
How can humans leverage machine learning?
From Medical Data Wrangling to
Learning to Defer to Multiple Experts

AUTHOR:

Daniel Barrejón Moreno

A thesis submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy in

Multimedia and Communications

Universidad Carlos III de Madrid

Advisors:

Pablo Martínez Olmos

Antonio Artés Rodríguez

July 2023

This thesis is distributed under the *Creative Commons* license
Attribution - Non-commercial - No Derivative Works.



*Lo que dejo por escrito,
no está tallado en granito.
Yo apenas suelto en el viento,
presentimientos.
Pido lo que necesito.
Tinta y tiempo, tinta y tiempo.
Tinta y tiempo, tinta y tiempo.*

Tinta y tiempo - *Jorge Drexler* ▶

Acknowledgements

La verdad es que no sé cómo empezar. Parecía fácil, pero agradecer a todos los que te acompañan en un camino (*emocionalmente*) tan duro no lo es, os lo aseguro. La verdad es que envidio un poco, o admiro, a aquellos que acaban con muchísimas ganas e ilusión, porque lo que más me apetece a mí ahora mismo es descansar y tirarme tres meses tocando la guitarra. Quizás eso es lo que haga. No lo sé. Pero bueno, aquí van esos agradecimientos.

En primer lugar me gustaría agradecer a mis directores de tesis, Antonio y Pablo, por brindarme la oportunidad de hacer esta tesis y por guiarme en este camino. Creo que el eco de la risa de Pablo por el pasillo me va a resonar durante bastante años más en mi cabeza. Y a parte, me voy sin saber a cuánto volumen se puede poner el equipo de sonido del despacho de Antonio. No obstante, sí que me voy con muchos aprendizajes que me han brindado durante estos años. Gracias por confiar en mí, y hacerme ver *the bright side of investigation* cuando yo no podía.

Continúo mis agradecimientos con todos aquellos que me han acompañado estos años en los *cubículos* del lab. Tanto a los que siguen por Madrid, como a los que echan de menos Madrid. Gracias por hacer de la investigación algo más ameno, por corroborar que también hay gente en este mundillo que le gusta salir a tomar unas cervezas y por compartir en las comidas la indignación por la precariedad laboral de los doctorandos en España. A todos os deseo un futuro lleno de felicidad y, seguro que después de este camino, además, con más paz.

I would also like to express my gratitude to all the people I met in Amsterdam. First of all, I am deeply thankful to Eric Nalisnick for giving me the chance to do a research visit in Amsterdam and for his belief in my abilities, showing interest in my work and potential since the very beginning. I could not feel more welcome. Also thanks to all the lab members and other visiting researchers who made the stay more enjoyable. And of course, to all the wonderful people that I was able to meet outside UvA.

Tampoco puedo dejar de agradecer a todas las personas que me llevo de estos años en Madrid. En especial, los que me han acompañado desde los primeros años de resi: a los *chorbos* y a los *no chorbos*. Gracias por hacer durante todos estos años que Madrid sea más casa y por cederme parte de vuestros acentos y jergas que ahora definen mi forma de hablar (sobre todo a partir de media noche). Gracias a cada uno de vosotros.

Obviamente también me gustaría agradecer a todas las personas que me han apoyado desde Toledo. En primer lugar, a los que considero mis amigos de toda la vida. A los que empezasteis conmigo en la UNI y a día de hoy seguís a mi lado. Cada vez es más difícil que todos coincidamos, pero lo bonito de estas amistades es que, a pesar de todo, sabemos que siempre van a estar ahí. Gracias. Me gustaría agradecer también al resto de personas que he podido conocer en Toledo

durante estos años. Me habéis ayudado de manera sutil pero igualmente importante a desconectar del trabajo en pequeñas cápsulas de felicidad: en viajes, en casas rurales, en la Pozuela, en conciertos. Guardo todos esos momentos de felicidad como instantáneas en mi recuerdo.

Quien me conoce sabe que lo que me hace más feliz, lo que más me mueve y lo que más me emociona es la música. No sé que sería de mi sin la música. Probablemente nada. Lo que si tengo seguro es que la música ha sido, y será, mi principal ansiolítico. Por eso, agradezco profundamente y de todo corazón a cada uno de los que me han acompañado en lo musical. Primero, a Oplutón: Ruli, Chenchó y Miguel. No tengo palabras para describir lo mucho que me llevo de este proyecto musical. Todas las alegrías, chanzas, y algún que otro resquemor de los ensayos. Las noches de verano en la Pozu hablando y escuchando música con una tortilla y unas cervezas. La sensación de sinergia y conexión en los conciertos. El fin de semana del refugio en Gredos. La culminación de *Lar y Lumbre*. Todo esto me lo llevo para toda la vida. Gracias. Tampoco puedo olvidarme de mis padrinos en la música: Fran y Ritx. Desde que os conocí un verano mientras tocaba en una terraza de Toledo nunca habéis dudado de mi potencial en la música. Me adoptasteis en Dos Incautos y habéis permanecido a mi lado durante muchos años. Me habéis demostrado que la humildad va por encima de todo y que lo más importante en esto es disfrutar. De corazón, gracias.

Antes de pasar a mi familia, no puedo olvidarme de las dos personas que más me han ayudado durante estos años de doctorado. En primer lugar, a Ángela. Has demostrado ser una de esas personas talismán muy difíciles de encontrar en la vida. Me has ayudado y escuchado en todo momento y has estado cuando más lo necesitaba. Gracias de corazón. Y a quién no podré olvidar nunca de estos años es a Lorena. Desde los primeros momentos has demostrado ser la persona más empática y cercana del laboratorio. Siempre has estado cuando lo he necesitado, y solo los dos sabemos lo *jodido* que ha sido el doctorado para cada uno. Sin embargo, siempre hemos estado para apoyarnos el uno al otro, desde los cafés imbebibles de la cafetería hasta los paseos con Kala por Polvoranca. Sabes lo mucho que te admiro como persona (y como matemática) y solo espero que podamos disfrutar muchos años más.

Por último, gracias a mi familia. A mis padres por el amor incondicional que nunca ha faltado y del que siempre he podido contar en los momentos más difíciles. A mi hermano Lionel por el cariño y el apoyo desde que era pequeño. Y también a los más pequeños de la familia: a mini Lio y Olivia, que habéis llegado para iluminar más aún esta familia.

Y bueno, en verdad, más que un gracias, dar un abrazo sentido a todas las versiones de mi mismo que han dado color a estos años. Si de algo ha servido todo esto es para darme cuenta de una cosa. Y es que ahora, lo que hay que buscar, es la calma.

Published and presented contents

The following list of works is a short bibliography of journal articles, conference articles and workshop articles included as part of this thesis. In each contribution we explicitly specify whether the contribution is partially or fully present in the chapters of this thesis. Additionally, all publications are referred in the introductory paragraphs of the chapters of this thesis. I hereby declare that: *The material from this source included in this thesis is not singled out with typographic means and references.*

CONFERENCES & WORKSHOPS

1. Rajeev Verma*, **Daniel Barrejón***, and Eric Nalisnick. Learning to Defer to Multiple Experts: Consistent Surrogate Losses, Confidence Calibration, and Conformal Ensembles. *In International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023. [\[pdf\]](#) - This article is *fully* included in Chapter 5 and *partially* included in Chapter 4.
2. Rajeev Verma, **Daniel Barrejón**, and Eric Nalisnick. On the Calibration of Learning to Defer to Multiple Experts. *In Human-Machine Collaboration and Teaming Workshop (HMCaT) at ICML*, 2022 [\[pdf\]](#) - This article is *fully* included in Chapter 5 and *partially* included in Chapter 4.

JOURNAL

3. **Daniel Barrejón**, Pablo M. Olmos, and Antonio Artés-Rodríguez. Medical data wrangling with sequential variational autoencoders. *In IEEE Journal of Biomedical and Health Informatics (JBHI)*, 2021. 26(6), 2737-2745. [\[pdf\]](#) - This article is *fully* included in Chapter 3.

*Co-first authors, with equal contribution. Order determined by coin flip.

Abstract

The irruption of the smartphone into everyone's life and the ease with which we digitise or record any data supposed an explosion of quantities of data. Smartphones, equipped with advanced cameras and sensors, have empowered individuals to capture moments and contribute to the growing pool of data. This data-rich landscape holds great promise for research, decision-making, and personalized applications. By carefully analyzing and interpreting this wealth of information, valuable insights, patterns, and trends can be uncovered.

However, big data is worthless in a vacuum. Its potential value is unlocked only when leveraged to drive decision-making. In recent times we have been participants of the outburst of artificial intelligence: the development of computer systems and algorithms capable of perceiving, reasoning, learning, and problem-solving, emulating certain aspects of human cognitive abilities. Nevertheless, our focus tends to be limited, merely skimming the surface of the problem, while the reality is that the application of machine learning models to data introduces is usually fraught. More specifically, there are two crucial pitfalls frequently neglected in the field of machine learning: the quality of the data and the erroneous assumption that machine learning models operate autonomously. These two issues have established the foundation for the motivation driving this thesis, which strives to offer solutions to two major associated challenges: 1) dealing with irregular observations and 2) learning when and who should we trust.

The *first* challenge originates from our observation that the majority of machine learning research primarily concentrates on handling regular observations, neglecting a crucial technological obstacle encountered in practical big-data scenarios: the aggregation and curation of heterogeneous streams of information. Before applying machine learning algorithms, it is crucial to establish robust techniques for handling big data, as this specific aspect presents a notable bottleneck in the creation of robust algorithms. Data wrangling, which encompasses the extraction, integration, and cleaning processes necessary for data analysis, plays a crucial role in this regard. Therefore, the first objective of this thesis is to tackle the frequently disregarded challenge of addressing irregularities within the context of medical data. We will focus on three specific aspects. Firstly, we will tackle the issue of *missing* data by developing a framework that facilitates the imputation of missing data points using relevant information derived from alternative data sources or past observations. Secondly, we will move beyond the assumption of homogeneous observations, where only one statistical data type (such as Gaussian) is considered, and instead, work with *heterogeneous* observations. This means that different data sources can be represented by various statistical likelihoods, such as Gaussian, Bernoulli, categorical, etc. Lastly, considering the *temporal* enrichment of today's collected data and our focus on medical data, we will develop a

novel algorithm capable of capturing and propagating correlations among different data streams over time. All these three problems are addressed in our first contribution which involves the development of a novel method based on Deep Generative Models (DGM) using Variational Autoencoders (VAE). The proposed model, the Sequential Heterogeneous Incomplete VAE (Shi-VAE), enables the aggregation of multiple heterogeneous data streams in a modular manner, taking into consideration the presence of potential missing data. To demonstrate the feasibility of our approach, we present proof-of-concept results obtained from a real database generated through continuous passive monitoring of psychiatric patients.

Our *second* challenge relates to the misbelief that machine learning algorithms can perform independently. However, this notion that AI systems can solely account for automated decision-making, especially in critical domains such as healthcare, is far from reality. Our focus now shifts towards a specific scenario where the algorithm has the ability to make predictions independently or alternatively defer the responsibility to a human expert. The purpose of including the human is not to obtain just better performance, but also more reliable and trustworthy predictions we can rely on. In reality, however, important decisions are not made by one person but are usually committed by an ensemble of human experts. With this in mind, two important questions arise: 1) *When* should the human or the machine bear responsibility and 2) among the experts, *who* should we trust? To answer the first question, we will employ a recent theory known as *Learning to defer* (L2D). In L2D we are not only interested in abstaining from prediction but also in understanding the humans confidence for making such prediction. thus deferring only when the human is more likely to be correct. The second question about who to defer among a pool of experts has not been yet answered in the L2D literature, and this is what our contributions aim to provide. First, we extend the two yet proposed consistent surrogate losses in the L2D literature to the multiple-expert setting. Second, we study the frameworks ability to estimate the probability that a given expert correctly predicts and assess whether the two surrogate losses are confidence calibrated. Finally, we propose a conformal inference technique that chooses a subset of experts to query when the system defers. Ensembling experts based on confidence levels is vital to optimize human-machine collaboration.

In conclusion, this doctoral thesis has investigated two cases where humans can leverage the power of machine learning: first, as a tool to assist in data wrangling and data understanding problems and second, as a collaborative tool where decision-making can be automated by the machine or delegated to human experts, fostering more transparent and trustworthy solutions.

Resumen

La irrupción de los *smartphones* en la vida de todos y la facilidad con la que digitalizamos o registramos cualquier situación ha supuesto una explosión en la cantidad de datos. Los teléfonos, equipados con cámaras y sensores avanzados, han contribuido a que las personas puedan capturar más momentos, favoreciendo así el creciente conjunto de datos. Este panorama repleto de datos aporta un gran potencial de cara a la investigación, la toma de decisiones y las aplicaciones personalizadas. Mediante el análisis minucioso y una cuidada interpretación de esta abundante información, podemos descubrir valiosos patrones, tendencias y conclusiones

Sin embargo, este gran volumen de datos no tiene valor por si solo. Su potencial se desbloquea solo cuando se aprovecha para impulsar la toma de decisiones. En tiempos recientes, hemos sido testigos del auge de la inteligencia artificial: el desarrollo de sistemas informáticos y algoritmos capaces de percibir, razonar, aprender y resolver problemas, emulando ciertos aspectos de las capacidades cognitivas humanas. No obstante, solemos centrarnos solo en la superficie del problema mientras que la realidad es que la aplicación de modelos de aprendizaje automático a los datos presenta desafíos significativos. Concretamente, se suelen pasar por alto dos problemas cruciales en el campo del aprendizaje automático: la calidad de los datos y la suposición errónea de que los modelos de aprendizaje automático pueden funcionar de manera autónoma. Estos dos problemas han sido el fundamento de la motivación que impulsa esta tesis, que se esfuerza en ofrecer soluciones a dos desafíos importantes asociados: 1) lidiar con datos irregulares y 2) aprender cuándo y en quién debemos confiar.

El *primer* desafío surge de nuestra observación de que la mayoría de las investigaciones en aprendizaje automático se centran principalmente en manejar datos regulares, descuidando un obstáculo tecnológico crucial que se encuentra en escenarios prácticos con gran cantidad de datos: la agregación y el curado de secuencias heterogéneas. Antes de aplicar algoritmos de aprendizaje automático, es crucial establecer técnicas robustas para manejar estos datos, ya que esta problemática representa un cuello de botella claro en la creación de algoritmos robustos. El procesamiento de datos (en concreto, nos centraremos en el término inglés *data wrangling*), que abarca los procesos de extracción, integración y limpieza necesarios para el análisis de datos, desempeña un papel crucial en este sentido. Por lo tanto, el primer objetivo de esta tesis es abordar el desafío normalmente pasado por alto de tratar datos irregulares. Específicamente, bajo el contexto de datos médicos. Nos centraremos en tres aspectos principales. En primer lugar, abordaremos el problema de los datos *perdidos* mediante el desarrollo de un marco que facilite la imputación de estos datos perdidos utilizando información relevante obtenida de fuentes de datos de diferente naturaleza u observaciones pasadas. En segundo lugar, iremos más allá de

la suposición de lidiar con observaciones homogéneas, donde solo se considera un tipo de dato estadístico (como Gaussianos) y, en su lugar, trabajaremos con observaciones *heterogéneas*. Esto significa que diferentes fuentes de datos pueden estar representadas por diversas distribuciones de probabilidad, como Gaussianas, Bernoulli, categóricas, etc. Por último, teniendo en cuenta el enriquecimiento *temporal* de los datos hoy en día y nuestro enfoque directo sobre los datos médicos, propondremos un algoritmo innovador capaz de capturar y propagar la correlación entre diferentes flujos de datos a lo largo del tiempo. Todos estos tres problemas se abordan en nuestra primera contribución, que implica el desarrollo de un método basado en Modelos Generativos Profundos (Deep Generative Model en inglés) utilizando Autoencoders Variacionales (Variational Autoencoders en inglés). El modelo propuesto, Sequential Heterogeneous Incomplete VAE (Shi-VAE), permite la agregación de múltiples flujos de datos heterogéneos de manera modular, teniendo en cuenta la posible presencia de datos perdidos. Para demostrar la viabilidad de nuestro enfoque, presentamos resultados de prueba de concepto obtenidos de una base de datos real generada a través del monitoreo continuo pasivo de pacientes psiquiátricos.

Nuestro *segundo* desafío está relacionado con la creencia errónea de que los algoritmos de aprendizaje automático pueden funcionar de manera independiente. Sin embargo, esta idea de que los sistemas de inteligencia artificial pueden ser los únicos responsables en la toma de decisiones, especialmente en dominios críticos como la atención médica, está lejos de la realidad. Ahora, nuestro enfoque se centra en un escenario específico donde el algoritmo tiene la capacidad de realizar predicciones de manera independiente o, alternativamente, delegar la responsabilidad en un experto humano. La inclusión del ser humano no solo tiene como objetivo obtener un mejor rendimiento, sino también obtener predicciones más transparentes y seguras en las que podamos confiar. En la realidad, sin embargo, las decisiones importantes no las toma una sola persona, sino que generalmente son el resultado de la colaboración de un conjunto de expertos. Con esto en mente, surgen dos preguntas importantes: 1) ¿Cuándo debe asumir la responsabilidad el ser humano o cuándo la máquina? y 2) de entre los expertos, ¿en quién debemos confiar? Para responder a la primera pregunta, emplearemos una nueva teoría llamada *Learning to defer* (L2D). En L2D, no solo estamos interesados en abstenernos de hacer predicciones, sino también en comprender cómo de seguro estará el experto para hacer dichas predicciones, diferiendo solo cuando el humano sea más probable en predecir correctamente. La segunda pregunta sobre a quién deferir entre un conjunto de expertos aún no ha sido respondida en la literatura de L2D, y esto es precisamente lo que nuestras contribuciones pretenden proporcionar. En primer lugar, extendemos las dos primeras surrogate losses consistentes propuestas hasta ahora en la literatura de L2D al contexto de múltiples expertos. En segundo lugar, estudiamos la capacidad de estos modelos para estimar la probabilidad de que un experto dado haga predicciones correctas y evaluamos si estas surrogate losses están calibradas en términos de confianza. Finalmente, proponemos una técnica de conformal inference que elige un subconjunto de expertos para consultar cuando el sistema decide diferir. Esta combinación de expertos basada en los respectivos niveles de confianza es fundamental para optimizar la colaboración entre humanos y máquinas.

En conclusión, esta tesis doctoral ha investigado dos casos en los que los humanos pueden aprovechar el poder del aprendizaje automático: primero, como herramienta para ayudar en problemas de procesamiento y comprensión de datos y, segundo, como herramienta colaborativa en la que la toma de decisiones puede ser automatizada para ser realizada por la máquina o delegada a expertos humanos, fomentando soluciones más transparentes y seguras..

