows:

Naturally Uninformative Evidence. This group consists of evidence sources which do not convey relevant information for the primary task, due to inherent properties, such as *internal noise* or *uncorrelated tasks*. Internal noise involves categorical variables with high entropy, i.e., uncertainty due to noise found during the collection procedure, such as noise from sensors or labelling errors. On the other hand, uncorrelated tasks entail categorical variables, that convey irrelevant information. Despite, as outcomes of meaningful processes, they may represent non-complementary or redundant relations. Internal noise evidence, often tends to have similar distribution characteristics to uniformly distributed features, while uncorrelated tasks are harder to detect by observation of its data features.

Maliciously Uninformative Evidence. While the presence of evidence sources found within the previous group is a naturally occurring phenomenon, this group involves artificially manifested categorical variables with malicious intent, such as *artificial noise* or *tampered tasks*. These two evidence sources, are simulation of the above naturally occurring phenomenons. Artificial noise involves artificially manufactured samples, that mimic the feature characteristics of inherent noise. Tampered tasks involve tampering with meaningful auxiliary tasks, in order to introduce them in an uncorrelated manner. Example of such tampering may be for example, reorganising the order of evidence samples, in such way that is no longer corresponding with primary data samples. Both cases aim to disrupt the learning process.

3.3.3 Incomplete Evidence Transfer

In addition to feature characteristics of the evidence sources, e.g., feature values or data distribution, other characteristics can potentially impact the learning process of the primary task in a negative manner. Incomplete evidence transfer refers to the use-case of EviTraN, where introduced auxiliary task outcomes do not necessarily have a full correspondence with the primary dataset. Such evidence sources, may potentially behave similarly as inaccurate evidence sources.

Having incomplete correspondence with the primary task, incomplete evidence sources, may potentially introduce implicit bias to the learning process. As evidence source indicates external information in the form of supervision, introducing incomplete evidence sets would highlight semantically high-level features only for a portion

of the primary dataset. At the same time, if the evidence sources is missing samples only from particular external classes, it would heavily bias the learning of these classes, as the representations of their counterparts will lack high-level information. Examples of incomplete evidence sources are:

Uniformly missing task samples. This group contains all the cases where the amount of missing correspondence is present across all auxiliary classes, thus auxiliary task samples are uniformly missing. Uniformly missing samples is often a naturally occurring phenomenon. The procedure of acquiring labels for a dataset is often costly or timely. Even for cases that require simpler labelling procedure, i.e., weak labelling, one may introduce a preliminary version of the evidence source. Missing samples across evidence classes could also be a result of tampering.

Biased task outcomes. Biased task outcomes, refer to incomplete evidence sources that are missing samples from specific auxiliary classes. Having certain auxiliary classes unrepresented in the evidence source, can lead to extremely biased outcomes. Biased task outcomes, may often be attributed to certain classes being harder to identify (for manual labelling procedures) or due to missing a certain classifier (for automatic labelling procedures). However, biased task outcomes may also be manifested for malicious activities.

3.4 Deep Learning Framework

This section entails implementation details of EviTraN method. It includes description of the training strategy, that consists of three distinct steps. The above training strategy covers all the aforementioned learning settings, i.e., hybrid, inaccurate and incomplete learning. In addition, it includes description of the involved deep learning models, as well as, a step-by-step description of translating the objective of EviTraN into a deep learning framework.

3.4.1 Translating the High-Level Objective Into Deep Learning Solution

As mentioned in previous sections, the objective of EviTraN is to improve the unsupervised representation learning process by introducing auxiliary task outcomes – external evidence. However, the proper way of incorporating external knowledge within a deep neural network solution is an open question within the scientific community. Despite the lack of a universal approach for the above task [110], transfer learning is a deep learning approach capable of incorporating external knowledge.

However, the act of improving the learning process of a model is ambiguous. For example, consider the learning of a binary task with two possible class outcomes: *positive* and *negative*. Furthermore, let the evaluation criteria of the process be the F1-score. Since F1-score also involves precision and recall metrics (more detail in Chapter 6), improving the learning of the binary task can be interpreted in two ways. Either to improve the correct classification rate, i.e., the detection of true positives and true negatives or to reduce the misclassification rate, i.e., falsely classified positives and negatives.

In practice these two cases are not independent (assuming a fixed set of data instances), meaning that improving the misclassification rate should also lead to improved correct classification rate and vice versa. However, it successfully lends itself as an example of the complexity of translating high-level descriptions of objectives into deep learning training objectives and solutions.

Despite the case, one may perceive these actions as an act of conditioning an initial learning outcome. Starting from a baseline performance, one should adjust its initial mechanism into producing a different outcome. Such conditioning, should either lead to improvement of decision boundaries or in reduction of errors. To this end, cross entropy lends itself to the training objective of EviTraN, in order to translate the high-level goal into deep learning implementation.

Cross-entropy stems from the definition of the established "Kullback-Leibler Divergence (KLD)" by Kullback and Leibler [111]. Kullback-Leibler Divergence is an information theoretic metric, that measures the divergence between two data distributions. Equation 3.1 describes Kullback-Leibler Divergence, using the same notation as in Deep Learning Book [1], for consistency purposes with following definitions[1].

$$D_{KL}(P||Q) = \mathbb{E}_{x\sim P}[\log \frac{P(x)}{Q(x)}] = \mathbb{E}_{x\sim P}[\log P(x) - \log Q(x)] \tag{3.1}$$

$$H(x) = \mathbb{E}_{x\sim P}[I(x)] = -\mathbb{E}_{x\sim P}[\log P(x)] \tag{3.2}$$

$$\begin{aligned} D_{KL}(P||Q) &= \mathbb{E}_{x\sim P}[\log P(x) - \log Q(x)] \\ &= \mathbb{E}_{x\sim P}[\log P(x)] - \mathbb{E}_{x\sim P}[\log Q(x)] \\ &= \underbrace{H(P)}_{\text{Entropy}} - \underbrace{H(P,Q)}_{\text{Cross-entropy}} \end{aligned} \tag{3.3}$$

KLD involves the quantification of uncertainty within a distribution, also known as the Shannon entropy (as depicted in Equation 3.2). The Shannon entropy of a

[1]The same notation as in Deep Learning Book [1] is also used for Equations 3.2, 3.3 and 3.4

distribution is defined as: "*the expected amount of information in an event drawn from that distribution. It gives a lower bound on the number of bits (if the logarithm is base 2, otherwise the units are different) needed on average to encode symbols drawn from a distribution P*" [1].

Using Equation 3.2, KLD can be rewritten as shown in Equation 3.3. Therefore, cross-entropy (defined in Equation 3.4)[2] similar to the above self-entropy definition, it can be considered as a lower bound on the number of bits needed on average to encode symbols drawn from distribution P, where the encoding involves distribution Q. Cross-entropy is not symmetric, since it involves the Kullback-Leibler Divergence which is asymmetrical. Thus, cross-entropy quantifies the self-entropy of true distribution P and the divergence of estimated distribution Q from true distribution P.

$$H(P,Q) = -\mathbb{E}_{x\sim P}[\log Q(x)] = H(P) + D_{KL}(P||Q) \tag{3.4}$$

Since cross-entropy is asymmetrical the selection of true and estimated distribution is critical. For simplicity purposes, let the set of external categorical evidence sources be $\mathcal{V} = \{V_1\}$ and thus $L = 1$. V_1 is a task outcome retrieved from a decision model f_1, through observation of external dataset ϵx_1, as well as, involving external model parameters ψ_1. Despite V_1 being an observable variable, the rest of the components are unobserved, i.e., f_1, ϵx_1 and ψ_1. Thus, from the perspective of EviTraN, V_1 is fixed. In addition, training iterations of primary task learning does not affect the auxiliary task outcomes. Thus, selecting evidence sources as the true distribution is appropriate. On the other hand, latent representations Z are a product of a generative model. Generative model G, produces latent representations from a parametric family of distributions that involve trainable parameters θ. Since training iterations affect the outcome of primary task, selecting Z as the estimated distribution is fitting.

Cross-entropy is an appropriate metric that allows for simultaneous satisfaction of both the effectiveness and robustness criteria. Ultimately, the cross-entropy $H(V, Z)$ between true distribution V_1 and estimated distribution $G(X, \theta) = Z$ quantifies the entropy within external evidence source V_1, as well as, the divergence of estimated distribution from true distribution.

Taking into account the cross-entropy between evidence source(s) and the latent space can lead to two possible outcomes: minimisation of cross-entropy or decaying. If V_1 is an evidence source that conveys relevant information to the primary task, then the entropy should be a constant number, since V_1 outcome is invariant to training iterations. At the same time, latent representations Z are conditioned to minimise their divergence to V_1. Consequently, the minimisation of the objective should lead

[2]Not to be confused with **joint entropy** that is frequently notated in the same manner.

to improved maximum likelihood estimation of the primary task. Therefore, minimisation of cross-entropy allows the satisfaction of effectiveness criterion.

Alternatively, if V_1 does not convey any useful information in regard to the primary task, its entropy remains unchanged and therefore is constant. However, the divergence between V_1 and Z should remain stable or increase, since conditioning with irrelevant task outcomes is not feasible. Stability or growth of cross-entropy is a good indication of low quality evidence. This indication, can be utilised in order to satisfy the robustness criterion (more details are following in Section 3.4.2).

To satisfy the Modularity criterion, EviTraN as a transfer learning method should learn to adapt into new auxiliary tasks without forgetting the primary task. EviTraN should be carefully designed in order to avoid *Catastrophic Forgetting*. Catastrophic forgetting [112] is a phenomenon that occurs in transfer learning methods. Target models that suffer from catastrophic forgetting, underperform in the original task after being trained to learn new auxiliary ones. To this end, the training strategy of EviTraN belongs to families that do not suffer from catastrophic forgetting (more details are following in Subsection 3.4.3). Therefore, satisfying the modularity criterion by not requiring the presence of previously introduced evidence sources and iteratively learning with each new available evidence source.

3.4.2 Training Strategy

The training strategy of EviTraN is depicted in Figure 3.3, presented in the logic of a workflow diagram. EviTraN consists of three steps: initialisation, intermediate and evidence transfer steps.

The first step in the training strategy of EviTraN is the *initialisation* step. During initialisation the generative model of choice, i.e., an autoencoder, learns initial latent representations. In this step, the base autoencoder is trained in a completely unsupervised manner. Similar to other representation learning methods, the learned representations are repurposed towards an end-goal. Repurposing latent representations without the introduction of external evidence sources is a baseline solution. The training objective that is used during initialisation step, is the reconstruction of input depicted in Equation 3.5.

$$L_{init} = MSE(X_{in}, X_{out}) = \frac{1}{N}\sum_{i=1}^{N}(x_{in}^{(i)} - x_{out}^{(i)})^2 \tag{3.5}$$

The subsequent steps depend on the presence of available external evidence sources. After the initialisation step, an intermediate step between evidence transfer and initialisation is present. *Intermediate* step aims to filter each individual evidence source before its use in evidence transfer. Although cross-entropy may ensure some satis-

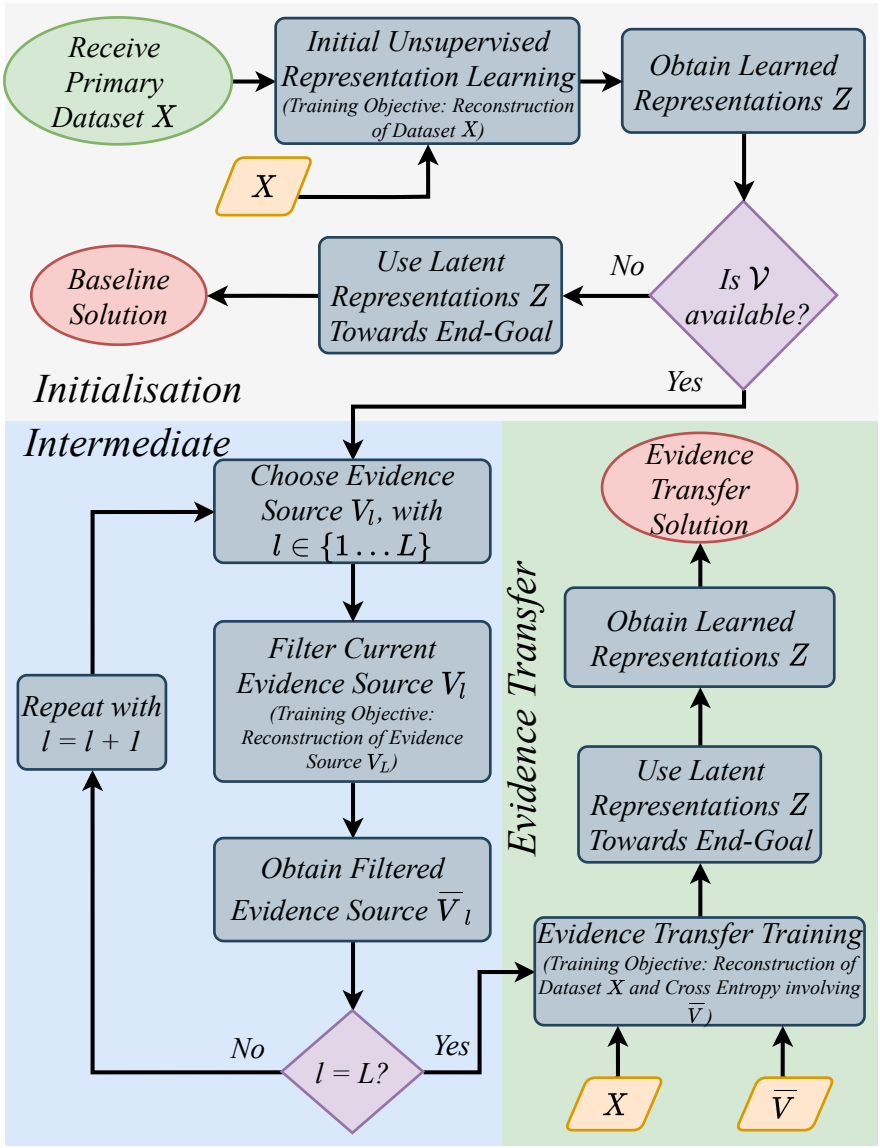


Figure 3.3: Training strategy of EviTraN depicted in the logic of a Workflow diagram.

faction of the predefined criteria, it focuses on the feature distribution properties of external evidence. Due to that, certain non-meaningful evidence may not be detected.

A clear example of such case is maliciously uncorrelated tasks. Tampered meaningful auxiliary tasks introduced in an uncorrelated manner, may appear as normal task outcomes from a statistical perspective. Studying its properties should bear no differences from any other task outcome. However, from a high-level perspective, e.g., semantic correlation, the introduced task may be completely uncorrelated or be introduced in a non-corresponding manner. Thus, a mechanism that will aid the quantification of the cross-entropy in such cases, is required.

To this end, intermediate step aims to transform evidence with high-level incon-

sistencies into samples with appropriate distribution, that will aid their identification through cross-entropy. Evidence autoencoder (Figure 3.6) is build with restricted capacity. It is a shallow model with a small number of nodes within each layer. In addition, evidence autoencoder is trained in a biased manner. Evidence autoencoder is optimised only for a small amount of epochs. The idea behind bias training is based on the assumption that evidence sources which represent meaningful relations, are characterised by consistency. On the other hand, low quality of evidence is inconsistent. In addition, it can either be observed through its distributional properties (inherent noise) or by its inconsistencies (uncorrelated tasks).

For each evidence source, a shallow evidence autoencoder is trained with bias. After training, the intermediate step produces filtered evidence sources, which are extracted from the bottleneck of evidence autoencoder. In other words, the filtered evidence sources are latent representations acquired from a biased shallow autoencoder.

If the evidence source represents a meaningful relation, then the distribution of shallow autoencoder bottleneck is similar to the distribution of raw evidence. Due to consistency in meaningful relations, shallow autoencoder can learn to reconstruct the evidence distribution even for low amounts of epochs. On the other hand, if the evidence distribution is of low quality, e.g., random values or white noise, then the biased autoencoder can not generalise and produces representations with distribution properties similar to Uniform distribution. If the evidence source represents low quality of evidence from a high-level perspective, e.g., uncorrelated tasks, then the reconstruction also leads to a uniform-like distribution, similar to white noise type of evidence. The ability of biased evidence autoencoder to behave similarly in white noise and uncorrelated tasks, is due to shallow evidence autoencoder not being able to generalise for inconsistent input. The evidence autoencoder training objective is depicted in Equation 3.6.

$$L_{intermediate} = MSE(Vin_l, Vout_l) = \frac{1}{M}\sum_{j=1}^{M}(Vin_l^{(j)} - Vout_l^{(j)})^2 with\ l\ \in \{1, \dots, L\} \tag{3.6}$$

$$L_{ET} = \underbrace{L_{primary}}_{=L_{init}} + \lambda * L_{aux} \tag{3.7}$$

After all evidence sources have been filtered through the intermediate step, evidence transfer step involves filtered evidence values, instead of their raw features. The last step acts as the process of involving external knowledge, in the form of auxiliary tasks in the initially unsupervised learning process. Evidence transfer step transfers

knowledge from external evidence to the learning process of the primary task. The training objective of evidence transfer consists of a composite learning objective. The objective of evidence transfer (Equation 3.7) simultaneously minimises the initial objective, which is the input reconstruction (L_{init}), and the cross-entropy between an extension of latent space (auxiliary introduced layers) and filtered evidence sources (Equation 3.8).

$$L_{aux} = \frac{1}{L} \sum_{l=1}^{L} H(\overline{V}_l, Q_l) \tag{3.8}$$

Evidence transfer introduces an additional hyperparameter to regulate the minimisation of composite objective. Alternatively, if a particular use-case requires more stabilisation regarding the involvement of two objectives two hyperparameters can be used (Equation 3.9).

$$L_{ET} = \lambda_1 * \underbrace{L_{primary}}_{=L_{init}} + \lambda_2 * L_{aux} \tag{3.9}$$

The composite objective of evidence transfer serves a dual purpose. During cases where the auxiliary task outcome represents a meaningful relation, then the composite objective enables the autoencoder to not forget its primary task. In other words, the autoencoder is restricted from the composite objective to incorporate auxiliary information into the representation learning process, instead of lazily being led to only learn auxiliary introduced tasks. In addition, the composite training objective enables modularity, by assimilating the auxiliary task into the primary one. On the other hand, if low quality of evidence is introduced instead, the cross entropy should remain stable due to weight decaying of newly introduced auxiliary layers Q (presented in Section 3.4.3). Therefore, minimisation of the composite objective falls back on minimisation of the primary task objective. Thus, the training procedure is similar to the initial step, which allows the model to return in its original state without disturbance of the original effectiveness.

Incomplete Training. During incomplete evidence transfer the correspondence between evidence sources and primary dataset is incomplete. In other words, $M < N$ where M is the total number of available samples for each evidence source and N is total amount of available primary data samples. Similar to the notation that is frequently used during incomplete supervision frameworks, let a subset of the primary dataset X be $X_{unlabelled} = \{x^{(1)}_{unlabelled}, \ \cdots, \ x^{(N-M)}_{unlabelled}\}$. In this case the term unlabelled refers to primary data samples that have no corresponding evidence. Alternatively, evidence samples that correspond to unlabelled data samples

are missing, i.e., $V_{unlabelled} = \{\ \}$. Also let a subset of the primary dataset X be $X_{labelled} = \{x^{(1)}_{labelled},\ \dots,\ x^{(M)}_{labelled}\}$. The term labelled refers to the labelled subset having corresponding evidence sources $V_l = \{V_l^{(1)},\ \dots,\ V_l^{(M)}\}$ with $l \in \{1, \cdots, L\}$ and $V_{labelled} = \{V_1,\ \dots,\ V_L\}$. The training strategy during incomplete training is similar to previously defined training strategy. First, initialisation step is carried out for all data samples in primary dataset X, since it is an unsupervised procedure. Intermediate step is also carried out similarly for all available evidence sources. However, during evidence transfer step only the labelled data samples are used (Equation 3.10).

$$L_{ET-incomplete} = MSE(Xin_{labelled}, Xout_{labelled}) + \lambda * L_{aux} \tag{3.10}$$

Alternatively if necessary two hyperparameters can be used:

$$L_{ET-incomplete} = \lambda_1 * MSE(Xin_{labelled}, Xout_{labelled}) + \lambda_2 * L_{aux} \tag{3.11}$$

3.4.3 Models

EviTraN uses Autoencoders in order to generate latent representations. For consistency with previously depicted EviTraN overview (Figure 3.1), $G(X, \theta)$ is an autoencoder with trainable parameters θ. Autoencoders are feed-forward neural networks that are widely used in order to learn dimensionally lower, frequently linear and meaningful latent representations from raw data. They are trained in an unsupervised manner, by training to minimise input reconstruction error.

As EviTraN is designed for general applications, multiple variations of autoencoders can be involved. The experimental evaluation that is following in Chapter 5, involves a Convolutional and a Stacked denoising autoencoder variation. Figure 3.4 and 3.5 depict both autoencoder variations.

The selection of hyperparameters in the topology of Convolutional Autoencoder, is based on the same idea as the base Convolutional Autoencoder used in DCEC [44]. The idea is to utilise convolutional layers in a hierarchy where the layers move from bigger convolutional windows to smaller, as they approach the autoencoder bottleneck. The convolutional autoencoder variation is regularised with inner dropout, i.e., transforming a random portion of the convolutional filter windows before being fully connected to a dense layer. Stabilisation of the training procedure, requires the use of Batch Normalisation layers. Batch normalisation layers, perform mini-batch normalisation to enable stochastic optimisation.

The stacked denoising autoencoder variation consists mainly of fully connected

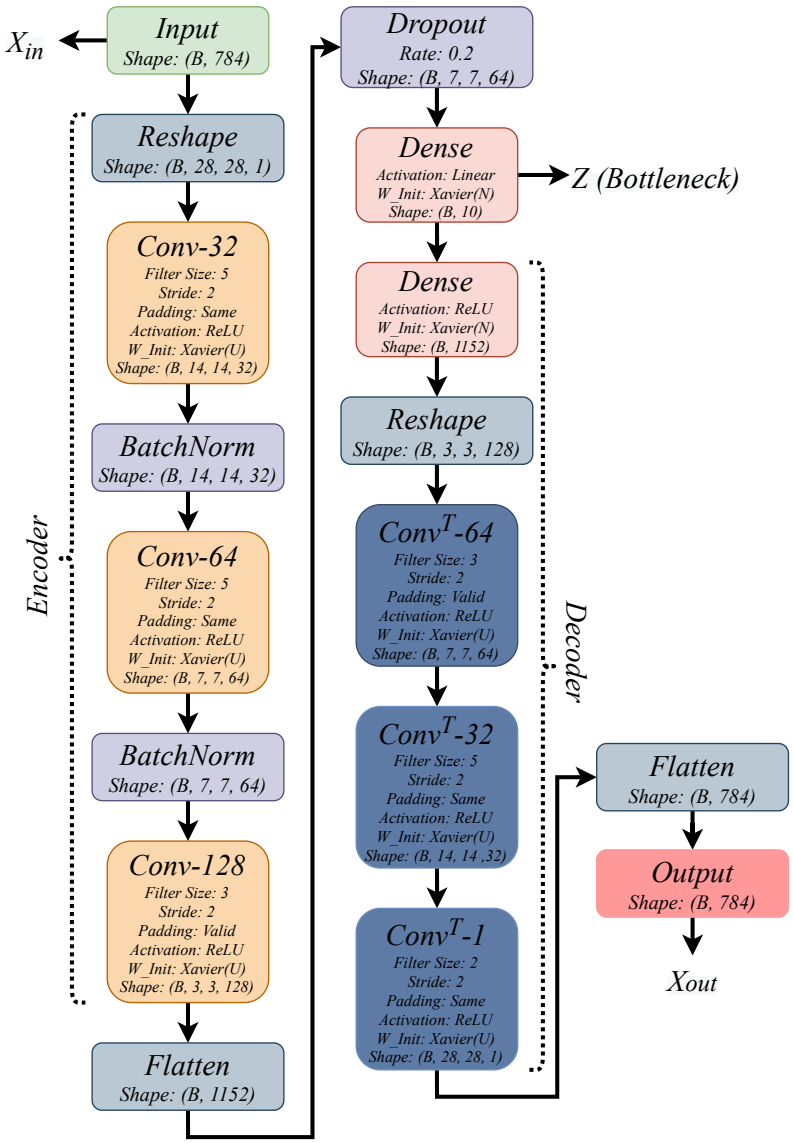


Figure 3.4: Convolutional variation of base autoencoder involved in the primary task of EviTraN. *Conv* represents a convolutional layer along with its number of filters. Similarly, $Conv^T$ are transpose convolutional layers. This neural network configuration is used as is during initialisation step.

layers (also known as dense). Dropout is following the input layer. However, some use-cases may require a batch normalisation layer before dropout. The selection of hyperparameters in the topology of stacked denoising autoencoder, is based on the same principal as the autoencoder used in DEC [24]. The training procedure is the standard process of training stacked denoising autoencoders. It consists of performing greedy layer-wise pre-training [26], followed by end-to-end training. During greedy layer-wise training, the autoencoder is trained in pairs of shallower autoencoders, that are part of the deeper end-to-end autoencoder. Greedy layer-wise autoencoder

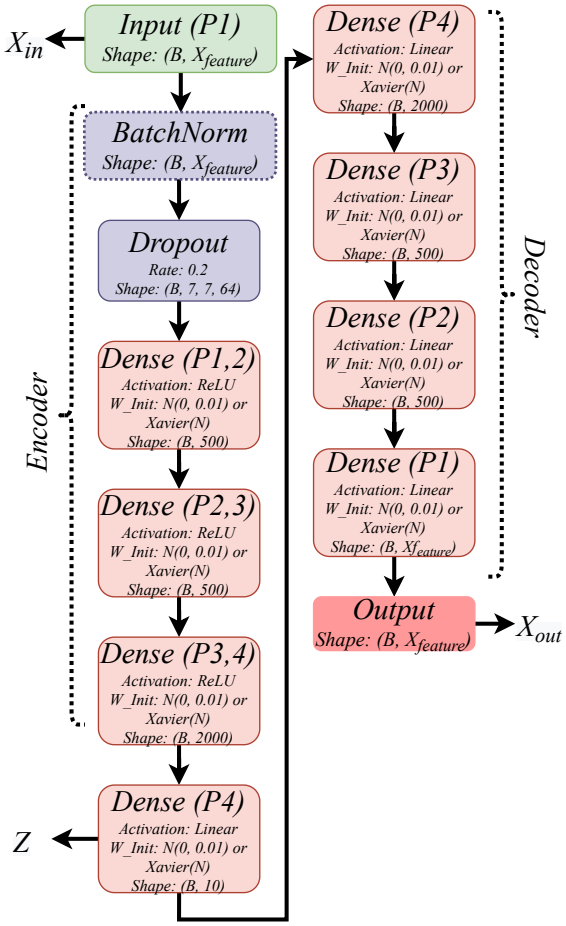


Figure 3.5: Stacked denoising variation of base autoencoder involved in the primary task of EviTraN. Dotted line around the first batch normalisation layer indicates that its use, depends on the use-case. Notation P represents smaller encoder pairs that are deployed during greedy layer-wise training [26]. This neural network configuration is used as is during initialisation step.

has been shown to lead in better performance.

Evidence autoencoder (as shown in Figure 3.6) used during the intermediate step between initialisation and evidence transfer steps is a simple shallow autoencoder. It consists of three fully connected layers. Evidence autoencoder is restricted in terms of capacity, i.e., network depth and layer width, in order to effectively transform certain cases of low quality evidence sources into new "filtered" distributions.

Additional layers are introduced to the decoders of the base autoencoder models, during evidence transfer step (as shown in Figures 3.8 and 3.7). Additional layers serve the purpose of incorporating external knowledge introduced in the form of external evidence. Depending on the width of the output layer, additional compression in the form of layers may be required, e.g., additional dense layers.

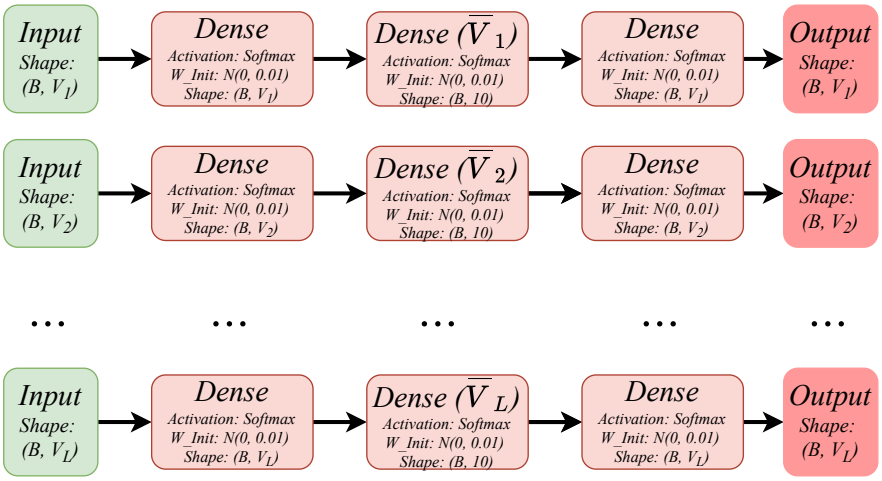


Figure 3.6: Shallow "biased" evidence autoencoder present during the intermediate step of EviTraN.