Contents

1	Introduction	1
1.1	All that glitters is not gold	2
1.1.1	Promises of Big Data	3
1.1.2	Pitfalls of Big Data	3
	Data quality is overlooked	4
	AI is not an independent agent	4
1.2	Research Challenges	4
1.2.1	Challenge I: Dealing with Irregular Observations	5
	Data Wrangling	5
	Irregular observations	5
1.2.2	Challenge II: When and Who Should We Trust?	6
	Who should bear responsibility?	7
	Learning to defer	7
	Who should we trust?	8
1.3	Thesis Organization	8
1.3.1	Part I: Medical Data Wrangling using VAEs	8
	Chapter 2: Handling irregular observations using VAEs	8
	Chapter 3: Medical Data Wrangling With Sequential Variational Autoencoders	9
1.3.2	Part II: Learning to Defer to Multiple Experts	9
	Chapter 4: Learning to Defer	9
	Chapter 5: Learning to Defer to Multiple Experts	9
	Chapter 6: Conclusions and Future Work	10
2	Handling Irregular Observations using VAEs	13
2.1	Our learning framework: Variational Autoencoders	14
2.1.1	Latent Variable Models	14
2.1.2	Variational Inference and ELBO	16
2.1.3	Variational Autoencoders	18
2.2	Handling Missing Data	21
2.2.1	Imputation and Deletion Techniques	21
2.2.2	Generative Models for Missing Data	22
2.2.3	HI-VAE: Our Basis for Missing data Handling	23

2.3	Handling Heterogeneous Data	24
2.3.1	The Challenge of Modeling Heterogeneous Distributions	24
2.3.2	VAEs for Heterogeneous Data	25
2.3.3	HI-VAE: Our Basis for Heterogeneous Data Handling	26
	1) Factorized Decoder	26
	2) Data Normalization	27
	3) Gaussian Mixture Prior	28
2.4	Handling Temporal Data	29
2.4.1	RNNs: Our Toolkit for Temporal Data	29
2.4.2	RNNs Handling Missing Data in Medical Context	31
2.4.3	VRNN: Our Basis for Temporal Data Handling	32
	Generative Model	32
	Inference Model	33
	Learning	33
2.5	Summary of the Chapter	33
3	Medical Data Wrangling With Sequential Variational Autoencoders	35
3.1	Introduction	36
3.2	A human monitoring database	38
3.3	Proposed Model	39
3.3.1	Notation	39
3.3.2	The Sequential Heterogeneous Incomplete VAE (Shi-VAE)	40
3.3.3	Heterogeneous Decoder	41
3.3.4	Model Training with Variational Inference	42
3.3.5	The GP-VAE Probabilistic Model	43
3.4	Experimental Results	44
3.4.1	Evaluation Metrics	45
3.4.2	Synthetic Data set	46
3.4.3	Physionet	48
3.4.4	Human Monitoring Database	49
3.5	Discussion	50
4	Learning to Defer	53
4.1	General Classification Problem	55
4.1.1	Binary Classification	57
4.1.2	Multiclass Classification: Cross-entropy loss	58
4.1.3	Multiclass-to-binary reduction: Code Based Surrogates	59
4.1.4	Can we abstain to predict? A motivating example towards L2D	60
4.2	Learning to Defer Background and Related Work	61
4.2.1	Learning to Defer within Rejection Learning	62

4.2.2	Learning to Defer in the Context of Human-Machine Collaboration	63
4.3	Learning to Defer	64
4.3.1	Preliminaries	65
4.3.2	Softmax Surrogate Loss: Single Expert	67
4.3.3	One-vs-All Surrogate Loss: Single Expert	68
4.3.4	Realizable-Surrogate Loss: Complement when deferring	68
4.3.5	Toy example for Learning to Defer surrogate losses	69
4.4	Confidence Calibration in Learning to Defer	70
4.4.1	Our Notion of Confidence Calibration	71
4.4.2	Softmax Parametrization: Single Expert	72
4.4.3	One-vs-All Parameterization: Single Expert	73
4.5	Summary of the Chapter	75
5	Learning to Defer to Multiple Experts	77
5.1	L2D To Multiple Experts	79
5.1.1	Softmax Surrogate Loss: Multiple Experts	80
5.1.2	One-vs-All Surrogate Loss: Multiple Expert	80
5.1.3	Toy example with Multiple Experts L2D	81
5.1.4	Inconsistency of Mixture of Experts	82
5.2	Confidence Calibration of Expert Confidence	82
5.2.1	Softmax Parameterization: Multi-Expert	83
5.2.2	One-vs-All Parameterization: Multi-Expert	84
5.3	Ensembling Expert with Conformal Inference	84
5.3.1	Conformal Inference	84
5.3.2	Conformal Inference on Sets of Experts	85
5.3.3	Choice of Hyperparameters for Regularized Conformal Ensembles	86
5.4	Related Work	87
5.5	Experiments	88
5.5.1	Overall System Accuracy	88
5.5.2	Confidence Calibration	90
5.5.3	Conformal Ensembles	92
5.6	Conclusions	94
6	Conclusions and Future Lines of Work	95
6.1	Summary of Methods and Contributions	95
6.1.1	Part I: Medical Data Wrangling using VAEs	95
6.1.2	Part II: Learning to Defer to Multiple Experts	96
6.2	Suggestions for Future Research	97
6.2.1	Next Steps in VAEs for Irregular Data	97
	Heterogeneous Likelihoods	97

Modeling data-relationship, rather than data itself	98
Advanced Techniques for Temporal Data	98
Active Learning	99
6.2.2 Next Steps in Learning to Defer	99
Application of Learning to Defer <i>to Multiple Experts</i>	99
Temporal L2D with Conformal Guarantees	100
New Forms of Calibration	100
Conformalize L2D	100
Access to Real Experts' Correctness Probabilities	101
Balancing Reliance on AI	101

Appendices

A ELBO Derivation for Shi-VAE	105
A.1 Shi-VAE <i>without</i> discrete latent space \mathbf{s}_t	105
A.2 Shi-VAE <i>with</i> discrete latent space \mathbf{s}_t	107
B Proofs and Derivations for Multiple Expert L2D	111
B.1 Bayes Rule for Learning to Defer with Multiple Experts	111
B.2 Proof of Theorem 3.1: Consistency of ψ_{SM}^J	114
B.3 Proof of Theorem 3.2: Consistency of ψ_{OVA}^J	115
B.4 Inconsistency of the Mixture of Experts Formulation (Hemmer et al., 2022)	116
Bibliography	119

Doesn't have a point of view
Knows not where he's going to
Isn't he a bit like you and me?

Nowhere Man — The Beatles ▶

1

Introduction

Contents

1.1	All that glitters is not gold	2
1.1.1	Promises of Big Data	3
1.1.2	Pitfalls of Big Data	3
1.2	Research Challenges	4
1.2.1	Challenge I: Dealing with Irregular Observations	5
1.2.2	Challenge II: When and Who Should We Trust?	6
1.3	Thesis Organization	8
1.3.1	Part I: Medical Data Wrangling using VAEs	8
1.3.2	Part II: Learning to Defer to Multiple Experts	9

LIFE is uncertain. No one can deny that. From the moment you are born throughout your whole life, paradoxically, one thing is for certain: life is uncertain. “I’m going on holidays, will it rain tomorrow?”. “The championship final is tomorrow, who will win?”. “Are you expecting a child? Congratulations! Is it a boy or a girl?” We are always asking these kind of questions, and there is always *uncertainty* about the possible answers one could expect. Inevitably, these questions need answers, and the answers do not appear spontaneously. There must be a prior processing step of the question considering the possible outcomes using *contextual information*, and subsequently, the resulting *decision*.

In order to create an abstract route that leads to the final decision, first we need **contextual information**. However, this final decision is not only influenced by the present information, but most often greatly influenced by the *past*. In the case of humans, this information retrieval is done by memory. But we sometimes have lapses and forget: “what did I do in summer two years ago?”. In such cases, we first try to carve inside our memories to get a hint, a clue, an echo that can lead us to that *missing* part. But sometimes we do not succeed, and instead we need to form an abstract memory, that is, we fill those *missing* gaps with the merging of the surrounded memories, which might be composed of conceptual representations of images, sounds, smells still present in our memory. Precisely, this mingling of concepts is *heterogeneous* since each conceptual representation might be of different nature. Hence, we can conclude that our

conceptual representations are inherently *irregular*. In the following sections, we will comment how this view that we humans have about information relates to *data* in our modern times.

Once the information has been partially characterized, we can proceed to the next step: **decision-making**. The process of making decisions is not exclusively confined to individuals, as it can be shaped by external factors and involve the collaboration of other individuals. Let's consider the scenario of a patient visiting a hospital with a skin lesion. The doctor in charge would need to prescribe medication or determine the next steps for an accurate diagnosis. In this case, the responsibility for making the decision lies with the doctor. However, for certain high-stakes tasks, decisions are often made by a team of doctors. Typically, these decisions are reached through consensus within the committee. Nonetheless, in practical terms, certain doctors may exert greater influence in the decision-making process due to their extensive experience and heightened confidence in their judgment. In the present era, these decisions are further supported by machine-generated feedback. Doctors utilize advanced equipment to extract medical information, such as CTA scans, in order to facilitate and alleviate the weight of the ultimate decision. This posits yet another question that this thesis seeks to answer: *For a specific scenario, should the system or human(s) bear the responsibility of decision-making?*

These two preceding examples have been presented as the primary motivation and starting point for this thesis. Firstly, we hypothesize that, just as local-abstract concepts can be combined to form deep-abstract concepts, allowing us to discern missing patterns in our memory, we can design models that leverage probabilistic methods to encode data information from diverse data types. By doing so, we can generate meaningful latent representations to facilitate tasks such as missing data imputation and data wrangling. Secondly, we believe in designing machine learning methods in a more synergistic manner, wherein machines and humans can effectively complement each other's abilities. Our objective is to establish a framework that not only determines whether a decision should be made by a machine learning system or deferred to a single human, but also discerns *which specific human or humans*, from a set of individuals, should hold the responsibility. Furthermore, we also aim for this deferral decision-making process to be reliable, transparent, and trustworthy.

1.1 All that glitters is not gold

In ancient times, the only way to prevent information to be lost was text¹. From the very first poems ever written (Tigay, 2002)² to ancient Chinese agricultural texts (Needham, 1974)³, humans have worried about protecting their ideas and discoveries over time. This worry drove the development of technologies like Johannes Gutenberg's printing press and continues to be relevant in our current era. Nowadays, almost anything can be transformed into a digital format - everything can be saved as a combination of 0s and 1s. No matter the format, whether it is a photo, a music clip, a video, or an electronic health record (EHR), everything is transformed into an abstract concept called **data**.

So far we talked about the content: text, books, photos, *i.e.* data. But we lack the container. A very widespread term that no one can help to notice is **big data**. The term big data can be perceived as ambiguous because it encompasses not only the notion of vast quantities of data, as the term suggests, but also the associated storage and processing technologies that leverage this data (Gandomi and Haider, 2015). It is a common belief that we are living the fourth industrial revolution thanks to big data, the improvement in computational power and the rapid development of artificial intelligence algorithms. While there is undeniable truth to this, we can still acknowledge some promises and pitfalls.

¹Although one could also think about *folklore* or painting.

²https://en.wikipedia.org/wiki/Epic_of_Gilgamesh

³https://en.wikipedia.org/wiki/Qimin_Yaoshu

1.1.1 Promises of Big Data

The irruption of the smartphone into everyone’s life and the ease with which we digitise or record any data supposed an explosion of quantities of data. Smartphones, equipped with advanced cameras and sensors, have empowered individuals to capture moments and contribute to the growing pool of data. The digitized information that captures our interactions with the world can be utilized to establish a characterization of an individual, commonly referred to as the digital phenotype (Jain et al., 2015). What’s more, this vast amount of information captured by personal technology such as social media, online banking, shopping and mobile devices can add immense value to many fields like security, marketing or healthcare. The digital footprint (Bidargaddi et al., 2017) we generate through the use of personal technologies in our daily lives is remarkably diverse, encompassing various types of data and exhibiting temporal dynamics. This includes information such as visited websites, browser queries, interactions on social networks, call logs, physical activity recorded by wearables or mobile phones, GPS location, text data from messaging applications, sleep patterns, and more.

This data-rich landscape holds great promise for research, decision-making, and personalized applications. By carefully analyzing and interpreting this wealth of information, valuable insights, patterns, and trends can be uncovered. However, *big data is worthless in a vacuum*. Its potential value is unlocked only when leveraged to drive decision making. In our recent times we have been participants of the outburst of artificial intelligence: the development of computer systems and algorithms capable of perceiving, reasoning, learning, and problem-solving, emulating certain aspects of human cognitive abilities. We experienced how the AlphaGo (Silver et al., 2016) defeated the world champion of the ancient game Go using deep learning and reinforcement learning, and how this inspired new works like AlphaZero, mastering board games (Silver et al., 2018) or AlphaTensor (Fawzi et al., 2022), an algorithm to discover faster matrix multiplications. In the context of protein folding, AlphaFold (Jumper et al., 2021) represents significant advancement in the field of protein folding as it introduces a groundbreaking approach to predict protein structures. The impact of accurate protein structure prediction extends across diverse domains such as drug discovery, enzyme engineering, and comprehending disease mechanisms. Other works like DALL-E, a deep learning model introduced by Ramesh et al. (2021), has pushed the boundaries of generative AI by enabling the creation of highly realistic and imaginative images from textual descriptions. Or the GPT (Stiennon et al., 2020) model has revolutionized natural language processing by demonstrating impressive capabilities in generating coherent and contextually relevant text based on given prompts. But these works just represent the tip of the iceberg, the accomplished promises of big data and artificial intelligence. But, when embarking on a problem involving big data and algorithm development, the landscape becomes hazier.

1.1.2 Pitfalls of Big Data

If the renowned playwright William Shakespeare was alive today, he would reiterate the timeless sentiment he once wrote: *All that glitters is not gold*. That is, not everything that looks precious or true turns out to be completely so. In our concerning problem, we will talk about two main pitfalls that appear in any problem involving data: data quality, and algorithms as independent agents.

Data quality is overlooked

The very first thing we need to worry about, even before algorithm design, is data itself. Because of the ease of capturing data on today's days, we can have data indiscriminately. But having data is not the same as having *good-quality data*. Actually, the assessment of data quality can become one of the biggest problems in practical big-data scenarios. Quoting Lawrence (2017): *'water, water everywhere and not a drop to drink', we have 'data, data everywhere and not a set to process'. Just as extracting drinkable water from the real ocean requires the expensive process of desalination, extracting usable data from the data-ocean requires a significant amount of processing.* This data processing and data preparation prior to the actual data modeling can be very time-consuming and tedious. Some authors comment that the task of data preparation constitutes approximately 80% of the work of data scientists⁴. This highlights the necessity for effective frameworks that can alleviate the burden on data scientists and big data practitioners by automating data cleaning and enhancing data understanding. In the subsequent sections of the thesis, we will present our proposed solution, which not only addresses the data cleaning aspect but also incorporates comprehensive data understanding capabilities using probabilistic methods based on deep generative models.

AI is not an independent agent

The next pitfall comes with the common belief that artificial intelligence behaves completely as an independent agent, to the extent that some argue these agents could potentially replace humans entirely. Obviously, this is far from being true. The question of whether statistical methods or the human brain excel in predicting certain outcomes has long been a topic of discussion in research. The rapid advancements in artificial intelligence, as demonstrated by the aforementioned achievements, have triggered a lively debate. However, the most probable scenario for the future is not one where humans and machines operate independently, but rather one of mutual complementarity, where they collaborate, learn from each other. This perspective is encapsulated in the concept of Hybrid Intelligence (HI) (Dellermann et al., 2019), which seeks to combine human intelligence with AI to achieve performance beyond what either humans or machines could achieve alone (Kamar, 2016b). In this thesis, we embrace the paradigm of human-machine collaboration and present a solution that revolves around determining when the machine should make predictions versus when a human should intervene. In particular, our focus is on efficiently discerning *when* we should defer to individuals and *to whom* individuals within a pool of humans who exhibit the highest levels of confidence and expertise.

1.2 Research Challenges

This section will discuss the research challenges that provided the motivation for this thesis and highlight the contributions we made in addressing these issues.

⁴<https://whatsthebigdata.com/2016/05/01/data-scientists-spend-most-of-their-time-cleaning-data/>

1.2.1 Challenge I: Dealing with Irregular Observations

Designing algorithms to process data and provide meaningful results can be nerve-wracking. In the context of machine learning, we can define a universal formula (Lawrence, 2017)

$$\text{data} + \text{model} \rightarrow \text{prediction} \quad (1.1)$$

By looking at the formula, we can deduce that having a good prediction depends on having both a good model and good data. The common trend in the machine learning community is to exert a great effort on the design of the best model. This emphasis is particularly noticeable in academic settings, where research ideas are often evaluated using benchmark datasets that have minimal artifacts or noise. But the real-life case is completely different. Real-life datasets frequently contain artifacts due to a range of factors. These factors include noise and errors originated from data collection through sensors or measurement devices, the presence of missing data due to sensor failures or incomplete processes, data corruption during storage or transmission, sampling bias that leads to data collection that is not fully representative of the entire population, human errors during data entry or interpretation, and the challenges associated with integrating datasets of different types.

Data Wrangling

Before applying machine learning algorithms, it is crucial to establish robust big data wrangling techniques, since this particular aspect poses a significant bottleneck in the development of robust algorithms. *Data wrangling* encompasses the extraction, integration, and cleaning processes necessary for data to be analyzed. Classical data wrangling problems include addressing missing data, detecting outliers, identifying data errors, and cleaning dirty data (Kandel et al., 2011). It is often estimated that data scientists spend more than half of their time on data wrangling, emphasizing the urgent need for systematic and cost-effective techniques to support wrangling activities (Furche et al., 2016). This demand has led to a recent surge of interest from both industry and academia in these problems, resulting in the exploration of new abstractions, interfaces, scalable approaches, and statistical techniques (Abiteboul et al., 2017; Chu et al., 2016).

Irregular observations

The majority of machine learning research primarily concentrates on handling regular observations, neglecting a crucial technological obstacle encountered in practical big-data scenarios: the aggregation and curation of heterogeneous information streams. This issue involves tackling various challenges, including temporal alignment of different information streams, resolving discrepancies in measurements (such as inconsistencies between step counts from wearables and smartphones or conflicting activity reports), managing missing or corrupted data, and effectively processing the information streams to construct features that accurately capture human behavior properties. It is imperative to address these challenges to adequately capture the complexity of real-world data in big-data applications

In this thesis, we aim to address the often overlooked challenge of handling irregularities within the context of medical data. We will focus on three specific aspects. Firstly, we will tackle the issue of **missing** data by developing a framework that enables us to impute (or *marginalize* in more statistical terms) missing points using relevant information from other data sources or

past observations. Secondly, we will move beyond the assumption of homogeneous observations, where only one statistical data type (such as Gaussian) is considered, and instead, work with **heterogeneous** observations. This means that different data sources can be represented by various statistical likelihoods, such as Gaussian, Bernoulli, categorical, etc. Lastly, considering the **temporal** enrichment of today’s collected data and our focus on medical data, we will develop a novel algorithm capable of capturing and propagating correlations among different data streams over time. These three properties of the data involve determining how to temporally align the different information streams, dealing with contradictory measurements (*e.g.*, inconsistencies between step counts from wearables and smartphones), handling missing or corrupted data, and processing the information streams together to construct features that adequately capture the correlation between different sources. All this process of data wrangling and data understanding will be done in a fully *unsupervised* way using deep probabilistic methods.

Contribution Our first contribution in this thesis involves the development of a novel method based on Deep Generative Models (DGM) using Variational Autoencoders (VAE). This method enables the aggregation of multiple heterogeneous data streams in a modular manner, taking into consideration the presence of potential missing data. The research was conducted within the scope of the *Deep-DARWIN*⁵ project, which focuses on constructing human behavior models using indirect data measurements obtained from personalized technology. The data is aggregated based on simpler conceptual levels such as mobility data, social network interactions, physical activity, and emotions. To demonstrate the feasibility of our approach, we present proof-of-concept results obtained from a real database generated through continuous passive monitoring of psychiatric patients. This research collaboration with the Hospital Universitario Fundacion Jimenez Diaz involved the utilization of mobile phones, wearables and data aggregated from social networks and fitness platforms.

1.2.2 Challenge II: When and Who Should We Trust?

The previous challenge was focused on the search of a novel method to *help* the human, alleviating the data processing burden, and simultaneously facilitating a comprehensive understanding of the data. In this section we go one step further. Now we will not visualize our AI agent as a practical tool that can only help us, but we want to interact with this agent. Our goal aligns with the search of a *collective intelligence* that tries to answer the following question: *How can people and computers be connected so that collectively they act more intelligently than any individuals, groups, or computers have ever done before?* (Leimeister, 2010).

Referring back to Equation 1.1, we can now add the human into the equation

$$\text{data} + \boxed{\text{model} \text{ } \text{🧑} \text{ human}} \rightarrow \text{safer prediction} \quad (1.2)$$

The purpose of including the human is not just to obtain better performance, but also to obtain more reliable and trust-worthy predictions we can rely on. The notion that AI systems can solely account for automated decision-making, especially in critical domains such as healthcare, is far from reality. To motivate why *only* machine learning models cannot decide independently, we take the study conducted by (Beede et al., 2020) as an example. In this work, it was observed that the model declined to make predictions for 20% of the samples due to blurry

⁵<http://deep-darwin.webs.tsc.uc3m.es/>

images. By eliminating ophthalmologists from the system, important safety checks against model failure (such as distribution shift) and input issues are eliminated as well. Similarly, in another study for antidepressant prescription, the performance degrade notably when clinicians received recommendations from machine learning models. Similarly, in another study for antidepressant prescription, the performance degrade notably when clinicians received recommendations from ML models (Jacobs et al., 2021).

These works effectively unveil the important flaw of not integrating the human into algorithm design. In the latter part of this thesis, our emphasis shifts towards the development of human-centric approaches within the framework of *hybrid intelligence* Kamar (2016a); Dellermann et al. (2019); Akata et al. (2020). Hybrid intelligence is characterized by its capability to attain complex objectives through the combination of human and artificial intelligence, thereby achieving superior results compared to what each entity could achieve independently.

Who should bear responsibility?

In the pursuit of finding the optimal balance between machines and humans, an important question arises: who should shoulder the responsibility? Now, our focus turns to a specific scenario where the algorithm has the ability to make predictions independently or alternatively defer the responsibility to a human expert. One intuitive approach to address this problem is to consult the expert whenever the model lacks confidence (Chow, 1957). However, this simplistic approach overlooks the actual level of human confidence. For instance, a study conducted by Tschandl et al. (2020) examined the use of AI feedback for clinicians in diagnosing various skin cancer lesions. The findings demonstrated that experienced clinicians derived less benefit from AI assistance, while inexperienced raters significantly benefited from regular AI guidance but were adversely affected by an inaccurate AI model. In order to faithfully rely on the model, we need the model to understand what the human knows, when the human is confident and the model can predict with a higher confidence than the human.

Learning to defer

All the above-mentioned concerns are tackled in a recent theory called Learning to defer (L2D) (Madras et al., 2018). The concept of learning to defer involves the AI system acquiring the ability to identify specific instances or circumstances where it lacks confidence or faces uncertainty in generating accurate predictions. Rather than offering potentially flawed or unreliable predictions, the system delegates the responsibility of decision-making to a human expert. This approach ensures that the AI system can seek input or guidance from humans when needed, leading to more precise and dependable outcomes. For instance, in the context of semi-autonomous driving, the AI system relies on sensing the state of the human to effectively distribute tasks between itself and the human driver.

Who should we trust?

The issue of *trust* in AI poses a significant obstacle to its widespread adoption. Establishing a delicate balance between trust and distrust is crucial to leverage the benefits of AI without succumbing to over reliance (Lee and See, 2004). Consequently, this thesis also focuses on ensuring the validity of our model’s confidence to promote transparency and trust. That is, we want our L2D models to be confidence-calibrated. However, we go beyond the common assumption of working with only one expert in Learning to Defer (L2D) scenarios and extend it to involve multiple experts. Therefore, we aim to determine not only when to seek human expertise but also which specific human expert to consult. This notion of ensembling experts based on their confidence levels is essential for optimizing the human-machine combination, as machine learning models can be influenced by less-experienced clinicians, which we aim to avoid.

Contribution The second contribution of this thesis is the extension of the two most recent works of L2D, namely the works by Mozannar and Sontag (2020) and Verma and Nalisnick (2022) to the multiple expert setting. Our study (Verma et al., 2023) investigates the statistical properties of learning to defer (L2D) to multiple experts. We address the challenges of consistent surrogate loss, confidence calibration, and expert ensembling. We derive two surrogate losses based on Mozannar and Sontag (2020) and Verma and Nalisnick (2022) works, and analyze their ability to estimate expert prediction probabilities. The ensembling of experts is performed using a conformal inference technique that is applied on the deferred experts, to retrieve those experts that are expected to be correct.

1.3 Thesis Organization

The structure of this doctoral manuscript comprises two primary parts, each dedicated to addressing one of the main challenges mentioned earlier. The first part focuses on medical data wrangling through the utilization of sequential variational autoencoders, while the second part dives into the topic of learning to defer to multiple experts. Chapters 2 and 4 primarily serve as introductions and do not present any novel contributions. They set the stage for our main contributions, which are presented in Chapters 3 and 5.

1.3.1 Part I: Medical Data Wrangling using VAEs

Chapter 2: Handling irregular observations using VAEs

This chapter serves as an introduction to our used learning framework, variational autoencoders (VAE), which we will use for medical data wrangling. We present a comprehensive overview of the general VAE theory, covering important notation and fundamental concepts. Moreover, we highlight the three main challenges we aim to tackle in our research: handling missing data, dealing with heterogeneous data, and addressing temporal data. For each of these challenges, we review relevant works from the literature that have attempted to tackle them. This chapter serves as a motivating and introductory section, setting the stage for the proposed model described in the following chapter.

Chapter 3: Medical Data Wrangling With Sequential Variational Autoencoders

In this chapter, we present our novel contribution that addresses the challenge of medical data wrangling using sequential variational autoencoders. We propose an innovative approach to model medical data records that consist of heterogeneous data types and exhibit bursty missing data using sequential variational autoencoders (VAEs). Building upon the work of [Nazabal et al. \(2020\)](#), we introduce a new methodology called Shi-VAE, which extends the capabilities of VAEs to effectively handle sequential data streams with missing observations. To assess the effectiveness of our proposed model, we compare it against state-of-the-art solutions using both an intensive care unit database (ICU) and a dataset of passive human monitoring. Furthermore, we demonstrate that conventional error metrics like RMSE are inadequate for evaluating temporal models. Hence, we incorporate the cross-correlation between the ground truth and the imputed signal into our analysis. Our experimental results reveal that Shi-VAE outperforms other methods in terms of both metrics, showcasing its superior performance.

1.3.2 Part II: Learning to Defer to Multiple Experts

Chapter 4: Learning to Defer

Moving on from the first part, our focus now shifts to the section devoted to human-machine collaboration. This section serves as an introduction to the theory of learning to defer, which will be the central theme of the final contribution in this thesis. Initially, we present the notation and theoretical background of a standard classification problem, encompassing both multiclass and binary scenarios. The purpose is to emphasize that in certain situations, relying solely on a machine learning classifier may not be sufficient. This realization leads us to introduce the context and background for learning to defer, placing it within the framework of rejection learning theory ([Chow, 1957](#)), and providing an overview of recent related works. Next, we introduce the two consistent surrogate losses proposed in the existing learning to defer literature. These losses will be referred as the *softmax* loss by [Mozannar and Sontag \(2020\)](#) and the *OvA* loss by [Verma and Nalisnick \(2022\)](#). To illustrate the behavior of these losses, we include an example using a Mixture of Gaussians dataset. Additionally, we present the degenerate behavior discovered by [Verma and Nalisnick \(2022\)](#) for the softmax surrogate loss. This finding serves as further motivation for our contribution, as described in Chapter 5.

Chapter 5: Learning to Defer to Multiple Experts

In this thesis, we present the final chapter, which focuses on analyzing the statistical properties of learning to defer (L2D) to multiple experts. Our main contributions involve addressing challenges related to consistent surrogate loss derivation, confidence calibration, and principled expert ensembling. We derive two consistent surrogate losses for the multi-expert setting, one using softmax parameterization and the other employing a one-vs-all (OvA) parameterization, similar to the single expert losses proposed by [Mozannar and Sontag \(2020\)](#) and [Verma and Nalisnick \(2022\)](#) respectively. Through our analysis, we find that the softmax-based loss leads to the propagation of mis-calibration among the estimates, while the OvA-based loss does not, although practical trade-offs exist. Lastly, we introduce a conformal inference technique to selectively query a subset of experts when the system defers, and we empirically validate our approach using tasks such as galaxy, skin lesion, and hate speech classification.

Chapter 6: Conclusions and Future Work

The chapter marks the culmination of the doctoral manuscript. We summarize the novel contributions presented in this thesis: 1) the application of sequential variational autoencoders for medical data wrangling, and 2) the extension of learning to defer to multiple experts, accounting for confidence calibration and conformal ensembles. Additionally, we offer future guidelines for further research ideas in the context of VAEs handling irregular observations and learning to defer.

Part I

Medical Data Wrangling using VAEs

El sueño va sobre el tiempo
 flotando como un velero.
 Nadie puede abrir semillas
 en el corazón del sueño.

La Leyenda del Tiempo — Federico García Lorca ▶

2

Handling Irregular Observations using VAEs

Contents

2.1	Our learning framework: Variational Autoencoders	14
2.1.1	Latent Variable Models	14
2.1.2	Variational Inference and ELBO	16
2.1.3	Variational Autoencoders	18
2.2	Handling Missing Data	21
2.2.1	Imputation and Deletion Techniques	21
2.2.2	Generative Models for Missing Data	22
2.2.3	HI-VAE: Our Basis for Missing data Handling	23
2.3	Handling Heterogeneous Data	24
2.3.1	The Challenge of Modeling Heterogeneous Distributions	24
2.3.2	VAEs for Heterogeneous Data	25
2.3.3	HI-VAE: Our Basis for Heterogeneous Data Handling	26
2.4	Handling Temporal Data	29
2.4.1	RNNs: Our Toolkit for Temporal Data	29
2.4.2	RNNs Handling Missing Data in Medical Context	31
2.4.3	VRNN: Our Basis for Temporal Data Handling	32
2.5	Summary of the Chapter	33

NOWADAYS we have access to a massive amount of data from which we can extract very valuable information. In particular, in this part of the thesis we focus on *medical data*. Our major goal now is to answer the following question: *what can we do to understand medical data?*

First, we need to *learn* from this data. That is, we want to design a model that is able to capture the information and all those hidden correlations that might be present within the data. In more mathematical words, we want to find a set of parameters θ that define a probabilistic model $p_{\theta}(\mathbf{x})$ that is able to approximate the true distribution of the data $p^*(\mathbf{x})$, *i.e.* $p_{\theta}(\mathbf{x}) \approx p^*(\mathbf{x})$.

Nevertheless, the process of learning becomes significantly difficult when dealing with *irregular* observations. In practical scenarios, particularly in healthcare, it is common to encounter missing values within datasets. These *missing* values may arise due to various factors such as sensor failures during laboratory measurements. Additionally, we often come across *heterogeneous* observations where the data instance \mathbf{x} consists of a combination of variables $x_d, d = 1, \dots, D$ belonging to

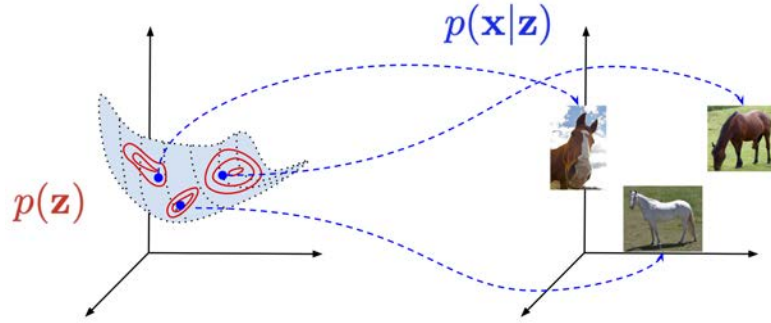


Figure 2.1: *Latent variable model diagram:* Diagram showing a latent variable model and the corresponding generative process. \mathbf{z} lies in a 2D manifold embedded in a high dimensional space. Figure extracted from this [blog](https://jmtomczak.github.io/blog/4/4_VAE.html)¹ by Jakub M. Tomczak

different statistical types. This combination may include a mixture of real-valued, binary, and categorical data. Moreover, medical data typically exhibits a *temporal* aspect due to the inherent nature of the data, such as electrocardiogram (EKG) readings, or because it is associated with a patient’s medical history. In order to handle all these irregularities inherent to the data, we will use Variational Autoencoders (VAE), which are a class of *Deep Generative Models* (DGM). We hypothesize that medical data exhibits strong hidden correlations that can be captured in a common lower dimensional space which can be then used to generate data in a discriminative way.

The outline for this chapter is the following: in Section 2.1 we present our learning framework, the VAEs: we briefly present *latent variable models* (LVM), which sets the basis for our lower dimensional space \mathbf{z} , briefly continue with the theory of Variational Inference (VI) and finally present VAEs. We continue on how we can extend VAEs to 1) handle missing data (Section 2.2), 2) handle heterogeneous data (Section 2.3) and 3) handle temporal data (Section 2.4).

Comments on contributions This chapter does not include any novel contribution. It was written drawing inspiration from the following amazing works: VAE Section 2.1 from David Blei’s introduction to VI (Blei et al., 2017), Kingma’s and Welling’s introduction to VAEs (Kingma et al., 2019) and Jakub Tomczak’s VAEs blog. Missing data Section 2.2 from Chao Ma’s thesis (Ma, 2022). Temporal data Section 2.4 from Lipton et al. (2015)’s review of RNN for sequence learning.

2.1 Our learning framework: Variational Autoencoders

2.1.1 Latent Variable Models

A *latent variable model* (LVM) is a statistical framework that incorporates hidden or unobserved variables, referred to as *latent variables*, to capture complex relationships and underlying structures in the data. These latent variables are not directly observed but are inferred from the available data. We assume that there is some unobservable information that influences the observed data. The model assumes that the observed data is generated as a result of the interaction between the latent variables and other known variables or parameters. For a given data point $\mathbf{x} \in \mathcal{X}^D$ e.g. an image, a sentence, etc. the latent variable are often denoted as $\mathbf{z} \in \mathcal{Z}^K$, where it is often assumed that \mathbf{z} is a low-dimensional representation of \mathbf{x} , i.e. $K \ll D$. \mathcal{Z}^K can

¹https://jmtomczak.github.io/blog/4/4_VAE.html

be referred as a low-dimensional *manifold* (see Figure 2.1 for a better understanding). The generative process that defines \mathbf{x} is therefore

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z}, \quad (2.1)$$

which is usually called the *marginal likelihood* or the *model evidence*, when taken as a function of $\boldsymbol{\theta}$. The relationship between \mathbf{x} and \mathbf{z} allows the distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ to be quite flexible.

The next natural questions is how we can calculate the integral in Equation 2.1. This integral is generally very difficult to solve. First, we present an example where this integral is tractable and that will be very much related to VAEs. Second, we present variational inference, which will be used as the learning framework to handle the above-mentioned equation when the integral is intractable

Probabilistic PCA (pPCA) The probabilistic Principal Component Analysis (pPCA) (Tipping and Bishop, 1999) model is defined as follows. First, let us consider now that $\mathbf{x} \in \mathbb{R}^D$ and $\mathbf{z} \in \mathbb{R}^K$ are continuous variables. We choose a standard Gaussian prior $p_{\boldsymbol{\theta}}(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, and the dependency between \mathbf{x} and \mathbf{z} be defined as

$$\mathbf{x} = \mathbf{W}\mathbf{z} + \mathbf{b} + \epsilon, \quad (2.2)$$

where $\epsilon \sim \mathcal{N}(\epsilon|0, \sigma^2\mathbf{I})$. Then we know that

$$p_{\boldsymbol{\theta}}(\mathbf{x} | \mathbf{z}) = \mathcal{N}(\mathbf{x} | \mathbf{W}\mathbf{z} + \mathbf{b}, \sigma^2\mathbf{I}). \quad (2.3)$$

Since the prior $p_{\boldsymbol{\theta}}(\mathbf{z})$ and the likelihood $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ are Gaussian, we can exploit the properties of the linear combination of two Gaussians (Bishop, 2006) and obtain a closed-form solution for the evidence as:

$$\begin{aligned} p_{\boldsymbol{\theta}}(\mathbf{x}) &= \int p_{\boldsymbol{\theta}}(\mathbf{x} | \mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{z}) d\mathbf{z} \\ &= \int \mathcal{N}(\mathbf{x} | \mathbf{W}\mathbf{z} + \mathbf{b}, \sigma^2\mathbf{I}) \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{I}) d\mathbf{z} \\ &= \mathcal{N}(\mathbf{x} | \mathbf{b}, \mathbf{W}\mathbf{W}^{\top} + \sigma^2\mathbf{I}). \end{aligned} \quad (2.4)$$

And the same applies for the posterior

$$p_{\boldsymbol{\theta}}(\mathbf{z} | \mathbf{x}) = \mathcal{N}(\mathbf{M}^{-1}\mathbf{W}^{\top}(\mathbf{x} - \mathbf{b}), \sigma^{-2}\mathbf{M}), \quad (2.5)$$

where $\mathbf{M} = \mathbf{W}^{\top}\mathbf{W} + \sigma^2\mathbf{I}$. Once we found the optimal parameters by maximizing the log-likelihood we can calculate the distributions over the latent variables \mathbf{z} . But this closed-form expression comes at the cost of using linear combinations, that is, by using \mathbf{W} . Next question is: *is it still tractable* if we apply non-linear combinations; or if other distributions different from the Gaussian are used? The answer is no. The integral would not be tractable any more, and therefore we need a new learning framework that handles this problem.

Deep Latent Variables For probabilistic Principal Component Analysis (pPCA), we introduced a latent variable model that relied on a fully Gaussian assumption, which facilitated the tractability of solving complex integrals. However, it is also possible to parameterize the latent variables using neural networks. In such cases, we refer to the model as a *deep latent variable model* (DLVM). The advantage of using neural networks is that we can approximate very complex distributions for $p_{\theta}(\mathbf{x})$. The common assumption is to factorize the joint distribution as follows

$$p_{\theta}(\mathbf{x}, \mathbf{z}) = p_{\theta}(\mathbf{z})p_{\theta}(\mathbf{x}|\mathbf{z}) \quad (2.6)$$

The computational challenge of evaluating the probability distribution $p_{\theta}(\mathbf{x})$ (Equation 2.1) is connected to the computational challenge of estimating the posterior distribution $p_{\theta}(\mathbf{z}|\mathbf{x})$. It is worth noting that the joint distribution $p_{\theta}(\mathbf{x}, \mathbf{z})$ can be computed efficiently, and the densities are related by Bayes' theorem:

$$p_{\theta}(\mathbf{z}|\mathbf{x}) = \frac{p_{\theta}(\mathbf{x}, \mathbf{z})}{p_{\theta}(\mathbf{x})}. \quad (2.7)$$

When the marginal likelihood $p_{\theta}(\mathbf{x})$ is computationally tractable, it implies that the posterior distribution $p_{\theta}(\mathbf{z}|\mathbf{x})$ is also tractable, and conversely, a tractable posterior leads to a tractable marginal likelihood. At first, the reader might wonder whether the integral for $p_{\theta}(\mathbf{x})$ can be solved by numerical approximations. Indeed, the simplest scenario would be approximating the integral by *Monte Carlo* (MC) samples

$$p_{\theta}(\mathbf{x}) = \int p_{\theta}(\mathbf{x}|\mathbf{z})p_{\theta}(\mathbf{z})d\mathbf{z} = \mathbb{E}_{\mathbf{z} \sim p_{\theta}(\mathbf{z})}[p_{\theta}(\mathbf{x}|\mathbf{z})] \approx \frac{1}{J} \sum_j p_{\theta}(\mathbf{x}|\mathbf{z}_j) \quad (2.8)$$

where we take J samples from the prior. Intuitively this approach seems simple. But if the dimensionality of \mathbf{z} grows, then we fall into the *curse of dimensionality* and we would need an exponentially increasing number of samples to cover the whole space. On the contrary, few samples would result in suboptimal solutions. Next we present how the notion of *variational inference* is a right tool to handle Equation 2.1.

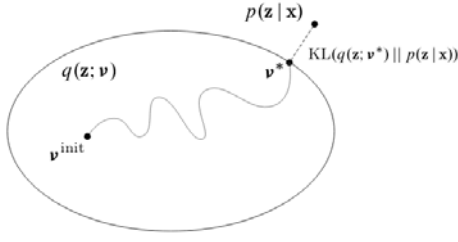
2.1.2 Variational Inference and ELBO

Learning: Variational Inference Approximating the posterior distribution $p_{\theta}(\mathbf{z}|\mathbf{x})$ and the marginal likelihood $p_{\theta}(\mathbf{x})$ has been proven to be a difficult task for DLVMs. However, approximate inference techniques can be employed to circumvent the problem. In the literature we find different options (maximum a posteriori (MAP), Variational EM (Neal and Hinton, 1998)), however we will use a different learning technique, the so called *variational inference* (VI) (Jordan et al., 1999).

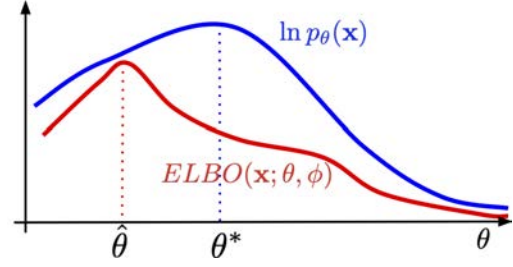
In variational inference the goal is turning the intractable posterior inference into a tractable problem. To do so, we introduce a variational family with variational parameters ϕ . These variational parameters ϕ define a parametric inference model denoted as $q_{\phi}(\mathbf{z}|\mathbf{x})$, the so called *encoder* or inference model. As depicted in Figure 2.2a, we want to optimize the variational parameters to approximate the true posterior, *i.e.*

$$q_{\phi}(\mathbf{z}|\mathbf{x}) \approx p_{\theta}(\mathbf{z}|\mathbf{x}). \quad (2.9)$$

The variational distribution $q_{\phi}(\mathbf{z}|\mathbf{x})$ is defined explicitly — for example, we can assume a spherical Gaussian distribution with parameters $\phi = \{\mu, \sigma^2\}$.



(a) Variational inference. Figure courtesy of David Blei from these [VI slides](#)².



(b) ELBO gap. Figure from Jakub M. Tomczak VAEs' blog³.

Figure 2.2: *Variational inference and ELBO gap:* In Figure (a) we depict the variational inference procedure: starting from some initial variational parameters \mathbf{v}^{init} , we try to approximate from the variational family $q(\mathbf{z}; \mathbf{v})$ to the true posterior $p(\mathbf{z}|\mathbf{x})$. The distance between the true posterior and our best approximation (\mathbf{v}^*) is the KL divergence. In Figure (b) we depict the gap between the ELBO and the true log-likelihood resulting from choosing a bad approximation $q_\phi(\mathbf{z}|\mathbf{x})$ of $p_\theta(\mathbf{z}|\mathbf{x})$. A very simple variational posterior would result in a very high value for $D_{KL}[q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})]$ in Equation 2.12 — the predicted parameter $\hat{\theta}$ is far from the true parameter θ^* .