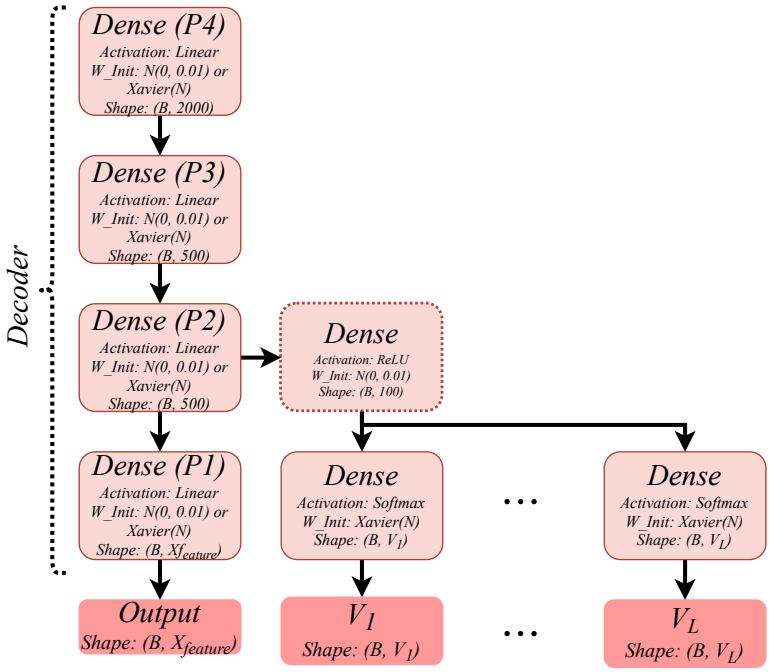


Figure 3.7: Decoder adaptation during evidence transfer step of EviTraN method. This neural network configuration depicts the adjustment of layers of the stacked denoising variation.

Li and Hoiem [113] studied the ability of certain transfer learning setups to retain knowledge from their initial task, after introduction of a new task. Evidence transfer is similar to the joint training family of methods, i.e., layers from the initial task are fine-tuned along with randomly initialised layers involved in the new task. The authors conclude that joint optimisation frameworks are able to retain knowledge in

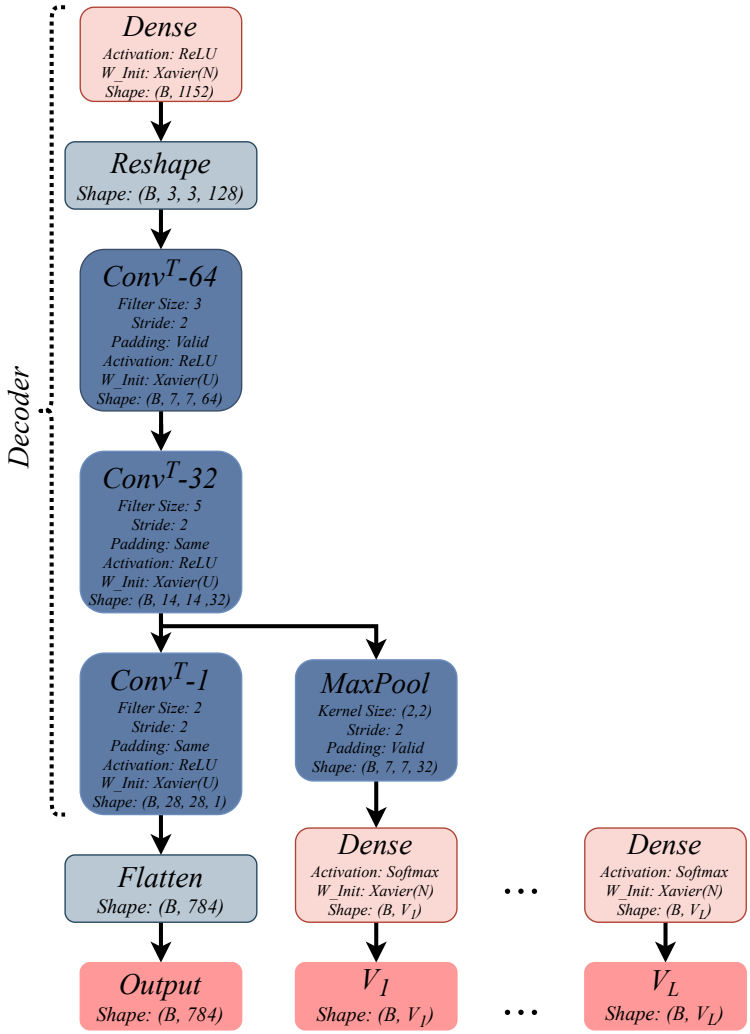


Figure 3.8: Decoder adaptation during evidence transfer step of EviTraN method. This neural network configuration depicts the adjustment of layers of the convolutional variation.

the initial performance. In other words, joint optimisation frameworks do not suffer for catastrophic forgetting (also indicated from qualitative evaluation in Chapter 5).

The next chapter presents a theoretical interpretation of the effects of EviTraN in the latent features, through comparison with the well-received information theoretic *information bottleneck* method.

Chapter 4

Effects of Evidence Transfer

This chapter contains an information theoretic interpretation of the effects of EviTraN on the latent space of an autoencoder. To this end, it contains an introduction to the concepts of model interpretability and explainability, which is otherwise known as *explainable artificial intelligence*. In addition, it includes an introduction to an information theoretic method called *Information Bottleneck*. Information bottleneck lends itself to an interpretation of the effects of EviTraN, through comparison of their common properties.

4.1 Towards Explainable Models

This section includes motivation regarding the use and manifestation of explainable and interpretable models. Furthermore, it includes introduction to Information Bottleneck, that lends itself as a mean of interpreting the inner workings of deep learning methods.

4.1.1 Interpretability of Deep Learning

Deep learning models are notorious for being very effective in a variety of applications, including but not limited to biological data mining [114], drug-target interactions [115] or medicine [116]. Despite their popularity, due to being more effective than their traditional machine learning counterparts, deep learning models are infamous for being *black box* systems. A black box system [117] is a system that includes inner processes with unknown functionality. In turn, unknown functionality of the inner processes, constraints the perception of these systems into a set of decisions, produced based on observation of a corresponding set of inputs. Despite the use of deep learning in a plethora of applications, the cultivated black box perception of deep learning models is not suitable for some applications. Explaining why a deep learning

model is effective or what logic hides behind each prediction outcome, has become quite complex for researchers, machine learning engineers or practitioners. Yet, to some extent the attributes that make deep learning models effective are known.

A relevant study regarding the use of deep learning for computer vision [118], has highlighted a multitude of aspects that make deep learning effective. The experimental study, suggests that deep learning is effective due to: representation learning, training with voluminous data and model capacity. As discussed in Section 2.1.6, representation learning is a vital process of deep learning. Deep learning models are able to learn increasingly high-level representations, thanks to their incremental depth and aggregation of previous layers. In turn, it allows the learning of complex concepts, such as shapes within an image, from observation of low-level features such as colour values.

The rapid progress of deep learning methods and the rise of *big data*, played a vital role in the advancements of artificial intelligence over the years [119, 120]. Big data is a rather popular term which is frequently connected with the notion of dealing with large-scale datasets. Although frequent use of the term in multiple scenarios, suggested the requirement of a more accurate definition. Numerous organisations and individuals have proposed various definitions of big data [121]. However, the most widespread one, is characterising big data by Vs. The definition of Vs suggest that big data is a multi-dimensional notion that involves three constituent concepts: **volume, variety and velocity** [122, 123]. A fourth and a fifth V have also been considered, namely **veracity** [124] and **value** [125].

Each component describes a different characteristic of the data. *Volume* refers to the size of the data collection. *Variety* refers to the availability of a multitude of data types. *Velocity* refers to the pace of generation, collection, and process of the data. *Veracity* refers to uncertainty or ambiguity found within a data collection, while *Value* refers the procedures revolving around repurposing of the outcomes produced by processing of big data. The incorporation of multi-faceted data collections such as big data, allows deep learning models to effectively generalise, due to high amount of data samples, multiple perspectives from various data types and accurate sources.

Models with high enough capacity are necessary for the learning of complex tasks. Yet, there are indications that increasing the amount of training data leads to more scalable solutions, than increasing the capacity of the model [126]. To this end, an appropriate level of capacity should be considered, instead. Models with low capacity are computationally inexpensive, as they consist of shallow layer compositions. However, they require multiple training iterations in order to reach convergence of the training objective. On the other hand, models with high capacity, converge quicker but are computationally expensive. Selecting appropriate levels of capacity is the middle ground in the trade-off between training iterations and required computational

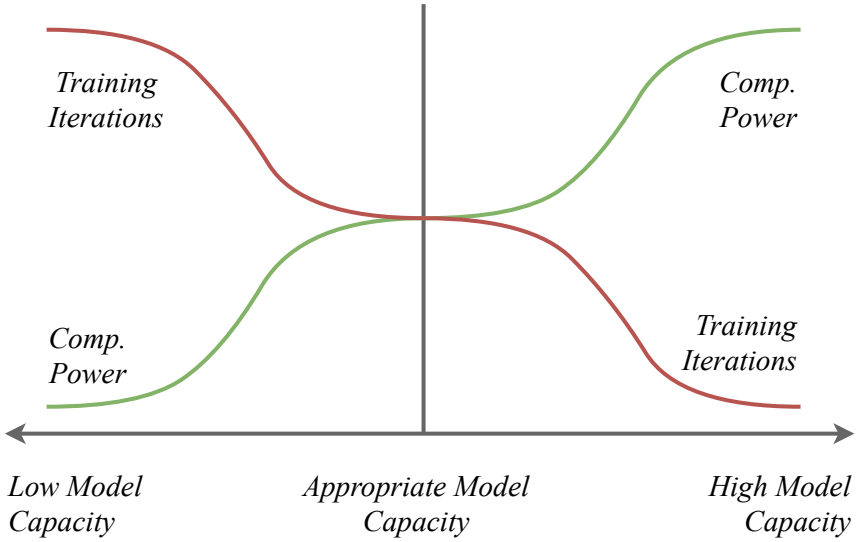


Figure 4.1: Visual example depicting relation between model capacity, computational power required and training iterations. Computational power required for training increases with higher model capacity, while training iterations required to reach convergence decrease with higher model capacity. Appropriate model capacity is the middle ground between both. Note that lower capacity models may never reach appropriate levels of convergence, despite the plethora of training iterations.

power. A visual example of the relations between model capacity, computational power required and training iterations is depicted in Figure 4.1.

The deep learning aspects highlighted by Sun et al. [118], are generally accepted by the scientific community as attributes that make deep learning effective. However, these aspects are rather broad, and do not provide any knowledge regarding the functionality of the model mechanisms. Model capacity and training with big data are aspects of training rather than the learning process. Representation learning, although it provides some insight regarding the learning mechanisms of deep learning (learning complex concepts, incrementally, from learning of simpler constituent parts), it does not allow backtracing to relevant features. To some applications, where incorrect predictions may lead to serious implications such as healthcare [127, 128] or industry [129]. Being knowledgable regarding inner procedures of the model, such as learning mechanisms, most relevant features or features involved in the prediction, is critical. Samek et al. [130] suggested that explainable artificial intelligence enables the following properties: "*verification of the system, improvement of the system, learning from the system*" and "*compliance to legislation*".

However, as suggested by Section 3.4.1, translating high-level objectives into deep learning solutions involves ambiguity. In turn, the ambiguity leads to dissociation between the training objectives that guide the process of learning latent representa-

tions and the end-goal (application). In practice, this means that the minimisation of such objectives is not an indication of relevance. For example, minimising the reconstruction error, which is frequent in unsupervised learning, does not indicate which features are relevant to later repurposing of the learned representations, such as, compression, clustering or disentanglement. This is also suggested by Lipton [131], that motivates "*Model Interpretability*" as an urgency that originates from the discrepancy between training objective and application/real world/human-interpretable objective. In addition, the author indicates that the model interpretability should be humanly interpretable as it aims to stakeholders.

An information theoretic method which can contribute to explaining or interpreting the learning process of deep learning models is *Information Bottleneck*.

4.1.2 Information Bottleneck

The information bottleneck (IB) method [39] originated from the domain of information theory. Information bottleneck is a formalisation of a rather simple problem: How to preserve the "relevant" information of a signal after compressing into a short code? However, the solution to such a problem is more complex. In order to preserve the "relevant" information of a signal, one should first quantify the relevance of a signal. Yet, according to Tishby et al. [39], original formulations of information theory [132] did not include concepts of relevance, but rather concentrated on the act of transmitting information, independent of its relevance.

To preserve relevant information, one should understand the concepts that make a signal meaningful. Signals by themselves are neither relevant nor irrelevant. Measuring the relevance of a signal depends on predefined criteria. For example, retrieving the most relevant articles for an academic study, depends on the scope of the study. Drawing inspiration from various applications and domains, Tishby et al. [39] conclude that the criterion based on which one may classify a signal as relevant, depends on auxiliary (or additional) variables. In other words, a signal is relevant, if it transmits information that is significant for the task of predicting an auxiliary variable. For instance, an image is relevant to the task of detecting household items, if it contains data features that depict a household item.

Information bottleneck formalises the trade-off between compression and preservation of relevant information. Transforming a high-dimensional signal into a low-dimensional code, that only contains relevant information is desirable for two reasons. First, lower-dimensional codes are more compact, and thus dealing with them is more efficient in terms of computational resources. Second, preserving only the information that is relevant, reduces the presence of redundant features. As discussed in Section 2.1.11, dealing with redundant features or signals is not desirable.

The implementation of the method involves three components: a primary signal, a code and an additional variable. Primary signal consists of a *signal space* X, probability measure $p(x)$ and values $x \in X$. Information bottleneck aims to find a mapping between x and $\tilde{x} \in \tilde{X}$, where $\tilde{X}$ is the code space. The mapping between x and $\tilde{x}$ is a conditional probabilistic distribution function $p(\tilde{x}|x)$. Mutual information $I(X;\tilde{X})$ determines the quality of mapping. Yet, without involving an additional variable, the mapping between primary signal X and code $\tilde{X}$ only represents a compression process that may discard meaningful aspects of the primary signal. Thus, introducing auxiliary variable Y.

The introduction of Y formalises a new objective. In order to condition the original compression process between X and $\tilde{X}$ to also capture relevant information to Y, an additional term is required. Mutual information $I(X,Y)$ determines the relevant information between two signals. However, since the aim of the method is to find a relevant and compressed code, the mutual information $I(\tilde{X};Y)$ is more appropriate. Therefore, the method aims to find optimal mapping $p(\tilde{x}|x)$ by minimisation of Equation 4.1, with respective terms found in Equation 4.2 and 4.3. Note that β is a Lagrange multiplier [133], that regularises the trade-off between compression and preservation.

$$\mathcal{L}[p(\tilde{x}|x)] = I(\tilde{X};X) - \beta I(\tilde{X};Y) \tag{4.1}$$

$$I(X;\tilde{X}) = \sum_{x\in X}\sum_{\tilde{x}\in\tilde{X}} p(x,\tilde{x}) \log\left[\frac{p(\tilde{x}|x)}{p(\tilde{x})}\right] \tag{4.2}$$

$$I(\tilde{X};Y) = \sum_{y}\sum_{\tilde{x}} p(y,\tilde{x}) \log\frac{p(y,\tilde{x})}{p(y)p(\tilde{x})} \leq I(X;Y) \tag{4.3}$$

The above notation and terminology is consistent to that found in the work of Tishby et al. [39].

Information bottleneck lend itself as an explanation for the effectiveness of deep learning. Shwartz-Ziv and Tishby [134] suggested that the process of learning optimal lower-dimensional representations, which are repurposed for the task of predicting an auxiliary variable is the same principal as information bottleneck. From their study of the "*Information Plane*" of deep neural networks, among others, found that most effort of the training process is spent on compression and that input-intermediate and intermediate-output mappings satisfy the equations of information bottleneck. Information plane is the plane formed by mapping several latent representations from various layers (output of hidden layers) into points with coordinates ($I(X;T)$, $I(T;Y)$) [135]. Variable T represents the current depth representation. The above

points in plane follow the data processing inequality [56], as presented in Equations 4.4 and 4.5.

$$I(X;Y) \geq I(T_1;Y) \geq I(T_2;Y) \geq ... \geq I(T_k;Y) \geq I(\hat{Y};Y) \tag{4.4}$$

$$H(X) \geq I(X;T_1) \geq I(X;T_2) \geq ... \geq I(X;T_k) \geq I(X;\hat{Y}) \tag{4.5}$$

The above notation and terminology is consistent to that found in the work of Shwartz-Ziv and Tishby [134].

4.2 Information Theoretic Interpretation of Evidence Transfer

This section includes the investigation of the similarities between EviTraN and IB, which share common characteristics. In order for the two methods to lead to similar effects in the latent space (or short code, as mentioned in IB), their objectives should be equivalent.

IB involves the concept of a "short code". The concept of code is an analogous to a latent representation. In the original work of Tishby et al. [39], an implicit mention to this analogy is present[1], while explicit mentions of that analogy are made in the later work of explaining the effectiveness of deep neural networks [134]. Therefore, both methods are representation learning methods. They aim to learn a mapping process that converts data samples from a high-dimensional space to a lower-dimensional, compact and meaningful space.

$$\begin{aligned} L_{IB} &= \underbrace{I(\tilde{X};X)}_{compression} - \underbrace{\beta I(\tilde{X};Y)}_{relevance} \\ L_{ET} &= \underbrace{MSE(X_{in},X_{out})}_{L_{init}} + \lambda * \underbrace{H(\overline{V},Q)}_{L_{aux}} \end{aligned} \tag{4.6}$$

At the same time, both methods utilise auxiliary variables that condition a primary task. Such condition is able to steer the learning process of the primary task, to more efficient outcomes. With the concept of efficiency varying between the two methods. For IB, the conditioning of learning a mapping of high-dimensional data into short codes, is required in order to provide guidelines as to what features are relevant. On the other hand, EviTraN requires conditioning of learning representations that will incorporate external knowledge from auxiliary tasks. Such knowledge may

[1] "*For each value of $x \in X$ we seek a possibly stochastic mapping to a* ***representative****, or a codeword in a codebook, $\tilde{x} \in \tilde{X}$*" [39]

lead to more accurate results during repurposing of learned representations in other down-stream tasks.

The investigation of the hypothesis regarding the equivalency of the two training objectives, requires their comparison, which is shown in Equation 4.6. The confirmation of the hypothesis, requires investigation regarding the equivalency of the two individual terms found in both objectives. Therefore, to investigate if EviTraN is able to compress the original information found in primary dataset X into latent code Z through learning of L_{init} objective. At the same time, if L_{aux} is able to preserve relevant information dictated by external evidence V. Assuming that the short code of IB is the latent space of an autoencoder the objective is transformed as shown in Equation 4.7.

$$L_{IB} = I(Z;X) - \beta I(Z;Y) \tag{4.7}$$

$$\begin{aligned}
&\xrightarrow{(4.11)} I(Z;X) = H(Z) - H(Z|X) = H(X) - H(X|Z) \\
&\xrightarrow{(4.12)} = H(X) - \left[- \sum_{(z \in Z, x \in X)} p(z,x) \log \frac{p(z,x)}{p(z)} \right] \\
&= H(X) + \sum_{(z,x)} p(z,x) \log \frac{p(z,x)}{p(z)} \\
&\xrightarrow{(4.13)} = H(X) + \sum_{(z,x)} p(z,x) \log p(z,x) - \sum_{(z,x)} p(z,x) \log p(z) \\
&= H(X) + \underbrace{\sum_{(z,x)} p(z,x) \log p(z,x)}_{\mathbb{E}_{p(z,x)}} - \underbrace{\sum_{(z,x)} p(z,x) \log p(z)}_{\mathbb{E}_{p(z,x)}} \\
&= H(X) + \mathbb{E}_{p(z,x)} \log p(z,x) - \mathbb{E}_{p(z,x)} \log p(z) \\
&\xrightarrow{(4.14)} = H(X) + \mathbb{E}_{p(z,x)} \log p(x|z)p(z) - \mathbb{E}_{p(z,x)} \log p(z) \\
&\xrightarrow{(4.13)} = H(X) + \mathbb{E}_{p(z,x)} \log p(x|z) + \mathbb{E}_{p(z,x)} \log p(z) - \mathbb{E}_{p(z,x)} \log p(z) \\
&= \underbrace{H(X)}_{const.} + \mathbb{E}_{p(z,x)} \log p(x|z) \\
&= \underbrace{H(X)}_{const.} + \mathbb{E}_{p(x,z)} \log p(x|z)
\end{aligned} \tag{4.8}$$

The primary training objective of EviTraN is the reconstruction error of the autoencoder framework. Autoencoders lend themselves to compression applications [136, 137, 138, 139], as they are able to learn meaningful mappings from high-

dimensional data to a lower-dimensional space. Due to the learning of these mappings (through reduction of the reconstruction error), they can convert high-dimensional data into lower-dimensional codes and vice versa. As EviTraN utilises autoencoders as the generative model of choice, it also inherits such properties.

Unravelling of Equation 4.7 should indicate if the compression term of IB is equivalent to the reconstruction error objective of EviTraN. Equation 4.8 depicts the unravelling process. To increase reading comprehension and self-containtment of this Section, information theory equations presented during Section 2.1.10, are also repeated in Equations 4.11, 4.12, 4.13, 4.14.

Mutual information can be expressed as other well-known metrics from information theory, such as self-entropy and conditional entropy (as defined in Section 2.1.10 and shown in Equation 4.11). In addition, mutual information is symmetric. Unravelling the conditional entropy term further and utilising logarithmic properties, leads to the composition of two terms. Conditional entropy consists of the expected value of joint probability between latent representations z and primary data samples x minus expected value of the prior of latent representations z, with the expectation being taken over joint probability of z and x. Using the Bayes' rule, results in the expected value of conditional probability of x given representation z. Therefore, the mutual information between primary data and latent representations is a composition of the self-entropy of primary data and the expected value of conditional probability $p(x|z)$.

Therefore, optimisation of mutual information between latent representations and primary data depends on the optimisation of self-entropy $H(X)$ and $\mathbb{E}_{p(z,x)} \log p(x|z)$. Primary data X are not affected by training and therefore is considered as constant. Thus, in order to find a successful mapping between primary data and latent space, we should increase the conditional log-probability $p(x|z)$ over joint probability expectation.

A deterministic view of autoencoders was prevalent, however recent advances in the autoencoder framework perceive them from a probabilistic point of view [15, 1, 140]. Perceiving the encoder and decoder functions as two probabilistic models (Equation 4.9), leads to two distinct objectives. The encoder aims to maximise the probability of $p_{encoder}(Z = z|X = x)$. Which is the probability of producing a specific representation z from observation of primary data sample x. At the same time, the decoder aims to maximise the probability of $p_{decoder}(X = x|Z = z)$. Deep learning optimisation of the above of objectives with the use of stochastic gradient descent would require minimising:

$$\begin{aligned} \mathcal{L}(\theta) &= -\mathbb{E} \log p(z|x) \\ \mathcal{L}(\phi) &= -\mathbb{E} \log p(x|z) \end{aligned} \tag{4.9}$$

With θ and ϕ being the trainable parameters of encoder and decoder respectively. Autoencoder training aims to learn minimise both objectives simultaneously. Since, for the decoder to produce realistic data samples, requires the encoder to produce successful mappings.

EviTraN makes use of denoising autoencoder. Denoising autoencoders are generative models [51], which means that learned latent code, aims to approximate the true generative distribution of the primary data. Equation 4.10 depicts the objective of denoising autoencoders as iterated by both Bengio et al. [51] and Goodfellow et al. [1]. The notation found in Equation 4.10 is similar to the respective references, for consistency purposes.

$$\begin{aligned} \mathcal{L}(\theta) &= -\mathbb{E}_{P(X,\tilde{X})}\left[\log P_\theta(X|\tilde{X})\right] \text{ [51]} \\ &= -\mathbb{E}_{x\sim\hat{p}_{data}(x),\tilde{x}\sim C(\tilde{x}|x)} \log p_{decoder}(x|h = f(\tilde{x})) \text{ [1]} \end{aligned} \tag{4.10}$$

Probability $P_\theta(X|\tilde{X})$ represents the reconstruction distribution of the autoencoders which utilises the corrupted version of X. Parameters θ represent the trainable parameters of the autoencoder. Despite not being explicitly mentioned in the work of Bengio et al. [51], the latent representations are involved in the training objective, since the process of decoding requires encoding first. Goodfellow et al. [1] explicit mentions the involvement of latent representations z. $\tilde{X}$ represents the corrupted version of primary data, $C(\tilde{x}|x)$ is the corruption function, while h is the output of hidden layer (z latent representations). Function $f(\tilde{x})$ represents a deterministic version of the encoding process. The above comparison of the training objective of denoising autoencoders (Equation 4.10) and unravelled objective of IB (Equation 4.8) indicates that the compression term of IB is equivalent to the minimisation of reconstruction error performed in EviTraN.

The following chapter, includes the experimental evaluation of EviTraN, as well as, investigation of the equivalence between auxiliary learning objective of EviTraN and relevance term of IB through empirical analysis.

Repeated Equations From Section 2.1.10

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ I(Y;X) &= H(Y) - H(Y|X) \end{aligned} \tag{4.11}$$

$$\begin{aligned} H(Y|X) &= -\sum_{(x,y)} p(x,y)\log p(y|x) \\ \xrightarrow{(4.14)}\ H(Y|X) &= -\sum_{(x,y)} p(x,y)\log\frac{p(x,y)}{p(x)} \end{aligned} \tag{4.12}$$

$$\begin{aligned} \log_a(x * y) &= log_a x + log_a y \\ \log_a\left(\frac{x}{y}\right) &= log_a x - log_a y \end{aligned} \tag{4.13}$$

$$\begin{aligned} P(x|y) = \frac{P(x)P(y|x)}{P(y)} = \frac{P(x,y)}{P(y)} \rightarrow P(x,y) &= P(x|y)P(y) \\ P(y|x) = \frac{P(y)P(x|y)}{P(x)} = \frac{P(x,y)}{P(x)} \rightarrow P(x,y) &= P(y|x)P(x) \end{aligned} \tag{4.14}$$

Chapter 5

Experimental Evaluation

This chapter contains the experimental evaluation of EviTraN. In more details, it includes the experimental setting, the quantitative and qualitative results of the experimental evaluation, a discussion of the aforementioned results and an empirical analysis of the correlation between the relevance term of Information Bottleneck and auxiliary objective of EviTraN.

5.1 Experimental Setting

This section includes the experimental setting of the evaluation, such as involved datasets, evidence sources and metrics. The experimental setting not only provides insight necessary for the purposes of understanding the results of the evaluation, but also acts as a guideline for reproduction of the experiments. To this end, it includes detailed description of pre-processing techniques, the groups that each evidence sources yields, etc.

5.1.1 Datasets

As mentioned in Chapter 3, EviTraN is a representation learning method guided by unrestrained auxiliary task outcomes. To this end, its design and thus its evaluation should be widely applicable. In addition, representation learning is a domain invariant task. Therefore, the learning process of latent representations performed by EviTraN should be evaluated for a variety of datasets. As a result, the experimental evaluation of EviTraN includes two types of primary data sources: images and text, which are described as follows:

1. *MNIST* (non-coloured images)
2. *CIFAR-10* (coloured images)

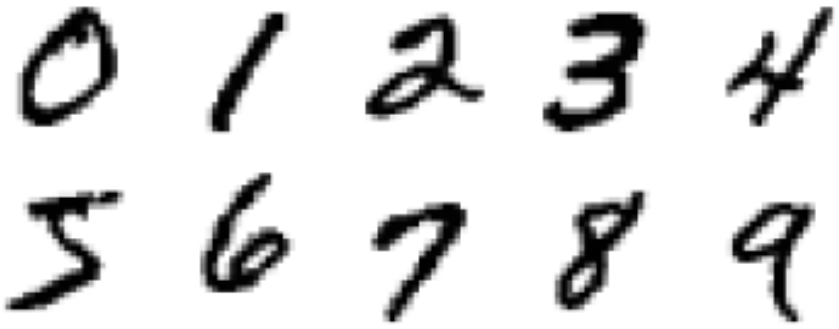


Figure 5.1: Sample from training set of MNIST [2], depicting images from all classes.

3. *20newsgroups* (newsgroup posts)

4. *RCV1* (newswire stories)

MNIST [2] is one of the most widely used datasets in the domain of machine learning. Lecun et al. [2] created a modified version of NIST's Special Database 1 and 3[1], by performing scaling, normalisation and re-positioning of the original images. MNIST contains black and white images that depict handwritten digits. Each image has a 28×28 resolution, with a 20×20 centred pixel box depicting a digit with an 8×8 padding. The centred pixel box has no visible boundaries. The dataset is connected with the task of identifying the class of the handwritten digit. Therefore, the associated labels represent the numerical value of the depicted image, i.e., from 0 to 9. It contains 70,000 handwritten digits, split in a 60,000 training and 10,000 test images. A sample from the training set of MNIST is shown in Figure 5.1.

CIFAR-10 [27] is a coloured image dataset. The 10 label variation (a 100 label variation called CIFAR-100, also exists) contains divisions of two superclasses: vehicles and animals. It contains 60,000 coloured images of 32×32 resolution split in 50,000 training and 10,000 test images. Human annotators have manually annotated the labels of the dataset. An instruction sheet given to annotators (for more details please refer to its technical report [27]), guided the process of labelling and including images. The 10 categories of CIFAR-10 are: airplane (0), automobile (1), bird (2), cat (3), deer (4), dog (5), frog (6), horse (7), ship (8), truck (9). The classes are mutually exclusive, meaning that there is no overlap between the classes, i.e., automobile class does not involve trucks. CIFAR-10 is connected with the task of identifying the object or the animal that is depicted in the coloured image. CIFAR-10 and CIFAR-100 are well-known within the community of computer vision and have been used for evaluation of multiple models. A sample from the training set of CIFAR-10 is shown in Figure 5.2.