ELBO In our attempt to calculate the marginal likelihood, we can take the logarithm of the marginal likelihood and we get the following expression

$$\begin{aligned}
 \log p_\theta(\mathbf{x}) &= \log \int p_\theta(\mathbf{x} | \mathbf{z}) p_\theta(\mathbf{z}) d\mathbf{z} \\
 &= \log \int \frac{q_\phi(\mathbf{z})}{q_\phi(\mathbf{z})} p_\theta(\mathbf{x} | \mathbf{z}) p_\theta(\mathbf{z}) d\mathbf{z} \\
 &= \log \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z})} \left[\frac{p_\theta(\mathbf{x} | \mathbf{z}) p_\theta(\mathbf{z})}{q_\phi(\mathbf{z})} \right] \\
 &\geq \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z})} \log \left[\frac{p_\theta(\mathbf{x} | \mathbf{z}) p_\theta(\mathbf{z})}{q_\phi(\mathbf{z})} \right] \\
 &= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z})} [\log p_\theta(\mathbf{x} | \mathbf{z}) + \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z})] \\
 &= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z})} [\log p_\theta(\mathbf{x} | \mathbf{z})] - \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z})} [\log q_\phi(\mathbf{z}) - \log p_\theta(\mathbf{z})] \\
 \text{ELBO} &:= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z})} [\log p_\theta(\mathbf{x} | \mathbf{z})] - \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z})} [\log q_\phi(\mathbf{z}) - \log p_\theta(\mathbf{z})],
 \end{aligned} \tag{2.10}$$

where we applied Jensen's inequality in the fourth line. This final expression is the so called Evidence Lower Bound (ELBO). As the name suggests, this new equation defines a lower bound for the true log likelihood of the model. It can be checked in Figure 2.2b.

Amortized Inference In standard variational inference we would have a variational distribution for each data sample. However, we could instead allow the inference model $q_\phi(\mathbf{z}|\mathbf{x})$ to use a set of parameters to model the relationship between the input \mathbf{x} and the latent variables \mathbf{z} , that is, $q_\phi(\mathbf{z}|\mathbf{x})$ instead of $q_\phi(\mathbf{z})$. This is called *amortized inference* (Gershman and Goodman, 2014) (other works such as Kim et al. (2018) have investigated semi-amortized inference techniques; however, we will always stick to fully amortized inference). With amortized inference we train one single network and we can obtain the parameters of the variational distribution rather fast. With this new scheme, the ELBO can be expressed as

$$\begin{aligned}
 \log p_\theta(\mathbf{x}) &\geq \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x} | \mathbf{z})] - \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log q_\phi(\mathbf{z}|\mathbf{x}) - \log p_\theta(\mathbf{z})] \\
 &= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x} | \mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}))
 \end{aligned} \tag{2.11}$$

²http://www.cs.columbia.edu/~blei/talks/Blei_VI_tutorial.pdf

³https://jmtomczak.github.io/blog/4/4_VAE.html

Notice that we just substituted $q_\phi(\mathbf{z}|\mathbf{x})$ by $q_\phi(\mathbf{z})$ in Equation 2.10. The first term for this ELBO $q_\phi(\mathbf{z}|\mathbf{x})[\log p_\theta(\mathbf{x} | \mathbf{z})]$ is the expected conditional log-likelihood, which can be seen as the (negative) reconstruction error, since the input \mathbf{x} is encoded to \mathbf{z} and then decoded back. The second part of the ELBO, $D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}))$ is the KL divergence between the variational posterior $q(\mathbf{z}|\mathbf{x})$, and the prior $p(\mathbf{z})$. This KL term is often seen as a *regularizer* that forces the variational distribution towards the prior.

Important of having a good variational posterior The ELBO can be derived in different ways (Hoffman and Johnson, 2016; Alemi et al., 2018). In the following we derive the ELBO as follows

$$\begin{aligned}
\log p_\theta(\mathbf{x}) &= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x})] \\
&= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{z} | \mathbf{x}) p_\theta(\mathbf{x})}{p_\theta(\mathbf{z} | \mathbf{x})} \right] \\
&= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x} | \mathbf{z}) p_\theta(\mathbf{z})}{p_\theta(\mathbf{z} | \mathbf{x})} \right] \\
&= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x} | \mathbf{z}) p_\theta(\mathbf{z})}{p_\theta(\mathbf{z} | \mathbf{x})} \frac{q_\phi(\mathbf{z} | \mathbf{x})}{q_\phi(\mathbf{z} | \mathbf{x})} \right] \\
&= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x} | \mathbf{z}) \frac{p_\theta(\mathbf{z})}{q_\phi(\mathbf{z} | \mathbf{x})} \frac{q_\phi(\mathbf{z} | \mathbf{x})}{p_\theta(\mathbf{z} | \mathbf{x})} \right] \\
&= \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x} | \mathbf{z}) - \log \frac{q_\phi(\mathbf{z} | \mathbf{x})}{p_\theta(\mathbf{z})} + \log \frac{q_\phi(\mathbf{z} | \mathbf{x})}{p_\theta(\mathbf{z} | \mathbf{x})} \right] \\
&= \underbrace{\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x} | \mathbf{z})]}_{\text{ELBO}} - D_{KL}[q_\phi(\mathbf{z} | \mathbf{x}) || p_\theta(\mathbf{z})] + D_{KL}[q_\phi(\mathbf{z} | \mathbf{x}) || p_\theta(\mathbf{z} | \mathbf{x})].
\end{aligned} \tag{2.12}$$

In this reformulation we get the same as Equation 2.11, with an additional term at the end. This term is the Kullback-Leibler divergence between the variational posterior and true posterior. However, we don't have access to the real posterior. As shown Figure 2.2b, we will obtain a gap between the log-likelihood and the ELBO, and this gap will be tight or loose depending on the variational posterior, and therefore on the capacity of the *encoder* architecture of our VAE, as noted by Cremer et al. (2018) and Mattei and Frellsen (2018).

2.1.3 Variational Autoencoders

So far the reader might notice that we have followed an encoder-decoder scheme, where we first encode our samples \mathbf{x} into a latent space \mathbf{z} and then decode back to \mathbf{x} , or the approximation $\hat{\mathbf{x}}$ (ideally we would expect \mathbf{x} , however there is always some loss of information). However, our encoder-decoder mechanism is not deterministic, we allow the encoder and the decoder to be stochastic, that is, to explicit set them to follow a certain distribution. The idea of using an autoencoder with variational inference as learning framework is called *Variational Autoencoder* (VAE) (Kingma and Welling, 2014; Rezende et al., 2014).

Variational Autoencoders (VAE) are composed of a stochastic encoder or *inference* model, and a stochastic decoder or *generative* model. These two modules are interconnected yet parametrized independently. The encoder defines the variational posterior $q_\phi(\mathbf{z}|\mathbf{x})$ that approximates the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$ and the stochastic decoder defines the conditional distribution of the data given the latent variable $p_\theta(\mathbf{x}|\mathbf{z})$. Figure 2.3 depicts the overall framework. The parameters ϕ and θ are often parametrized using neural networks, thereby defining a DLVM. The last choice when

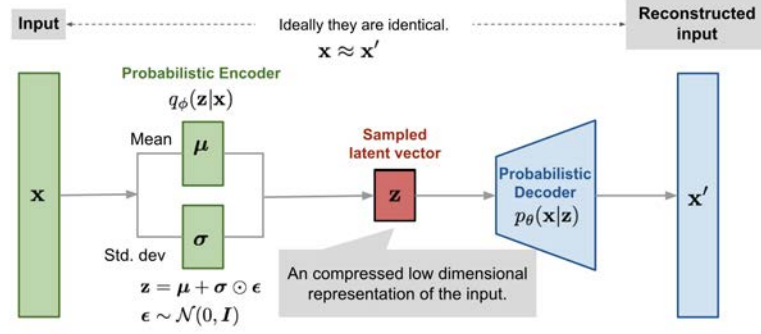


Figure 2.3: VAE framework: In this figure we depict the VAE framework with a Gaussian latent variable \mathbf{z} . The input \mathbf{x} is encoded to \mathbf{z} through some neural networks with parameters ϕ and then decoded back with a different neural network with parameter θ . \mathbf{z} is commonly a low-dimensional representation of \mathbf{x} . Also notice that \mathbf{z} is calculating using the *reparametrization trick* $\mathbf{z} = \mu + \sigma \odot \epsilon$. Figure extracted from (Weng, 2018).

designing VAEs is the prior $p(\mathbf{z})$, which we will briefly comment later. VAEs are optimized using the ELBO as objective function. For the sake of clarity, we rewrite the equation again

$$\text{ELBO} : \log p_{\theta}(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x} | \mathbf{z})] - D_{KL}(q_{\phi}(\mathbf{z}|\mathbf{x}) || p_{\theta}(\mathbf{z})) \quad (2.13)$$

Decoder (Generative model) parametrization The choice of the decoder is rather flexible and will be fully dependent on the problem. We could choose a *categorical* distribution $p_{\theta}(\mathbf{x}|\mathbf{z}) = \text{Categorical}(\mathbf{x}|\theta(\mathbf{z}))$ where the probabilities for each category are the output of a neural network $\theta(\mathbf{z}) = \text{softmax}(\psi(\mathbf{z}))$, where ψ is the neural network, *e.g.* a multi-layer perceptron (MLP), convolutional neural network (CNN), *etc.* Another flexible choice is the Gaussian distribution $p_{\theta}(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mu_{\theta}(\mathbf{z}), \text{diag}[\sigma_{\theta}^2(\mathbf{z})])$, where $\mu_{\theta}(\mathbf{z})$ and $\sigma_{\theta}^2(\mathbf{z})$ are the outputs of a neural network, and $\text{diag}(\cdot)$ is the diagonal operator.

Encoder (Recognition model) parametrization The variational distribution for the encoder can also be parametrized by neural networks. A common choice is defining a continuous latent space with a Gaussian distribution

$$q_{\phi}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\mu_{\phi}(\mathbf{x}), \text{diag}[\sigma_{\phi}^2(\mathbf{x})]) \quad (2.14)$$

where again $\mu_{\phi}(\mathbf{x})$ and $\sigma_{\phi}^2(\mathbf{x})$ are the outputs of the neural network. Here we assumed a diagonal covariance matrix, but we could also choose a full covariance matrix (details can be found in Kingma and Welling (2014)).

Choosing the distribution $q_{\theta}(\mathbf{z}|\mathbf{x})$ is fully linked to choosing the prior $p(\mathbf{z})$. The first VAE Kingma and Welling (2014) assumed an isotropic Gaussian prior $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, \mathbf{I})$. It is simple and no extra parameters must be learned. However, we could also use more sophisticated prior such as the Mixture of Gaussians Prior $p_{\lambda}(\mathbf{z}) = \sum_{k=1}^K w_k \mathcal{N}(\mathbf{z} | \mu_k, \sigma_k^2)$ with new trainable parameters $\lambda = \{w_k, \mu_k, \sigma_k^2\}$ for all mixture components k , or the VampPrior (Tomczak and Welling, 2018), which we refer the reader to the original paper for full details. The right choice of priors in VAEs is a whole field of research on its own (Chen et al., 2017; Gatopoulos and Tomczak, 2021).

Reparametrization trick By looking at the ELBO in Equation 2.11 we see that we need to calculate an integral with respect to the posterior $q_\phi(\mathbf{z}|\mathbf{x})$. However, we can instead take Monte Carlo samples. However, if you sample \mathbf{z} from $q_\phi(\mathbf{z}|\mathbf{x})$, calculate the ELBO and take gradients with respect to the parameters ϕ defined by the neural network, we observe that the variance of the gradient is very large. Indeed, this was pointed out by Rezende et al. (2014) and Kingma and Welling (2014). Hopefully, they came out with a solution which is probably one of the most significant contributions of the VAE framework. This solution, called *reparametrization trick* is based on reparametrizing the distribution $q_\phi(\mathbf{z}|\mathbf{x})$ using transformations of an independent random variable with a simple distribution (Devroye, 1996). That is, instead of expressing $\mathbf{z} \sim q_\phi(\mathbf{z} | \mathbf{x})$, we express \mathbf{z} as $\mathbf{z} = \mathbf{g}(\epsilon, \phi, \mathbf{x})$, where \mathbf{g} is a differentiable (invertible) transformation, *e.g.* sum, log, and the independent variable ϵ is independent of the input \mathbf{x} and the variational parameters ϕ . For example, we can choose

$$\mathbf{z} = \mu + \sigma \odot \epsilon \quad (2.15)$$

where $\epsilon \sim \mathcal{N}(\epsilon|0,1)$. With this trick we reduce the variance of the gradients because now the stochasticity is given by ϵ and therefore we can alleviate the computation of the gradients with respect to the parameters of the neural network, which are deterministic. For other variational distributions $q_\phi(\mathbf{z} | \mathbf{x})$ assuming different distributions from the Gaussian we would need to follow a similar procedure — for the categorical distribution, a common practice is to use the Gumbel Softmax reparametrization (Jang et al., 2017). Additionally, if the categorical variable has a high dimensionality, alternative techniques like the REINFORCE algorithm (Williams, 1992) can be employed to approximate the gradient.

Common problems in VAEs VAEs are very powerful models able to outperform other competitive state-of-the-art models in today’s machine learning literature. However, they also present some drawbacks. Here we briefly comment a few of these that we personally experienced when working with VAEs, or other authors pointed out in similar works. The most common problem is the so called *posterior collapse* (Bowman et al., 2015). As the name suggest, this problem occurs when the variational posterior collapses to the prior, *i.e.* $q_\phi(\mathbf{z}|\mathbf{x}) = p(\mathbf{z})$. If the decoder is very powerful, it will discard the information from \mathbf{z} , treating it as noise.

Another problem arises when the aggregated posterior (the average of the variational posteriors over all data points $q_\phi(\mathbf{z}) = \frac{1}{N} \sum_n q_\phi(\mathbf{z} | \mathbf{x}_n)$) and the prior $p(\mathbf{z})$. There might be regions in the latent space where the prior assigns high probability and the posterior assigns low probability. Or the opposite. This problem is called the *hole problem* (Rezende and Viola, 2018).

Lastly, we present the *out-of-distribution* (OOD) problem that appears in deep generative models in general. In Nalisnick et al. (2019), the authors test the following: they train a deep generative model on a dataset, MNIST for example, and then test on samples from other dataset, FashionMNIST for instance. Intuitively, we would expect that the model assigns more probability to MNIST samples and low probability to FashionMNIST samples. Paradoxically, this is not the case.

2.2 Handling Missing Data

The common trend in the machine learning literature is assuming we have fully observed data. But in real-world scenarios, this is rarely the case (Schafer and Graham, 2002; Rubin, 1976). Missing data might appear due to human errors, failure in measurement sensors, non-responses, *etc.* In scenarios where there are missing entries in our dataset, it is crucial to acknowledge our uncertainty about these missing values. Neglecting this uncertainty can potentially impact the performance of machine learning models and subsequent tasks reliant on these models. Hence, it becomes essential to conduct learning and inference while considering missing data and accurately quantify the uncertainties stemming from these missing values.

Notation Assume that for each data sample $\mathbf{x} \in \mathbb{R}^d$ it can be further decomposed into the observed values \mathbf{x}_o and the missing values \mathbf{x}_m . We can denote as \mathcal{O} the index set of observed values and \mathcal{M} the index set of missing values, such that $\mathcal{O}_t \cap \mathcal{M}_t = \emptyset$. Furthermore, we can define a binary mask vector \mathbf{m} which indicates whether a value in \mathbf{x} is observed $\mathbf{m}_d = 1$ or missing $\mathbf{m}_d = 0$. With this notation, we can define the missing mechanism with the conditional distribution $p(\mathbf{m}|\mathbf{x})$, where the dependency between \mathbf{m} and \mathbf{x} defines the following missing data assumption (Rubin, 1976):

- If $p(\mathbf{m}|\mathbf{x}) = p(\mathbf{m})$, the data is missing completely at random (MCAR).
- If $p(\mathbf{m}|\mathbf{x}) = p(\mathbf{m}|\mathbf{x}_o)$, the data is missing at random (MAR). That is, the cause of missingness \mathbf{m} is observed.
- Otherwise, the data is missing not at random (MNAR). That is, the cause of missingness is unobserved.

Most works often assume MCAR or MAR mechanism due to the simplicity. But in reality the MNAR assumption is more common: the failure of a sensor along time might be periodic, or the hospital record for a patient might have missing days because patients have doctor’s appointment with a certain frequency. Some works have already proposed solutions to the MNAR problem using DGM (Ma and Zhang, 2021; Collier et al., 2020). However, the ideal choice of the missing data mechanism is not straight-forward and new assumptions have been proposed recently. For instance, Berrevoets et al. (2023) propose a new mechanism called *mixed confounded missingness* (MCM) for treatment effect estimation where some missingness *determines* treatment selection and other missingness *I* by treatment selection. With these missing data mechanisms in mind, we can distinguish different techniques to handle missing data: 1) imputation/deletion techniques and 2) generative models for missing data.

2.2.1 Imputation and Deletion Techniques

The most naïve approach one could very first think is discard those samples with missing information: we could employ listwise deletion (Allison, 2001) or pairwise deletion (Marsh, 1998) to name a few. Obviously, these techniques induce a clear bias, specially for MAR and MNAR assumptions. The next intuitive procedure would be to replace the missing values with imputed values from certain methods. The most standard techniques are replacing with zero, mean, mode or other statistics. For regression problems we can also apply interpolation techniques, or backward/forward imputation in a temporal signal. We can also apply non-parametric methods based on k-nearest neighbors (Keerin et al., 2012), random forest (Stekhoven and Bühlmann,

2012), *etc.* These are *single-imputation* techniques because they only produce one imputed value for each data sample. Single-imputation techniques just produce a point estimate, therefore they do not quantify missing data uncertainty unfortunately. To account for this problem we can employ *multiple imputation* methods. The most used multiple-imputation method is the MICE (White et al., 2011), a procedure that imputes missing data through an iterative series of predictive models. In each iteration, each variable \mathbf{x}_d from the data is imputed using the other variables in the dataset until convergence.

2.2.2 Generative Models for Missing Data

The problem of missing data has been tackled before the recent explosion of DGM: the extension of the Expectation-Maximization (EM) algorithm with missing data (Ghahramani and Jordan, 1994) or the application of the EM with missing data in time series, (Bashir and Wei, 2018) to name a few. However, the flexibility and scalability of DGMs make them favorable for the missing data problem. The assumption is the following: if we have a sufficiently powerful generative model $p_{\theta}(\mathbf{x})$, then theoretically we could expect our inferred posterior over the unobserved data $p_{\theta}(\mathbf{x}_m|\mathbf{x}_o)$ should be accurate too. However, the usage of these methods posit two questions:

1. **Learning:** How can we estimate the optimal parameters θ given that we have partial observations \mathbf{x}_o ?
2. **Inference:** How can we quantify missing data uncertainty, that is, how can we calculate $p_{\theta}(\mathbf{z}|\mathbf{x}_o)$ and how can we impute $p_{\theta}(\mathbf{x}_m|\mathbf{x}_o)$?

The standard VAE framework does not consider missing data: we cannot incorporate samples with missing entries *per se* into the model. The most common missing assumption in VAEs handling missing data is MCAR because one can integrate out the missing variables from the ELBO and calculate it only on the observed variables. This approach was followed by Nazabal et al. (2020) and similarly by Mattei and Frellsen (2019), where they adapt the importance weighted autoencoder (IWAE) (Burda et al., 2015) to missing data. But the missing entries must be handled somehow: the encoder of the VAE usually requires a fixed-length input. One common step often used in the literature is to apply a *zero-filling* mechanism, where we replace the missing values with zero values (Vedantam et al., 2018; Nazabal et al., 2020; Mattei and Frellsen, 2019; Ma et al., 2019). This is a naive procedure that works well in practice, however there is one evident problem: there is no way to distinguish between a missing value and an observed value 0.

In order to circumvent this issue, other works have proposed different inference model mechanisms to handle the missing entries. In Vedantam et al. (2018) propose a product of Gaussian factorization in the inference model, requiring a separate encoder for each dimension \mathbf{x} . While more expressive, this approach does not scale well to higher dimensions, and one could argue that we lose the *beauty* of VAEs to elegantly capture the hidden correlations of the data within a common and shared latent space. Ma et al. (2019), within their proposed EDDI framework (Efficient Dynamic Discovery of high-value Information), yet propose a novel VAE architecture, called Partial VAE. The partial VAE draws inspiration from the *Point Net* approach for point cloud classification, and proposes a VAE encoder which is permutation invariant and depends only on \mathbf{x}_o , whose dimensionality may vary among different samples.

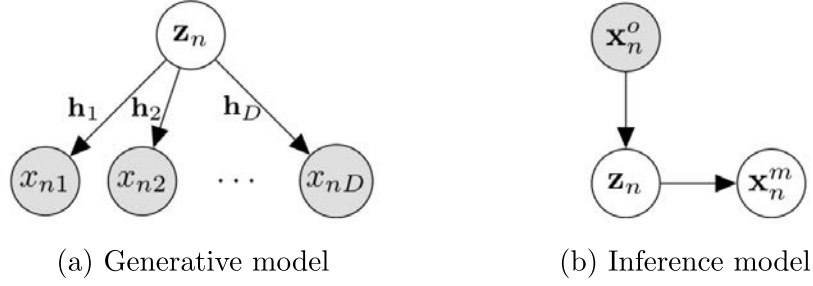


Figure 2.4: *Generative and Inference models for the HI-VAE:* In Figure (a) we show the generative model for the HI-VAE, where each dimension is model independently from a shared latent space \mathbf{z} . Figure (b) depicts the inference model where we see that \mathbf{z} only depends on the observed values \mathbf{x}_o , and the missing values \mathbf{x}_m only depend on \mathbf{z} . Figure adapted from [Nazabal et al. \(2020\)](#).

2.2.3 HI-VAE: Our Basis for Missing data Handling

In this thesis, our reference for will be the HI-VAE ([Nazabal et al., 2020](#)), and we will follow their proposed strategy to handle missing data. To get a better illustration of the problem, we will briefly present their approach. Again, we will be using a VAE. Hence, we need to define the likelihood $p_{\theta}(\mathbf{x}|\mathbf{z})$, the variational posterior $q_{\phi}(\mathbf{z}|\mathbf{x}_o)$ and the prior $p_{\theta}(\mathbf{z})$. As we commented above, we can split \mathbf{x} into and observed \mathbf{x}_o and missing \mathbf{x}_m partitions, which defines the following likelihood

$$p_{\theta}(\mathbf{x} | \mathbf{z}) = \prod_{d \in \mathcal{O}} p_{\theta}(x_d | \mathbf{z}) \prod_{d \in \mathcal{M}} p_{\theta}(x_d | \mathbf{z}). \quad (2.16)$$

The inference model from Figure 2.4 (b) also shows how the the latent space \mathbf{z} only depends on the observed values

$$q_{\phi}(\mathbf{z}, \mathbf{x}_m | \mathbf{x}_o) = q_{\phi}(\mathbf{z} | \mathbf{x}_o) \prod_{d \in \mathcal{M}} p_{\theta}(x_d | \mathbf{z}). \quad (2.17)$$

The objective we will use for training our VAE is therefore the ELBO computed only on the observed variables. For a sample $\mathbf{x} \in \mathcal{X}$, the ELBO on the observed variable \mathbf{x}_o has the following form

$$\begin{aligned} \log p_{\theta}(\mathbf{x}_o) &= \log \int p_{\theta}(\mathbf{x}_o, \mathbf{x}_m, \mathbf{z}) d\mathbf{z} d\mathbf{x}_m \\ &\geq \mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x}_o)} \left[\sum_{d \in \mathcal{O}} \log p_{\theta}(x_d | \mathbf{z}) \right] - D_{KL}(q_{\phi}(\mathbf{z} | \mathbf{x}_o) \| p_{\theta}(\mathbf{z})). \end{aligned} \quad (2.18)$$

Notice that the only difference with respect to the original ELBO from Equation 2.10 is that we marginalize over the observed variables. In [Nazabal et al. \(2020\)](#) they choose an isotropic prior for $p_{\theta}(\mathbf{z})$ and a Gaussian distribution for the inference model $q_{\phi}(\mathbf{z} | \mathbf{x}_o)$, where the input \mathbf{x} is transformed into a new $\tilde{\mathbf{x}}$ resulting from filling the missing entries with zeros. But the choice of the prior and the variational posterior is flexible. In the upcoming section, we discuss the use of a Gaussian prior mixture to address the challenge of handling heterogeneous data, as suggested in the HI-VAE paper. This approach, which is also employed in our own contribution outlined in Chapter 3, is a reasonable option for effectively tackling the issue.

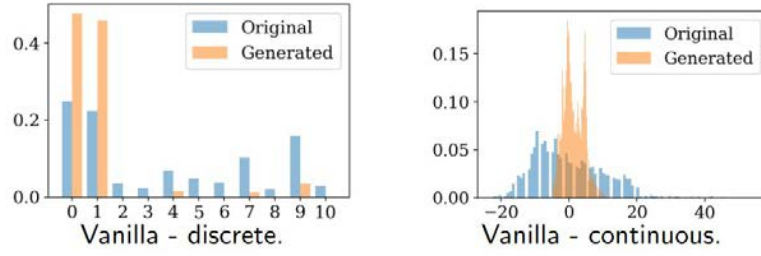


Figure 2.5: *Likelihood imbalance problem:* From the figure we can see how a Vanilla VAE with an heterogenous decoder trained on both discrete and continuous variables fails at properly fitting the data. Blue is the original distribution, and orange the generated distribution from the vanilla VAE. Figure courtesy of Adrián Javaloy, from this [tweet](#)⁴, and associated to [Javaloy et al. \(2022\)](#).

2.3 Handling Heterogeneous Data

When we think about *data* nowadays, we do not just think about the nature of the data, that is, whether a datum is a photo, a video, a document, an audio, or any other data format. We also think about the container of the data, which is the database. Hence, a database will be usually composed of different datums of different nature. For instance, imagine a hospital where, for each patient, we can have the Electronic Health Records (EHR) with medical different measurements, diagnosis, genomic information, electrocardiograms (EKG) or CT scans, just to name a few. But surprisingly, most of the literature has been primarily focused on modeling under the *homogeneous* assumption.

Therefore, without loss of generality, we can conclude that databases will usually be *heterogeneous*. Some related works in the literature also refer to this problem as *mixed-type* data problem, where we assume some data can have different statistical distributions at the same time (a tabular database with continuous and discrete variables), or also *multi-view* problem ([Damianou et al., 2012, 2021](#); [Guerrero-López et al., 2022](#)), where we can have access to different modalities of the data (a meeting can be defined by th audio and the video feed). While the literature of these two concepts complement each other, we will focus on the mixed-type data.

2.3.1 The Challenge of Modeling Heterogeneous Distributions

In an heterogeneous problem, we define our data sample $\mathbf{x} = [x_1, \dots, x_D]$, where each x_d will be modeled by a certain distribution depending on the data type: **continuous variables** which can be categorized into real-valued data ($x_d \in \mathbb{R}$), positive real-valued data ($x_d \in \mathbb{R}^+$), *etc* and **discrete data**, including categorical data, which comprises values in a finite unordered set ($x_d \in \{\text{'walking'}, \text{'sit'}, \text{'laying'}\}$) or other discrete distributions. Continuous and discrete distributions are defined in different domains that require different likelihood functions (*e.g.* Gaussian likelihoods for real-valued variables and Bernoulli likelihoods for binary variables). As a consequence, the impact of each likelihood on the training objective can vary significantly, resulting in complex optimization challenges ([Kendall et al., 2018](#)). Consequently, certain data dimensions may be inadequately represented in favor of others, or in other words, *likelihood imbalance* of certain distributions might appear, as shown in Figure 2.5

The most standard solution is obviously not worrying about this heterogeneity, and treat all variables either continuous or discrete, as Gaussian. Or we could also scale the likelihood of each

⁴<https://twitter.com/javaloyML/status/1536299712881500163?s=20>

data type in the ELBO to compensate for penalizations; however it is not intuitive how to do it properly. Luckily, some works focused on dealing with heterogeneity directly. One example is the work by [Valera et al. \(2020\)](#), where they proposed the general latent feature model (GLFM) suitable for heterogeneous datasets where the attributes describing each object can be either discrete, continuous or mixed variables. The proposed model extends the Indian Buffet Process (IBP), a Bayesian nonparametric latent feature model, to handle heterogeneous datasets. In [Valera and Ghahramani \(2017\)](#) they further propose a model that automatically detects the type of variable in a dataset. We find other interesting works for heterogeneous distributions in other machine learning areas. In the Gaussian process (GP) community, [Moreno-Muñoz et al. \(2018\)](#) extended a multi-output GP for handling heterogeneous outputs; in the change-point detection community [Romero-Medrano and Artés-Rodríguez \(2023\)](#) present a new change-point detection methodology with adaptive factorizations mechanisms that is able to handle multi-source observations with different statistical properties.

2.3.2 VAEs for Heterogeneous Data

While traditional latent variable models have dealt with the heterogeneous problem before ([Valera and Ghahramani, 2017](#); [Dhir et al., 2018, 2020](#)), deep generative models have been shown to outperform these methods.

Provably the first work to shed light on this topic was the Heterogeneous-Incomplete VAE (HI-VAE) ([Nazabal et al., 2020](#)) presented earlier. In this work, a part from proposing an approach to deal with missing data in VAEs, they also propose a VAE framework able to handle heterogeneous data. By using a factorized decoder where every dimension x_d is parametrized by an independent neural network (depicted in Figure 2.4(a)) and a common latent space which captures the underlying correlations among variables they are able to outperform state-of-the-art methods in missing data estimation. However, they also remark the importance of preventing certain dimensions to dominate the training when designing models for heterogeneous data. Further details for the decoder will be explained in next section.

This likelihood imbalance problem pointed out by [Nazabal et al. \(2020\)](#) in the HI-VAE was effectively confirmed by [Ma et al. \(2020a\)](#). To remedy this, the same authors propose a new model, the Variational Auto-encoder for heterogeneous mixed type data (VAEM) ([Ma et al., 2020b](#)). In this work, the authors present a new two-stage approach: in the first stage they train marginal VAEs for every variable independently, projecting into a Gaussian uni-dimensional space, and in the second stage they use a dependency VAE that fuses every marginal VAE using balanced Gaussian likelihoods. The VAEM is shown to achieve very competitive results with other models such as the HI-VAE in similar prediction tasks, and they also test in on sequential feature selection tasks. This model was later improved by [Peis et al. \(2022\)](#) with the HH-VAEM, an extension of the VAEM with a novel hierarchical VASE-based architecture that leverages the improved approximate inference of using Hamiltonian Monte Carlo methods. Also in a recent study by [Gong et al. \(2021\)](#), a VAE-based model called the variational selective autoencoder (VSAE) was introduced as a comprehensive framework for learning representations from partially observed heterogeneous data. VSAE tackles the challenge of capturing latent dependencies within such data by modeling the joint distribution of observed data, unobserved data, and an imputation mask that indicates the missing data patterns. This approach facilitates the learning of informative representations in the presence of missing or incomplete information, enabling a more comprehensive understanding of

heterogeneous datasets. The VAE, HH-VAE and VSAE emerge as interesting works because they tackle both the heterogeneous and the missing data problems.

These works have contributed to a significant increase of interest in this topic. Continuing with VAEs, Javaloy et al. (2022) conjecture that VAEs designed for multimodal tasks *e.g.* trained for both image and caption, often suffer from *modality collapse*, which happens when the VAE only focuses on a subset of modalities. *e.g.* fitting the image and neglecting the caption. This problem directly relates to the likelihood imbalance problem commented before. In Javaloy et al. (2021) they further analysed the feature overlooking symptom that VAEs suffer from, and relate it to the problem of *negative transfer* and gradient interaction using multitask learning (MTL) theory.

2.3.3 HI-VAE: Our Basis for Heterogeneous Data Handling

To cope with heterogeneous observations, we will also use the HI-VAE (Nazabal et al., 2020) as our basis framework. Hence, we proceed to explain how the Hi-VAE addresses the heterogeneous problem, which we will also use in our work in Chapter 3: 1) how the factorized decoder accommodates to heterogeneous observations, 2) how a applying a normalization per variable prevents some variables to dominate the training and 3) how a hierarchical model using a Gaussian mixture prior can facilitate obtaining a rich posterior and hence alleviate the heterogeneous problem.

1) Factorized Decoder

From Figure 2.4 (a) we can see how the decoder can be factorized in the following way

$$p_{\theta}(\mathbf{x}, \mathbf{z}) = p_{\theta}(\mathbf{z}) \prod_d p_{\theta}(x_d | \mathbf{z}) \quad (2.19)$$

where again $\mathbf{z} \in \mathbb{R}^K$ is the latent K -dimensional vector for \mathbf{x} , and $p_{\theta}(\mathbf{z}) = \mathcal{N}(\mathbf{z}|0, \mathbf{I}_K)$. The likelihood for each variable is parametrized as

$$p_{\theta}(x_d | \mathbf{z}) = p_{\theta}(x_d | \gamma_d = \mathbf{h}_d(\mathbf{z})) \quad (2.20)$$

where $\mathbf{h}_d(\cdot)$ is an independent DNN, one for each variable, that transforms the global latent variable \mathbf{z} into the specific domain for x_d . Next step is to define the likelihood functions that will be used to model continuous and discrete data types. For each data type then, we have the following parametrizations :

- **Real-valued data** (Normal): We assume a Gaussian likelihood

$$p(x_d | \gamma_d) = \mathcal{N}(x_d | \mu_d(\mathbf{z}), \sigma_d^2(\mathbf{z})), \quad (2.21)$$

where $\gamma_d = \{\mu_d(\mathbf{z}), \sigma_d^2(\mathbf{z})\}$ are the output of a DNN.

- **Positive real-valued data** (log-Normal): We assume a log-normal likelihood

$$p(x_d | \gamma_d) = \log \mathcal{N}(x_d | \mu_d(\mathbf{z}), \sigma_d^2(\mathbf{z})), \quad (2.22)$$

where $\gamma_d = \{\mu_d(\mathbf{z}), \sigma_d^2(\mathbf{z})\}$ are the output of a DNN corresponding to the mean and variance of the variable after applying the natural logarithm.

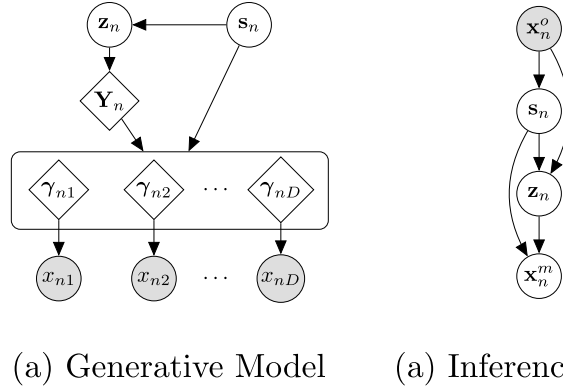


Figure 2.6: Overall HI-VAE generative and inference model: On Figure (a) we present the generative model described in Equation 2.26, where we can see the factorized decoder and how the parameters γ_d are generated for each statistical data type x_d . On figure (b) the inference model is presented following Equation 2.28. Figure from original paper (Nazabal et al., 2020).

- **Binomial (Bernoulli):** We assume a Bernoulli likelihood distribution, *i.e.*,

$$p(x_d|\gamma_d) = \text{Be}(p_d(\mathbf{z})), \quad (2.23)$$

and $\gamma_d = p_d(\mathbf{z}) = \sigma(\mathbf{h}_d(\mathbf{z}))$ is the probability parameter of the Bernoulli distribution and σ is the sigmoid function.

- **Categorical (Mult. logit):** We assume a multinomial likelihood distribution where the parameters of the likelihood are the C -dimensional output of a DNN with a log-softmax output

$$\log p(x_d = c|\gamma_{dc}) = h_d(\mathbf{z}_d)|_c \quad \text{for } c = [1, \dots, C]. \quad (2.24)$$

where $\gamma_d = \{h_{d0}(\mathbf{z}_d), \dots, h_{d(C-1)}(\mathbf{z}_d)\}$. To ensure identifiability, we fix the value of $h_{d0}(\mathbf{z})$ to zero.

where we changed notation from $\boldsymbol{\theta}$ to $\boldsymbol{\gamma}$ for each data type.

2) Data Normalization

Another strategy to prevent the likelihood imbalance problem is to implement a normalization technique on the input data before feeding it into the model, along with a corresponding denormalization process to map the learned values back to the original domain. In the case of real-valued data, a standard normalization technique is applied, which involves shifting and scaling the variables such that they have a mean of zero and a variance of one. The parameters for shifting and scaling, denoted as μ' and σ' , are then utilized for denormalization. *i.e.* $x_d \sim \mathcal{N}(x_d|\sigma'\mu_d(\mathbf{z}) + \mu', \sigma'^2\sigma_d^2(\mathbf{z}))$. For positive real-valued data, a similar approach is followed, but after applying the natural logarithm to the data. In the case of binary and categorical data, a one-hot encoding scheme is used. This normalization strategy ensures a more fair training between all the statistical data types.

3) Gaussian Mixture Prior

A vanilla VAE equipped with a simple continuous latent space might fail at handling both missing and heterogeneous observations. Namely, a standard Gaussian prior $p(\mathbf{z})$ might be too restrictive to model complex and high-dimensional data (Tomczak and Welling, 2018). Plus, the factorized decoder of the HI-VAE loses the properties of an amortized decoder where each dimension x_d is parametrized by the same DNN. To overcome this issues, Nazabal et al. (2020) propose to use a Gaussian mixture prior (Li et al., 2019b) of the form

$$\begin{aligned} p_{\theta}(\mathbf{s}) &= \text{Categorical}(\mathbf{s} | \boldsymbol{\pi}) \\ p_{\theta}(\mathbf{z} | \mathbf{s}) &= \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}_{\theta}(\mathbf{s}), \mathbf{I}_K) \end{aligned} \quad (2.25)$$

where \mathbf{s} is the one-hot encoding indicating the component that generates \mathbf{z} , and $\pi_l = 1/L, l = 1, \dots, L$ for L Gaussian components.

Generative Model As depicted in Figure 2.6, the corresponding joint probability would be

$$p_{\theta}(\mathbf{x}, \mathbf{z}, \mathbf{s}) = p_{\theta}(\mathbf{s}) p_{\theta}(\mathbf{z} | \mathbf{s}) \prod_d p_{\theta}(x_d | \gamma_d = h_d(\mathbf{y}_d, \mathbf{s})) \quad (2.26)$$

where $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_D] = \mathbf{g}(\mathbf{z})$ defines a hierarchical structure that allows the generative model to share parameters across different data types from $\mathbf{g}(\cdot)$, and therefore favor the capturing of hidden correlations between variables. This is also depicted in Figure 2.6.

Inference Model From Figure 2.6 we notice that the variational distributions are formulated as follows

$$\begin{aligned} q_{\phi}(\mathbf{s} | \mathbf{x}^o) &= \text{Categorical}(\mathbf{s} | \boldsymbol{\pi}_{\phi}(\tilde{\mathbf{x}})) \\ q_{\phi}(\mathbf{z} | \mathbf{x}^o, \mathbf{s}) &= \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}_{\phi}(\tilde{\mathbf{x}}, \mathbf{s}), \boldsymbol{\Sigma}_{\phi}(\tilde{\mathbf{x}}, \mathbf{s})). \end{aligned} \quad (2.27)$$

where $\tilde{\mathbf{x}}$ is the sample \mathbf{x} with the missing values filled by zeros. $q_{\phi}(\mathbf{s} | \mathbf{x}^o)$ is a categorical distribution with probability parameters $\boldsymbol{\pi}_{\phi}$ coming from a DNN, and $q_{\phi}(\mathbf{z} | \mathbf{x}^o, \mathbf{s})$ follows a Gaussian distribution where the parameters are given by two distinct DNNs whose input is the concatenation of $\tilde{\mathbf{x}}$ and \mathbf{s} . Assuming that the missing values \mathbf{x}^m are conditionally independent on the observed attributes \mathbf{x}^o , we finally write the whole variational distribution as

$$q_{\phi}(\mathbf{s}, \mathbf{z}, \mathbf{x}^m | \mathbf{x}^o) = q_{\phi}(\mathbf{s} | \mathbf{x}^o) q_{\phi}(\mathbf{z} | \mathbf{x}^o, \mathbf{s}) \prod_{d \in \mathcal{M}} p_{\theta}(x_d | \mathbf{z}, \mathbf{s}), \quad (2.28)$$

where \mathcal{M} denotes the missing attributes for \mathbf{x} . Finally, the training of the HI-VAE model is done applying the ELBO marginalizing out the missing observations as described in Equation 2.18, but now incorporating the discrete latent space \mathbf{s} . To draw samples from $q(\mathbf{s} | \mathbf{x}^o)$ we apply the Gumbel-softmax reparametrization trick (Li et al., 2019b). We will follow the same procedure in our model in Chapter 3.

2.4 Handling Temporal Data

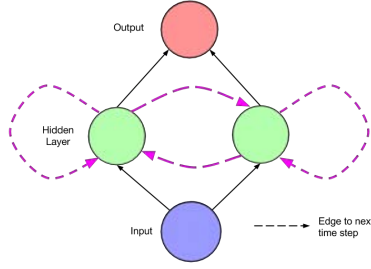
Before, we commented that the first common assumption about data is assuming fully observed data, and talked about how can we deal with missing data in the learning process. We continued presenting the next common assumption that data is assumed homogeneous, when in real-case scenarios this will rarely be the case, and how assuming heterogeneous likelihood is a very challenging problem. Lastly, we present the last common assumption: assuming that data is *independent*.