[1] https://www.nist.gov/srd/shop/special-database-catalog

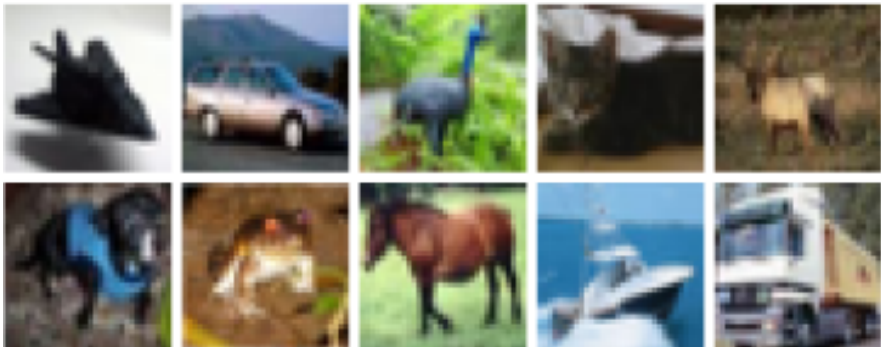


Figure 5.2: Sample from training set of CIFAR-10 [27], depicting images from all classes.

Listing 5.1: Training example from training set of 20newsgroups [141, 142] the text belongs in category *rec.autos*. The user information has been retrated for privacy.

```
Subject: Re: Is car saftey important?

 USER RETRACTED  writes:
>Is it only me, or is
>safety not one of the most important factors when buying a car?

It depends on your priorities.  A lot of people put higher priorities
on gas mileage and cost than on safety, buying "unsafe" econoboxes  …
```

20newsgroups [141, 142] is a collection of newsgroup documents. It contains close to 20,000 documents (approximately 1,000 documents for each topic). 20newsgroups is connected with the task of identifying the topic of each news document. It consists of 20 categories that can be seen as divisions of 6 supergroups. 20newsgroups is a popular dataset for text applications in machine learning. Table 5.1 depicts the categories/groups of 20newsgroups dataset. Listing 5.1 depicts an example from the category of *rec.autos*.

RCV1 [143] is a collection of newswire stories. It contains 804,414 documents. RCV1 is connected with the task of identifying the topic of each document. It consists of 103 categories which originate from 4 root categories: CCAT (Corporate/Industrial), ECAT (Economics), GCAT (Goverment/Social), MCAT (Markets). Evaluating EviTraN requires the creation and use of a subset that consists of 10 topics (subcategories), for consistency purposes. The final dataset called *Reuters-100k* consists of 96,933 documents. The labels of Reuters-100k are: C15 (Performance), C151 (Accounts/Earnings), GPOL (Domestic Politics), GSPO (Sports), GDIP (International Relations), E51 (Trade Reserves), M11 (Equity Markets), M14 (Commodity Markets), E21 (Goverment Finance), E41 (Employment/Labour). Table 5.2 depicts the categories/groups of Reuters-100k dataset. An example of RCV1 data can be found

Table 5.1: 20newsgroups root and subcategories.

Root Group	Subcategories	Root Group	Subcategories
Comp(uters).	comp.graphics (1) comp.os.ms-windows.misc (2) comp.sys.ibm.pc.hardware (3) comp.sys.mac.hardware (4) comp.windows.x (5)	Talk	talk.politics.misc (18) talk.politics.guns (16) talk.politics.mideast (17) talk.religion.misc (19)
Rec(reational).	rec.autos (7) rec.motorcycles (8) rec.sport.baseball (9) rec.sport.hockey (10)	Misc	misc.forsale (6) alt.atheism (0) soc.religion.christian (15)
Sci(ence).	sci.crypt (11) sci.electronics (12) sci.med (13) sci.space (14)		

Table 5.2: Reuters-100k root and subcategories.

Root Group	Subcategories	Root Group	Subcategories
CCAT	C15 (Performance) (0) C151 (Earnings) (1)	GCAT	GSPO (Sports) (3) GPOL (Dom. Politics) (2) GDIP (Int. Relations) (4)
ECAT	E51 (Trade Reserves) (5) E21 (Gov. Finance) (8) E41 (Employment) (9)	MCAT	M11 (Equity Markets) (6) M14 (Comm. Markets) (7)

in the original work [143].

5.1.2 Pre-processing Techniques

The experimental evaluation for all the above datasets except MNIST, involves embeddings instead of raw data. Using some form of embedding for text datasets is an intuitive procedure. Due to lack of numerical values, their original format is not suitable for machine learning methods. Word2Vec [28] or TD-IDF features [144, 145, 146, 147] are procedures capable of transforming text into vectors, which are more appropriate for machine learning. Despite CIFAR-10 being represented by numerical values, coloured images consist of a plethora of features. Even though the resolution of the images is low, each image contains thrice as many features as its greyscale or black and white counterpart. The experimental evaluation with CIFAR-10 involves

embeddings extracted from a pre-trained VGG-16 network [148]. Such embeddings reduce the amount of effort required for tinkering with the hyperparameters of the network, in order to efficiently learn from data of that dimensional merit. At the same time, such reduction enables more reasonable training times.

TF-IDF is a composite metric. It consists of two individual metrics: Term Frequency (TF) and Inverse Document Frequency (IDF). Consider the task of classifying/categorising documents into semantically relatable topics, in an automated manner. One would suggest that the observation of the most frequent words or terms that appear within the document may indicate the topic of the document. For example, a document from the domain of medicine or health, would include domain specific terms such as "Pulmonology" or "Respirology", which are characteristics of the topic. Term frequency lends itself to the above idea. Term frequency (as shown in Equation 5.1) measures how frequent a term t is found in a document d. The function f is the count of occurrences of t within d, i.e., its *frequency*. For normalisation purposes, one may divide the term frequency with the maximum term frequency found between any term (t') in the document.

$$TF_{t,d} = \frac{f_{t,d}}{\max_{t'} f_{t',d}} \tag{5.1}$$

The above notation is similar to notation found within the work of Rajaraman and Ullman [147], for consistency purposes. To aid reading comprehension, the terms i and j in the original work, are swapped with t and d.

As expected however, most frequent terms found within a document are rather generic. Common words such as "The" or "And" are very frequent and do not contribute any insight regarding the context of the document. One simple solution to this problem, would be to remove such common words, which is a data pre-processing procedure known as "*Eliminating/Removing Stop Words*" [147]. However, certain terms may not be quite as common as stop words, yet they do not provide any insight regarding the contents of the document, such as "vice versa". Thus, term frequency requires further conditioning.

Inverse Document Frequency (IDF) is the necessary conditioning to term frequency. IDF measures how frequent a term is across all documents. As shown in Equation 5.2, IDF is a logarithmic ratio of the number of documents (N) to the number of documents where a term t appears (n_t).

$$IDF_t = \log \frac{N}{n_t} \tag{5.2}$$

The above notation is similar to notation found within the work of Rajaraman and Ullman [147], for consistency purposes. To aid reading comprehension, the terms

i and n_i in the original work, are swapped with t and n_t.

Therefore, TF-IDF (Equation 5.3) quantifies how frequent is a term within a document, while at the same time being weighted by how frequent that term is across the collection of documents. Thus, frequent terms have high term frequency but low inverse document frequency. At the same time, rare terms have high inverse document frequency but low term frequency. This composition of metrics, allows a more balanced representation regarding the ranking of words based on their frequency and insight.

$$TF\text{-}IDF_{t,d} = TF_{t,d} \cdot IDF_t \tag{5.3}$$

Unlike its previous counterparts, e.g., N-gram model [149], Word2Vec aims to learn continuous (dense) representations of words. Despite that, it also aims to produce word representations influenced by their context. Word2Vec is a deep neural network that is trained to produce word embeddings. Mikolov et al. [28] proposed two similar but distinct variations of the training strategy. The first training strategy called Continuous Bag-of-Words Model, also known as *CBOW*, is based on predicting a target word through observation of its neighbour words. On the other hand, during Continuous Skip-gram Model the model aims to correctly predict the neighbour words through observation of the current word. For both cases a hyperparameter, C, is used to represent the maximum distance between words, where for each word a random amount of words between 1 and C are selected as neighbours. Both variations receive as input a sparse one-hot vector encoding of words, called "*1-of-V*", with V being the size of the vocabulary. Figure 5.3 depicts the training variations of Word2Vec.

During the experimental evaluation with CIFAR-10, transformation of raw data leads to images with shape: (N, 32, 32, 3) into single dimensional feature vectors of 4,096 features. A pre-trained VGG-16 neural network on ImageNet database [150], lends itself to the transformation process. The VGG-16 network receives as input rescaled CIFAR-10 images (to fit the input layer expectations) and produces embeddings extracted from the inner-most dense layer. The above pre-processing is similar to that found in Xie et al. [24]. During evaluation with 20newsgroups, the pre-processing procedure of Spathis et al. [151] and Spathis et al. [152] is followed[2]. The pre-processing procedure involves tokenisation, removal of stop words, removal of empty documents and documents with words that do not exist in the vocabulary of the word2vec model. Then, a pre-trained word2vec model, trained on Google News

[2]More regarding the implementation can be found here: https://github.com/sdimi/average-word2vec

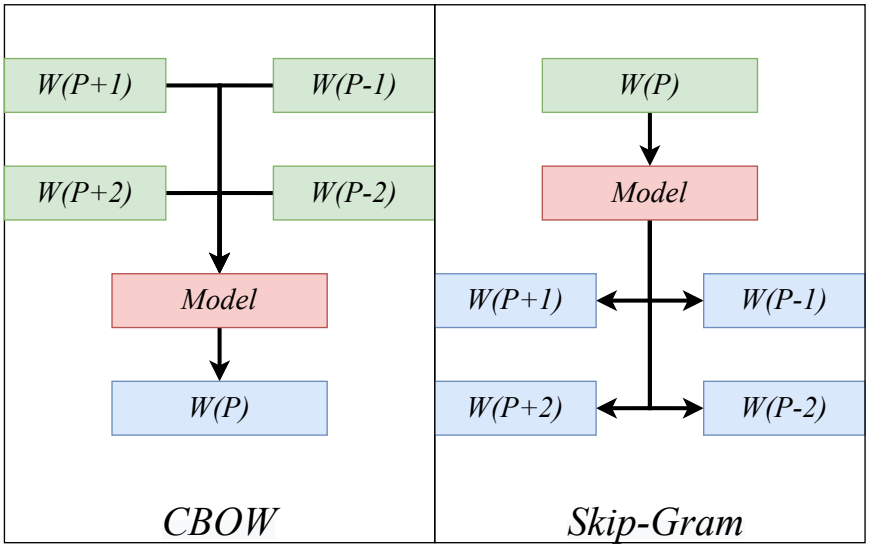


Figure 5.3: Training strategy variations of Word2Vec model [28]. P represents the current position, while $W(P)$ represents the word/token at current position.

Table 5.3: Final dimensions of the datasets involved in the experimental evaluation of EviTraN.

Dataset	Original Dims	PPS Technique	Final Dims
MNIST	(N, 28, 28, 1)	-	(N, 784)
CIFAR-10	(N, 32, 32, 3)	VGG-16 embeddings [24]	(N, 4096)
20newsgroups	-	Word2Vec embeddings [151]	(N, 300)
Reuters-100k	-	TF-IDF features [24]	(N, 2000)

N: Total amount of samples found within each dataset.
PPS: pre-processing

corpus[3] (about 100 billion words), lends itself to the extraction of 20newsgroups embeddings. The final outcome is a single feature vector of 300 features by averaging the word embeddings (extracted from Word2Vec) of each document. During evaluation with Reuters-100k, the pre-processing procedure of Xie et al. [24] is followed. The pre-processing procedure involves transformation of raw text into TF-IDF features of the 2000 most frequent word stems[4]. Table 5.3, shows the final dimensions and pre-processing techniques for each dataset.

[3] https://code.google.com/archive/p/word2vec/

[4] More regarding the implementation can be found here: https://github.com/XifengGuo/DEC-keras. **DISCLAIMER: This is not the official code repository**.

5.1.3 Evidence Sources

In order to evaluate the ability of EviTraN to utilise external categorical evidence for the purposes of guiding the learning process. The evaluation process requires the creation and introduction of a plethora of external evidence sources for each dataset. The experimental evaluation includes three types of external evidence: meaningful, inaccurate and incomplete evidence, which are tested in three different quantities: single, double and triple. Triple evidence sources are involved mostly in CIFAR-10 dataset, as its labelset groups allow for the creation of more meaningful evidence variations, in comparison to the other three datasets.

During experiments with MNIST, meaningful evidence represents relations that involve the numerical values of the image. The first meaningful evidence, consists of 3 auxiliary classes and represents the relation of $y \ mod \ 3$, with y being the numerical value of the depicted digit, i.e., the label. Similarly, meaningful evidence that consists of 4 auxiliary classes, represents the relation of $hash(y) \ mod \ 4$, where $hash$ is the hashing function of *Python*. Furthermore, the labelset of MNIST is also introduced as meaningful evidence. The labelset consists of 10 auxiliary classes. Table 5.4 (a) contains the auxiliary classes yielded from each aforementioned evidence source.

In CIFAR-10 experiments, meaningful evidence represents alternative divisions of the labelset into various supergroups. The first meaningful evidence, consists of 3 auxiliary classes and represents supergroups: *vehicles*, *pets* and *wild animals*. Similarly, meaningful evidence with 4 and 5 auxiliary classes consists of groups: *vehicles*, *indoor pets*, *outdoor pets*, *wild animals* — *road vehicles*, *other vehicles*, *indoor pets*, *outdoor pets*, *wild animals*. Additionally, the labelset of CIFAR-10 is also introduced as meaningful evidence. The labelset consists of 10 auxiliary classes. Table 5.4 (b) contains the auxiliary classes yielded from each aforementioned evidence source.

Throughout experiments with 20newsgroups, meaningful evidence represents alternative divisions of its labelset into supergroups. The first meaningful evidence, consists of 5 auxiliary classes and represents supergroups: *computers*, *recreational*, *science*, *talk* and *misc*. Similarly, meaningful evidence with 6 auxiliary classes consists of: *sports*, *politics*, *religion*, *vehicles*, *systems* and *science*. Additionally, the labelset of 20newsgroups is also introduced as meaningful evidence. The labelset of 20newsgroups consists of 20 auxiliary classes. Table 5.5 (a) contains the auxiliary classes yielded from each aforementioned evidence source.

In Reuters-100k experiments, meaningful evidence represents alternative divisions of its labelset into supergroups. The first meaningful evidence, consists of 4 auxiliary classes, which are the four root categories of RCV1. Similarly, meaningful evidence of 5 auxiliary classes is a re-categorisation of 10 subcategories into 5 groups. The 5 groups do not represent any particular group relation. Additionally, the labelset of

Table 5.4: Detailed description of groups yielded by meaningful evidence sources for image datasets: MNIST and CIFAR-10

(a) MNIST

External Evidence	Auxiliary Classes	
M3	0:	[0, 3, 6, 9]
	1:	[1, 4, 7]
	2:	[2, 5, 8]
M4	0:	[0, 4, 8]
	1:	[1, 5, 9]
	2:	[2, 6]
	3:	[3, 7]
M10	V:	[0, 1, 2, 3, 4 5, 6, 7, 8, 9]

(b) CIFAR-10

External Evidence	Auxiliary Classes	
M3	Vehicles:	[0, 1, 8, 9]
	Pets:	[3, 5, 7]
	Wild Animals:	[2, 4, 6]
M4	Vehicles:	[0, 1, 8, 9]
	Indoor Pets:	[3, 5]
	Outdoor Pets:	[4, 7]
	Wild Animals:	[2, 6]
M5	Road Vehicles:	[1, 9]
	Other Vehicles:	[0, 8]
	Indoor Pets:	[3, 5]
	Outdoor Pets:	[4, 7]
	Wild Animals:	[2, 6]
M10	V:	[0, 1, 2, 3, 4 5, 6, 7, 8, 9]

Reuters-100k is also introduced as meaningful evidence. The labelset of Reuters-100k consists of 10 auxiliary classes. Table 5.5 (b) contains the auxiliary classes yielded from each aforementioned evidence source.

The exploitation of meaningful evidence sources by EviTraN is crucial for the evaluation of the effectiveness criterion. As a task outcome, meaningful evidence sources represent semantically high information. They represent high-level concepts that can not be inferred only through observation of the data features. For instance, identifying the class of each handwritten digit in MNIST (i.e., predicting y) is feasible through observation of the image features, since each label can be inferred through the depicted shape. However, production of $y \; mod \; 3$ requires the understanding that each shape in MNIST represents a value. And thus, repurpose that value towards the computation of the modulo operation. Therefore, steering of the unsupervised learning process with such information should lead to increased performance, since it involves additional unobservable insight from external sources.

In addition, meaningful evidence sources simulate the process of utilising task outcomes extracted from unobserved datasets. There may exist an alternative version of MNIST connected with the task of identifying the groups yielded from relation

Table 5.5: Detailed description of groups yielded by meaningful evidence sources for text datasets: 20newsgroups and Reuters-100k

(a) 20newsgroups

External Evidence	Auxiliary Classes
M5	V: Root groups in Table 5.1
M6	Sports: [9, 10] Politics: [16, 17, 18] Vehicles: [6, 7, 8] Systems: [2, 3, 4, 5] Science: [1, 11, 12, 13, 14] Religion: [0, 15, 19]
M20	V: Subcategories in Table 5.1

(b) Reuters-100k

External Evidence	Auxiliary Classes
M4	V: Root groups in Table 5.2
M5	0 [0, 5] 1 [1, 6] 2 [2, 7] 3 [3, 8] 4 [4, 9]
M10	V: Subcategories in Table 5.2

Table 5.6: Example of meaningful, inaccurate and incomplete evidence sources. Meaningful evidence is $y \ mod \ 3$ or $M3$ of MNIST.

External Evidence	Auxiliary Classes	External Evidence	Auxiliary Classes
Meaningful (M3)	Y: 0 [1, 0, 0] (a) Y: 1 [0, 1, 0] (b) Y: 2 [0, 0, 1] (c) Y: 3 [1, 0, 0] (d) Y: 0 [1, 0, 0] (e) Y: 0 [1, 0, 0] (f) Y: 2 [0, 0, 1] (g) ...	Inaccurate	$RV3$: [0.3, 0.3, 0.3] $RV10$: [0.1, 0.1, 0.1, 0.1, 0.1 0.1, 0.1, 0.1, 0.1, 0.1] $RI3$: Y: 0 →[0, 0, 1] (c) Y: 1 →[1, 0, 0] (a) Y: 2 →[0, 1, 0] (d) ...
Incomplete (Uniform)	Y: 0 [1, 0, 0] (a) Y: 1 [0, 1, 0] (b) Y: 2 [0, 0, 1] (c) Y: 3 [1, 0, 0] (d) ...	Incomplete (Bias)	Y: 1 [0, 1, 0] (a) Y: 2 [0, 0, 1] (b) ...

$y \ mod \ 3$. Involving only the outcome of the above task, simulates a realistic scenario where the alternative MNIST version is unobserved or unavailable. Furthermore, it reduces introduction of redundant features.

The experimental evaluation also includes a set of non-meaningful evidence sources for the above datasets, as shown in Table 5.6. Non-meaningful evidence sources are

part of the evaluation in the inaccurate learning setting. Two additional sources with the same width, i.e., the same auxiliary classes, are created for each meaningful evidence source. One consists of random values drawn from a uniform distribution (white noise). *Random values* evidence is a simulation of inherent/artificial noise or highly uncertain task outcomes (which are characterised by high entropy, i.e., uniform samples). The other consists of the meaningful evidence introduced in a randomised order. *Random index* evidence is a simulation of uncorrelated tasks or tampered evidence. Random index has similar distribution of features to meaningful evidence sources, however, there is no correspondence with the primary data.

Furthermore, two versions of each meaningful evidence source that aim to simulate incomplete learning setting scenarios are also created. *Uniformly missing* samples evidence represents scenarios where an amount of samples is missing from all auxiliary classes. This can be perceived as having a lower amount of total samples. This can be a case of the external evidence being incomplete due to malicious activity or using evidence that is still in the process of gathering. On the other hand, *biased* evidence sources represent scenarios where some auxiliary classes are lacking. In other words, are not represented within the evidence samples. Similarly to uniformly missing, this can be a case of malicious activity or incomplete collection process.

5.1.4 Metrics

Unlike the evaluation process of other tasks, e.g., down-stream tasks, the evaluation process of learning representations is not as straightforward. This is a consequence of representation learning being disconnected from an end-goal. Most often the training objective does not represent the end-goal. At the same time, the learning of representations may aim to multiple end-goals such as dimensionality reduction, clustering, compression, generation, etc. Thus, establishing universal criteria for evaluation of representation learning is not feasible.

During the experimental evaluation of EviTraN, clustering is the end-goal of the representation learning procedure. Clustering is essentially an unsupervised down-stream task, i.e., it involves unlabelled data samples. Clustering is an appropriate task that will highlight if EviTraN is able to is able to enclose high-level information, introduced from external evidence, into latent features. Since, clustering generates group memberships only through observation of data features, increased performance in the clustering task should indicate the successful translation of information from auxiliary task outcomes into latent features.

The evaluation procedure consists of two stages (similar to training strategy). The first stage consists of measuring the clustering performance during the initialisation step. Measuring the clustering performance with initial set of representations

Algorithm 2: Hungarian Algorithm, according to HungarianAlgorithm.com [154]

Data: M: square matrix with n by n dimensions.
Result: m: mapping of minimum cost
forall *rows of* M **do**
  row = row − min(row);
end
forall *columns of* M **do**
  column = column − min(column);
end
if n *vertical and horizontal lines cover all zeros* **then**
  m = matrix of (x, y) coordinates of zeros;
else
  k = minimum element of M not covered by a line;
  Subtract k from uncovered elements;
  Add k to elements covered by two lines;
  m = matrix of (x, y) coordinates of zeros;
end

is the *baseline* solution. The second step consists of measuring the clustering performance after the introduction of external evidence, i.e., after evidence transfer step. During both stages, a clustering algorithm receives latent representations as input, which are produced by the autoencoder (independent of the involvement of external evidence). For consistency purposes across solutions, a k-means algorithm with the same hyperparameters is deployed for both stages[5].

The experimental evaluation involves two metrics: unsupervised clustering accuracy (ACC) [24] and normalised mutual information (NMI). Xie et al. [24] proposed the unsupervised clustering accuracy, which measures the accuracy of the best mapping between a cluster membership and ground truth labels. The metric is defined as follows, in Equation 5.4.

$$ACC = \max_{m} \frac{\sum_{i=1}^{n} \mathbf{1}\{l_i = m(c_i)\}}{n} \tag{5.4}$$

In Equation 5.4, l_i is the ground truth label of the i-th data sample. Similarly, c_i is the cluster assignment of the i-th data sample, while m is a mapping between cluster assignment and ground truth label (which ranges over all mappings). According to the authors, the best mapping can be efficiently computed by the Hungarian algorithm [153] depicted in Algorithm 2.

[5] $K = |Y|$ with 20 initialisations, the rest of hyperparameters are the default ones from scikit-learn implementation.

Algorithm 3: Computation of Unsupervised Clustering Accuracy (ACC) [24].

Result: ACC: unsupervised clustering accuracy

```
L = ground truth labels;
C = clusters;
D = max(max(L), max(C));
W = Matrix(D, D);
Initialise W with zeros;
i = 0;
while i ≤ |C| do
    W[C_i, L_i] += 1;
    i += 1;
end
m = HungarianAlgorithm(−W + max(W)) ▷ Find assignment of maximum overlap between Clusters and Labels.;
ACC = (Σ_{i=1}^{n} 1{l_i=m(c_i)}) / n;
```

In practice, the unsupervised clustering accuracy metric is very similar to plain accuracy metric that is frequent in classification problems. However, unlike plain accuracy, the label assignment through mapping $m(c_i)$ of cluster c_i, is crucial to the performance of the metric. Assigning labels to clusters is an open research question with multiple proposed solutions. The implementation used in the experimental evaluation, follows the example of Xie et al. [24]. The mapping m is produced by the Hungarian algorithm.

Algorithm 3[6] depicts the process of computing unsupervised clustering accuracy. From vectors L and C that contain ground truth labels and clustering memberships respectively, a square matrix W with dimensions equal to the maximum element between maximum value of L and C is created. Matrix W at first contains only zeros. To populate matrix W, one should loop through the cluster membership of each data sample and increase the counter of position (C_i, L_i) by one. After population of matrix W, each row depicts the distribution of ground truth labels within each cluster. For instance, position (0, 0) in W represents how many data within cluster 0 have ground truth label 0. The Hungarian algorithm computes a mapping that yields the minimum cost. However, in this case the mapping that yields the maximum "cost" is required. The above procedure yields a majority vote mapping, i.e., assign ground truth label i to cluster that contains most i data. In order for the Hungarian Algorithm to produce the maximum cost, all elements of W are turned into negative values and the maximum element of W is added, in order to deal with non-negative

[6]The implementation can be found in the original code repository: https://github.com/piiswrong/dec

values.

The unsupervised clustering accuracy metric is the sum of all correctly assigned data. In other words, is the sum of all instances where elements from cluster i have been correctly mapped into ground truth label i. To normalise the result of the metric into a [0, 1] range, the sum is divided with the number of classes.

On the other hand, NMI is an information theoretic metric that does not require a mapping between cluster membership and ground truth labels. Multiple normalised variations of mutual information exists [155, 156, 157]. Equation 5.5 shows the variation [155, 156] used in the experimental evaluation of EviTraN[7].

$$NMI(L, P) = 2 \frac{I(L; P)}{H(L) + H(P)} \tag{5.5}$$

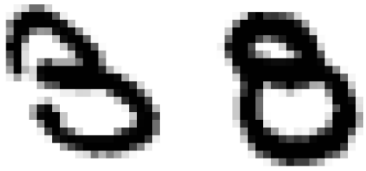
The above notation is similar to that found in the work of Maes et al. [156], for consistency purposes. To aid reading comprehension, notation A and B is switched to L and P that represent the class labels and predicted labels respectively.

NMI measures the mutual information between the clustering membership yielded from a clustering algorithm and ground truth labels. In other words, it quantifies "*how much one random variables tells us about another*" [159]. For practical purposes, since mutual information is a non-negative number, Astola and Virtanen [155] and Maes et al. [156] proposed a normalising term.

From the introduction of the above metrics, one might find the use of ground truth labels in experimental evaluation of an unsupervised learning process unorthodox. As mentioned before, evaluating the unsupervised clustering is not as straightforward as supervised tasks. To this end, two different approaches exist. Their major difference lies on whether they make use of ground truth labels or not. Metrics that utilise ground truth labels, such as ACC, NMI or Random Index Score [160], convey easier to digest insight regarding the outcome of clustering. Since the outcome of these metrics, is similar to that of evaluation metrics used in supervised tasks. However, the expectation of knowing the ground truth labels beforehand, defeats the purpose of unsupervised learning.

On the other hand, metrics that do not involve ground truth labels, such as Calinski-Harabasz score [161] or silhouette score [162], evaluate the quality of the separation of dataset into clusters. In other words, they study the similarity of data samples within a cluster, as well as, their dissimilarity to data samples from other clusters. The evaluation criteria of such metrics are vital. However, data samples with similar data features may have completely distinct high-level interpretations. For example, a 3 digit and an 8 digit from MNIST dataset might bear similar structure (as shown in Figure 5.4), but their labels are distinct. Therefore, a cluster that contains

[7]More regarding the implementation can be found at: `scikit-learn.org` [158]



(a) Digit: 3 **(b)** Digit: 8

Figure 5.4: Examples of digits: 3 and 8 from the testing set of MNIST [2].

both 3 and 8 digits satisfies the cluster quality criteria. However, 3 and 8 have different semantic meanings.

Since EviTraN aims to translate high-level information in the form of external evidence into latent features, evaluation metrics that involve ground truth labels are more fit for its evaluation. If the external evidence is meaningful and thus, provides valuable high-level semantic insight into the learning process, it should be reflected with appropriate evaluation metrics that also involve high-level information.

5.2 Evaluation Results

This section includes the results of the evaluation process. It includes both quantitative evaluation with the aforementioned metrics and qualitative evaluation through observation and study of the state of latent space before and after deployment of EviTraN. The quantitative evaluation includes the results for both ACC and NMI metrics, as well as, their respective boxplots for each dataset. Qualitative evaluation includes 2D plots of the latent space state for auxiliary classes, as well as, study of individual auxiliary classes.

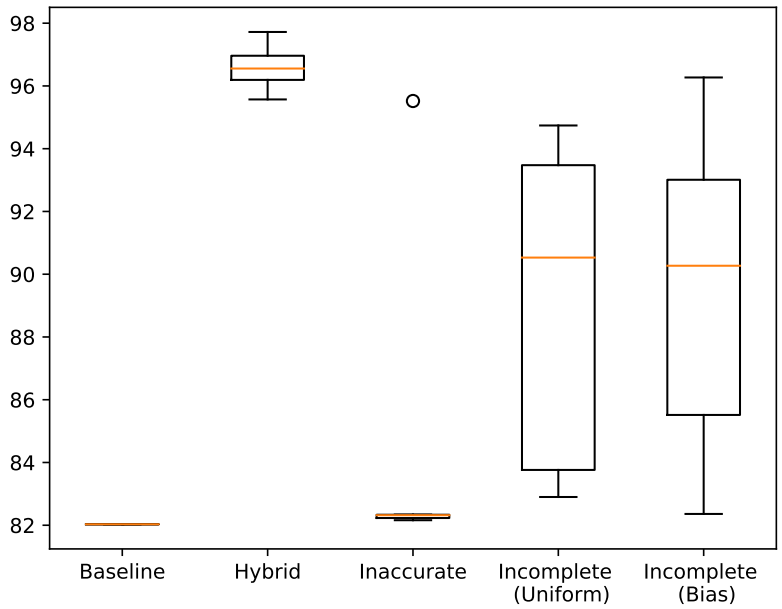
The quantitative evaluation involves the aforementioned datasets, evidence sources and metrics. The experimental setting aims to investigate the performance of EviTraN, in regard to both the effectiveness and robustness criteria, in three learning settings. Namely: hybrid, inaccurate and incomplete learning settings. The baseline solution, that does not utilise external evidence sources, is the evaluation of an unsupervised solution. The matching between scenarios and learning settings depends on the quality or type of the evidence source.