The independence assumption is arguably one of the first assumptions that any statistician (or machine learning researcher in our case) would make about the data, *i.e.* assuming independent and identically distributed (I.I.D) samples. Not surprisingly, even model assuming this complete independence are able to implicitly capture inner dependencies within the data *out-of-the-box*. Furthermore, perhaps this assumptions is one of the cornerstones of the recent machine learning succeed. But we can always do better. For instance, take an image classification problem. We could always assume independence among pixels, but convolutional neural networks (CNN) supposed a milestone in this field because they explicitly capture the spatial correlation between the pixels of the image (among other things, *of course*). That is why Recurrent Neural Networks (RNN) also supposed a milestone in those problems where the data follows a temporal distribution and we cannot assume independence between samples.

Obviously, there has been a vast amount of investigation in how can we properly model sequential data before the RNN (re)appearance. Provably one of the top competitors of RNNs are Markov models, which model transitions between states. Among Markov models, Hidden Markov Models (HMM) are one of the most used, because they model observations based on the transition between hidden states with a probabilistic dependence. These states are drawn from a discrete state S , and the transition between time-adjacent states is defined by a probability table of size $|S|^2$. For an increasing number of possible states the operations for HMMs becomes and feasible. And while it is true that the dependency along time can be made larger, usually the order of the HMM is set to 1, the previous state. All these limitations motivated us to use RNNs instead.

2.4.1 RNNs: Our Toolkit for Temporal Data

RNN Although there is a widespread belief that RNNs emerged contemporary to the massive usage of neural networks, many early works were already investigating about recurrent neural networks (Hopfield, 1982; Jordan, 1986; Elman, 1990). However, using RNNs for learning in temporal scenarios has always been considered difficult. RNNs offer the advantage of being fully differentiable from end to end when considering a fixed architecture comprising nodes, edges, and activation functions. This allows us to calculate gradients with respect to the RNN weights and utilize gradient-based optimization techniques. However, dealing with long-range dependencies becomes challenging (Bengio et al., 1994; Hochreiter et al., 2001). When dealing with very long temporal series, the propagation of gradients throughout the network can lead to two distinct issues. Firstly, the gradients may diminish and approach zero, known as the *vanishing* gradient problem. Alternatively, the gradients can become excessively large, escalating continuously, which is referred to as the *exploding* gradient problem. In Figure 2.7a we include a diagram for a simple RNN.



(a) RNN. Figure extracted from Lipton et al. (2015).

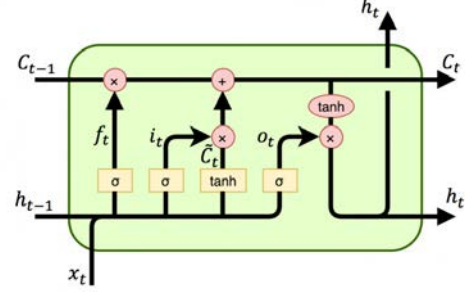
(b) LSTM. Figure extracted from this website⁵.

Figure 2.7: *Basic RNN model and LSTM cell:* In Figure (a), we depict a basic RNN, where at every time step t , the activation is propagated through solid edges, similar to a feedforward network. Dashed edges establish connections from a source node at each time t to a target node at the subsequent time $t + 1$. Figure (b) depicts a LSTM cell.

The RNN general formula can be described as follows. For a given sequence $\mathbf{x}_{1:T} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$, $\mathbf{x}_t \in \mathbb{R}^d$, an RNN can recursively update its internal hidden state \mathbf{h} with the following equation

$$\mathbf{h}_t = f_\theta(\mathbf{x}_t, \mathbf{h}_{t-1}), \quad (2.29)$$

where f can be any non-linear function with its corresponding parameters θ and $\mathbf{h}_t \in \mathbb{R}^k$, with k is usually lower-dimensional. In the following we will see how f_θ is implemented for the case of the LSTM.

LSTM In a successful attempt to mitigate these problem, Hochreiter and Schmidhuber (1997) proposed a novel RNN architecture, the so called Long Short-Term Memory (LSTM). The LSTM addresses the vanishing and exploding gradient problems by introducing specialized memory cells. It uses a sophisticated gating mechanism to regulate the flow of information within the network, allowing it to selectively remember or forget information over varying time intervals. The LSTM cell is defined by the following equations:

$$\begin{aligned} \mathbf{i}_t &= \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i) && \text{(Input Gate)} \\ \mathbf{f}_t &= \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f) && \text{(Forget Gate)} \\ \mathbf{o}_t &= \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_{t-1} + \mathbf{b}_o) && \text{(Output Gate)} \\ \tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c) && \text{(Candidate Memory Cell)} \\ \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t && \text{(Memory Cell)} \\ \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t) && \text{(Hidden State)} \end{aligned}$$

In these equations, \mathbf{x}_t represents the input at time step t , \mathbf{h}_t is the hidden state at time step t , \mathbf{c}_t denotes the memory cell state at time step t , σ represents the sigmoid activation function, \odot denotes element-wise multiplication, and \mathbf{W} and \mathbf{b} represent the weight and bias parameters, respectively, for the different gates and cell operations in the LSTM architecture. These equations

⁵<http://dprogrammer.org/rnn-lstm-gru>

can be better understood with Figure 2.7b. Contemporary to Hochreiter and Schmidhuber (1997), (Schuster and Paliwal, 1997) also proposed a bidirectional RNN to capture temporal dependencies from past and future states.

While there is an immeasurable amount of works in the line the LSTM (*e.g.* the Gated Recurrent Unit (Cho et al., 2014)), we decide to stick to the basic LSTM as our building toolkit for modeling temporal dependencies, and we will use it later in Chapter 3.

2.4.2 RNNs Handling Missing Data in Medical Context

It is not surprising that one of the principal areas where RNNs haven been tested is in health care applications. Medical information typically exhibits a temporal nature, either inherent in the type of data, such as certain lab measurements, or due to the frequency at which a patient visits the hospital. Therefore, RNNs emerge as a significant tool for capturing the temporal correlations from medical data. However, we know from previous sections that medical data, among from other artifacts, is often corrupted by missing data. This motivates the search of new ways to incorporate missing data into standard RNN architectures. In the following we will present a few works that inspired our work presented in Chapter 3, not only because they serve as reference works for handling missing data in temporal series, but also because their primary motivation is the direct healthcare application.

In a very first work, Lipton et al. (2016a) analyzed the effectiveness of LSTMs to recognize patterns in multivariate time series of clinical measurements. But it is in their follow-up paper where Lipton et al. (2016b) treat the missing data problem. They acknowledge what we have been commenting before about medical data: the irregular spacing between measurements leads to missingness patterns in temporally discretized sequences. Instead of imputing the missing values prior to predictive step, they treat the missing artifacts as binary features. These binary features can be viewed as indicator variables or *gating* variables, and they show that knowing which measurement tests are can be as predictive as the results themselves.

In a similar fashion, Che et al. (2018) also exploit the missing patterns present in the data and they propose a novel deep learning model, namely GRU-D, based on the GRU model. This model makes use of both the binary missing mask and the time interval between missing values and incorporates them into the GRU. This can be viewed as a next step from just incorporating the missing mask as a binary indicator, because now we also allow the model to know *how long* the last missing happened. Concurrently, Cao et al. (2018) propose BRITS, a model that handles both the imputation and the prediction at the same time using bidirectional RNNs. In Luo et al. (2018), they tackle the missing data imputation problem using Generative Adversarial Networks (GAN) and in the same spirit as Che et al. (2018), they again propose a new modified version of a GRU, namely GRUI, which incorporates the time lags between missing values to model the decaying influence of past observations when a value has been missing for a while.

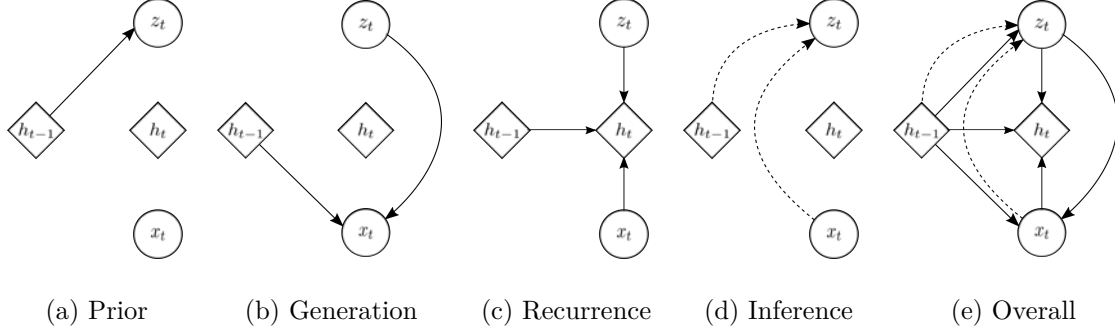


Figure 2.8: VRNN model with the corresponding steps: (a) Prior with Equation 2.31; (b) Generative model with Equation 2.32; (c) RNN hidden state updates with Equation 2.30; (d) Inference model with Equation 2.34; (e) overall VRNN graphical model. Figure from original paper (Chung et al., 2015).

2.4.3 VRNN: Our Basis for Temporal Data Handling

While the works commented above are of great interest for our problem, we would like to incorporate the temporal modeling using RNNs into our VAE learning framework presented above. Before we were commenting that a latent variable model allows to capture dependencies between the data withing a lower dimensional space. Now, we want to model the inner correlations within the data *across time steps* too.

Previous work already integrated stochasticity into the hidden states of the RNNs (Fabius and Van Amersfoort, 2014; Boulanger-Lewandowski et al., 2012) or proposed generative models for sequential data under the VAE framework (Fraccaro et al., 2016), we will mainly look at the work by Chung et al. (2015) and their proposed model Variational RNN (VRNN). Motivated by the succeed and flexibility of VAEs on modeling and capturing data variability for non-sequential data, they extend VAEs using a recurrent framework where the latent space at a given instant \mathbf{z}_t does not only depend on the data at that given time \mathbf{x}_t but also on the previous time $\mathbf{x}_{<t}$. The latent space captures the temporal dependencies from the data and propagates it along time.

Generative Model

In order to extend the a standard VAE to handle temporal data in a simple way, we present the approach from VRNN (Chung et al., 2015).

$$\text{RNN State Update: } \mathbf{h}_t = f_\theta(\varphi_\tau^{\mathbf{x}}(\mathbf{x}_t), \varphi_\tau^{\mathbf{z}}(\mathbf{z}_t), \mathbf{h}_{t-1}), \quad (2.30)$$

$$\begin{aligned} \text{Prior Distribution: } \mathbf{z}_t &\sim \mathcal{N}(\boldsymbol{\mu}_{0,t}, \text{diag}(\boldsymbol{\sigma}_{0,t}^2)) \\ &\text{, where } [\boldsymbol{\mu}_{0,t}, \boldsymbol{\sigma}_{0,t}] = \varphi_\tau^{\text{prior}}(\mathbf{h}_{t-1}), \end{aligned} \quad (2.31)$$

$$\begin{aligned} \text{Generative Distribution: } \mathbf{x}_t | \mathbf{z}_t &\sim \mathcal{N}(\boldsymbol{\mu}_{x,t}, \text{diag}(\boldsymbol{\sigma}_{x,t}^2)) \\ &\text{, where } [\boldsymbol{\mu}_{x,t}, \boldsymbol{\sigma}_{x,t}] = \varphi_\tau^{\text{dec}}(\varphi_\tau^{\mathbf{z}}(\mathbf{z}_t), \mathbf{h}_{t-1}), \end{aligned} \quad (2.32)$$

Let's analyze the equations from above. First, the RNN updates follows the recurrence equation where $\varphi_\tau^{\mathbf{x}}(\mathbf{x}_t)$ and $\varphi_\tau^{\mathbf{z}}(\mathbf{z}_t)$ are feature extractors for \mathbf{x}_t and \mathbf{s}_t respectively. The function f_θ is the same as Equation 2.29. Next, the prior distribution is not an isotropic Gaussian like in the standard VAE, since it needs to account for the temporal structure. This is achieved through $\varphi_\tau^{\text{prior}}$, which generates the Gaussian parameters based on the previous RNN state \mathbf{h}_{t-1} . Similarly for \mathbf{x}_t , the parameters of the Gaussian distribution come from a NN $\varphi_\tau^{\text{dec}}$ with the current feature from the latent variable $\varphi_\tau^{\mathbf{z}}(\mathbf{z}_t)$ and the previous RNN state \mathbf{h}_{t-1} .

The final joint distribution for all time instants $t = [1, \dots, T]$ has the following form:

$$p_{\theta}(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T}) = \prod_{t=1}^T p_{\theta}(\mathbf{x}_t | \mathbf{z}_{\leq t}, \mathbf{x}_{< t}) p_{\theta}(\mathbf{z}_t | \mathbf{x}_{< t}, \mathbf{z}_{< t}), \quad (2.33)$$

where $\mathbf{x}_{< t} = \mathbf{x}_{[1:t-1]}$ and $\mathbf{x}_{\leq t} = \mathbf{x}_{[1:t]}$. The same applies for \mathbf{z}_t . Notice that $p_{\theta}(\mathbf{x}_t | \mathbf{z}_{\leq t}, \mathbf{x}_{< t})$ refers to the likelihood of the current instant \mathbf{x}_t and depends on the past information ($< t$) and the current latent code \mathbf{z}_t . And the prior $p_{\theta}(\mathbf{z}_t | \mathbf{x}_{< t}, \mathbf{z}_{< t})$ depends *only* on all the past information ($< t$). The key point from this model is that we recover the past information at t using the a recurrent neural network — the RNN remembers the temporal dependencies encapsulated in the latent space.

Inference Model

Similarly, the variational distribution must also account for the temporal structure of the data. Therefore we can define it as

$$\mathbf{z}_t | \mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_{z,t}, \text{diag}(\boldsymbol{\sigma}_{z,t}^2)), \text{ where } [\boldsymbol{\mu}_{z,t}, \boldsymbol{\sigma}_{z,t}] = \varphi_{\tau}^{\text{enc}}(\varphi_{\tau}^{\text{x}}(\mathbf{x}_t), \mathbf{h}_{t-1}), \quad (2.34)$$

where the Gaussian parameterse $\boldsymbol{\mu}_{z,t}$ and $\boldsymbol{\sigma}_{z,t}$ are the output of a NN $\varphi_{\tau}^{\text{enc}}$ that takes as input the features from \mathbf{x}_t (*i.e.* $\varphi_{\tau}^{\text{x}}(\mathbf{x}_t)$) and the previous RNN state \mathbf{h}_{t-1} . The variational distribution for all time instants follows the factorization

$$q_{\phi}(\mathbf{z}_{\leq T} | \mathbf{x}_{\leq T}) = \prod_{t=1}^T q_{\phi}(\mathbf{z}_t | \mathbf{x}_{\leq t}, \mathbf{z}_{< t}) \quad (2.35)$$

Learning

With all these ingredients, the final ELBO used for training the VRNN model looks like this


$$\mathbb{E}_{q_{\phi}(\mathbf{z}_{\leq T} | \mathbf{x}_{\leq T})} \left[\sum_{t=1}^T \log p_{\theta}(\mathbf{x}_t | \mathbf{z}_{\leq t}, \mathbf{x}_{< t}) - \text{KL}(q_{\phi}(\mathbf{z}_t | \mathbf{x}_{\leq t}, \mathbf{z}_{< t}) || p_{\theta}(\mathbf{z}_t | \mathbf{x}_{< t}, \mathbf{z}_{< t})) \right] \quad (2.36)$$

All the update functions and the overall VRNN model are depicted in Figure 2.8. For further details, we refer the reader to the original paper (Chung et al., 2015) and advance that this work will serve as our reference work for handling the temporal dependencies in Chapter 3.

2.5 Summary of the Chapter

In this chapter we introduced VAEs, the learning framework which will be used in Chapter 3 for performing medical data wrangling. We also presented the three main problems covered in this thesis when working with irregular observations: missing data, heterogeneous observations and temporal data. For each problem we presented some background and related work, followed by the basis of our work, namely:

- To tackle the handling of **missing** and **heterogeneous** data , we introduced the HI-VAE (Nazabal et al., 2020) model, which will be used as our reference work.
- To tackle the handling **temporal** data, we introduced the VRNN (Chung et al., 2015) which will be used as our reference work.

I can't pin it
I can't pin it down
I can't pin it
But I think we've been here once before
I think we've been here once before
 Pin It Down — *Madison Cunningham* 

3

Medical Data Wrangling With Sequential Variational Autoencoders

Contents

3.1	Introduction	36
3.2	A human monitoring database	38
3.3	Proposed Model	39
3.3.1	Notation	39
3.3.2	The Sequential Heterogeneous Incomplete VAE (Shi-VAE)	40
3.3.3	Heterogeneous Decoder	41
3.3.4	Model Training with Variational Inference	42
3.3.5	The GP-VAE Probabilistic Model	43
3.4	Experimental Results	44
3.4.1	Evaluation Metrics	45
3.4.2	Synthetic Data set	46
3.4.3	Physionet	48
3.4.4	Human Monitoring Database	49
3.5	Discussion	50

MEDICAL data sets are usually corrupted by noise and missing data. These missing patterns are commonly assumed to be completely random, but in medical scenarios, the reality is that these patterns occur in bursts due to sensors that are off for some time or data collected in a misaligned uneven fashion, among other causes. This paper proposes to model medical data records with heterogeneous data types and bursty missing data using sequential variational autoencoders (VAEs). In particular, we propose a new methodology, the Shi-VAE, which extends the capabilities of VAEs to sequential streams of data with missing observations. We compare our model against state-of-the-art solutions in an intensive care unit database (ICU) and a dataset of passive human monitoring. Furthermore, we find that standard error metrics such as RMSE are not conclusive enough to assess temporal models and include in our analysis the cross-correlation between the ground truth and the imputed signal. We show that Shi-VAE achieves the best performance in terms of using both metrics, with lower computational complexity than the GP-VAE model, which is the state-of-the-art method for medical records.

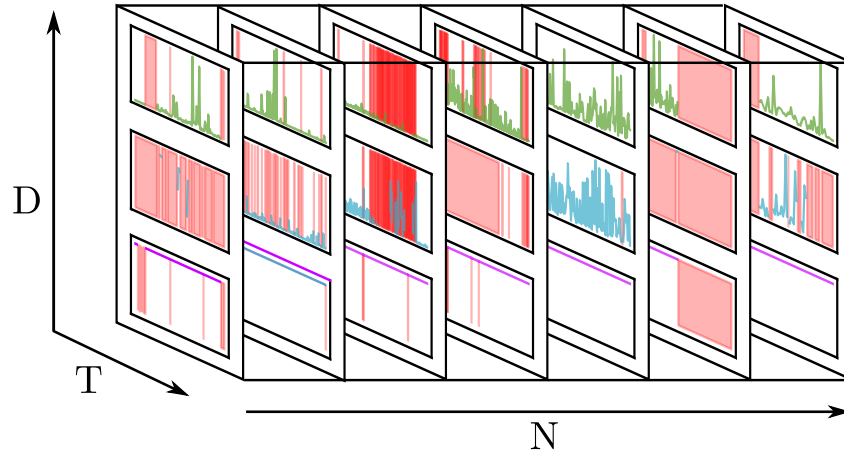


Figure 3.1: Example of heterogeneous streams of data with missing values from the medical data set. Red vertical lines correspond to missing values. Each row corresponds to a different type of data: the first two correspond to positive real-valued data and the third to binary data. D refers to dimensionality of the dataset, T to the temporal dimension and N to the number of samples.

Comments on contributions The results described in this chapter result from two main contributions. First, some preliminary results were shared at the [Bayesian Deep Learning workshop](http://bayesiandeeplearning.org/2020/index.html)¹ in the conference on Neural Information Processing Systems (NeurIPS) in 2020. Second, the final work ([Barrejón et al., 2021](#)) was accepted at the IEEE Journal of Biomedical and Health Informatics.

3.1 Introduction

Since machine learning emerged, all the primary attention focused on working with homogeneous data sets, where too few artifacts such as outliers or missing data barely appear. But real-world data sets are quite different. Data is usually organized in databases containing incomplete, noisy, and more critical, heterogeneous information sources. These scenarios are quite common in medical applications. For instance, Electronic Health Records (EHR) may contain information from monitoring sensors, different physicians' diagnoses, or visits to the hospital. A heterogeneous medical footprint hence defines each patient. This kind of information will exhibit missing data due to sensors' failures or due to temporal gaps between each visit to the hospital, to name a few.