The evaluation process involves full datasets during training (except evidence transfer step of incomplete, as presented in Section 3.4.2). The term full, refers to datasets that consists both of training and testing data. Involving full datasets is a realistic evaluation scenario that also enables better generalisation of the involved models. In a realistic unsupervised learning scenario, one would use all the available

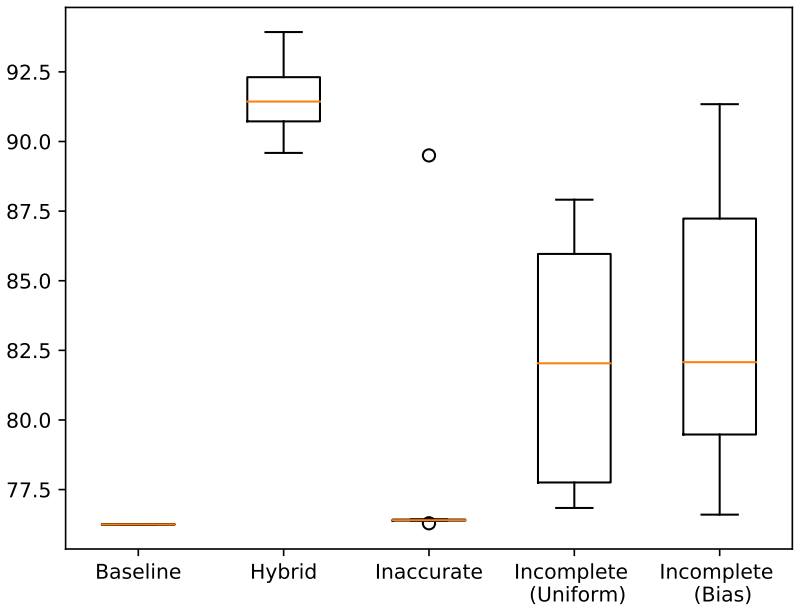
Table 5.7: Quantitative results of the experimental evaluation of EviTraN in MNIST.

Learning Setting	Evidence	ACC (%)	NMI (%)
Unsupervised (Baseline)	-	82.03	76.25
Hybrid	M3	95.57 (+13.54)	89.59 (+13.34)
	M4	96.40 (+14.37)	91.10 (+14.85)
	M10	96.71 (+14.68)	91.77 (+15.52)
	M3 & M4	97.72 (+15.69)	93.93 (+17.68)
Inaccurate	RV3	82.32 (+0.29)	76.40 (+0.15)
	RV10	82.32 (+0.29)	76.40 (+0.15)
	RV3 & RV4	82.20 (+0.17)	76.38 (+0.13)
	RI3	82.16 (+0.13)	76.29 (+0.04)
	RI10	82.34 (+0.31)	76.43 (+0.18)
	M3 & RV3	95.52 (+13.49)	89.50 (+13.25)
Incomplete (Uniform)	M3 ($M \downarrow_{70\%}$)	91.23 (+9.20)	82.93 (+6.68)
	M3 ($M \downarrow_{90\%}$)	82.90 (+0.87)	76.84 (+0.59)
	M4 ($M \downarrow_{70\%}$)	94.74 (+12.71)	87.91 (+11.66)
	M4 ($M \downarrow_{90\%}$)	89.83 (+7.80)	81.14 (+4.89)
	M10 ($M \downarrow_{70\%}$)	94.57 (+12.54)	87.68 (+11.43)
	M10 ($M \downarrow_{90\%}$)	84.02 (+1.99)	78.00 (+1.75)
	M3 & M4 ($M \downarrow_{70\%}$)	93.11 (+11.08)	85.39 (+9.14)
	M3 & M4 ($M \downarrow_{90\%}$)	82.99 (+0.96)	77.02 (+0.77)
Incomplete (Bias)	M3 (AC $\downarrow$ 1)	90.32 (+8.29)	82.19 (+5.94)
	M3 (AC $\downarrow$ 2)	82.38 (+0.35)	76.60 (+0.35)
	M4 (AC $\downarrow$ 1)	92.09 (+10.06)	86.21 (+9.96)
	M4 (AC $\downarrow$ 2)	86.56 (+4.53)	80.42 (+4.17)
	M10 (AC $\downarrow$ 1)	96.27 (+14.24)	91.34 (+15.09)
	M10 (AC $\downarrow$ 2)	95.77 (+13.74)	90.30 (+14.05)
	M3 & M4 (AC $\downarrow$ 1)	90.22 (+8.19)	81.96 (+5.71)
	M3 & M4 (AC $\downarrow$ 2)	82.36 (+0.33)	76.65 (+0.40)

$M\#$: Meaningful evidence, # the represents number of auxiliary classes — width of evidence samples.
RV: Random Values.
RI: Random Index.
($M \downarrow_{P\%}$): For consistency with Chapter 3, M represents the total amount of evidence samples with an evidence source. P represents the reduction percentage of the total amount of evidence samples.
AC $\downarrow$ #: Where # represents the reduction number of the total amount of auxiliary classes.



(a) ACC



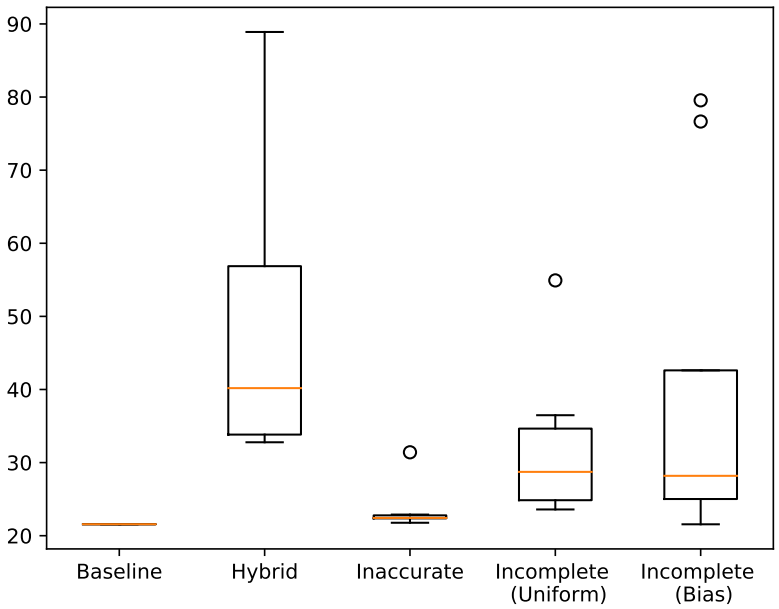
(b) NMI

Figure 5.5: Boxplots of ACC and NMI metrics for experimental evaluation with MNIST dataset in all learning settings.

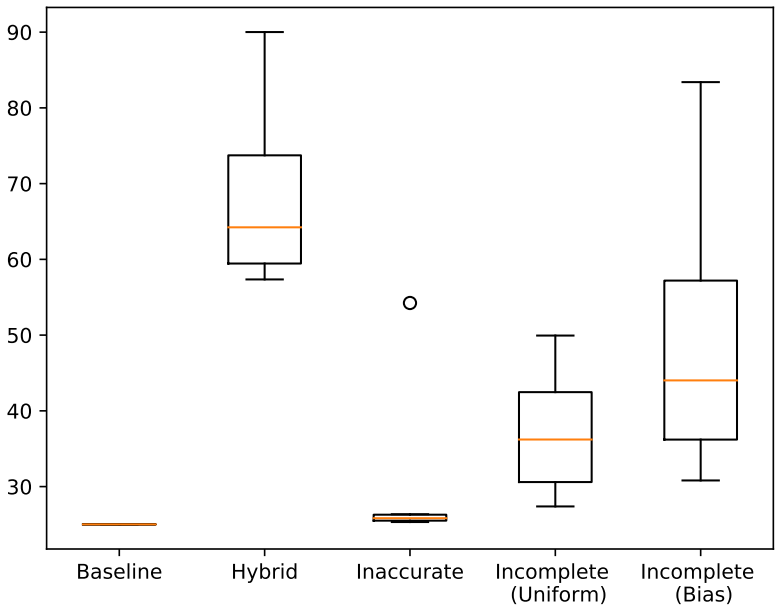
Table 5.8: Quantitative results of the experimental evaluation of EviTraN in 20newsgroups.

Learning Setting	Evidence	ACC (%)	NMI (%)
Unsupervised (Baseline)	-	21.19	25.01
Hybrid	M5	34.18 (+12.99)	57.35 (+32.34)
	M6	32.78 (+11.59)	60.15 (+35.14)
	M20	88.90 (+67.71)	90.01 (+65.00)
	M5 & M6	46.19 (+25.00)	68.31 (+43.30)
Inaccurate	RV3	22.36 (+1.17)	25.49 (+0.48)
	RV10	22.46 (+1.27)	26.11 (+1.10)
	RV3 & RV10	22.89 (+1.70)	26.35 (+1.34)
	RI5	21.77 (+0.58)	25.32 (+0.31)
	RI20	22.40 (+1.21)	25.54 (+0.53)
	M5 & RV3	31.41 (+10.22)	54.24 (+29.23)
Incomplete (Uniform)	M5 ($M \downarrow_{70\%}$)	30.42 (+9.23)	39.21 (+14.20)
	M5 ($M \downarrow_{90\%}$)	23.59 (+2.40)	29.27 (+4.26)
	M6 ($M \downarrow_{70\%}$)	34.04 (+12.85)	41.71 (+16.70)
	M6 ($M \downarrow_{90\%}$)	25.03 (+3.84)	31.04 (+6.03)
	M20 ($M \downarrow_{70\%}$)	54.92 (+33.73)	49.94 (+24.93)
	M20 ($M \downarrow_{90\%}$)	24.33 (+3.14)	27.38 (+2.37)
	M5 & M6 ($M \downarrow_{70\%}$)	36.48 (+15.29)	44.75 (+19.74)
	M5 & M6 ($M \downarrow_{90\%}$)	27.04 (+5.85)	33.22 (+8.21)
Incomplete (Bias)	M5 (AC $\downarrow$ 1)	31.27 (+10.08)	49.01 (+24.00)
	M5 (AC $\downarrow$ 2)	25.95 (+4.76)	35.59 (+10.58)
	M6 (AC $\downarrow$ 1)	30.43 (+9.24)	49.53 (+24.52)
	M6 (AC $\downarrow$ 2)	25.21 (+4.02)	36.41 (+11.40)
	M20 (AC $\downarrow$ 1)	79.55 (+58.36)	83.40 (+58.39)
	M20 (AC $\downarrow$ 2)	76.65 (+55.46)	80.20 (+55.19)
	M5 & M6 (AC $\downarrow$ 1)	21.56 (+0.37)	39.03 (+14.02)
	M5 & M6 (AC $\downarrow$ 2)	24.44 (+3.25)	30.81 (+5.80)

$M\#$: Meaningful evidence, # the represents number of auxiliary classes — width of evidence samples.
RV: Random Values.
RI: Random Index.
($M \downarrow_{P\%}$): For consistency with Chapter 3, M represents the total amount of evidence samples with an evidence source. P represents the reduction percentage of the total amount of evidence samples.
AC $\downarrow$ #: Where # represents the reduction number of the total amount of auxiliary classes



(a) ACC



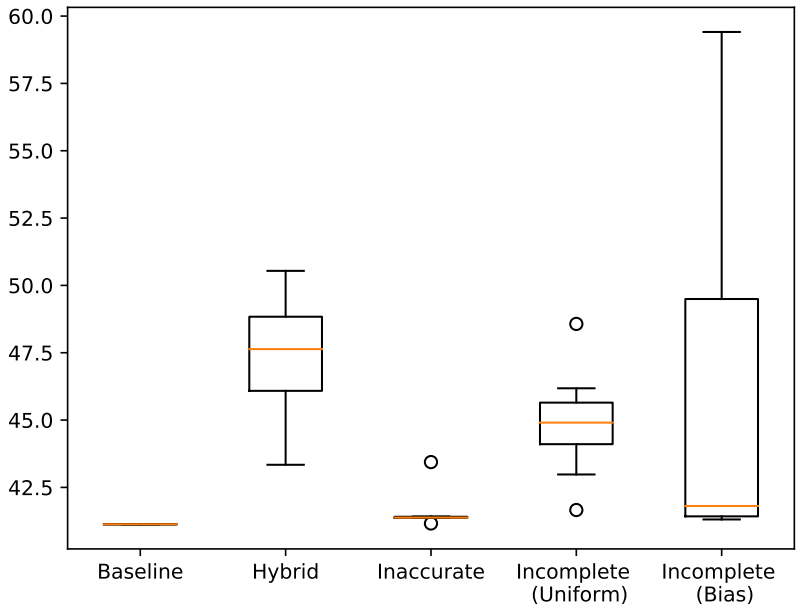
(b) NMI

Figure 5.6: Boxplots of ACC and NMI metrics for experimental evaluation with 20newsgroups dataset in all learning settings.

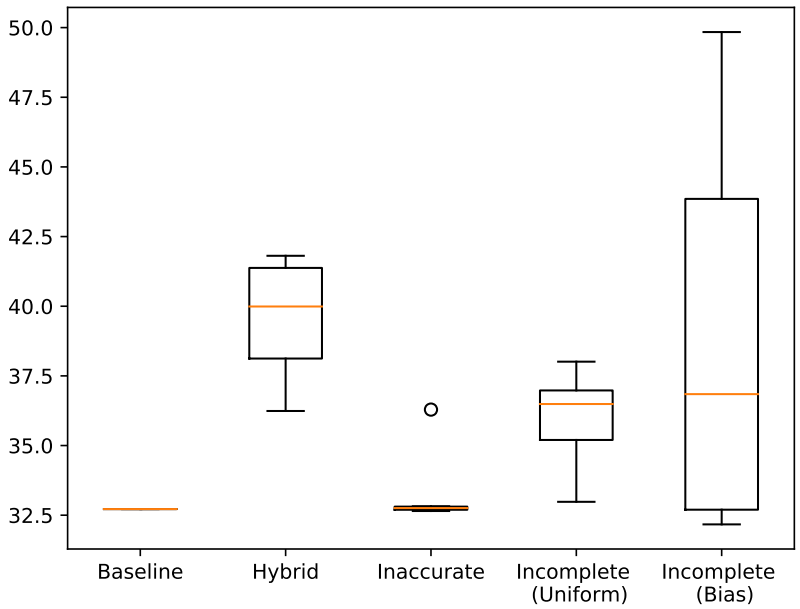
Table 5.9: Quantitative results of the experimental evaluation of EviTraN in Reuters-100k.

Learning Setting	Evidence	ACC (%)	NMI (%)
Unsupervised (Baseline)	-	41.12	32.72
Hybrid	M4	43.34 (+2.22)	36.24 (+3.52)
	M5	47.00 (+5.88)	38.75 (+6.03)
	M10	48.27 (+7.15)	41.23 (+8.51)
	M4 & M5	50.54 (+9.42)	41.81 (+9.09)
Inaccurate	RV3	41.42 (+0.30)	32.77 (+0.05)
	RV10	41.38 (+0.26)	32.74 (+0.02)
	RV3 & RV10	41.16 (+0.04)	32.65 (-0.07)
	RI4	41.37 (+0.25)	32.82 (+0.10)
	RI10	41.38 (+0.26)	32.68 (-0.04)
	M4 & RV3	43.44 (+2.32)	36.29 (+3.57)
Incomplete (Uniform)	M4 ($M \downarrow_{70\%}$)	44.48 (+3.36)	36.12 (+3.40)
	M4 ($M \downarrow_{90\%}$)	41.66 (+0.54)	32.98 (+0.26)
	M5 ($M \downarrow_{70\%}$)	42.98 (+1.86)	33.28 (+0.56)
	M5 ($M \downarrow_{90\%}$)	44.51 (+3.39)	35.84 (+3.12)
	M10 ($M \downarrow_{70\%}$)	46.18 (+5.06)	36.86 (+4.14)
	M10 ($M \downarrow_{90\%}$)	45.47 (+4.35)	37.18 (+4.46)
	M4 & M5 ($M \downarrow_{70\%}$)	48.57 (+7.45)	38.01 (+5.29)
	M4 & M5 ($M \downarrow_{90\%}$)	45.30 (+4.18)	36.91 (+4.19)
Incomplete (Bias)	M4 (AC $\downarrow$ 1)	41.63 (+0.51)	32.75 (+0.03)
	M4 (AC $\downarrow$ 2)	41.31 (+0.19)	32.17 (-0.55)
	M5 (AC $\downarrow$ 1)	46.53 (+5.41)	41.86 (+9.14)
	M5 (AC $\downarrow$ 2)	41.99 (+0.87)	40.87 (+8.15)
	M10 (AC $\downarrow$ 1)	59.41 (+18.29)	49.83 (+17.11)
	M10 (AC $\downarrow$ 2)	58.39 (+17.27)	49.84 (+17.12)
	M4 & M5 (AC $\downarrow$ 1)	41.32 (+0.20)	32.54 (-0.18)
	M4 & M5 (AC $\downarrow$ 2)	41.46 (+0.34)	32.82 (+0.10)

$M\#$: Meaningful evidence, # the represents number of auxiliary classes — width of evidence samples.
RV: Random Values.
RI: Random Index.
($M \downarrow_{P\%}$): For consistency with Chapter 3, M represents the total amount of evidence samples with an evidence source. P represents the reduction percentage of the total amount of evidence samples.
AC $\downarrow$ #: Where # represents the reduction number of the total amount of auxiliary classes



(a) ACC



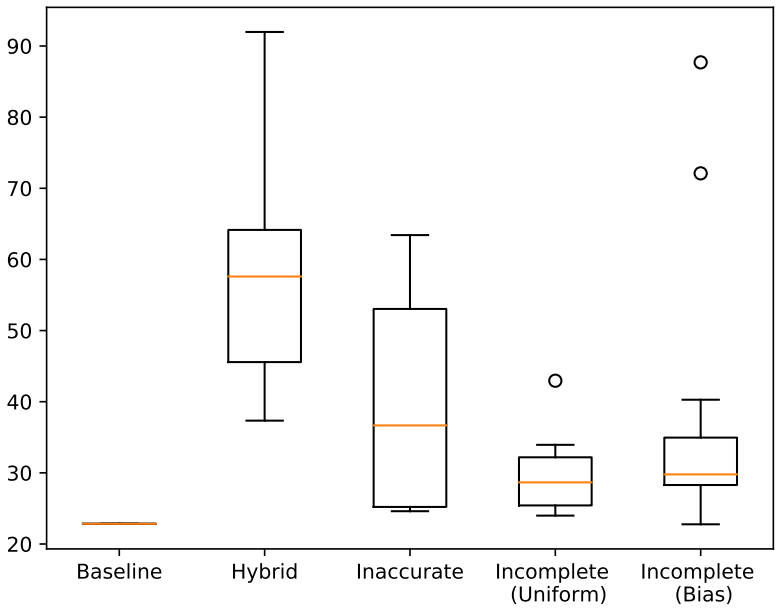
(b) NMI

Figure 5.7: Boxplots of ACC and NMI metrics for experimental evaluation with Reuters-100k dataset in all learning settings.

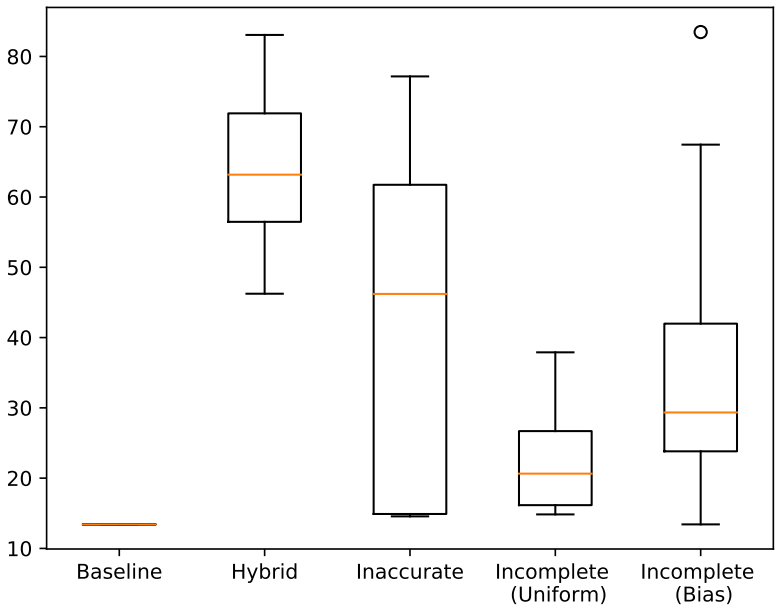
Table 5.10: Quantitative results of the experimental evaluation of EviTraN in CIFAR-10.

Learning Setting	Evidence	ACC (%)	NMI (%)
Unsupervised (Baseline)	-	22.79	13.44
Hybrid	M3	37.34 (+14.55)	46.24 (+32.80)
	M4	43.14 (+20.35)	54.81 (+41.37)
	M5	62.34 (+39.55)	64.92 (+51.48)
	M10	91.97 (+69.18)	83.06 (+69.62)
	M3 & M4	52.86 (+30.07)	61.44 (+48.00)
	M3 & M4 & M5	64.75 (+41.96)	74.23 (+60.79)
Inaccurate	RV3	24.62 (+1.83)	14.66 (+1.22)
	RV10	24.61 (+1.82)	14.56 (+1.12)
	RV3 & RV10	25.00 (+2.21)	14.80 (+1.36)
	RV3 & RV5 & RV10	25.21 (+2.42)	14.90 (+1.46)
	RI3	26.18 (+3.39)	15.35 (+1.91)
	RI10	26.01 (+3.22)	15.08 (+1.64)
	M3 & RV3	36.97 (+14.18)	46.22 (+32.78)
	M3 & RV3 & RV10	36.67 (+13.88)	46.21 (+32.77)
	M4 & RV3 & RV10	44.68 (+21.89)	54.37 (+40.93)
	M5 & RV3 & RV10	62.49 (+39.70)	65.58 (+52.14)
	M3 & M4 & RV3	53.04 (+30.25)	61.74 (+48.30)
	M3 & M5 & RV3	60.56 (+37.77)	71.39 (+57.95)
	M3 & M5 & RV3	63.42 (+40.63)	77.16 (+63.72)
Incomplete (Uniform)	M3 ($M \downarrow_{70\%}$)	30.81 (+8.02)	26.47 (+13.03)
	M3 ($M \downarrow_{90\%}$)	25.58 (+2.79)	16.29 (+2.85)
	M4 ($M \downarrow_{70\%}$)	32.78 (+9.99)	27.36 (+13.92)
	M4 ($M \downarrow_{90\%}$)	24.85 (+2.06)	15.73 (+2.29)
	M5 ($M \downarrow_{70\%}$)	28.44 (+5.65)	21.56 (+8.12)
	M5 ($M \downarrow_{90\%}$)	24.95 (+2.16)	15.76 (+2.32)
	M10 ($M \downarrow_{70\%}$)	33.94 (+11.15)	24.03 (+10.59)
	M10 ($M \downarrow_{90\%}$)	23.99 (+1.20)	14.84 (+1.40)
	M3 & M4 ($M \downarrow_{70\%}$)	31.99 (+9.20)	29.18 (+15.74)
	M3 & M4 ($M \downarrow_{90\%}$)	27.08 (+4.29)	17.87 (+4.43)
	M3 & M4 & M5 ($M \downarrow_{70\%}$)	42.95 (+20.16)	37.90 (+24.46)
	M3 & M4 & M5 ($M \downarrow_{90\%}$)	28.89 (+6.10)	19.72 (+6.28)
Incomplete (Bias)	M3 (AC $\downarrow$ 1)	29.16 (+6.37)	28.74 (+15.30)
	M3 (AC $\downarrow$ 2)	22.77 (-0.02)	13.42 (-0.02)
	M4 (AC $\downarrow$ 1)	33.18 (+10.39)	38.52 (+25.08)
	M4 (AC $\downarrow$ 2)	29.87 (+7.08)	26.94 (+13.50)
	M5 (AC $\downarrow$ 1)	40.28 (+17.49)	52.37 (+38.93)
	M5 (AC $\downarrow$ 2)	29.70 (+6.91)	35.38 (+21.94)
	M10 (AC $\downarrow$ 1)	87.70 (+64.91)	83.47 (+70.03)
	M10 (AC $\downarrow$ 2)	72.10 (+49.31)	67.45 (+54.01)
	M3 & M4 (AC $\downarrow$ 1)	28.78 (+5.99)	28.37 (+14.93)
	M3 & M4 (AC $\downarrow$ 2)	24.38 (+1.59)	13.79 (+0.35)
	M3 & M4 & M5 (AC $\downarrow$ 1)	29.90 (+7.11)	29.95 (+16.51)
	M3 & M4 & M5 (AC $\downarrow$ 2)	26.82 (+4.03)	14.44 (+1.00)

Notation is similar to previous Tables.

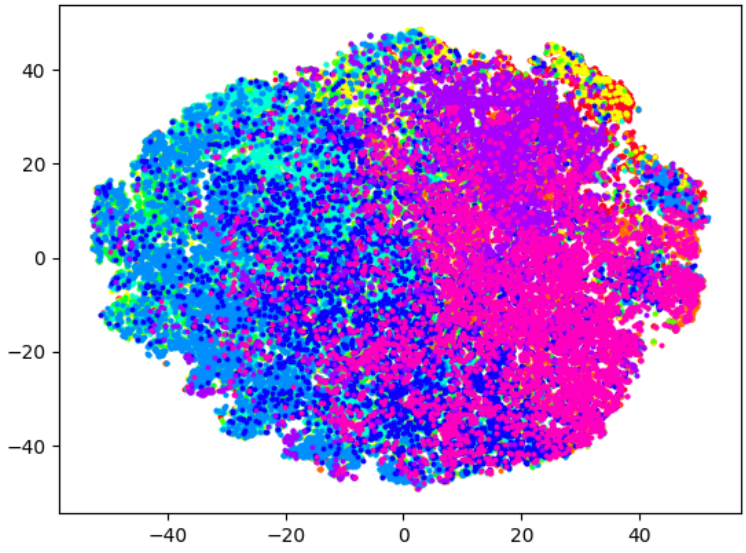


(a) ACC

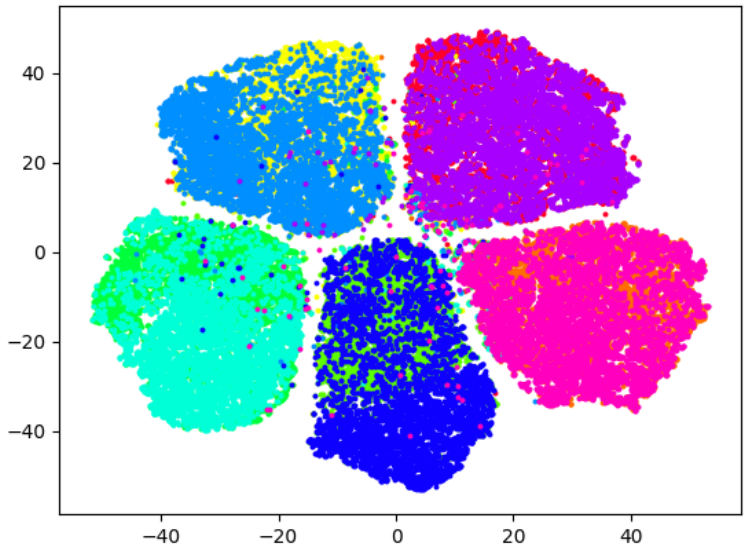


(b) NMI

Figure 5.8: Boxplots of ACC and NMI metrics for experimental evaluation with CIFAR-10 dataset in all learning settings.

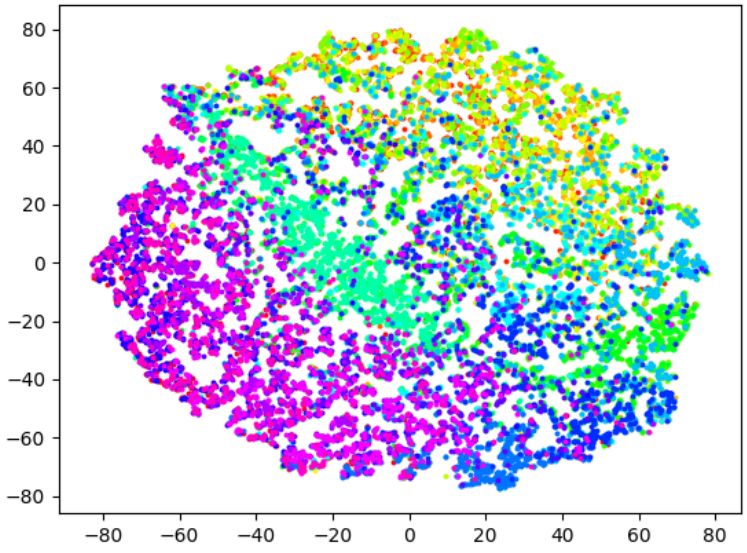


(a) Initial latent space of CIFAR-10

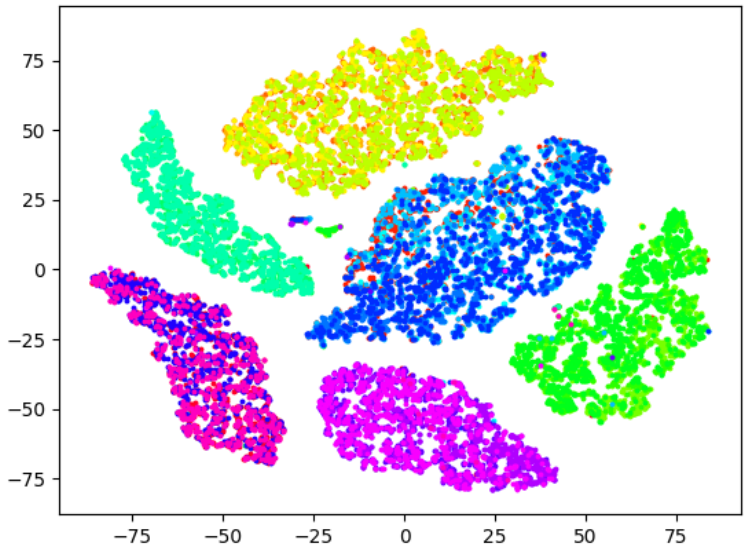


(b) Latent space of CIFAR-10 after with M5

Figure 5.9: State of latent space before (top figure) and after the introduction of external evidence sources (bottom figure), for CIFAR-10. The introduction of external evidence of 5 auxiliary classes, indicates the separation of the initial space into respective distinct groups. Appendix A includes similar figure for MNIST dataset.

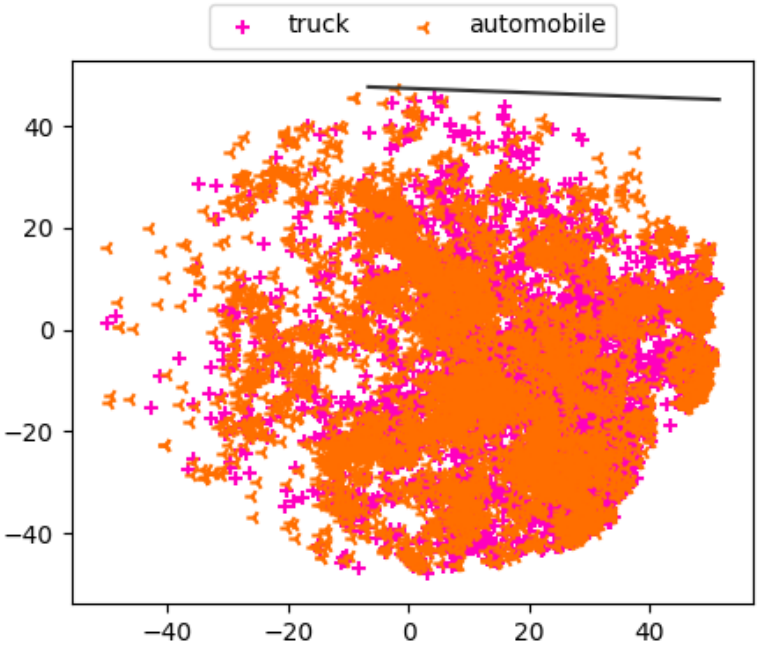


(a) Initial latent space of 20newsgroups

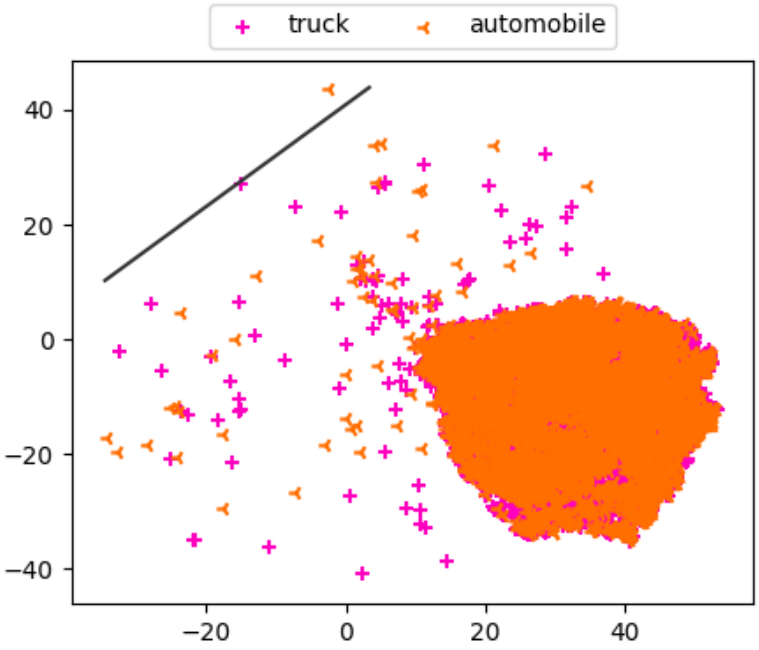


(b) Latent space of 20newsgroups after EviTraN with M6

Figure 5.10: State of latent space before (top figure) and after the introduction of external evidence sources (bottom figure), for 20newsgroups. The introduction of external evidence of 6 auxiliary classes, indicates the separation of the initial space into respective distinct groups. Appendix A includes similar figure for Reuters-100k dataset.

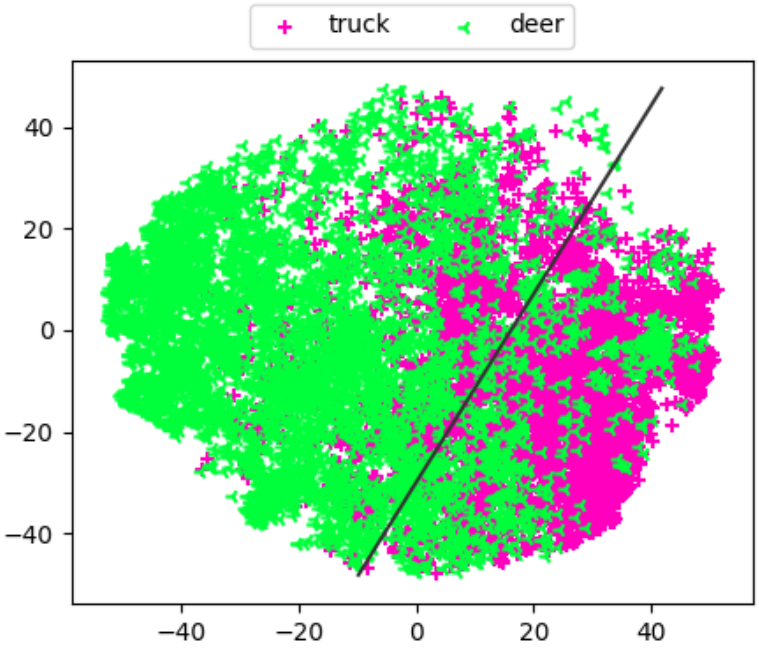


(a) Initial latent representations of Truck and Automobile

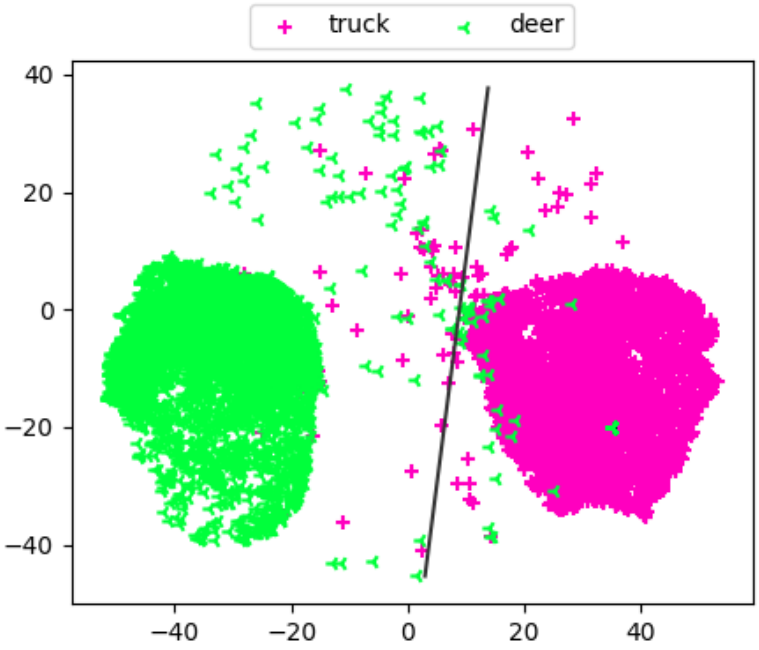


(b) Latent representations of Truck and Automobile after EviTraN with M5

Figure 5.11: State of latent representations of individual auxiliary classes: Truck and Automobile of CIFAR-10, before (top figure) and after EviTraN (bottom figure). Solid line represents the decision boundary predicted by an SVM classifier with a linear kernel.

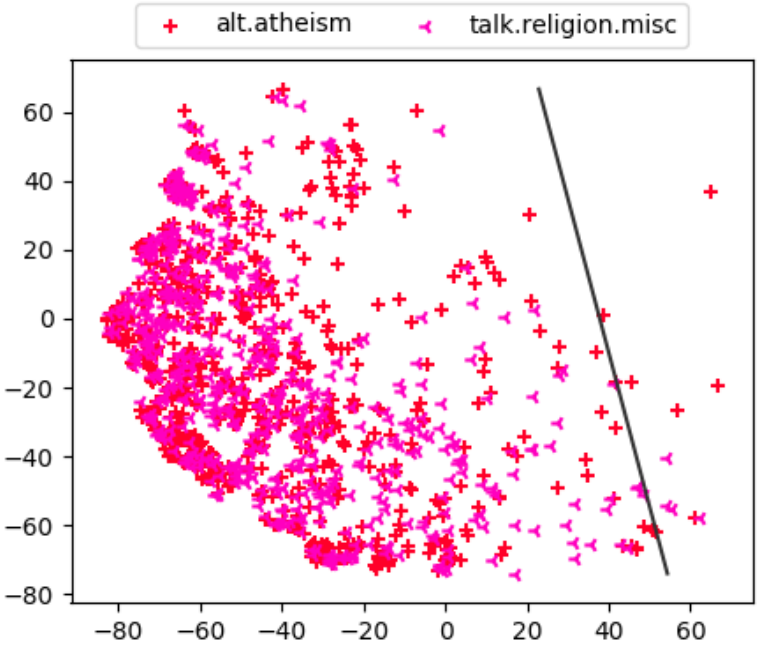


(a) Initial latent representations of Truck and Deer

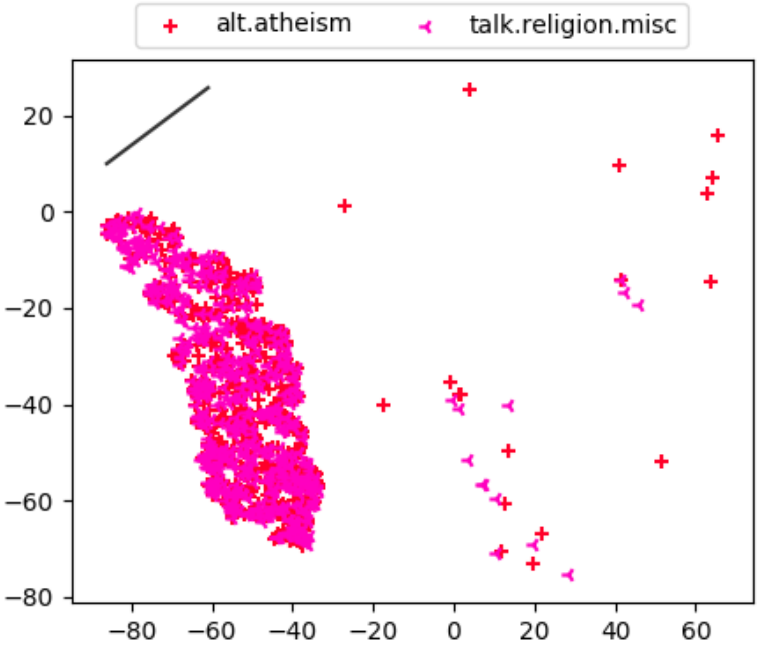


(b) Latent representations of Truck and Deer after EviTraN with M5

Figure 5.12: State of latent representations of individual auxiliary classes: Truck and Deer of CIFAR-10, before (top figure) and after EviTraN (bottom figure). Solid line represents the decision boundary predicted by an SVM classifier with a linear kernel.

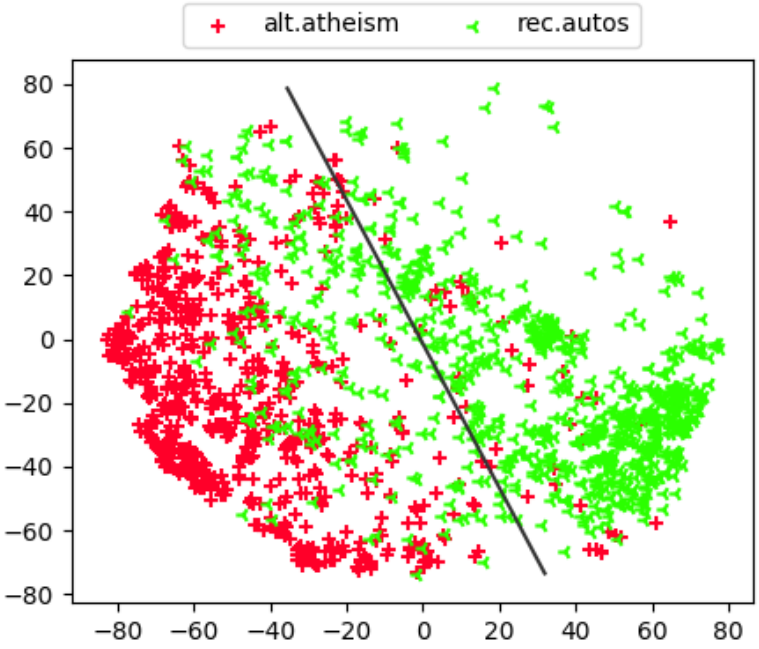


(a) Initial latent representations of Alt.Atheism and Talk.Religion.Misc

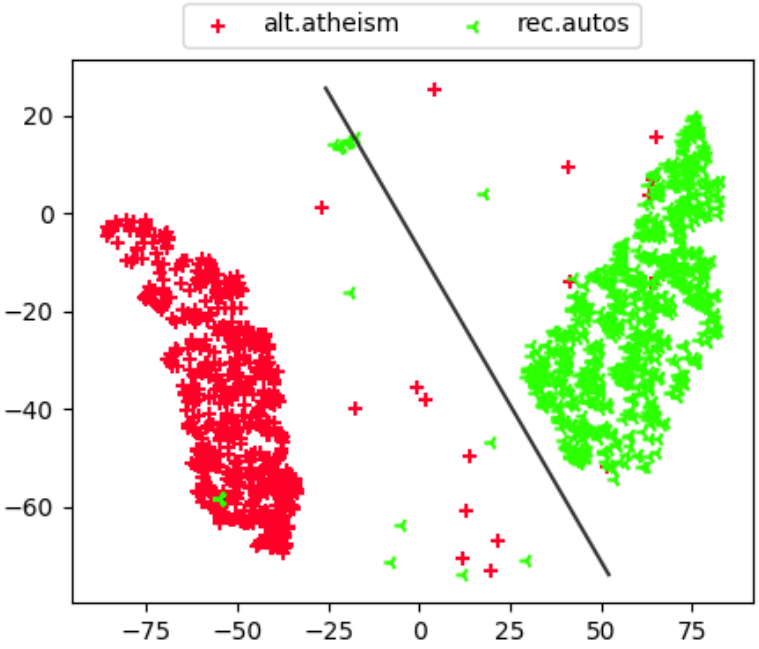


(b) Latent representations of Alt.Atheism and Talk.Religion.Misc after EviTraN with M6

Figure 5.13: State of latent representations of individual auxiliary classes: Alt.Atheism and Talk.Religion.Misc of 20newsgroups, before (top figure) and after EviTraN (bottom figure). Solid line in figures, represents the decision boundary predicted by an SVM classifier with a linear kernel.

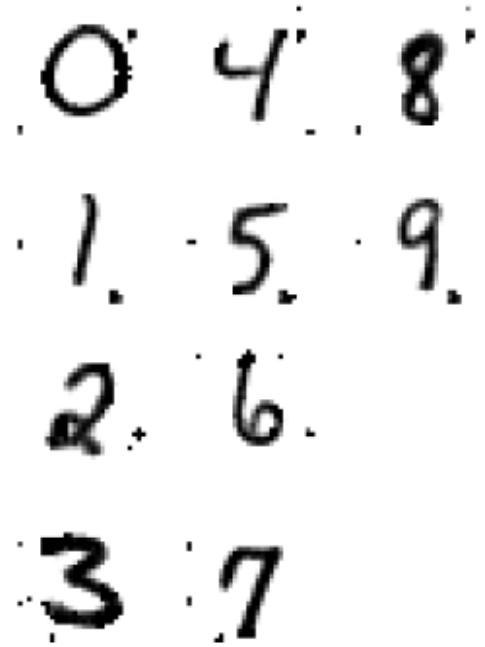


(a) Initial latent representations of Alt.Atheism and Rec.Autos

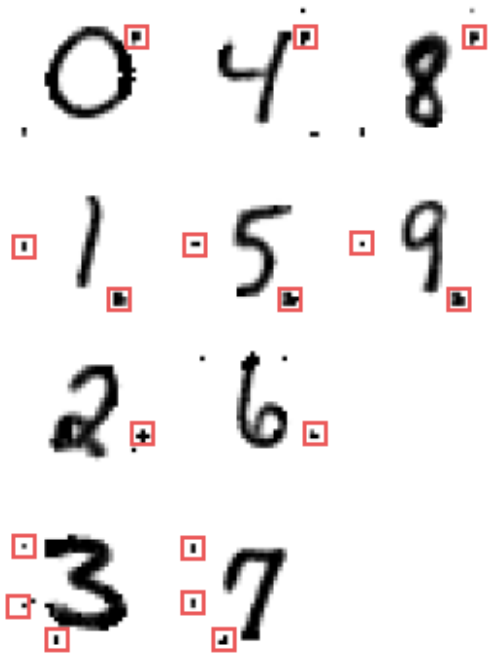


(b) Latent representations of Alt.Atheism and Rec.Autos after EviTraN with M6

Figure 5.14: State of latent representations of individual auxiliary classes: Alt.Atheism and Rec.Autos of 20newsgroups, before (top figure) and after EviTraN (bottom figure). Solid line in figures, represents the decision boundary predicted by an SVM classifier with a linear kernel.



(a) Reconstruction of M5



(b) Highlighted consistent marking

Figure 5.15: Reconstructed digits after introduction of M4 evidence with EviTraN. For visualisation purposes a different training strategy is deployed. Figure (b) highlights the consistent automated marking produced by the decoder. The automated marking is an outcome of M4 evidence influencing the initial latent space. The marking is consistent for digits that belong in the same group.

data in order to achieve better performance. Since clustering algorithms do not involve any trainable parameters, they rely on the availability of the data instances during prediction time in order to generalise. Using full datasets is a common strategy in deep clustering [24, 84, 44, 163, 164].

The results of unsupervised, hybrid and inaccurate learning settings are the average value of 4 runs. On the other hand, the results of incomplete learning setting are the outcome of a single run. The number of batch size for all experiments is 256. The amount of training epochs during the initialisation step for each dataset is 250, for all datasets. During evidence transfer step, the amount of training epochs is 200, 500, 500 and 200 for MNIST, CIFAR-10, 20newsgroups and Reuters-100k respectively. Greedy layer-wise training of stacked denoising autoencoders in CIFAR-10, 20newsgroups and Reuters-100k involves 30 training epochs for each individual shallow autoencoder. The choice of learning rate and learning rate decay varies depending on the dataset[8]. The choice of optimiser is Stochastic Gradient Descent (SGD) for all experiments.

Despite the presence of malicious evidence sources in the inaccurate learning setting, EviTraN is able to preserve its original effectiveness. While, in some exceptions the effectiveness is below the baseline, the discrepancy is not significant. At the same time during evaluations in hybrid learning setting, which involves meaningful evidence sources (i.e., task outcomes), EviTraN utilises these external relations in order to increase the performance of the process of learning latent representations. The above behaviour is compatible with both metrics. The above properties found in quantitative evaluation shown in Tables 5.7, 5.8, 5.9 and 5.10, support that EviTraN is robust and effective. Being effective and robust despite the introduction of inaccurate evidence sources is a result of the composite training objective and the intermediate step (as explained in Section 3.4.2).

EviTraN is also robust against incomplete evidence sources. In addition, certain cases indicate that incomplete evidence sources can lead to gain in performance. The performance heavily depends on the level of incompleteness found within each evidence source. Evidence with uniformly missing samples is mainly effective. These evidence sources yield effectiveness inversely proportionate to the amount of missing samples. In other words, evidence with uniformly missing sample with low amount of missing instances yields better performance and vice versa.

On the other hand, biased evidence sources (which are missing auxiliary classes) are more volatile. Whether they act as low quality evidence or not, depends on the amount of missing auxiliary classes. During experimental settings where the amount of total auxiliary classes is low, e.g., $y \ mod \ 3$ evidence of MNIST, the removal of

[8]More information regarding the hyperparameters can be found in the code repositories

one or two auxiliary classes deteriorates its performance. Yet, evidence sources with high total amount of classes, such as the labelset of MNIST which consists of 10 classes, removing one or two auxiliary classes does not deteriorate the performance as much. In any case, the performance does not decrease bellow the initial performance throughout the experimental evaluation.

Boxplots 5.5, 5.6, 5.7, 5.6 also support the conclusion that EviTraN is effective and robust. During all boxplots hybrid learning experiments severely outperforms the baseline solution. At the same time, inaccurate learning experiments are close to the value of baseline solution, but never lower. Outlier samples depicted as circles in inaccurate learning settings are experiments that involve meaningful evidence along with non-meaningful evidence (frequent in experiments with double and triple evidence sources). Incomplete learning experiments due to their volatility have a broad range of values. However, they consistently outperform the baseline solution.

Qualitative evaluation involves transformation of the original latent representations from feature vectors of 10 features into 2-dimensional vectors, with the use of t-SNE [165]. Hinton and Roweis [166] proposed Stochastic Neighbour Embedding (SNE), which aims to preserve the structure of data samples from a high-dimensional space into a low-dimensional manifold. SNE performs fit of a Gaussian distribution centred over each data sample in the high-dimensional space. The idea behind the fit of the Gaussian distribution is to find prospective neighbours of each data sample. Meaning that, data samples that are close in the high-dimensional space should also be close in the lower-dimensional space produced by SNE. The cost function for measuring the distance between samples is the Kullback-Leibler Divergence. Van der Maaten and Hinton [165] proposed the use of Student-t distribution instead, leading to t-Stochastic Neighbour Embedding (t-SNE).

During study of individual auxiliary classes (Figures 5.11, 5.12, 5.13 and 5.14), a SVM classifier with a linear kernel is trained with input each pair of latent representations and ground truth labels. The aforementioned figures depict the decision boundaries of the SVM classifier as a solid line. The idea behind the visualization of a linear classifier is to display how easy it is for an algorithm with linear distances to distinguish between latent representations before and after the use of EviTraN.

Qualitative evaluation, as shown in Figures 5.9, 5.10, 5.11, 5.12, 5.13 and 5.14, provides additional insight regarding the increased effectiveness of EviTraN. As shown in Figure 5.9 and 5.10, the initial state of the latent space (without incorporation of evidence) bears resemblance to a Gaussian distribution. In practice, this means that the latent representations tend to cluster close to a mean. From the perspective of clustering, the decision boundaries between classes is not clear, since the distances between latent samples are small. By incorporating external evidence, EviTraN is able to manipulate the initial latent space into a more appopriate space. EviTraN

preserves the original properties of the latent space, i.e., preserves distribution of samples close to a mean. Yet, the composite training objective manipulates the latent space to represent appropriate distances between individual auxiliary classes. Therefore, indicating distinction between auxiliary class samples.

As shown in Figures 5.11, 5.12, 5.13 and 5.14, studying the effects of EviTraN in pairs of auxiliary classes shows that an SVM classifier with linear kernel is able to clearly distinguish the decision boundaries between classes, where the evidence source indicates that they belong in different auxiliary classes. At the same time, classes where the evidence source does not indicate their separation, preserve a similar structure with their initial latent space counterpart. Since classes that do not indicate separation preserve their structure, the gain in effectiveness is an effect of reducing incorrect cluster assignment of the available data instances.

In an attempt to visualise the learning process of EviTraN, a different training strategy is deployed to produce Figure 5.15. During this process, auxiliary layers Q are repositioned after the decoder, i.e., the reconstruction layer. Furthermore, Adam optimiser is repurposed for more aggressive optimisation and scale the reconstructed data samples with `maxabs_scale` of scikit-learn, which scales a vector in [-1, 1] range without disturbing its sparsity. Figure 5.15 depict the results of the above visualisation process, with introduction of M4 evidence (meaningful evidence of 4 distinct groups representing $hash(y)\ mod\ 4$).

After introduction of M4 evidence, the reconstructed MNIST samples present a consistent automated marking. The marking is consistent with the distinct groups of M4. The automated marking indicates that despite some data instances bearing similar data features (as depicted in Section 5.1.4), M4 allows distinction of these samples based high-level semantic information (relation $hash(y)\ mod\ 4$). Therefore, any clustering algorithm should yield increased performance, as it should not produce clusters with similar data features but different semantic meanings (e.g., 3 and 8 digits).

5.3 Empirical Analysis of Relevance

As mentioned in Section 4.2, investigation of hypothesis regarding the auxiliary training objective of EviTraN being similar to information bottleneck term $I(Z;Y)$, requires an empirical analysis. This section includes the results and conclusions of said analysis.

The goal of the previous evaluation process was to investigate the satisfaction of effectiveness and robustness criteria. To this end, clustering metrics that involve ground truth labels were used. The satisfaction of effectiveness criterion, as well as,

the involved metrics, indicate that the evidence transfer step of EviTraN involves information regarding the ground truth labels associated with the dataset.

The empirical analysis of investigating the correlation between relevance term of IB and auxiliary learning term of EviTraN, requires an appropriate quantification metric. A well-received metric from the domain of feature selection lends itself to the above investigation. Feature selection aims to reduce the amount of features into maintaining the most relevant ones. For this purpose, the metric of minimum redundancy — maximum relevance (mRMR) [167] has been proposed.

MRMR suggests that the final outcome of a feature selection method should satisfy two criteria. First, each feature in the final set should have minimum redundancy along the features of the set. Second, each feature should have maximum relevance in comparison to the ground truth labels. Relevance, as shown in Equation 5.6 measures the mutual information between a set of features and a set of ground truth labels. Computation of relevance is complex, since mutual information involves the computation of conditional probability which is often intractable. The following empirical analysis involves two computational variations of relevance.

$$D(Z, Y) = \frac{1}{|Z|} \sum_{z_i \in Z} I(z_i; Y) \tag{5.6}$$

The above notation is similar to that found in the work of Peng et al. [167], for consistency purposes. To aid reading comprehension, notation S and c is switched to Z and Y that represent the set of learned representations and class labels respectively.

The first variation is based on K-Nearest Neighbour [168, 169] and computes the mutual information between discrete and continuous sets [170]. The other variation involves F-test values [171]. During the first variation, the mutual information is the average of the following metric over all points in the dataset:

$$I_i = \psi(N) - \psi(N_{x_i}) + \psi(K) - \psi(m_i) \tag{5.7}$$

The above notation is the same as in the work of Ross [170], ψ is the digamma function, N is the total number of data points and K is the hyperparameter of choice of finding k-nearest neighbours. While N_x are data points with the same discrete value as x_i and m_i is the number of neighbours within some distance d. On the other hand, computation with F-statistic requires computation of:

$$\begin{aligned} F(i, C) &= \frac{\left[\sum_c n_c(\overline{i_c} - \overline{i})/(C-1)\right]}{\sigma^2} \\ V_F &= \frac{1}{|S|} \sum_{i \in S} F(i, C) \end{aligned} \tag{5.8}$$

Table 5.11: Comparison of Relevance with ACC and NMI metrics for MNIST.

Learn. Set.	Evidence	Rel. MI	Rel. FT	ACC	NMI	R. Var.
Uns. (I)	-	0.47	0.25	0.82	0.76	-
Hybrid (II)	M3	0.58 (+0.11)	0.36 (+0.11)	0.96 (+0.14)	0.90 (+0.14)	0.4
(III)	M4	0.60 (+0.13)	0.35 (+0.12)	0.96 (+0.14)	0.91 (+0.15)	0.2
(IV)	M10	0.62 (+0.15)	0.40 (+0.15)	0.97 (+0.15)	0.92 (+0.16)	0.4
(V)	M3 & M4	0.66 (+0.19)	0.46 (+0.21)	0.98 (+0.16)	0.94 (+0.18)	0.6
Inacc. (VI)	RV3	0.48 (+0.01)	0.25	0.82	0.76	0
(VII)	RV10	0.48 (+0.01)	0.25	0.82	0.77 (+0.01)	0
(VIII)	RV3 & RV10	0.48 (+0.01)	0.25	0.82	0.76	0
(IX)	RI3	0.48 (+0.01)	0.25	0.82	0.76	0
(X)	RI10	0.48 (+0.01)	0.25	0.82	0.76	0
(XI)	M3 & RV3	0.58 (+0.11)	0.35 (+0.10)	0.96 (+0.14)	0.90 (+0.14)	0.4

$M\#$: Meaningful evidence, $\#$ the represents number of auxiliary classes — width of evidence samples.
RV: Random Values.
RI: Random Index.
$Rel.MI$: Relevance with Mutual Information implementation
$Rel.FT$: Relevance with F-test implementation
$R.Var.$: Rank Variation

The above notation is similar to that found in the work of Ding and Peng [171], $\bar{i}$ is the mean value of data point i and $\overline{i_c}$ is the mean value of data points within c class, while n_c is the total amount of points within c class. To aid reading comprehension, h was switched with C.

5.3.1 Results of Empirical Analysis

The implementation used during the first variation (from scikit-learn [158]) involves some stochasticity. To this end, the relevance metric that involves mutual information, as shown in Tables 5.11, 5.12, 5.13 and 5.14, are the average of 50 runs. The choice of K for the nearest neighbour algorithm is $K = 3$. The aforementioned tables contain the measurement of overall relevance, i.e., the average value of relevance between all latent features and ground truth labels (as shown in Equation 5.6). The measurements of relevance using F-test has been normalised using the L2 norm.

Independent of the implementation variation, relevance seems to follow the same pattern as ACC and NMI. During hybrid learning setting, which involves meaningful evidence, the relevance is increased (compared to the baseline result). At the same time, during inaccurate learning setting, the relevance is barely increased or remains completely stable. This behaviour is consistent with ACC and NMI metrics.

In practice, this means that the incorporation of meaningful evidence, enables the

Table 5.12: Comparison of Relevance with ACC and NMI metrics for 20newsgroups.