In the literature, the common assumption is that the lost information from a data set is Missing Completely at Random (MCAR). However, the most usual scenario is that missing data follows some kind of pattern. For example, in human monitoring applications the sensors tracking different sources might disconnect for some amount of time, not intermittently, generating *bursts of missing data*. For medical data sets missing patterns can appear simultaneously across different attributes as it is shown in Figure 3.1.

The recent literature on machine learning (ML) approaches to handle noise and missing data in medical records is dominated by deep learning methods. In this regard, recurrent neural networks (RNN) stand as one of the most popular approaches. In [Lipton et al. \(2016a\)](#) the authors propose Long-Short-Term Memory (LSTM) networks ([Hochreiter and Schmidhuber, 1997](#)), to recognize patterns in multivariate time series of clinical measurements. This work was extended in [Lipton et al. \(2016b\)](#) with binary indicators of missingness as features. A different approach is proposed

¹<http://bayesiandeeplearning.org/2020/index.html>

in Che et al. (2018), where Gated Recurrent Units (GRU) are modified to incorporate missing masks, hence modeling the time intervals between clinical appointments. Other works like BRITS (Cao et al., 2018) also look into the bidirectional capabilities of RNNs and exploit this property to impute missing values in time series with underlying nonlinear dynamics.

Although the above RNN-based methods show impressive results dealing with time series forecasting, they do not benefit from the flexibility and the underlying data correlations inferred by probabilistic deep generative models (DGMs). DGMs capture inner correlations that can be present in high-dimensional data employing a low-dimensional latent space. In the framework of VAEs, the heterogeneous incomplete variational autoencoder (HI-VAE) (Nazabal et al., 2020), the mixed VAE (VAEM) (Ma et al., 2020b), the MIWAE (Mattei and Frellsen, 2019), the Partial VAE presented in Ma et al. (2019) or similar works (Collier et al., 2020; Qiu et al., 2020) propose efficient methods to jointly model different data types and missing data in a single DGM. Among DGMs able to deal with sequential data, GP-VAE (Fortuin et al., 2020) stands out. GP-VAE implements a latent probabilistic model in which a Gaussian process captures the correlation of the low-dimensional latent variable along time, and this GP relies on a VAE to implement the observation model. However, GP-VAE cannot deal with heterogeneous observations. Finally, DGM-like solutions to deal with tabular or sequential based on generative adversarial networks (GANs), such as GAIN in Yoon et al. (2018), the gated recurrent GAN in Luo et al. (2018), MisGAN in Li et al. (2019a) and VIGAN (Shang et al., 2017) do not show to outperform the imputation ability of other VAE-based methods and are harder to train due to the min-max underlying optimization problem.

In this paper, we consider modeling sequential heterogeneous data when missing data comes in bursts, a scenario in which none of the previous DGMs have been tested to date. On the one hand, we show that when errors come in bursts, standard error metrics such as normalized mean-squared error (NRMSE) do not reflect well the imputation accuracy, and we study the correlation between the ground-truth signal and the imputed one. In this setup, we demonstrate that GP-VAE struggles to deal with long-missing data bursts since the underlying GP correlation quickly decays, driving the GP posterior to a non-informative mean and large variance.

To better deal with bursty missing patterns, we propose the sequential heterogeneous incomplete VAE (Shi-VAE). This model generalizes the HI-VAE model in Nazabal et al. (2020) presented in Chapter 2.3.3, including a latent temporal structure driven by LSTMs following a similar idea as in Chung et al. (2015), presented in Chapter 2.4.3. The extended memory properties of these networks provide a more robust ability to cope with missing bursts, efficiently capturing into the low-dimensional latent projection the correlation to past observations. Besides, Shi-VAE comes with efficient training methods based on amortized variational inference that can handle massive data sets. As a representative example of a medical database, we demonstrate the superior ability of Shi-VAE to deal with complex time-series using two real data sets. First, we consider the data set from the 2012 Physionet Challenge (Silva et al., 2012) which contains measurements of 35 electrophysiological signals for 12,000 patients monitored during 48 on the intensive care unit (ICU). Second, we consider a data set of human passive monitoring coming from mobile devices. It contains heterogeneous attributes (distance travelled, mobile phone usage, quality of sleep, etc.) and a challenging presence of bursty missing data. The Shi-VAE code to reproduce our experiments can be found in <https://github.com/dbarrejon/Shi-VAE>. Overall, we claim the following contributions:

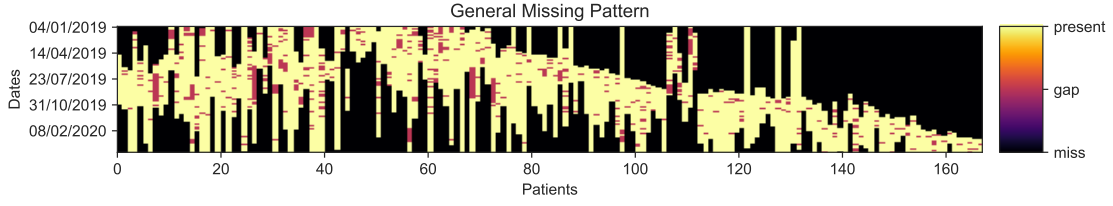


Figure 3.2: Overall view of the human monitoring database. Each patient has a given sequence length. Black means no record of that patient, magenta means a complete missing day and yellow that at least there is one variable present at that day.

- We propose Shi-VAE as a robust generative model to handle heterogeneous time series corrupted with missing data.
- We demonstrate that NRMSE is a partial metric when it comes to compare imputation models in the presence of missing data in bursts.
- We propose to use a temporal correlation metric to compare the different models. This metric is more sensitive to detect over-smooth solutions.

We organize the paper as follows. Firstly, Section 3.2 introduces the problem statement we want to tackle. Section 3.3 presents Shi-VAE. In Section 3.4 we present the two data sets we have used to validate our model and the results we have found. Section 3.5 presents our final remarks.

3.2 A human monitoring database

Through patients' mobile phones and other wearable devices, continuous sensor data can be collected in a non-invasive manner, providing valuable information about everyday activity patterns. The possibility of inferring emotional states by analyzing smartphone usage data LiKamWa et al. (2013), Mehrotra et al. (2017), GPS traces of movement Canzian and Musolesi (2015), social media data De Choudhury et al. (2013), and even sound recordings Lu et al. (2012) has become a growing research focus over the past decade.

One of the databases that we use in this paper was collected using the mobile application eB2 MindCare² in a collaboration we carried out with two public mental health hospitals in Madrid (Hospital Universitario Fundación Jiménez Díaz and Hospital Universitario Rey Juan Carlos). This study was approved by the Fundación Jiménez Díaz Research Ethics Committee (Study code: LSRG-1-005 16). We periodically capture passive monitoring information from $N = 170$ psychiatric patients using eB2 MindCare, thus registering different signals for every user. In particular, we are working with daily summary representations of every variable. The seven attributes we work with are listed in Table 3.1, along with the fraction of missing values across all patients.

Regarding the positive variables, distance, steps total, and vehicle are related to the patient's mobility. App usage is a positive variable that measures the total amount of active time the user has been using the phone, with social applications, phone calls, etc. Sleep is a positive variable that counts the total time a person has slept during a day. Regarding binary variables, sport explains whether the person has done any sport $x_t = 1$ or not $x_t = 0$ during the day and steps home states whether the person was at home $x_t = 1$ or not $x_t = 0$ at that particular day.

²Available at: <https://eb2.tech/>

Variable	Type	Missing Percentage [%]
Distance	Positive	42
Steps Home	Binary	66
Steps Total	Positive	22
App Usage	Positive	38
Sport	Binary	62
Sleep	Positive	31
Vehicle	Positive	44

Table 3.1: Human Monitoring data set.

Finally, we remark that, although the number D of attributes is the same for every patient ($D = 7$), the signal length T per patient is very diverse. The average sequence length is 233. Figure 3.2 illustrates the whole population and the missing pattern. From the Figure 3.2 we can observe that almost any day comes with missing values, and hence we can expect long bursts of missing attributes.

In this paper, we demonstrate the superior ability of the proposed Shi-VAE to capture the non-trivial correlations among the database attributes and accurately impute missing values.

3.3 Proposed Model

We first introduce a general notation of the problem and then present the Shi-VAE model.

3.3.1 Notation

We define our data set as $\mathcal{D} = \{\mathbf{X}^1, \dots, \mathbf{X}^N\}$, where N corresponds to the total number of samples in the data set. Each sample $\mathbf{X}^n \in \mathbb{R}^{T^n \times d}$ has T^n observations $\mathbf{x}_t = [x_{t1}, \dots, x_{td}]^\top \in \mathbb{R}^d$, where d refers to the dimension or attribute. From now on, we use $\mathbf{X}^n = \mathbf{X}$ in order to relax notation. We consider heterogeneous attributes:

- **Continuous Variables:**

1. **Real-valued data:** Data taking real values, *i.e.*, $x_{td} \in \mathbb{R}$.
2. **Positive-valued data:** Data taking only positive values, *i.e.*, $x_{td} \in \mathbb{R}^+$.

- **Discrete Variables:**

1. **Binary Data:** Data can only be either 1 or 0, *i.e.*, $x_{td} \in [0, 1]$.
2. **Categorical data:** Data taking values in a finite unordered set, *i.e.*, $x_{td} \in \{-1, 0, 1\}$, or $x_{td} \in \{\text{'negative', 'neutral', 'positive'}\}$.

Furthermore, we assume that any \mathbf{x}_t can have both observed values and missing values. Let us define \mathcal{O}_t as the index set for the observed attributes at time t and \mathcal{M}_t as the missing index at the same time. Hence $\mathcal{O}_t \cap \mathcal{M}_t = \emptyset$. With this notation, we can split this sentence into a vector containing observed attributes \mathbf{x}_t^o , and a complementing vector containing missing attributes \mathbf{x}_t^m .

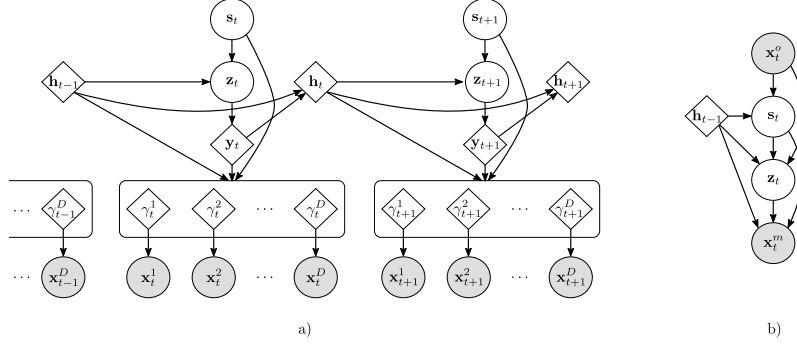


Figure 3.3: On a), Shi-VAE generative model. On b), Shi-VAE inference model.

3.3.2 The Sequential Heterogeneous Incomplete VAE (Shi-VAE)

This section presents the Shi-VAE probabilistic generative model, which extends the capabilities of a standard VAE to sequential heterogeneous data streams and handles missing data. In Shi-VAE, the temporal dependencies and shared correlations among attributes are captured by a latent hierarchy of low-dimensional latent variables: a continuous latent variable $\mathbf{z}_t \in \mathbb{R}^K$, which follows a Mixture of Gaussian's (MoG) Prior distribution (Dilokthanakul et al., 2016), and a discrete latent variable \mathbf{s}_t that represents the component of the MoG³. We model the dependence between these two latent variables and the temporal data as follows:

$$p(\mathbf{X}, \mathbf{Z}, \mathbf{S}) = \prod_{t=1}^T p_{\theta_x}(\mathbf{x}_t | \mathbf{z}_{\leq t}, \mathbf{s}_t) p_{\theta_z}(\mathbf{z}_t | \mathbf{z}_{< t}, \mathbf{s}_t) p_{\theta_s}(\mathbf{s}_t), \quad (3.1)$$

where $\mathbf{Z} = \mathbf{z}_{\leq T}$ and $\mathbf{S} = \mathbf{s}_{\leq T}$. The joint probability density function is parameterized by $\theta = \{\theta_x, \theta_z, \theta_s\}$. From now on, we omit this dependency to further relax notation. Following Nazabal et al. (2020), we assume that given the latent variable \mathbf{z}_t encodes all the correlation among attributes and hence they are all conditionally independent

$$p(\mathbf{x}_t | \mathbf{z}_{\leq t}, \mathbf{s}_t) = \prod_{d \in \mathcal{O}_t} p(x_{td} | \mathbf{z}_{\leq t}, \mathbf{s}_t) \prod_{d \in \mathcal{M}_t} p(x_{td} | \mathbf{z}_{\leq t}, \mathbf{s}_t). \quad (3.2)$$

Notice this is the temporal extension of Equation 2.16 with the additional discrete latent variable \mathbf{s}_t . The actual expression for each of the likelihood factors $p(x_{td} | \mathbf{z}_{\leq t}, \mathbf{s}_t)$ depends on the data-type of every attribute, as we develop in the next sub-section.

Temporal continuous latent variable \mathbf{z}_t Following the VRNN motivation presented in Section 2.4.3, the temporal dependency is encoded into the term $p_{\theta_z}(\mathbf{z}_t | \mathbf{z}_{< t}, \mathbf{s}_t)$, which implements a RNN-based model to capture the temporal data correlation along time:

$$p(\mathbf{z}_t | \mathbf{z}_{< t}, \mathbf{s}_t) = \mathcal{N}(\mathbf{z}_t | \boldsymbol{\mu}_{0,t}, \boldsymbol{\Sigma}_{0,t}), \quad (3.3)$$

where $\boldsymbol{\mu}_{0,t}$ and $\boldsymbol{\Sigma}_{0,t}$ define the parameters of the conditional prior distribution, and they are obtained as the output of a deep neural network (DNN) $\varphi_{\omega}^{\text{prior}}(\cdot)$ that extracts features from the past hidden state \mathbf{h}_{t-1} and the current discrete state \mathbf{s}_t :

$$[\boldsymbol{\mu}_{0,t}, \boldsymbol{\Sigma}_{0,t}] = \varphi_{\omega}^{\text{prior}}(\mathbf{h}_{t-1}, \mathbf{s}_t), \quad (3.4)$$

³Another option for the prior would be to use a mixture of posteriors as prior also known as VampPrior (Tomczak and Welling, 2018). However, to us it is more reasonable to use a prior that is not dependent on the posteriors distributions, due to the implicit dependencies present in the model.

where $\Sigma_{0,t}$ is considered a diagonal matrix. Notice this is similar to Equation 2.31 from the VRNN model. The hidden state \mathbf{h}_{t-1} encodes the information of the process \mathbf{z} up to time $t-1$, and it is updated along time using an LSTM with the following state update recurrence

$$\mathbf{h}_{t-1} = f_{\tau}(\mathbf{y}_{t-1}, \mathbf{h}_{t-2}), \quad (3.5)$$

where $\mathbf{y}_{t-1} = \varphi_{\omega}^{\mathbf{z}}(\mathbf{z}_{t-1})$ is the output of a DNN with input \mathbf{z}_{t-1} . We choose to work with LSTM (Hochreiter and Schmidhuber, 1997) due to the ability to better cope with long sequences, but any other RNN architectures such as GRU (Cho et al., 2014) could be used. Besides, in order to prevent the exploding gradient problem that can arise in RNNs, we clip the gradients to 0.5.

Discrete latent variable \mathbf{s}_t Finally, for the discrete latent variable \mathbf{s}_t we assume an informative time-independent prior:

$$p(\mathbf{s}_t) = \text{Categorical}(\mathbf{s}_t | \boldsymbol{\pi}), \quad (3.6)$$

where $\pi_k = 1/L$, where L is the number of components in the mixture.

3.3.3 Heterogeneous Decoder

We propose to use a factorized decoder that can handle different data-types for each attribute. A DNN is used to provide the likelihood parameters, e.g. mean and variance of a Gaussian distribution, given $\mathbf{h}_{t-1}, \mathbf{s}_t$, and \mathbf{y}_t . We denote the likelihood parameters for the d -th attribute at time t as $\gamma_t^d = \varphi_{\omega,d}^{\text{dec}}(\mathbf{h}_{t-1}, \mathbf{s}_t, \mathbf{y}_t)$, where $\varphi_{\omega,d}^{\text{dec}}$ is the *decoder* DNN, as it translates latent information into the observed variable space. Hence, extending the HI-VAE heterogeneous decoder presented in Section 2.3.3 to the temporal domain, the general likelihood expression derives as follows:

$$p(x_{td} | \mathbf{z}_{\leq t}, \mathbf{s}_t) = p(x_{td} | \gamma_t^d) \quad (3.7)$$

We consider the following data-types and associated likelihood forms:

1. **Real-valued data:** We assume a Gaussian likelihood distribution, *i.e.*,

$$p(x_{td} | \gamma_t^d) = \mathcal{N}(\mu_{x,t}^d, \sigma_{x,t}^{2,d}), \quad \text{where } [\mu_{x,t}^d, \sigma_{x,t}^{2,d}] = \varphi_{\omega,d}^{\text{dec}}(\mathbf{y}_t, \mathbf{s}_t, \mathbf{h}_{t-1}). \quad (3.8)$$

2. **Positive real-valued data:** We assume a log-Gaussian likelihood distribution, *i.e.*,

$$p(x_{td} | \gamma_t^d) = \log \mathcal{N}(\mu_{x,t}^d, \sigma_{x,t}^{2,d}), \quad \text{where } [\mu_{x,t}^d, \sigma_{x,t}^{2,d}] = \varphi_{\omega,d}^{\text{dec}}(\mathbf{y}_t, \mathbf{s}_t, \mathbf{h}_{t-1}). \quad (3.9)$$

3. **Binomial data:** We assume a Bernoulli likelihood distribution, *i.e.*,

$$p(x_{td} | \gamma_t^d) = \text{Be}(p_{x,t}^d), \quad \text{where } p_{x,t}^d = \sigma(\varphi_{\omega,d}^{\text{dec}}(\mathbf{y}_t, \mathbf{s}_t, \mathbf{h}_{t-1})), \quad (3.10)$$

and $p_{x,t}^d$ is the probability parameter of the Bernoulli distribution and σ is the sigmoid function.

4. **Categorical data:** We assume a multinomial likelihood distribution where the parameters of the likelihood are the C -dimensional output of a DNN with a log-softmax output

$$\log p(x_{td} = c | \gamma_t^d) = \varphi_{\omega,d}^{\text{dec}}(\mathbf{y}_t, \mathbf{s}_t, \mathbf{h}_{t-1})|_c \quad \text{for } c = [1, \dots, C]. \quad (3.11)$$

The left part of Figure 3.3 illustrates the generative model defined by Equations 3.1-3.7. From this figure we can see the motivation of having a shared latent space on \mathbf{z} and \mathbf{s} but an independent heterogeneous decoder where each likelihood for x_t^d is parameterized by γ_t^d .

3.3.4 Model Training with Variational Inference

Variational training (Kingma and Welling, 2014) involves optimizing a parameterized family of distributions $q_\eta(\cdot)$ that approximate the latent posterior distribution given the observed data. This optimization is carried out by maximizing the well-known evidence lower bound (ELBO).