Learn. Set.	Evidence	Rel. MI	Rel. FT	ACC	NMI	R. Var.
Uns. (I)	-	0.28	0.05	0.21	0.25	-
Hybrid (II)	M5	0.87 (+0.59)	0.39 (+0.34)	0.34 (+0.13)	0.58 (+0.33)	0.8
(III)	M6	0.97 (+0.69)	0.42 (+0.37)	0.33 (+0.12)	0.60 (+0.35)	0.8
(IV)	M20	1.14 (+0.86)	0.55 (+0.50)	0.87 (+0.66)	0.90 (+0.65)	0.7
(V)	M5 & M6	1.08 (+0.80)	0.57 (+0.52)	0.47 (+0.26)	0.68 (+0.43)	0.9
Inacc. (VI)	RV3	0.29 (+0.01)	0.06 (+0.01)	0.22 (+0.01)	0.25	0
(VII)	RV10	0.30 (+0.02)	0.06 (+0.01)	0.23 (+0.02)	0.26 (+0.01)	0.2
(VIII)	RV3 & RV10	0.29 (+0.01)	0.06 (+0.01)	0.23 (+0.02)	0.26 (+0.01)	0.4
(IX)	RI5	0.29 (+0.01)	0.06 (+0.01)	0.21	0.25	0.2
(X)	RI20	0.28	0.06 (+0.01)	0.22 (+0.01)	0.26 (+0.01)	0.2
(XI)	M5 & RV3	0.83 (+0.55)	0.44 (+0.39)	0.32 (+0.11)	0.54 (+0.29)	0.6

$M\#$: Meaningful evidence, $\#$ the represents number of auxiliary classes — width of evidence samples.
RV: Random Values.
RI: Random Index.
$Rel.MI$: Relevance with Mutual Information implementation
$Rel.FT$: Relevance with F-test implementation
$R.Var.$: Rank Variation

Table 5.13: Comparison of Relevance with ACC and NMI metrics for Reuters-100k.

Learn. Set.	Evidence	Rel. MI	Rel. FT	ACC	NMI	R. Var.
Uns. (I)	-	0.28	0.24	0.41	0.33	-
Hybrid (II)	M4	0.36 (+0.08)	0.35 (+0.11)	0.43 (+0.02)	0.36 (+0.03)	0.5
(III)	M5	0.39 (+0.11)	0.37 (+0.15)	0.47 (+0.06)	0.39 (+0.06)	0.7
(IV)	M10	0.38 (+0.10)	0.44 (+0.20)	0.48 (+0.07)	0.41 (+0.08)	0.7
(V)	M4 & M5	0.43 (+0.15)	0.49 (+0.25)	0.51 (+0.10)	0.42 (+0.09)	0.7
Inacc. (VI)	RV3	0.28	0.24	0.41	0.33	0.2
(VII)	RV10	0.28	0.23 (-0.01)	0.42 (+0.01)	0.33	0.4
(VIII)	RV3 & RV10	0.28	0.23 (-0.01)	0.41	0.33	0.4
(IX)	RI4	0.28	0.23 (-0.01)	0.41	0.33	0.2
(X)	RI10	0.28	0.23 (-0.01)	0.41	0.33	0
(XI)	M4 & RV3	0.35 (+0.07)	0.34 (+0.10)	0.43 (+0.02)	0.36 (+0.03)	0.4

$M\#$: Meaningful evidence, $\#$ the represents number of auxiliary classes — width of evidence samples.
RV: Random Values.
RI: Random Index.
$Rel.MI$: Relevance with Mutual Information implementation
$Rel.FT$: Relevance with F-test implementation
$R.Var.$: Rank Variation

latent space to increase its mutual information (or relevance) with the ground truth labels, while remaining stable during introduction of malicious evidence. Therefore,

Table 5.14: Comparison of Relevance with ACC and NMI metrics for CIFAR-10.

Learn. Set.	Evidence	Rel. MI	Rel. FT	ACC	NMI	R. Var.
Uns. (I)	-	0.11	0.04	0.23	0.13	-
Hybrid (II)	M3	0.59 (+0.48)	0.43 (+0.39)	0.38 (+0.15)	0.46 (+0.33)	0.8
(III)	M4	0.59 (+0.48)	0.23 (+0.21)	0.43 (+0.20)	0.55 (+0.42)	0.8
(IV)	M5	0.54 (+0.43)	0.19 (+0.17)	0.62 (+0.39)	0.65 (+0.52)	0.7
(V)	M10	0.58 (+0.47)	0.39 (+0.35)	0.92 (+0.69)	0.83 (+0.70)	0.7
(VI)	M3 & M4	0.80 (+0.69)	0.68 (+0.64)	0.53 (+0.30)	0.61 (+0.48)	1
(VII)	M3 & M4 & M5	0.83 (+0.72)	0.47 (+0.45)	0.65 (+0.42)	0.74 (+0.61)	0.8
Inacc. (VIII)	RV3	0.11	0.04	0.25 (+0.02)	0.15 (+0.02)	0
(IX)	RV10	0.11	0.04	0.25 (+0.02)	0.15 (+0.02)	0
(X)	RV3 & RV10	0.12 (+0.01)	0.04	0.25 (+0.02)	0.15 (+0.02)	0
(XI)	RV3 & RV5 & RV10	0.12 (+0.01)	0.02	0.25 (+0.02)	0.15 (+0.02)	0.2
(XII)	RI3	0.12 (+0.01)	0.04	0.26 (+0.03)	0.16 (+0.03)	0.2
(XIII)	RI10	0.11	0.04	0.26 (+0.03)	0.15 (+0.02)	0
(XIV)	M3 & RV3	0.59 (+0.48)	0.44 (+0.40)	0.37 (+0.14)	0.46 (+0.33)	0.9
(XV)	M3 & RV3 & RV10	0.60 (+0.49)	0.27 (+0.25)	0.37 (+0.14)	0.46 (+0.33)	0.8
(XVI)	M3 & M4 & RV3	0.79 (+0.68)	0.40 (+0.38)	0.53 (+0.30)	0.62 (+0.49)	0.9
(XVII)	M3 & M5 & RV3	0.78 (+0.67)	0.45 (+0.43)	0.60 (+0.37)	0.71 (+0.58)	0.7
(XVIII)	M4 & RV3 & RV10	0.59 (+0.48)	0.29 (+0.27)	0.45 (+0.22)	0.54 (+0.41)	0.8
(XIX)	M4 & M5 & RV3	0.77 (+0.66)	0.46 (+0.44)	0.63 (+0.40)	0.77 (+0.64)	0.9
(XX)	M5 & RV3 & RV10	0.54 (+0.43)	0.23 (+0.21)	0.62 (+0.39)	0.65 (+0.52)	0.7

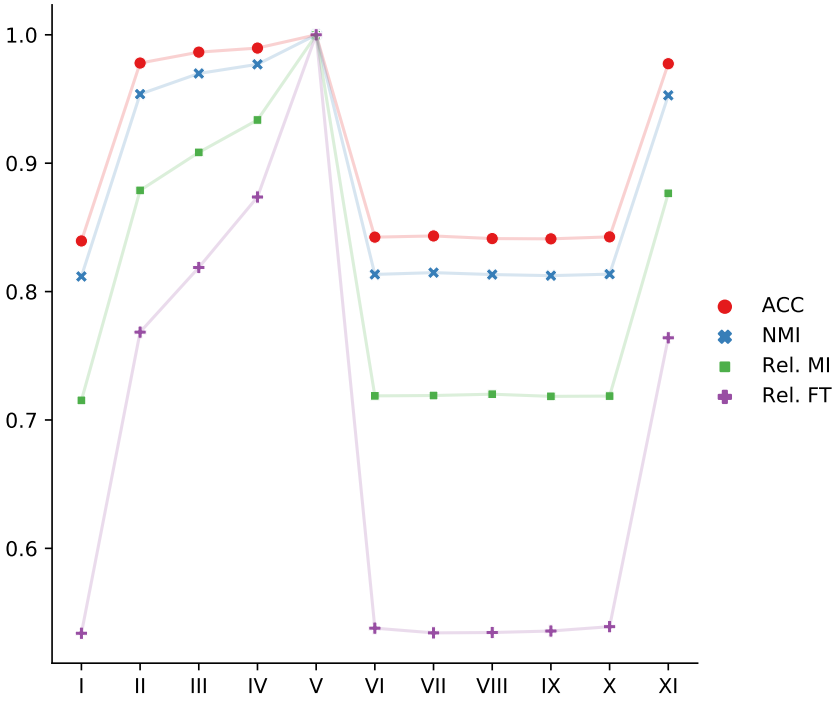
$M\#$: Meaningful evidence, $\#$ the represents number of auxiliary classes — width of evidence samples.
RV: Random Values.
RI: Random Index.
$Rel.MI$: Relevance with Mutual Information implementation
$Rel.FT$: Relevance with F-test implementation
$R.Var.$: Rank Variation

the increased performance of EviTraN in clustering metrics that involve ground truth labels (only during evaluation), such ACC and NMI, is explained by the increase of relevance.

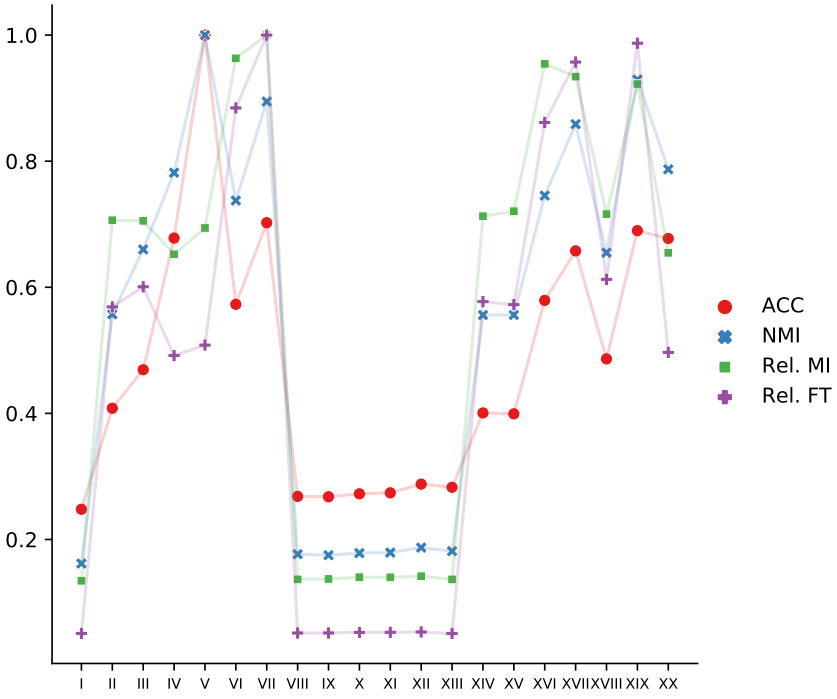
Additional qualitative evaluation provided in Figures 5.16 and 5.17, suggests that the four metrics: ACC, NMI, Relevance with mutual information implementation and Relevance with F-test implementation, follow similar patterns. Although some discrepancy between values exists, the increase and decrease in relevance and clustering performance is consistent. For visualisation purposes, all metrics have been normalised using the `max` method from sklearn `normalise`, in range of [0,1].

The above empirical analysis, indicates that EviTraN has the same effects on the latent space of the autoencoder as the proposed Information Bottleneck method. As shown in Chapter 4, the reconstruction objective of the autoencoder is similar to the compression term of IB. At the same time, as shown in this analysis, the relevance of the latent space with the ground truth labels (which are external to the unsupervised learning of the autoencoder) is increased during hybrid learning with meaningful evidence sources.

In order to study the effects of EviTraN on individual latent features, the proposal and measurement of a new metric called "Rank Variation" is required. Rank variation

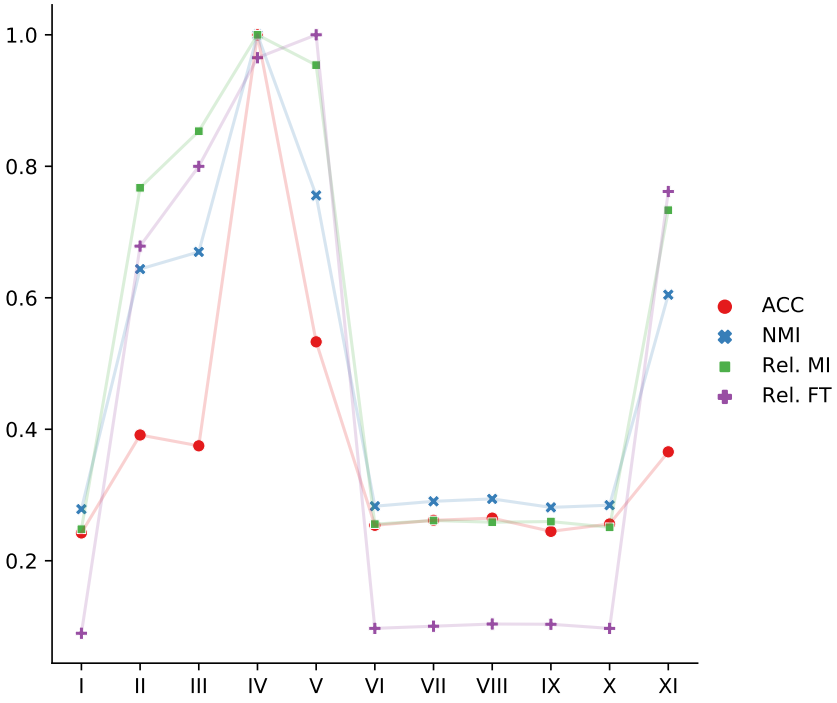


(a) MNIST

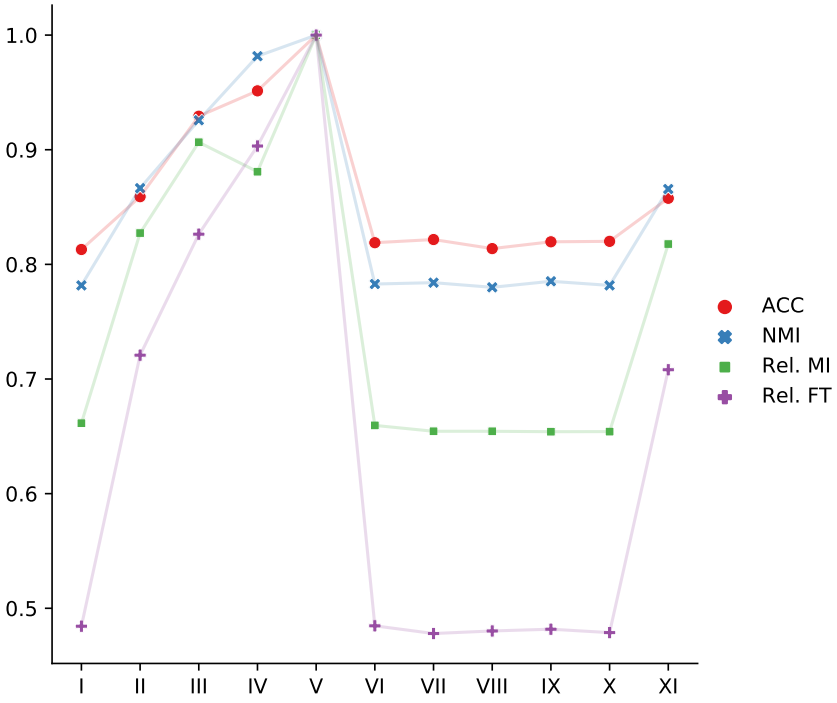


(b) CIFAR-10

Figure 5.16: Comparison between Relevance (both implementations), ACC and NMI metrics for MNIST and CIFAR-10. Despite the value discrepancy between the metrics (for visualisation purposes the metrics have been normalised to [0, 1]) consistent fluctuations are present in all four metrics.

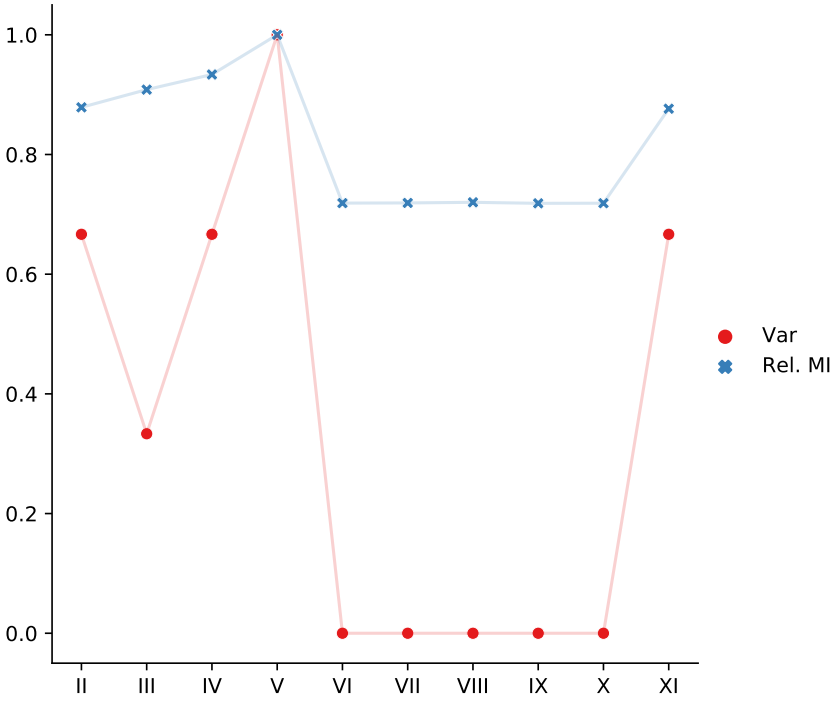


(a) 20newsgroups

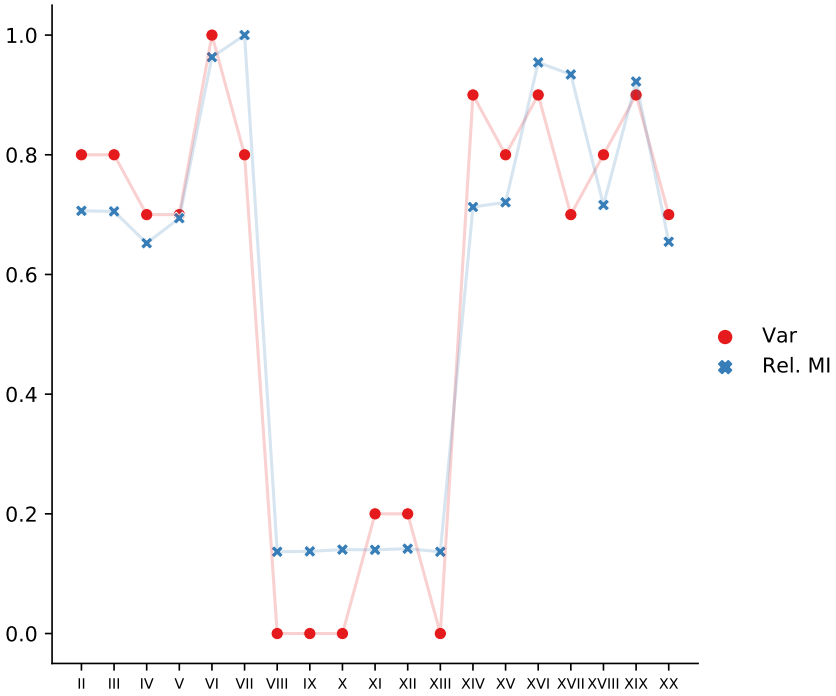


(b) Reuters-100k

Figure 5.17: Comparison between Relevance (both implementations), ACC and NMI metrics for 20newsgroups and Reuters-100k. Despite the value discrepancy between the metrics (for visualisation purposes the metrics have been normalised to [0, 1]) consistent fluctuations are present in all four metrics.

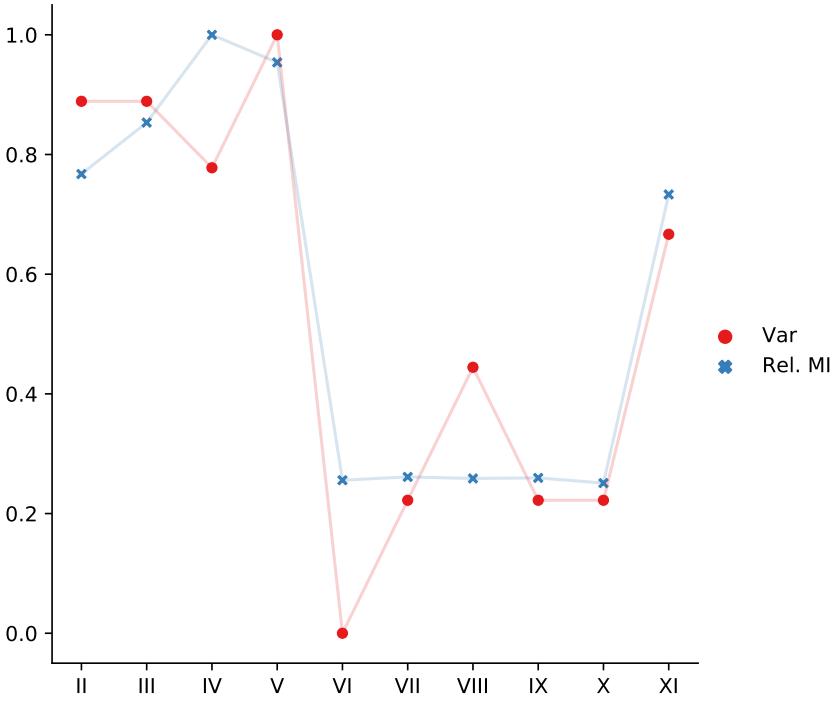


(a) MNIST

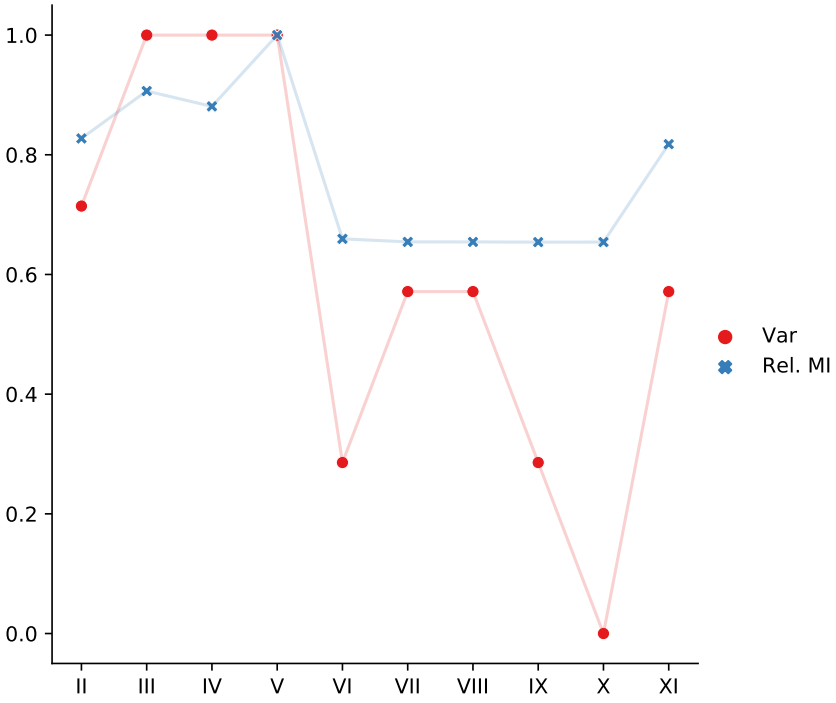


(b) CIFAR-10

Figure 5.18: Comparison between Relevance (Mutual Information implementation) and Rank Variation for MNIST and CIFAR-10. Both metrics have been normalised for visualisation purposes into a [0, 1] range. Consistent fluctuations are present in both metrics.



(a) 20newsgroups



(b) Reuters-100k

Figure 5.19: Comparison between Relevance (Mutual Information implementation) and Rank Variation for 20newsgroups and Reuters-100k. Both metrics have been normalised for visualisation purposes into a [0, 1] range. Consistent fluctuations are present in both metrics.

is a simple metric that measures the shifts in rank before and after the introduction of evidence. First, an initial ranking of the most relevant features based on the relevance metric is created. After the evidence transfer step, a new ranking is created. The rank variation metric compares the shift of ranks from the initial ranking to the ranking after the evidence transfer step. Rank variation is divided by the total amount of features for normalisation purposes. The proposed rank variation is depicted in Equation 5.9.

$$RankVar = \frac{|\sum_{i \in N} R^{(i)}_{init} \neq R^{(i)}_{EviTraN}|}{N} \tag{5.9}$$

Rank variation is shown in Tables 5.11, 5.12, 5.13 and 5.14 for all experiments. The variation of rankings is also consistent with the above metrics. This can be observed also in Figures 5.18 and 5.19. During hybrid learning setting, which increases the overall relevance, more features shift their ranks. While during inaccurate learning setting, the ranking remains stable or low variation is observed. Low variation during inaccurate setting is due to swapping between two or four latent features. Since these shifts do not produce any increased performance in any of the other metrics these re-rankings do not provide any additional insight, but are rather result of additional training with the reconstruction objective.

The following chapter includes experimental evaluation of EviTraN in a realistic scenarion of detecting severe weather in an unsupervised manner.

Chapter 6

Evaluation of Evidence Transfer in a Realistic Scenario

This chapter includes an investigation of the effectiveness of EviTraN in a realistic scenario of detecting severe weather, in an unsupervised manner. Furthermore, it includes an introduction to the use case and the main concepts of the task at hand (anomaly detection), the experimental setting, as well as, results of the experimental evaluation of severe weather detection.

6.1 Use Case: Unsupervised Severe Weather Detection

Weather is a complex concept that involves a plethora of variables, for multiple time instances and pressure levels. Accurately predicting weather variables is not straightforward due to being volatile. Its volatility results from the fact that is affected by a plethora of factors, such as geographic location, past conditions or season. However, one fact that is generally accepted is: that the weather conditions rarely shift rapidly or spontaneously. In other words, one expects the weather conditions to gradually change, or not to change at all for long periods, e.g., prolonged dry seasons. Therefore, creating the expectation of "normal" weather conditions. The criteria based on which one may characterise the weather as "normal", is heavily based on factors such as geographic location.

Despite, the observation of outliers in weather is a rare occasion. For instance, during summer in Mediterranean locations where the observation of high temperatures is common, the probability of low temperatures is small. Despite each day yielding different fluctuations in temperature, expecting a stable behaviour of high temperatures is a safe assumption. Yet, this indicates that the distinction between

normal and abnormal weather may be sensitive to subtle data features. In the same example, a drop of 3 or 4 degrees could be considered as abnormal for this time frame or season.

Finding abnormal, rare or anomalous occurrences is a task known as *anomaly detection*. The existence of these occurrences is ubiquitous in all data collections. Unless one explicitly wants to detect anomalies for a specific application, the presence of abnormal data is often an unwanted property in a data collection. For that reason, it is common to remove outliers during pre-processing of a data collection that aims to machine learning training. On the other hand, one may argue that since such data are ubiquitous in all data collection then their presence is normal and thus should be included in the training set. Despite, a frequent assumption regarding abnormal data is that they can be found in the outliers of the data distribution.

Yet, that is not always the case. First, in order to find anomalies[1], one must first define what an *anomaly* is. According to Chandola et al. [172], anomalies are data instances that diverge from an expected behaviour or notion of the data collection. Also, according to Chandola et al. [172] anomalies can be classified into three types: (i) "*point anomalies*": data instances that plainly differ from the majority of instances within a data collection, (ii) "*contextual anomalies*": data instances being anomalous only under particular circumstances, (iii) "*collective anomalies*": data instances being abnormal as a collection, but individually being considered as normal.

Point anomalies cultivate the assumption of anomalies existing in the outliers of the data distribution. Contextual anomalies can be defined from "*contextual attributes*" or "*behavioural attributes*". Contextual attributes are the observable circumstances that make the instance anomalous, for example a phone call with 1 hour duration, is normal during business hours but abnormal during the night. Therefore, the time frame that a phone call happens, is a contextual attribute. On the other hand, behavioural attributes are usually unobservable within the data collection.

Severe weather is a contextual anomaly with behavioural attributes. For certain locations such as tropical areas, the occurrence of tornados is very common. Even if a tornado occurs during a not so frequent time period, it does not automatically make it severe, since it may be a mild tornado. To decide whether a weather instance is severe or not, one should study its effects after its occurrence. Severe weather instances are the ones that lead to natural disasters and outcomes, such as damages, fatalities, or erosion. However, such properties are not observable from the data features of the weather instances. Thus, from an unsupervised perspective, accurately predicting such instances is not trivial.

In this realistic scenario, EviTraN transforms unobserved behavioural attributes,

[1]The term anomalies will now be used as an umbrella term that includes occurrences considered as abnormal, rare, anomalous, outliers, etc.

i.e., the impact of severe weather, into observable latent features. Incremental manipulation of EviTraN through joint representation learning with primary data and auxiliary task outcomes, should lead to manipulated latent representations that will highlight the necessary attributes capable of resulting in accurate distinguish between normal and severe weather.

6.2 Experimental Setting

This section includes the experimental setting of severe weather detection with EviTraN. It includes details regarding the weather dataset, extraction of evidence source from text, dealing with imbalanced classes and metrics utilised in the evaluation.

6.2.1 Weather Dataset

The experimental evaluation in the use case of severe weather detection, involves ERA-Interim re-analysis data [173], as the primary dataset. ERA-Interim re-analysis data consist of four dimensional weather variables. Examples of such variables are precipitation, temperature, etc. Re-analysis data include observation data along with prior information from a forecast model, in sequential data assimilation scheme. This assimilation scheme, aims to produce better representation of the atmospheric conditions. Re-analysis data have a grid structure. The two last dimensions represent longitude and latitude values, while the first two represent time and pressure level. Pressure levels represent a variety of atmospheric levels, with the minimum pressure level being measurements at sea level (1 hPa to 1000 hPa). ERA-interim covers a time period of approximately 40 years, with a spatial resolution of less than 1° and temporal resolution of 6 hours with global coverage.

The primary data cover a time period from 1st of January 1979 to 31st of May 2018[2], with spatial resolution of approximately 0.7°× 0.7°. However, raw ERA-Interim data have a global coverage. Studying weather events in such scale may lead to increased complexity, as multiple weather events can occur concurrently in multiple locations. To this end, a Cartesian domain that covers the European region is considered. WPS, which is the pre-processor of the well-known Weather Research Forecast (WRF) model [174], is utilised for the reduction. The new spatial resolution consists of 64 × 64 cells of 75 × 75 kilometres in the west-east and south-north axes.

[2]The data were retrieved from the Research Data Archive of National Center for Atmospheric Research in Boulder, Colorado. The archive can be found here: https://rda.ucar.edu/datasets/ds627.0/

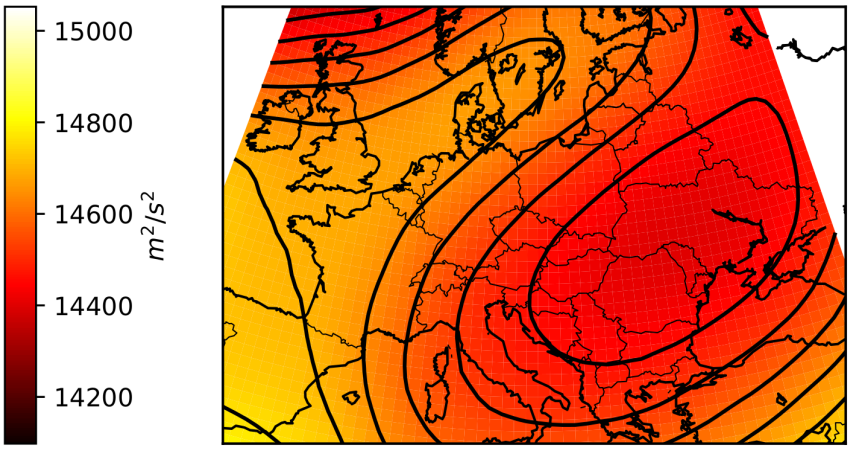


Figure 6.1: Data instance from primary dataset — ERA Interim. It depicts an instance of GHT variable @ 700 hPa.

During the experimental evaluation, the weather variable of choice is the geopotential height (GHT) which is a gravity-adjusted height (as shown in Figure 6.1). GHT is well-known for its predictive properties [175, 176, 177] and its use for extraction of weather patterns repurposed for emergency response [10]. Geopotential heights, depict sequences of patterns that can be utilised in the prediction of weather events, such as circular patterns for cyclones or tornados. Since GHT includes useful patterns, in order to further highlight such features, embeddings from a VGG-16 network are repurposed instead of raw data features, similar to the pre-processing method of CIFAR-10 in Section 5.1.2. To fit the input expectations of VGG-16, three pressure level of GHT (500, 700 and 900 hPa), are introduced in a similar scheme to RGB channels of an image. The final dimensions of the data are 4096 features (64×64, reduced from 3×64×64).

6.2.2 Wikipedia Evidence

Guiding the learning process with the use of EviTraN, requires the existence of external auxiliary outcomes. As mentioned in the previous Section, severe weather events are characterised as such based on the repercussions that they cause. As a result, searching for severe after-effects on Wikipedia can lead to extraction of auxiliary task outcomes. For instance, extraction of heavy rain occurrences can be performed from Wikipedia pages that mention the event of floods in Europe[3].

The experimental evaluation involves severe weather events based on four Wikipedia pages: (i) list of costly or deadly hailstorms, (ii) list of floods in Europe, (iii) list of

[3] https://en.wikipedia.org/wiki/List_of_floods_in_Europe, more regarding the sources of the Wikipedia evidence can be found in the code repositories.

European tornadoes and tornado outbreaks and (iv) list of European windstorms. These Wikipedia pages, allow for effortless correlation between text evidence and primary weather dataset. The date of occurrence mentioned within the pages, lends itself as a common reference between both datasets. The exact timeframe of each severe weather event is not reported in detail. For that reason, during the experiments it is expected that: the minimum period of an event is a single day or four 6-hour samples on the weather dataset.

The experimental evaluation yields the *severe weather* dataset[4]. The data collection includes information regarding severe weather incidents from 1st of January 1979 till May of 2018 (similar to the primary dataset). The collection includes: the name of the severe weather (event name), the type of severe weather (flood, tornado, etc.), countries affected by the severe weather, specific locations (if available), coordinates of the affected countries and description of the severe weather (if available). The coordinates are extracted by performing queries in the GeoNames API[5], the rest of information are found within the Wikipedia pages.

The above data collection, is utilised as a collection of individual binary tasks during the experiments. The experiments involve the use and creation of an auxiliary task outcome that indicates if a 6-hour increment of the primary dataset is normal or severe. This process yields four different binary groupings of the primary dataset (one for each severe weather type). Hail occurrences are not included due to their low total amount (5 incidents after 1979).

6.2.3 Class Balancing and Metrics

The task of anomaly detection suffers from a very specific and characteristic problem, which is unbalanced classes, since anomalies are rare. For any data collection it is safe to assume that most of the data samples within that collection would be not anomalous. This balance, or the lack thereof, leads to generalisation issues as the anomalous class is not sufficiently represented within the collection. In the severe weather use case, the primary dataset consists of approximately 60,000 data samples (57,584). The total amount of severe weather cases is 3,316, which is less than 6% of the total samples. Therefore, restoring balance in the classes is critical.

The investigation of the best balancing strategy, involves three distinct strategies based on re-sampling. The first one revolves around over-sampling the minority class. The second strategy is the inverse procedure of the first one, i.e., under-sampling the majority class. While the third strategy, is a combination of both methods. The SMOTE method [178] is selected as the over-sample method of choice. Synthetic

[4]The dataset can be found at: https://github.com/davidath/severe-weather-dataset
[5]https://www.geonames.org

Algorithm 4: Synthetic Minority Over-sampling TEchnique (SMOTE) [178].

Data: X, y
Result: Augmented X with increased data instances that belong in the anomalous class.

```
Extract K Nearest Neighbours of X with y = y_anom (anomalous class) ;
forall x ∈ X with y = y_anom do
    Randomly select k ∈ K ;
    c = random number ∈ [0, 1] ;
    Diff = x - k ;
    x_new = x + diff * c ;
    Add x_new to X collection ;
end
```

Minority Over-sampling TEchnique (SMOTE) is an over-sampling method based on k-nearest neighbours.

The process of SMOTE is described as follows: for each data instance in the minority class compute its k-nearest neighbours. Then for each data instance, select a random neighbour. Compute the difference between the feature vector of the data instance and the feature vector of the selected neighbour. Select a random number from the range [0, 1] and multiply the difference. Add the result of this process into the real minority instance to create a synthetic minority instance. Algorithm 4 presents the above process in pseudo language.

Random under-sampling, i.e., removing a random amount of instances that belong in the majority class, is selected as the under-sampling method of choice. To under-sample the majority class more sophisticated methods than random under-sampling exist. Edited Nearest Neighbours (EEN) [179] is an under-sampling method that is based on k-nearest neighbours. EEN removes data from the majority class based on a simple criterion. For each data instance extract the k-nearest neighbours of the majority class. Remove data instances which deviate from the majority of k-nearest neighbours. A logistic distribution lends itself as the criterion of deviation. The method performs better with larger instances of k.

A combination of under-sampling the majority class, as well as, over-sampling the minority class can be achieved by the SMOTEENN method [180]. SMOTEENN combines both methods which are based on k-nearest neighbours.

The realistic use case scenario of severe weather detection, involves only two classes: Normal and Anomalous. In anomaly detection task, the accurate prediction of the anomalous class is more important the prediction of the normal class. Unsupervised clustering accuracy and normalised mutual information are more fit for

Table 6.1: Example of true positive/negative and false positive/negative data instances in a binary task.

Ground Truth	Prediction	
	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

multi-class evaluation. *Precision*, *Recall* and *F1-score* metrics, lend themselves to the experimental evaluation of EviTraN in the severe weather scenario. To draw conclusions regarding the investigation of the best balancing strategy, the micro average of the aforementioned metrics is studied. While, during the performance evaluation of EviTraN, the aforementioned metrics are studied only for the anomalous class.

Independent of the task, *precision* and *recall* [181] should be studied in relation to each other. Individually extracting conclusions by only taking into account a single metric, e.g., precision, can lead to misleading outcomes. Equation 6.1 and 6.2 depict precision and recall metrics respectively, according to Ting [181]. Table 6.1 depicts the relation between true positive/negative and false positive/negative data instances, for a binary task. Positive and negative can be relabelled as normal and anomalous.

Precision is the ratio between true positive data instances and the sum of true positive and false positive samples. On the other hand, recall is the ratio between true positive and the sum of true positive and false negative samples. A more clear example from the domain of information retrieval [181], is that precision is equal to: the total number of documents retrieved that are relevant divided by the total number of documents retrieved. While recall is equal to: total number of documents retrieved that are relevant divided by total number of relevant documents in the database.

Consider the example of predicting spam and ham e-mails. In order to increase the precision of the prediction algorithm, one should label all e-mails as spam. However, by doing so, one ignores useful e-mails by categorising them as spam. Therefore, these two metrics should be studied in collaboration. Yet, studying and reporting two metrics at the same time, may be confusing. To this end, *F-score* or *F-measure* or *F1-score* can be studied instead, since it is the harmonic mean between precision and recall metrics. Equation 6.3 depicts F1-score.

$$P = \frac{TP}{TP + FP} \tag{6.1}$$

$$R = \frac{TP}{TP + FN} \tag{6.2}$$

$$F_1 = 2\,\frac{P * R}{P + R} \tag{6.3}$$

6.3 Evaluation of Evidence Transfer in Severe Weather Detection

This section includes the results of the experimental evaluation of detecting individual severe weather events, with the use of EviTraN. It additionally includes the results of the investigation for the best class balancing strategy.

6.3.1 Evaluation Overview

During evaluation of EviTraN in the severe weather detection use case, both the training strategy and evaluation process is similar to previous evaluation process in Chapter 5. First, an initial set of representations is extracted after initialisation training step from an autoencoder model, which is trained in an unsupervised manner. Since this scenario also involves a pre-processing method that produces embeddings from VGG-16, the stacked denoising autoencoder variation is selected. The initial representations are repurposed as input into an unsupervised detection algorithm, in order to infer the performance of the initialisation step. Similar to previous experiments, this is considered as a baseline solution to the task at hand.

Then, the autoencoder is trained according to the evidence transfer step of EviTraN. After, introduction of evidence, a new set of augmented latent representations is extracted. The new set of augmented representations is repurposed into an unsupervised detection algorithm, to infer the performance of evidence transfer step. During investigation of the most suitable class balancing strategy for this use case, a one class SVM method lends itself as the unsupervised detection method. During experimental setting of using EviTraN to improve the performance of severe weather detection, k-means clustering that predicts two clusters (k=2) is used instead. An exception to that is the evaluation pair: windstorm-tornado (more regarding the evaluation pairs are following), where agglomerative clustering is deployed, as it yields better performance for that particular case.

6.3.2 Investigation of Suitable Class Balancing Technique

The experimental investigation for the choice of best class balancing method involves all auxiliary task outcomes, i.e., all types of severe weather. To this end, the

Table 6.2: Results of the experimental investigation, regarding the choice of class balancing method.

(a) Baseline

Metric	SMOTE	Undersample	SMOTEENN
Precision	0.51	0.53	0.51
Recall	0.51	0.53	0.51
F1-Score	0.51	0.53	0.51

(b) Evidence Transfer

Metric	SMOTE	Undersample	SMOTEENN
Precision	0.59 (+0.08)	0.82 (+0.29)	0.55 (+0.04)
Recall	0.59 (+0.08)	0.82 (+0.29)	0.55 (+0.04)
F1-Score	0.59 (+0.08)	0.82 (+0.29)	0.55 (+0.04)

investigation includes the creation of an evidence sources that consists of four classes: normal, windstorm, tornado and flood. This evidence can be seen as the ground truth of the severe weather detection task. Having access to such auxiliary outcome is unrealistic. Despite, drawing conclusions regarding a suitable class balancing technique should be free from implicit uncertainty introduced from nitpicking auxiliary outcomes. Implicit uncertainty can be generated from the process of selecting which task outcome to involve. To test the generalisation of the balancing strategy, the evidence sources is split into 70%-30% train-test sample. During initialisation step, the full data is used for training. While during evidence transfer step, only the training part of the evidence set is involved (70% split).

Table 6.2, reports the quantitative evaluation of involving the above evidence source in EviTraN. In this table, the micro average of all involved metrics for the full dataset is reported. Micro average is more suitable for one-vs-all types of classification (normal vs all types of severe weather events), since it involves the summing of individual constituent parts (true positives, etc.). Equation 6.4 depicts the computation of micro and macro average for the precision metric (assuming two classes).

$$\begin{aligned} P_{micro} &= \frac{(TP1 + TP2)}{(TP1 + TP2 + FP1 + FP2)} \\ P_{macro} &= \frac{P1 + P2}{2} \end{aligned} \tag{6.4}$$

The three class balancing methods mentioned in Section 6.2.3 are preceding both initialisation and evidence transfer steps of EviTraN. Random under-sampling of the majority class, performs better than the rest of balancing techniques, for all

three metrics. Under-sampling is able to reduce the implicit bias found in the data collection, due to the majority class instances overpopulating the input and latent space. Through reduction of redundancy introduced from multiple data instances in the majority class, EviTraN is able to effectively augment and manipulate the initial latent representations into more distinct groups. This is evident both by quantitative results in Table 6.2 and qualitative evaluation in Figure 6.2. On the other hand, during combination and over-sampling, the bias and redundancy invoked by majority class aggressively conditions the latent space into similar structures.

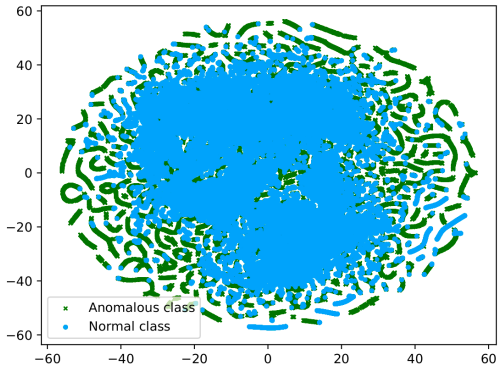
In addition, Figure 6.2 visualises the fact that severe weather detection is a contextual anomaly with behavioural attributes, which are not observable. Observation of the initial latent space for all class balancing strategies suggests that from an unsupervised perspective, severe weather detection can not be distinguished from normal instances. Since, severe weather instances and normal classes are closely tied in the latent space, i.e., their feature vector distance is low, accurately predicting severe weather instances is not feasible.

6.3.3 Individual Severe Weather Detection

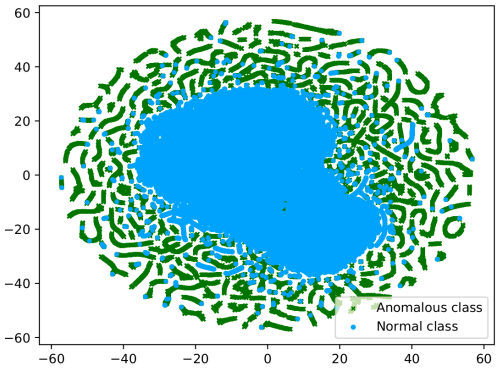
The experimental evaluation of EviTraN during the use case of severe weather detection involves rotation between using pairs of severe weather types as ground truth and evidence source. For instance, tornado data instances and an equal portion of non-severe data instances (after under-sampling) are selected as the primary data. In other words, the primary task is an anomaly detection task of predicting tornado weather instances from non-severe weather instances. At the same time, data instances from another severe weather type, e.g., windstorm are selected as evidence source.

Table 6.3, contains the results of the above evaluation process. It includes the precision, recall and F1-score measured for the anomalous class. During evaluation with all possible pairs of ground-truth and evidence, EviTraN effectively increases the detection of individual severe weather types. F1-score, that considers both recall and precision, increases after the introduction of evidence. Qualitative evaluation in Figure 6.3, suggests that EviTraN is able to effectively manipulate the latent space into distinct groups, that are linearly separable.

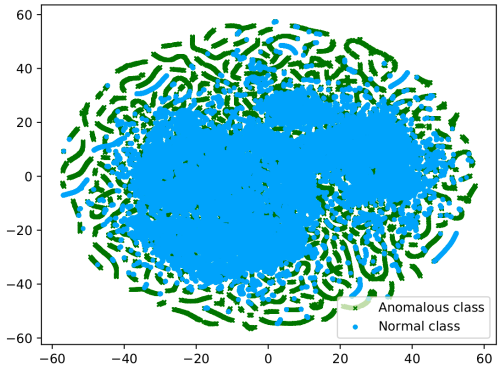
The evaluation individual severe weather detection does not involve more than one evidence sources. Due to severe under-sampling, the amount of involved data instances is low. By definition, instances from the anomalous class are low. Non-severe instances are undersampled two times in order to match the amount of each respective evidence source. The combination of low samples, as well as contradicting evidence sources makes the evaluation of additional sources less promising.



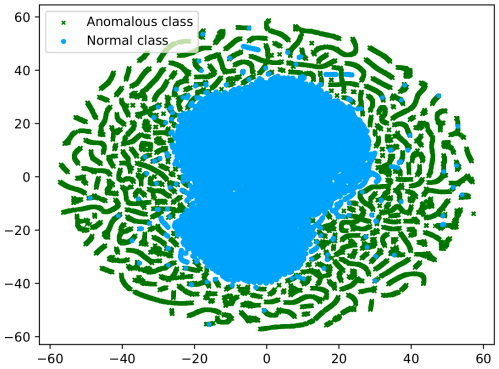
(a) SMOTE: Initialisation



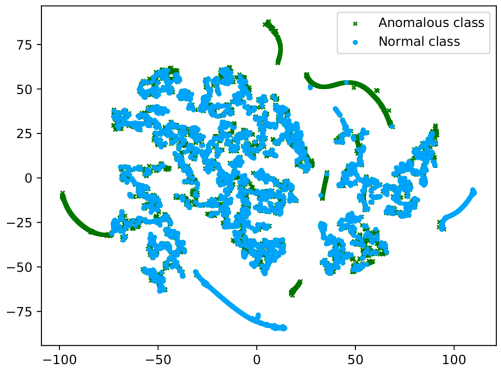
(b) SMOTE: EviTraN



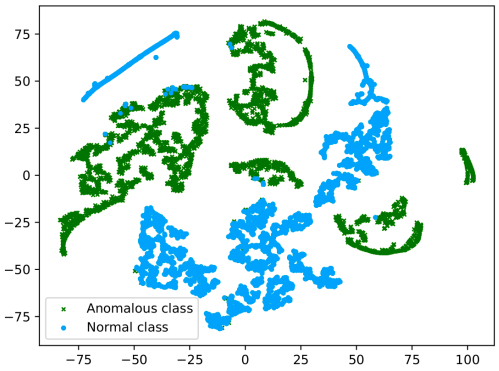
(c) SMOTEENN: Initialisation



(d) SMOTEENN: EviTraN

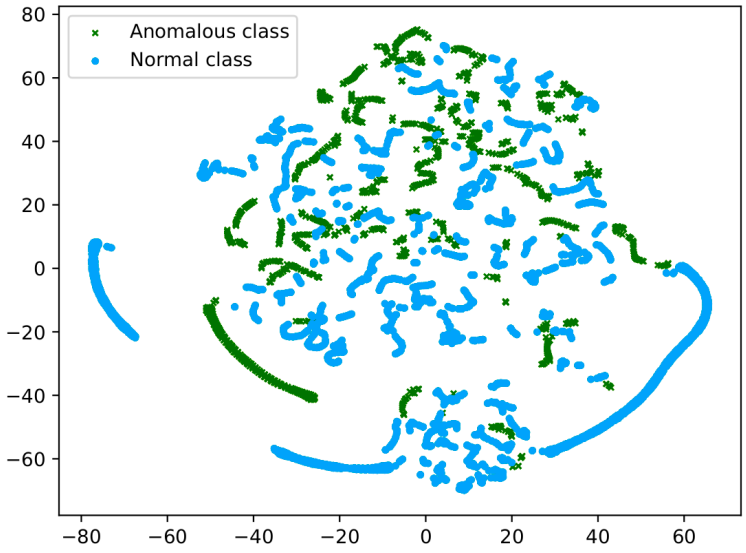


(e) Undersample: Initialisation

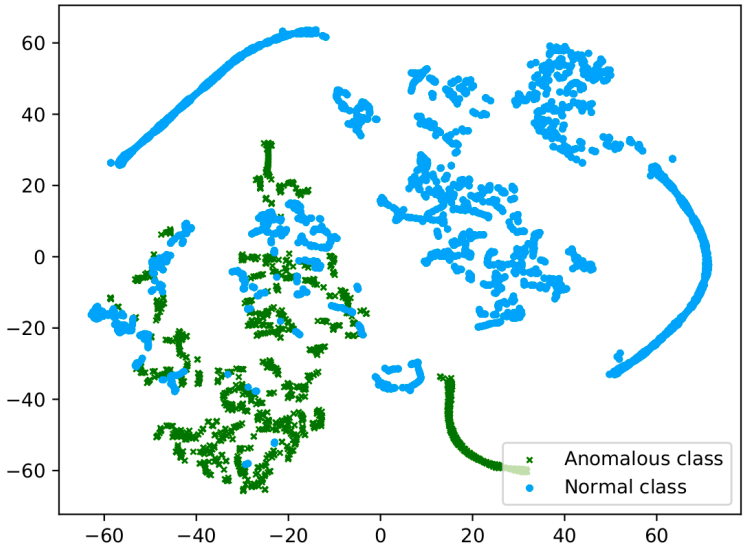


(f) Undersample: EviTraN

Figure 6.2: Qualitative evaluation of class balancing strategies. Left column depicts the state of latent space during initialisation, while the right column depicts the state of latent space after introduction of evidence with EviTraN.



(a) Initial latent space of pair: Flood-Windstorm



(b) Latent space of Flood after introduction of Windstorm evidence

Figure 6.3: State of latent space for evaluation pair with ground truth: Flood and evidence source: Windstorm. Top figure depicts the state of latent space during initialisation step. The bottom figure depicts the state of the latent space after introducing Windstorm evidence source. Introduction of Windstorm evidence allows for better distinction between normal and anomalous classes.

Table 6.3: Quantitative results of individual severe weather detection task with the use of EviTraN.

(a) Ground Truth: Flood

Metric	Baseline		Evidence Transfer	
	Windstorm	Tornado	Windstorm	Tornado
Precision	0.49	0.61	0.68 (+0.19)	0.72 (+0.11)
Recall	0.50	0.57	0.92 (+0.42)	0.69 (+0.12)
F1-Score	0.49	0.59	0.78 (+0.29)	0.71 (+0.12)

(b) Ground Truth: Tornado

Metric	Baseline		Evidence Transfer	
	Windstorm	Flood	Windstorm	Flood
Precision	0.26	0.24	0.32 (+0.06)	0.28 (+0.04)
Recall	0.62	1.00	0.98 (+0.36)	0.69 (-0.31)
F1-Score	0.36	0.38	0.49 (+0.13)	0.40 (+0.02)

(c) Ground Truth: Windstorm

Metric	Baseline		Evidence Transfer	
	Flood	Tornado	Flood	Tornado
Precision	0.49	0.61	0.84 (+0.23)	0.79 (+0.13)
Recall	0.50	0.57	0.74 (+0.03)	1.00 (+0.13)
F1-Score	0.49	0.75	0.79 (+0.13)	0.88 (+0.13)

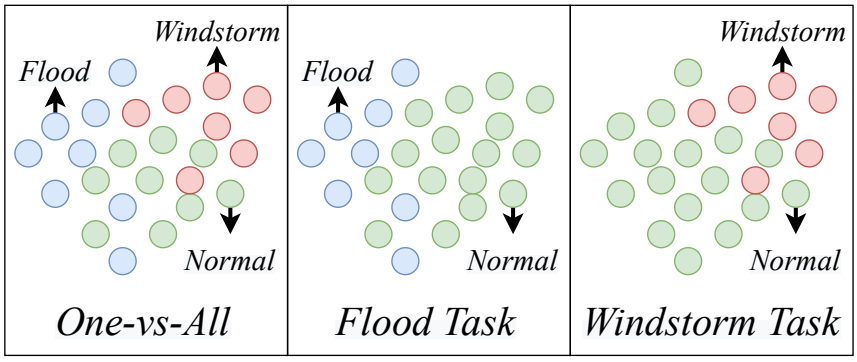


Figure 6.4: Visualisation of conflicting perspectives between pairs of ground-truth and evidence source tasks. One-vs-All depicts the groupings of the dataset during initial data collection.

The contradiction comes from the fact that only two classes are available for each auxiliary task. For instance, let tornado outbreaks be the current ground truth and windstorm be the evidence source. Then, the dataset consists of normal, tornado

and windstorm data instances. From the perspective of the ground truth, normal and windstorm instances are considered as non-severe, since only instances depicting tornadoes are severe. On the other hand, from the perspective of windstorm: normal and tornadoes are considered as normal. Which is the inverse case. Therefore, for a portion data instances, ground truth and evidence source contradict each other. Introducing additional evidence sources would further increase the contradictions between data instances. The problem of conflicting evidence sources is depicted in Figure 6.4.

The next chapter includes the conclusions of the thesis, as well as, discussion of future directions.

Chapter 7

Conclusions and Future Work

This thesis includes the investigation of the following hypothesis: "external data evidence improves deep representation learning". As the above hypothesis indicates the application of deep representation learning for data/information fusion, inspecting the relevant literature suggested certain limitations in previous approaches that mainly affected the inference of the suggested network. Examples of these limitations are requiring all data sources to be available during inference or dealing with incomplete or non-complementary datasets. To this end, it includes the proposal of evidence transfer (EviTraN), which a versatile, robust and effective deep representation learning fusion scheme.

Being versatile, robust and effective is an outcome of setting a set of evaluation criteria for deep representation learning fusion. The three proposed criteria, namely: effectiveness, robustness and modularity, cause EviTraN to be effective when presented with external categorical variables (external evidence), that represents meaningful relations to the primary dataset or task outcomes. At the same time, EviTraN is robust against disturbance introduced through low quality external evidence, such as random values, non-corresponding evidence or ill-intended evidence. The modularity criterion, enables EviTraN to overcome the above limitations, by being an iterative step that enriches initially learned weights with knowledge from auxiliary tasks.

From a data/information fusion perspective, EviTraN is a robust and effective fusion scheme that deals with the combination of diverse data sources, by involving task outcomes extracted on auxiliary datasets. The extraction of auxiliary task outcomes – external evidence, is an effortless process. Due to allowing arbitrary levels of supervision, such as weak or strong labels, it lends itself as a versatile fusion scheme that learns from heterogeneous data sources. Evidence transfer performs automatic correlation/association/alignment of data sources by introducing their task outcomes in the unsupervised representation learning process. The production of intermediate

features augmented from auxiliary data sources, allows multi-view perspective of the task at hand. This multi-view perspective could either be an outcome based on the primary dataset or other auxiliary datasets.

Due to its neural network architecture, EviTraN allows for hybrid learning of primary dataset's representations. Thus, being able to involve unlabelled and labelled instances whenever these are available. Due to the knowledge transfer accomplished from the hybrid modelling, it deals with limitations found in related work regarding the availability of data sources during inference. At the same time, its modular learning allows for iterative tinkering of representations in order to improve the quality of representations, according to new-found evidence. In addition, the learning framework allows the introduction of multiple task outcomes in a robust and effective manner. Due to only involving task outcomes, which are typically characterised by a low amount of features, EviTraN is scalable with multiple data sources.

The experimental evaluation of EviTraN, includes four datasets with artificially generated auxiliary evidence sources. It also includes learning in various settings, such as hybrid, inaccurate, uniformly incomplete and class biased incomplete learning. Furthermore, it includes three types of evidence sources: meaningful/real, random values/white noise and non-corresponding/random-index in three different quantities: use of single, double and triple evidence sources. Despite the artificial generated auxiliary evidence sources, the evalution also includes a realistic use case scenario. This scenario suggests the learning of improved representations for the task of unsupervised severe weather detection, through introduction of auxiliary binary task outcomes extracted from Wikipedia pages.

Through comparison with the information bottleneck method, as well as, the use of metrics from the domain of feature selection, a theoretical interpretation of the effects of EviTraN, on the latent space of the autoencoder is provided. The empirical analysis indicates that EviTraN increases the relevance between latent features and ground truth labels. The use of autoencoders enables the compression of raw observations into a smaller dimensional space. These two properties are very similar to information bottleneck. The comparison of EviTraN to the Information Bottleneck method not only provides insight regarding its inner workings, but also can be utilised for the purposes of adjusting the method for other domains or applications.

7.1 Future Work

Deep representation learning for fusion, requires multiple data sources for the purpose of the evaluation. It is most often motivated by a realistic application and therefore, finding data sources to include during evaluation is effortless. However, the

generation of new methods and adjustment of hyperparameters or training objectives, requires ready to use datasets, as in being normalised, adequate amount of samples, labels, etc., for rapid experimentation. Despite the evaluation of EviTraN involving such artificial data sources extracted from well-received datasets within the deep learning community, a complete testbed that would allow the timely evaluation of new methods is an interesting future work. Such testbed, would standardise the evaluation of deep representation learning for fusion independent of application, which would allow comparison by including similar data sources and evaluation metrics.

The generative and discriminative hybrid learning framework has proven to be robust and effective and therefore an appropriate choice for EviTraN. An interesting future direction would be the investigation of other hybrid learning methods, such as adversarial learning introduced in hybrid models that involve GANs. Discriminating between appropriate latent features, that will increase the relevance of the latent space, can be an intuitive way of transferring evidence. At the same time, adversarial training can be used before EviTraN, in order to conclude whether an evidence source is relevant or not.

Human in the loop. Human in the loop is also known as interactive machine learning. Which is a machine learning setting that involves querying a domain expert, in order to produce labels for a portion of the data samples. However, since EviTraN is able to utilise both weak and strong labels, it does not restrict such future direction to domain experts. Anyone, even users could provide feedback or indication of relation between data samples. Most likely, users not familiar with the intricate properties of the dataset would provide weak supervision, while domain experts would provide strong supervision. This interactive machine learning framework, would allow for explainable and direct hybrid learning of latent representations, as supervision would be extracted from decisions made by domain experts, policy makers and others. In any case, building appropriate tools capable of allowing humans to interact with the unsupervised learning process of EviTraN, would be an interesting future direction.

xAI. EviTraN is able to automatically associate/correlate/align data sources within a lower-dimensional latent and compact space. However, in the current version of EviTraN one is not able to explicit understand what is the relation between the data sources. One could only derive whether the task outcomes are relevant for the primary task or not. Incorporating explicit measures towards extracting the explicit relation between data sources, such as explicit training objectives, information theoretic metrics or use of semantic data, could aid the explicit discovery of relations across data sources. Being able to discover such relations, could potentially raise the effectiveness of the method. At the same time, being able to understand the encoding

of such relations within the neural network is an important stepping stone towards explainable neural networks or explainable artificial intelligence.

Negative Transfer Learning. To the extent which is capable, multiple versions of malicious, non-complementary, irrelevant or ill-intended evidence sources, are included in the experimental evaluation of EviTraN. In the domain of transfer learning, including such domains or tasks is called negative transfer learning. However, these relations are broad and are often application-specific. To this end, clear definition of these relations, categorisation and cataloguing is essential. Through the above actions, negative transfer learning would be standardised, and it would allow for appropriate selection of data sources during evaluation. This selection, would reduce current techniques of extracting data sources for evaluation in negative transfer learning, which require manual work or are an outcome of trial and error procedures.

Bibliography

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[2] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86 (11):2278–2324, 1998.

[3] Sinno J. Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[4] Risto Miikkulainen Elliot Meyerson. Beyond shared hierarchies: Deep multitask learning through soft layer ordering. *arXiv preprint arXiv:1711.00108*, 2018.

[5] Tong Meng, Xuyang Jing, Zheng Yan, and Witold Pedrycz. A survey on machine learning for data fusion. *Information Fusion*, 57:115 – 129, 2020.

[6] Xiaolong Dai and Siamak Khorram. Data fusion using artificial neural networks: A case study on multitemporal change analysis. *Computers, Environment and Urban Systems*, 23:19–31, 1999.

[7] Ren C. Luo, Chih-Chen Yih, and Kuo Lan Su. Multisensor fusion and integration: approaches, applications, and future research directions. *IEEE Sensors Journal*, 2(2):107–119, 2002.

[8] Sarath Chandar, Mitesh M. Khapra, Hugo Larochelle, and Balaraman Ravindran. Correlational neural networks. *Neural Computation*, 28(2):257–285, 2016.

[9] Natasha Jaques, Sara Taylor, Akane Sano, and Rosalind Picard. Multimodal autoencoder: A deep learning approach to filling in missing sensor data and enabling better mood prediction. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 202–208, 2017.

[10] Iraklis A. Klampanos, Athanasios Davvetas, Spyros Andronopoulos, Charalambos Pappas, Andreas Ikonomopoulos, and Vangelis Karkaletsis. Autoencoder-

Driven Weather Clustering for Source Estimation during Nuclear Events. *Environmental Modelling & Software*, 102:84–93, April 2018. ISSN 1364-8152.

[11] Yucan Zhou, Qinghua Hu, Jie Liu, and Yuan Jia. Combining heterogeneous deep neural networks with conditional random fields for chinese dialogue act recognition. *Neurocomputing*, 168:408 – 417, 2015.

[12] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, page 568–576, Cambridge, MA, USA, 2014. MIT Press.

[13] Petar Velickovic, Duo Wang, Nicholas D. Laney, and Pietro Lio. X-cnn: Cross-modal convolutional neural networks for sparse datasets. 2017.

[14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[15] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2014.

[16] Durk P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 3581–3589. Curran Associates, Inc., 2014.

[17] Weidi Xu, Haoze Sun, Chao Deng, and Ying Tan. Variational autoencoder for semi-supervised text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017.

[18] D. Zhang, S. Li, Q. Zhu, and G. Zhou. Multi-modal sentiment classification with independent and interactive knowledge via semi-supervised learning. *IEEE Access*, 8:22945–22954, 2020.

[19] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680. Curran Associates, Inc., 2014.

[20] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 2234–2242, Red Hook, NY, USA, 2016. Curran Associates Inc.

[21] Jost Tobias Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2016.

[22] Kumar Sricharan, Raja Bala, Matthew Shreve, Hui Ding, Kumar Saketh, and Jin Sun. Semi-supervised conditional gans. *arXiv preprint arXiv:1708.05789*, 2017.

[23] Wing W.Y. Ng, Guangjun Zeng, Jiangjun Zhang, Daniel S. Yeung, and Witold Pedrycz. Dual autoencoders features for imbalance classification problem. *Pattern Recognition*, 60:875 – 889, 2016.

[24] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 478–487. JMLR.org, 2016.

[25] Yu-Jhe Li, Fu-En Yang, Yen-Cheng Liu, Yu-Ying Yeh, Xiaofei Du, and Yu-Chiang Frank Wang. Adaptation and re-identification network: An unsupervised deep transfer learning approach to person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[26] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*, NIPS'06, page 153–160, Cambridge, MA, USA, 2006. MIT Press.

[27] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[28] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *ICLR Workshop*, 2013.

[29] Kenneth Cukier and Viktor Mayer-Schoenberger. *The rise of big data: How it's changing the way we think about the world*. Princeton University Press, 2014.

[30] Swati Sharma. Rise of big data and related issues. In *2015 Annual IEEE India Conference (INDICON)*, pages 1–6, 2015.

[31] The data age is here. are you ready? https://www.forbes.com/sites/splunk/2020/11/04/the-data-age-is-here-are-you-ready/, 2020. Accessed: 04-08-2021.

[32] Ross Towe, Graham Dean, Liz Edwards, Vatsala Nundloll, Gordon Blair, Rob Lamb, Barry Hankin, and Susan Manson. Rethinking data-driven decision support in flood risk management for a big data age. *Journal of Flood Risk Management*, 13(4):e12652, 2020.

[33] Hailong Li, Zhendong Wu, and Jianwu Zhang. Pedestrian detection based on deep learning model. In *2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 796–800, 2016.

[34] Babak Alipanahi, Andrew Delong, Matthew T. Weirauch, and Brendan J. Frey. Predicting the sequence specificities of dna- and rna-binding proteins by deep learning. *Nature Biotechnology*, 33(8):831–838, Aug 2015.

[35] Lei Zhou, Chu Zhang, Fei Liu, Zhengjun Qiu, and Yong He. Application of deep learning in food: A review. *Comprehensive Reviews in Food Science and Food Safety*, 18(6):1793–1811, 2019.

[36] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[37] Cassio Pennachin and Ben Goertzel. Contemporary approaches to artificial general intelligence. In *Artificial general intelligence*, pages 1–30. Springer, 2007.

[38] David G Stork. Scientist on the set: An interview with marvin minsky. *HAL's Legacy: 2001's Computer as Dream and Reality*, pages 15–31, 1997.

[39] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[40] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

[41] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938–e00938, Nov 2018.

[42] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[43] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[44] Xifeng Guo, Xinwang Liu, En Zhu, and Jianping Yin. Deep clustering with convolutional autoencoders. In Derong Liu, Shengli Xie, Yuanqing Li, Dongbin Zhao, and El-Sayed M. El-Alfy, editors, *Neural Information Processing*, pages 373–382, Cham, 2017. Springer International Publishing.

[45] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 843–852. JMLR.org, 2015.

[46] Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *J. Mach. Learn. Res.*, 15(1):3563–3593, January 2014. ISSN 1532-4435.

[47] Marc'aurelio Ranzato, Christopher Poultney, Sumit Chopra, and Yann Cun. Efficient learning of sparse representations with an energy-based model. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2007.

[48] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 833–840, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

[49] Ian T. Jolliffe. *Principal Component Analysis and Factor Analysis*, pages 115–128. Springer New York, New York, NY, 1986.

[50] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[51] Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In *NIPS*, pages 899–907, 2013.

[52] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015.

[53] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, page 841–848. MIT Press, 2001.

[54] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5, 08 2017.

[55] lexico.com. Definition of information theory.oxford university press. https://www.lexico.com/definition/information_theory, 2021.

[56] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

[57] Franklin E. White. Data fusion lexicon. Joint Directors of Labatories, Technical Panel for C_3, Data Fusion Sub-Panel, 1991.

[58] Pek H. Foo and Gee-Wah Ng. High-level information fusion: An overview. *Journal of Advances in Information Fusion*, 8:33–72, 06 2013.

[59] Henrik Boström, Sten F. Andler, Marcus Brohede, Ronnie Johansson, Alexander Karlsson, Joeri van Laere, Lars Niklasson, Marie Nilsson, Anne Persson, and Tom Ziemke. *On the Definition of Information Fusion as a Field of Research*. 2007. QC 20180122.

[60] Alan N. Steinberg, Christopher L. Bowman, and Franklin E. White. Revisions to the JDL data fusion model. In Belur V. Dasarathy, editor, *Sensor Fusion: Architectures, Algorithms, and Applications III*, volume 3719, pages 430 – 441. International Society for Optics and Photonics, SPIE, 1999.

[61] Martin E. Liggins, Chee-Yee Chong, Ivan Kadar, Mark G. Alford, Vincent Vannicola, and Stelios Thomopoulos. Distributed fusion architectures and algorithms for target tracking. *Proceedings of the IEEE*, 85(1):95–107, 1997.

[62] Justin Chuang, Leonard J. Cimini, and Nelson Sollenberger. Chapter 13 - wideband wireless packet data access. In JERRY D. GIBSON, editor, *Multimedia Communications*, Communications, Networking and Multimedia, pages 221–246. Academic Press, San Diego, 2001.

[63] Bowen Pan, Rameswar Panda, Camilo Fosco, Chung-Ching Lin, Alex Andonian, Yue Meng, Kate Saenko, Aude Oliva, and Rogerio Feris. Va-red^2: Video adaptive redundancy reduction. *arXiv preprint arXiv:2102.07887, Accepted in ICLR 2021*, 2021.

[64] Koji Kamma and Toshikazu Wada. Reconstruction error aware pruning for accelerating neural networks. In *International Symposium on Visual Computing*, pages 59–72. Springer, 2019.

[65] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[66] Edgar Anderson. The species problem in iris. *Annals of the Missouri Botanical Garden*, 23(3):457–509, 1936.

[67] Jiyun Cui, Hu Han, Shiguang Shan, and Xilin Chen. Rgb-d face recognition: A comparative study of representative fusion schemes. In *Biometric Recognition*, pages 358–366, Cham, 2018. Springer International Publishing.

[68] Jaewoo Kang and Jeffrey F. Naughton. On schema matching with opaque column names and data values. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, SIGMOD '03, page 205–216, New York, NY, USA, 2003. Association for Computing Machinery.

[69] Anastasios Zafeiropoulos, Nikolaos Konstantinou, Stamatios Arkoulis, Dimitrios-Emmanuel Spanos, and Nikolas Mitrou. A semantic-based architecture for sensor data fusion. In *2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pages 116–121, 2008.

[70] Denis Savenkov and Eugene Agichtein. EviNets: Neural networks for combining evidence signals for factoid question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 299–304, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[71] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52, 2010.

[72] Marta Amorim, Frederico D. Bortoloti, Patrick M. Ciarelli, Evandro O. T. Salles, and Daniel C. Cavalieri. Novelty detection in social media by fusing text and image into a single structure. *IEEE Access*, 7:132786–132802, 2019.

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

[74] Michele Volpi and Devis Tuia. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55:881–893, 2 2017.

[75] Zewei Xu, Kaiyu Guan, Nathan Casler, Bin Peng, and Shaowen Wang. A 3d convolutional neural network method for land cover classification using lidar and multi-temporal landsat imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 144:423–434, 10 2018.

[76] Du Yun-Mei, Allam Maalla, Liang Hui-Ying, Huang Shuai, Liu Dong, Lu Long, and Liu Hongsheng. The abnormal detection of electroencephalogram with three-dimensional deep convolutional neural networks. *IEEE Access*, 8:64646–64652, 2020.

[77] Eunbyung Park, Xufeng Han, Tamara L. Berg, and Alexander C. Berg. Combining multiple sources of knowledge in deep cnns for action recognition. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8, 2016.

[78] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 2048–2057. JMLR.org, 2015.

[79] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

[80] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 2095–2102. AAAI Press, 2018.

[81] Alireza Makhzani and Brendan J Frey. Pixelgan autoencoders. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 1975–1985. Curran Associates, Inc., 2017.

[82] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. volume 70 of *Proceedings of Machine Learning Research*, pages 1558–1567, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[83] Jiajie Peng, Hansheng Xue, Zhongyu Wei, Idil Tuncali, Jianye Hao, and Xuequn Shang. Integrating multi-network topology for gene function prediction using deep neural networks. *Briefings in Bioinformatics*, 4 2020.

[84] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5147–5156, 2016.

[85] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5880–5888, 2017.

[86] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, NIPS'93, page 737–744, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[87] Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[88] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France, 07–09 Jul 2015. PMLR.

[89] Kaisar Kushibar, Sergi Valverde, Sandra González-Villà, Jose Bernal, Mariano Cabezas, Arnau Oliver, and Xavier Lladó. Supervised domain adaptation for automatic sub-cortical brain structure segmentation with minimal user interaction. *Scientific Reports*, 9(1):6742, May 2019.

[90] Fayez W. Zaki, A.I. Abd el-Fattah, Yehia M. Enab, and Sayed H. el Konyaly. An ensemble average classifier for pattern recognition machines. *Pattern Recognition*, 21(4):327 – 332, 1988. ISSN 0031-3203.

[91] David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241 – 259, 1992. ISSN 0893-6080.

[92] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, Aug 1996. ISSN 1573-0565.

[93] Robert E. Schapire. A brief introduction to boosting. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'99, page 1401–1406, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[94] Wenbin Wu, Mugen Peng, Wenyun Chen, and Shi Yan. Unsupervised deep transfer learning for fault diagnosis in fog radio access networks. *IEEE Internet of Things Journal*, 7(9):8956–8966, 2020.

[95] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4293–4302, 2016.

[96] Dong-Jin Kim, Jinsoo Choi, Tae-Hyun Oh, Youngjin Yoon, and In S. Kweon. Disjoint multi-task learning between heterogeneous human-centric tasks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1699–1708, 2018.

[97] Roberto Cipolla, Yarin Gal, and Alex Kendall. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018.

[98] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5580–5590, Red Hook, NY, USA, 2017. Curran Associates Inc.

[99] Yi-Chen Chen, Sung-Feng Huang, Chia-Hao Shen, Hung yi Lee, and Lin shan Lee. Phonetic-and-semantic embedding of spoken words with applications in spoken content retrieval. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 941–948, 2018.

[100] Xiao Song, Xu Zhao, Liangji Fang, Hanwen Hu, and Yizhou Yu. Edgestereo: An effective multi-task learning network for stereo matching and edge detection. *International Journal of Computer Vision*, 128:910–930, 4 2020.

[101] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[102] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S. Kankanhalli. Learning to learn from noisy labeled data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5046–5054, 2019.

[103] Anders Søgaard and Yoav Goldberg. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235, Berlin, Germany, August 2016. Association for Computational Linguistics.

[104] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1923–1933, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[105] Hakan Bilen and Andrea Vedaldi. Universal representations:the missing link between faces, text, planktons, and cat breeds. *arXiv preprint arXiv: 1701.07275*, 2017.

[106] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[107] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 630–645, Cham, 2016. Springer International Publishing.

[108] Michael Rosenstein, Zvika Marx, and Leslie Kaelbling. To transfer or not to transfer. In *The International Conference on Neural Information Processing Systems Workshop Inductive Transfer: 10 Years Later*, 2005.

[109] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning (ICML)*, 2019.

[110] Tirtharaj Dash, Sharad Chitlangia, Aditya Ahuja, and Ashwin Srinivasan. Incorporating domain knowledge into deep neural networks. *arXiv preprint arXiv: 2103.00180*, 2021.

[111] Solomon Kullback and Richard Arthur Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86, 1951.

[112] Ian Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgeting in gradientbased neural networks. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2014.

[113] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[114] Mufti Mahmud, Mohammed Shamim Kaiser, Thomas Martin McGinnity, and Amir Hussain. Deep learning in mining biological data. *Cognitive Computation*, 13(1):1–33, 2021.

[115] Karim Abbasi, Parvin Razzaghi, Antti Poso, Saber Ghanbari-Ara, and Ali Masoudi-Nejad. Deep learning in drug target interaction prediction: Current and future perspectives. *Current Medicinal Chemistry*, 28(11):2100–2113, 2021.

[116] Francesco Piccialli, Vittorio Di Somma, Fabio Giampaolo, Salvatore Cuomo, and Giancarlo Fortino. A survey on deep learning in medicine: Why, how and when? *Information Fusion*, 66:111–137, 2021.

[117] Mario Bunge. A general black box theory. *Philosophy of Science*, 30(4):346–358, 1963. ISSN 00318248, 1539767X.

[118] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[119] Jiahong Wu, He Zheng, Bo Zhao, Yixin Li, Baoming Yan, Rui Liang, Wenjia Wang, Shipei Zhou, Guosen Lin, Yanwei Fu, Yizhou Wang, and Yonggang Wang. Large-scale datasets for going deeper in image understanding. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1480–1485, 2019.

[120] Irina Pencheva, Marc Esteve, and Slava Jankin Mikhaylov. Big data and ai – a transformational shift for government: So, what next for research? *Public Policy and Administration*, 35(1):24–44, 2020.

[121] Jonathan Stuart Ward and Adam Barker. Undefined by data: a survey of big data definitions. *arXiv preprint arXiv:1309.5821*, 2013.

[122] Bill Gerhardt, Kate Griffin, and Roland Klemann. Unlocking value in the fragmented world of big data analytics. *Cisco Internet Business Solutions Group*, 7, 2012.

[123] Seref Sagiroglu and Duygu Sinanc. Big data: A review. In *2013 International Conference on Collaboration Technologies and Systems (CTS)*, pages 42–47, 2013.

[124] Michael Schroeck, Rebecca Shockley, Janet Smart, Dolores Romero-Morales, and Peter Tufano. Analytics: The real-world use of big data. *IBM Global Business Services*, 12(2012):1–20, 2012.

[125] Richard K. Lomotey and Ralph Deters. Towards knowledge discovery in big data. In *2014 IEEE 8th International Symposium on Service Oriented System Engineering*, pages 181–191, 2014.

[126] Xiangxin Zhu, Carl Vondrick, Charless C. Fowlkes, and Deva Ramanan. Do we need more training data? *International Journal of Computer Vision*, 119(1): 76–92, Aug 2016. ISSN 1573-1405.

[127] Muhammad Aurangzeb Ahmad, Carly Eckert, and Ankur Teredesai. Interpretable machine learning in healthcare. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, BCB '18, page 559–560, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450357944.

[128] Sana Tonekaboni, Shalmali Joshi, Melissa D. McCradden, and Anna Goldenberg. What clinicians want: Contextualizing explainable machine learning for clinical end use. In Finale Doshi-Velez, Jim Fackler, Ken Jung, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 4th Machine Learning for Healthcare Conference*, volume 106 of *Proceedings of Machine Learning Research*, pages 359–380, Ann Arbor, Michigan, 09–10 Aug 2019. PMLR.

[129] Krishna Gade, Sahin Cem Geyik, Krishnaram Kenthapadi, Varun Mithal, and Ankur Taly. Explainable ai in industry. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 3203–3204, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016.

[130] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *ITU Journal: ICT Discoveries*, 1:39–48, 2018.

[131] Zachary C. Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3): 31–57, June 2018.

[132] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[133] Kazufumi Ito and Karl Kunisch. *Lagrange Multiplier Approach to Variational Problems and Applications*. Society for Industrial and Applied Mathematics, USA, 2008. ISBN 0898716497.

[134] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[135] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015.

[136] Chun Chet Tan and Chikkannan Eswaran. Using autoencoders for mammogram compression. *Journal of Medical Systems*, 35(1):49–58, Feb 2011.

[137] Davide Del Testa and Michele Rossi. Lightweight lossy compression of biometric patterns via denoising autoencoders. *IEEE Signal Processing Letters*, 22(12): 2304–2308, 2015.

[138] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Performance comparison of convolutional autoencoders, generative adversarial networks and super-resolution for image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2613–2616, 2018.

[139] Amirhossein Habibian, Ties van Rozendaal, Jakub M Tomczak, and Taco S Cohen. Video compression with rate-distortion autoencoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7033–7042, 2019.

[140] Vanessa Böhm and Uroš Seljak. Probabilistic auto-encoder. *arXiv preprint arXiv: 2006.05479*, 2020.

[141] Ken Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995.

[142] Tom M. Mitchell et al. Machine learning. 1997.

[143] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, December 2004.

[144] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, Jan 1972.

[145] Stephen Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of Documentation*, 60(5):503–520, Jan 2004.

[146] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Scoring, term weighting, and the vector space model*, page 100–123. Cambridge University Press, 2008. doi: 10.1017/CBO9780511809071.007.

[147] Anand Rajaraman and Jeffrey David Ullman. *Data Mining*, page 1–17. Cambridge University Press, 2011.

[148] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *The International Conference on Learning Representations (ICLR)*, 2015.

[149] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, December 1992.

[150] 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*, pages 248–255, 2009.

[151] Dimitris Spathis, Nikolaos Passalis, and Anastasios Tefas. Fast, visual and interactive semi-supervised dimensionality reduction. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[152] Dimitris Spathis, Nikolaos Passalis, and Anastasios Tefas. Interactive dimensionality reduction using similarity projections. *Knowledge-Based Systems*, 165: 77–91, 2019.

[153] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[154] HungarianAlgorithm.com. https://hungarianalgorithm.com, 2013.

[155] Jaakko Astola and Ilkka Virtanen. *Entropy correlation coefficient, a measure of statistical dependence for categorized data*. Lappeenrannan teknillinen korkeakoulu, 1981.

[156] Frederik Maes, Andre Collignon, Dirk Vandermeulen, Guy Marchal, and Paul Suetens. Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16(2):187–198, April 1997.

[157] Colin Studholme, Derek L.G. Hill, and David J. Hawkes. An overlap invariant entropy measure of 3d medical image alignment. *Pattern Recognition*, 32(1): 71–86, 1999.

[158] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

[159] Peter Latham and Yasser Roudi. Mutual information. *Scholarpedia*, 4(1): 1658, 2009. doi: 10.4249/scholarpedia.1658. URL https://doi.org/10.4249/scholarpedia.1658.

[160] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, December 1985.

[161] Tadeusz Calinski and Joachim Harabasz. A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, 3(1):1–27, 1974.

[162] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[163] Kamran G. Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5747–5756, 2017.

[164] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1965–1972, 2017.

[165] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[166] Geoffrey Hinton and Sam T. Roweis. Stochastic neighbor embedding. In *NIPS*, volume 15, pages 833–840, 2002.

[167] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[168] Evelyn Fix and J. L. Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3):238–247, 1989.

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

[170] Brian C Ross. Mutual information between discrete and continuous data sets. *PloS one*, 9(2):e87357, 2014.

[171] Chris Ding and Hanchuan Peng. Minimum Redundancy Feature Selection From Microarray Gene Expression Data. *Journal of Bioinformatics and Computational Biology*, 03(02):185–205, April 2005.

[172] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), July 2009.

[173] D. P. Dee, S. M. Uppala, A. J. Simmons, P. Berrisford, P. Poli, S. Kobayashi, U. Andrae, M. A. Balmaseda, G. Balsamo, P. Bauer, P. Bechtold, A. C. M. Beljaars, L. van de Berg, J. Bidlot, N. Bormann, C. Delsol, R. Dragani, M. Fuentes, A. J. Geer, L. Haimberger, S. B. Healy, H. Hersbach, E. V. Hólm, L. Isaksen, P. Kållberg, M. Köhler, M. Matricardi, A. P. McNally, B. M. Monge-Sanz, J.-J. Morcrette, B.-K. Park, C. Peubey, P. de Rosnay, C. Tavolato, J.-N. Thépaut, and F. Vitart. The era-interim reanalysis: configuration and performance of the data assimilation system. *Quarterly Journal of the Royal Meteorological Society*, 137(656):553–597, 2011.

[174] John Michalakes, Jimy Dudhia, D. Gill, Tom Henderson, J. Klemp, W. Skamarock, and Wei Wang. The weather reseach and forecast model: Software architecture and performance. 01 2004.

[175] Arthur C Pike. Geopotential heights and thicknesses as predictors of atlantic tropical cyclone motion and intensity. *Monthly weather review*, 113(6):931–940, 1985.

[176] Xiaogu Zheng and Carsten S Frederiksen. Statistical prediction of seasonal mean southern hemisphere 500-hpa geopotential heights. *Journal of climate*, 20(12):2791–2809, 2007.

[177] Arkadiusz M Tomczyk and Ewa Bednorz. Heat waves in central europe and tropospheric anomalies of temperature and geopotential heights. *International Journal of Climatology*, 39(11):4189–4205, 2019.

[178] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, June 2002. ISSN 1076-9757.

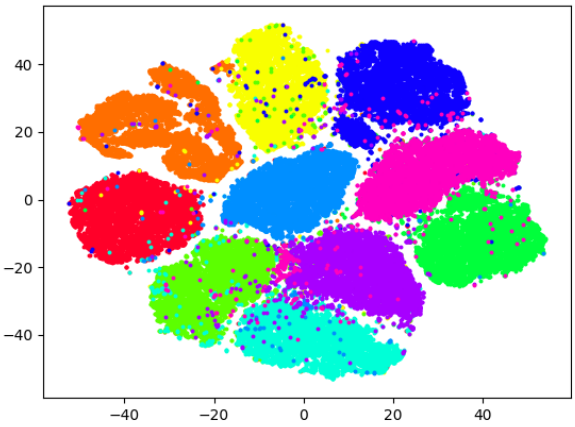
[179] Dennis L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-2(3):408–421, 1972.

[180] Gustavo EAPA Batista, Ronaldo C Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.

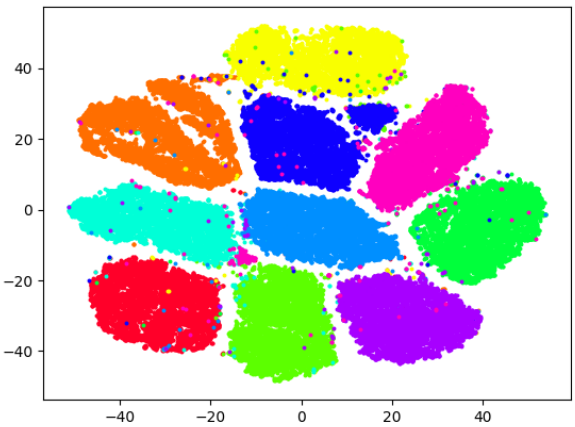
[181] Kai Ming Ting. *Precision and Recall*, pages 781–781. Springer US, Boston, MA, 2010.

Appendix A

This Appendix includes plots regarding the initial state and state of the latent space, after the introduction of evidence with EviTraN for MNIST and Reuters-100k. Unlike CIFAR-10 and 20newsgroups, MNIST and Reuters-100k do not have an initial latent space that resembles of a Gaussian distribution. Due to already having an initial structure, for example due to the use of Convolutional Autoencoder, the initial state of MNIST already highlights certain groups. Therefore, from a qualitative perspective the changes performed by EviTraN are very subtle, as shown in Figures 7.1 and 7.2

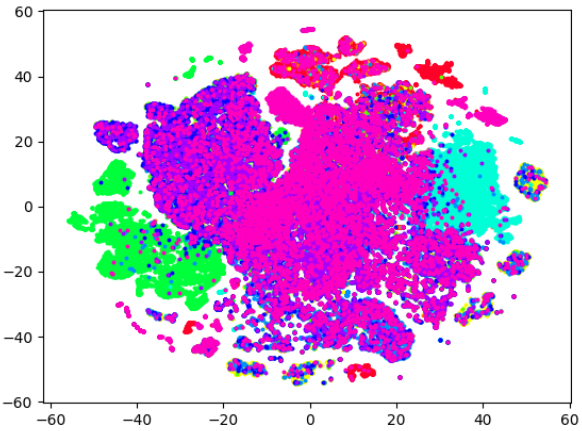


(a) Initial latent space of MNIST

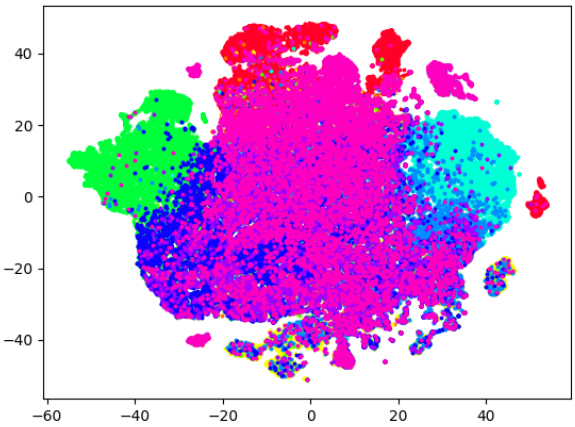


(b) Latent space of MNIST after M4

Figure 7.1: State of latent space before (top figure) and after the introduction of external evidence sources (bottom figure), for MNIST.



(a) Initial latent space of Reuters-100k



(b) Latent space of Reuters-100k after M4

Figure 7.2: State of latent space before (top figure) and after the introduction of external evidence sources (bottom figure), for Reuters-100k.