Variational distribution The variational distribution for our model is defined as

$$q_\phi(\mathbf{x}_{\leq T}^m, \mathbf{z}_{\leq T}, \mathbf{s}_{\leq T} | \mathbf{x}_{\leq T}^o) \quad (3.12)$$

and it only depends on the observed attributes. Firstly, we need to define the variational distribution over the latent variable \mathbf{z}_t

$$q_{\phi_z}(\mathbf{z}_t | \mathbf{z}_{<t}, \mathbf{s}_t, \mathbf{x}_t^o) = \mathcal{N}(\boldsymbol{\mu}_{z,t}, \boldsymbol{\Sigma}_{z,t}), \quad \text{where } [\boldsymbol{\mu}_{z,t}, \boldsymbol{\Sigma}_{z,t}] = \varphi_\omega^{\text{enc}}(\varphi_\omega^{\text{x}}(\tilde{\mathbf{x}}_t), \mathbf{h}_{t-1}, \mathbf{s}_t). \quad (3.13)$$

$\tilde{\mathbf{x}}_t$ denotes a D -dimensional vector where the missing dimensions have been replaced by zeros following the zero filling approach as described in Nazabal et al. (2020), $\boldsymbol{\mu}_{z,t}$ and $\boldsymbol{\Sigma}_{z,t}$ are the parameters of the variational distribution and $\varphi_\omega^{\text{x}}$ and $\varphi_\omega^{\text{enc}}$ are neural networks. $\boldsymbol{\Sigma}_{z,t}$ is a diagonal matrix. Notice that this derivation aligns with the variational distribution Equation 2.34 from the VRNN. The variational distribution for the discrete latent space \mathbf{s}_t is defined as

$$q_{\phi_s}(\mathbf{s}_t | \mathbf{x}_t^o, \mathbf{z}_{<t}) = \text{Categorical}(\boldsymbol{\pi}(\varphi_\omega^{\text{s}}(\tilde{\mathbf{x}}_t, \mathbf{h}_{t-1}))), \quad (3.14)$$

where the probability for each category is given by the output of the DNN $\varphi_\omega^{\text{s}}(\cdot)$ followed by a log soft-max function. The variational distribution will then be composed of the variational distribution from Equation 3.13, the variational distribution from Equation 3.14 and $p(\mathbf{x}_t^m | \mathbf{z}_{\leq t}, \mathbf{s}_t)$, *i.e.*

$$q_\phi(\mathbf{x}_{\leq T}^m, \mathbf{z}_{\leq T}, \mathbf{s}_{\leq T} | \mathbf{x}_{\leq T}^o) = \prod_{t=1}^T q_{\phi_z}(\mathbf{z}_t | \mathbf{z}_{<t}, \mathbf{s}_t, \mathbf{x}_t^o) q_{\phi_s}(\mathbf{s}_t | \mathbf{x}_t^o, \mathbf{z}_{<t}) p(\mathbf{x}_t^m | \mathbf{z}_{\leq t}, \mathbf{s}_t). \quad (3.15)$$

ELBO The inference model is shown at the right part of Figure 3.3. By expanding the following expression

$$\log p(\mathbf{x}^o) \geq \int q(\mathbf{X}^o, \mathbf{X}^m, \mathbf{Z}, \mathbf{S}) \log \frac{p(\mathbf{X}, \mathbf{Z}, \mathbf{S})}{q(\mathbf{X}^o, \mathbf{X}^m, \mathbf{Z}, \mathbf{S})} d\mathbf{Z} d\mathbf{S} d\mathbf{X}^m, \quad (3.16)$$

we obtain the ELBO objective training function

$$\begin{aligned} \log p(\mathbf{X}^o) \geq & \sum_{t=1}^T \left[\underbrace{\frac{\mathbb{E}_{q(\mathbf{s}_t | \mathbf{x}_t^o, \mathbf{z}_{<t})}}{q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{z}_{<t}, \mathbf{s}_t)} [\log p(\mathbf{x}_t^o | \mathbf{z}_{\leq t}, \mathbf{s}_t)]}_{\text{Reconstruction}} \right. \\ & \left. - \underbrace{\frac{\mathbb{E}_{q(\mathbf{s}_t | \mathbf{x}_t^o, \mathbf{z}_{<t})}}{q(\mathbf{s}_t | \mathbf{x}_t^o, \mathbf{z}_{<t})} [\beta \text{KL}(q(\mathbf{z}_t | \mathbf{z}_{<t}, \mathbf{x}_t^o, \mathbf{s}_t) || p(\mathbf{z}_t | \mathbf{z}_{<t}, \mathbf{s}_t))] - \beta \text{KL}(q(\mathbf{s}_t | \mathbf{x}_t^o, \mathbf{z}_{<t},) || p(\mathbf{s}_t))]}_{\text{Regularization}} \right] \end{aligned} \quad (3.17)$$

The first term inside the sum in Equation 3.17 is the average reconstruction log-likelihood (e.g. how well we explain the observed data given the latent space induced by the approximated posterior), while the other two Kullback-Leibler (KL) divergence terms act like regularizers that penalize for posteriors far from the prior latent distributions. Although the expectation over $q(\mathbf{s}_t | \mathbf{x}_t^o)$ can be computed analytically, since \mathbf{s}_t is a discrete variable, due to the temporal

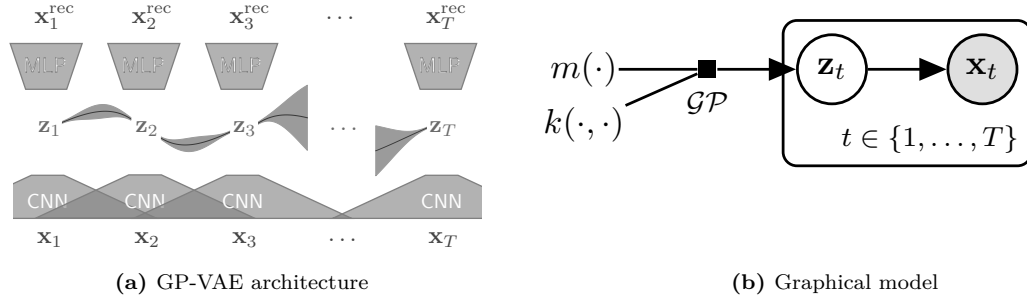


Figure 3.4: *Overview of the GP-VAE model:* In Figure (a) we show the GP-VAE architecture, where each z_t is modeled from the output of a CNN block and the GP on the latent space, and then decoded back to $\hat{\mathbf{x}}_t$ using MLPs. On Figure (b) we depict the graphical model, where the prior of \mathbf{z}_t comes from a GP with mean function $m(\cdot)$ and kernel $k(\cdot, \cdot)$. Figures from original paper (Fortuin et al., 2020).

dependencies encoded on the hidden state of the RNN \mathbf{h}_t we approximate such expectations at low complexity by sampling from $q(\mathbf{s}_t|\mathbf{x}_t^o)$ using the Gumbel-softmax trick (Jang et al., 2017). Finally, in Equation 3.17 β is a regularization parameter that we gradually increase during training, in a way the KL terms do not dominate over the reconstruction term during the earlier stages of training. Upon training, data is normalized following the same procedure as the HI-VAE described in Section 2.3.3: standard-scaling is used for real attributes, and also to the logarithm of positive attributes. Categorical data is one-hot encoded.

3.3.5 The GP-VAE Probabilistic Model

As discussed in the introduction, GP-VAE (Fortuin et al., 2020) stands out as the state-of-the-art VAE to handle temporal series. Before addressing the experimental section, it is relevant to compare at this point the GP-VAE probabilistic model with respect to Shi-VAE. In GP-VAE, the latent temporal variable \mathbf{z}_t is modeled with a Gaussian Process (GP) (Rasmussen et al., 2006), i.e., $\mathbf{z}_t \sim \mathcal{GP}(m_z(\cdot), k_z(\cdot, \cdot))$ (Figure 3.4b). The GP prior on the latent space is flexible and robust but it comes at the cost of inverting the kernel matrix, which has a time complexity of $\mathcal{O}(T^3)$. In contrast, the RNN-based correlation model in Equation 3.3 comes with a computational cost that grows linearly in T . Moreover, designing a kernel function for GP-VAE that accurately captures correlations in feature space and also in the temporal dimension is challenging. For this contribution (Barrejón et al., 2021) we compared to the GP-VAE from Fortuin et al. (2020). However, new followup works for the GP-VAE (Ashman et al., 2020; Jazbec et al., 2021b,a), based mainly on inducing points, have overcome the presented issues.

As in Shi-VAE, in GP-VAE given \mathbf{z}_t all the attributes are conditionally independent. Indeed, the GP-VAE and its inference machinery (Fortuin et al., 2020) does not consider heterogeneous observations, and all observations are modelled with real-valued Gaussian distributions. An overview of the GP-VAE model is depicted in Figure 3.4. Notice that the GP-VAE inference model is composed of CNN blocks which are more complex than our inference networks, only composed of basic MLP layers. Plus, these CNN blocks correlate neighboring samples, not only \mathbf{x}_t , which brings extra capabilities to the model. In the following experimental section we show how we obtain competitive results with our proposed Shi-VAE model in terms of error metrics for the reconstructed signals, but also how we obtain more correlated imputations with respect to the original signals.

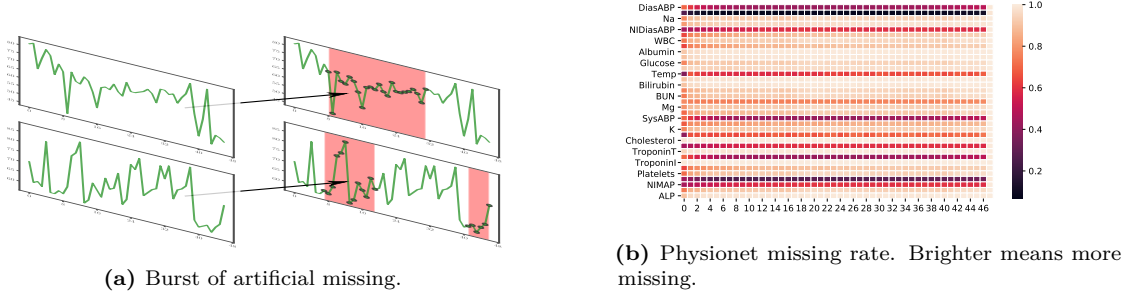


Figure 3.5: *Description of bursts of missing data and Physionet missing rates:* In Figure (a) we show how we generate missing artificial bursts for different sequences. The red masks and the corresponding missing entries (black markers) indicate the bursts of missing data. In Figure (b) we include the missing rates for Physionet. Notice that most of the variables are almost completely missing. The average missing rate is already 85% approximately.

3.4 Experimental Results

In this section we test the ability of Shi-VAE to exploit hidden correlations between attributes and infer trustworthy reconstructions in the presence of missing bursts. The following models are tested against Shi-VAE in the different experiments:

- **Mean:** We replace the missing values with the mean corresponding to the subsampled signal.
- **Last Obs Carried Forward (LOCF):** We impute using the last observed value for a given attribute.
- **KNN:** We use k -nearest neighbor with normalized Euclidean distance to find similar samples, and then impute with a weighted average of the neighbors.
- **Matrix Factorization (MF):** We subsample and factorize the data into two low-rank matrices and impute the missing entries with matrix completion [Friedman et al. \(2001\)](#).
- **MICE:** We use Multiple Imputation by Chained Equations (MICE), a very common method for missing value imputation which imputes those missing values from multiple imputations with chained equations [White et al. \(2011\)](#).
- **GP-VAE:** The GP-VAE described in Section 3.3.5.

We remark that both MF and MICE are “genie-aided” in the sense that they observe future values of the signal with-in a window to impute the results. The rest of the algorithms perform missing data imputation in an on-line fashion. Both GP-VAE and Shi-VAE reconstruct missing values by projecting the observed sequence to the latent space and then reconstruct the missing values using the generative model. The following python packages were used in order to implement the following methods: *fancyimpute* for Mean, KNN and MF; *autoimpute* for LOCF and *scikit-learn* for MICE ⁴.

⁴`autoimpute(0.12.1)`, `fancyimpute(0.5.5)`, `mice(0.23.2)` ([Pedregosa et al. \(2011\)](#))

Datasets We show results for three data sets. First, a synthetic data set generated by a heterogeneous HMM (Hidden Markov Model) with large hidden space, the human monitoring database described in Section 3.2, and the well known medical data set Physionet [Silva et al. \(2012\)](#). While in the first database, the generated data set does not contain any missing data, note that both Physionet and the human monitoring database have quite a lot of missing observations. We evaluate performance over artificial missing data that we further incorporate into the data streams in all cases.

Bursts of missing data We introduce missing sequences of random length for every variable to emulate missing bursts. A visual example can be seen in Figure 3.5a. Each burst is generated sampling a random length from a uniform distribution $\mathcal{U}(3, 10)$ and placing the burst in a random position given by an observed value. For every case (database and % introduced missing data), we create 10 random masks with a different missing pattern each, that we use to compute average errors and standard deviations around them. All masks implemented in the experiments are accessible in the code repository <https://github.com/dbarrejon/Shi-VAE>.

Experimental setup We used the default setups for all the baselines model except for the GP-VAE, where we set the latent dimension to 2 in the synthetic data set, to 35 for Physionet and to 5 for the other two databases, since these values provided optimized results after cross-validation. The cross-validated parameter configuration for the Shi-VAE is described in Table 3.2.

Parameter	Synthetic	Physionet	Human Monitoring
Epochs	100	100	100
Annealing Epochs	20	20	50
Dimension \mathbf{z}	2	35	5
Dimension \mathbf{h}	10	10	10
L	3	10	3
T	100	48	-
Optimizer	Adam	Adam	Adam
Learning Rate	$5e - 3$	$5e - 3$	$5e - 3$
Activation Layers	ReLU	ReLU	ReLU
Split Train/Val/Test	800/100/100	4K/4K/4K	135/15/17
Batch Size	64	64	64

Table 3.2: Parameter configuration for the different experiments.

3.4.1 Evaluation Metrics

In our experiments we found out that baseline models, even without explicitly modeling temporal dynamics, were able to obtain competitive results in terms of error metrics. However, when we compared our model with these baselines we notice their imputations were not faithfully correlated with respect to the original signal. Therefore, we will use two different types of metrics to compare our models: standard error metrics and cross-correlation metrics between the ground-truth sequence \mathbf{x} and the reconstructed one $\hat{\mathbf{x}}$. Before presenting the evaluation metrics, we will introduce some basic notation. Let us define \mathbf{X}_d as a $N \times T$ matrix where we compact the d -th attribute across all data points and time. This is the matrix before introducing the artificial missing bursts. The imputed matrix for such attribute is defined as $\hat{\mathbf{X}}_d$ (equal to \mathbf{X}_d for non-missing entries). Therefore, x_{td}^n is the entry at time t and data point n of \mathbf{X}_d . N_d is the number of missing entries in \mathbf{X}_d .

Error metrics We use a different type of error depending on the type of data:

- **Continuous data**, *i.e.* real and positive: we consider the normalized root mean squared error (NRMSE) evaluated only at missing entries

$$err(d) = \frac{\sqrt{1/N_d \sum_n \sum_t (x_{td}^n - \hat{x}_{td}^n)^2}}{\max(\mathbf{X}_d) - \min(\mathbf{X}_d)}. \quad (3.18)$$

- **Binary data and categorical data**: we consider the classification accuracy error evaluated at the missing entries.

$$err(d) = \frac{1}{N_d} \sum_n \sum_t \mathbb{I}(x_{td}^n \neq \hat{x}_{td}^n), \quad (3.19)$$

where $\mathbb{I}(\cdot)$ is the indicator function.

The average imputation error for all the attributes is given by $\text{Error} = 1/D \sum_d err(d)$, where D is the number of attributes.

Cross correlation On temporal data sets, evaluating the performance of a given model based on standard error metrics might not be conclusive enough, as our experiments demonstrate. We augment our experiments by analyzing $\phi(d)$, which is defined as the sum of the cross correlation between any missing burst in \mathbf{X}_d (a portion of a given row) and its corresponding imputation in $\hat{\mathbf{X}}_d$, normalized by the total number of missing entries N_d . To simplify notation, assume \mathbf{w} and $\hat{\mathbf{w}}$ are the true and imputed values of a missing burst respectively in \mathbf{X}_d , then we accumulate in $c(\mathbf{w}, \hat{\mathbf{w}})$ the maximum value of the normalized cross correlation, *i.e.*

$$c(\mathbf{w}, \hat{\mathbf{w}}) = \max[(\mathbf{w} - \mu_{\mathbf{w}}) \star (\hat{\mathbf{w}} - \mu_{\hat{\mathbf{w}}})], \quad (3.20)$$

\star is the cross correlation operator, and $\mu_{\mathbf{w}}$ is the average signal value during the burst. Hence

$$\phi(d) = \frac{\sum_{\mathbf{w}, \hat{\mathbf{w}} \in \mathbf{X}_d} c(\mathbf{w}, \hat{\mathbf{w}})}{N_d} \quad (3.21)$$

We also report the average correlation across all attributes, *i.e.* $\text{Cross. Corr} = 1/D \sum_d \phi(d)$.

3.4.2 Synthetic Data set

This data set is composed of $N = 1000$ samples of length $T = 100$ from a three-state HMM model. At each time instant the HMM produces four outputs of different nature: real, positive, binary and categorical. Each state is characterized by different emission distribution for each data type. The transition probabilities have been forced to be smooth, so that really abrupt changes are not likely to happen. Over the clean database, we generate missing masks with overall missing rates of 10%, 30% and 50%. This missing rates per variable, that is, for each variable we will have *e.g.* 10% missing on variable d. Therefore, the total missing per sample is the aggregated missing rate, which makes the problem more challenging. For all the baselines, including the GP-VAE, we work with subsampled slots of length $T = 50$ of every individual signal. For the Shi-VAE, we consider the whole signal.

In Figure 3.6 we display both reconstruction errors per attribute at different missing rates (a) and cross correlation for the real and positive attributes (b). In terms of reconstruction error, GP-VAE obtains the best results for the continuous variables by a small margin compared

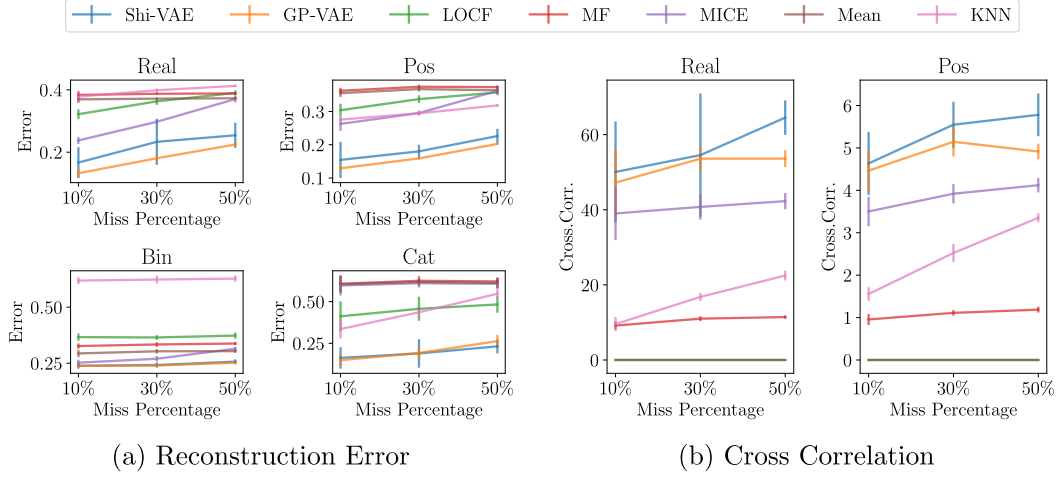


Figure 3.6: *Shi-VAE results for the synthetic data set:* On Figure (a) we show the imputation error for each variable and on Figure (b) the cross correlation for the continuous variables at different missing rates.

to Shi-VAE. This is due to the GP-VAE assuming a fully Gaussian distribution. However, for the binary and categorical view their performance is the same. Their distance with respect to the other baselines is remarkable. On the other hand, in terms of cross correlation, observe in Figure 3.6 (b) that Shi-VAE is able to reconstruct signals that are more correlated to the true distribution of the data. This raises an important question on how temporal models that are explicitly designed to impute missing values should be analyzed, whether it is more important to just focus on standard error metrics, or metrics considering temporal dependencies should be used when assessing the validity of temporal models.

We further illustrate that our model captured the temporal dynamics of the HMM state in Figure 3.7. The left-most figure depicts the real variable, in the center we show the HMM states and the latent space \mathbf{s} and the right-most figure shows the HMM states (3 possible states) and the two dimensions of the continuous latent embedding \mathbf{z} . The discrete latent variable \mathbf{s} is able to faithfully capture the HMM transitions, as well as the continuous latent variable \mathbf{z} , which only needs one dimension of the embedding to capture it. We argue that the continuous space in this case is well defined with one dimension since the latent space is already capable to discriminate between different types of dynamics within the sequences. Also notice that we successfully learnt for the discrete space because we obtain non-uniform probabilities for the categories of \mathbf{s} , and there is a clear structure directly related to the HMM states.

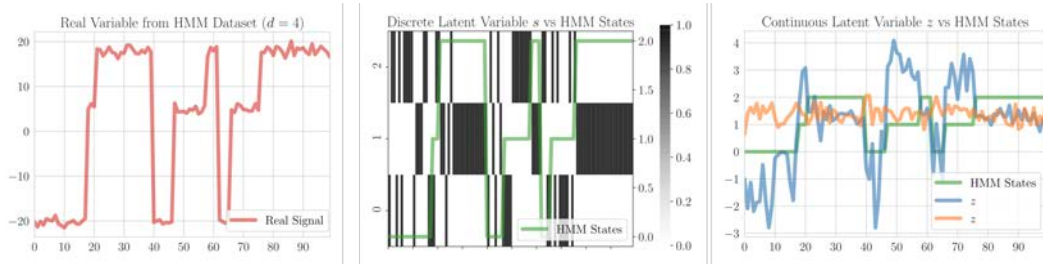


Figure 3.7: *HMM states versus continuous \mathbf{z}_t and discrete \mathbf{s}_t latent variables:* On the left, a real variable from the synthetic dataset. On the middle, a heatmap showing the probability for each mixture component from \mathbf{s} versus the hidden states of the HMM. On the right, the evolution of the continuous latent variable \mathbf{z} again versus the hidden states of the HMM. The non-identifiability problem comes from being an unsupervised problem of course.

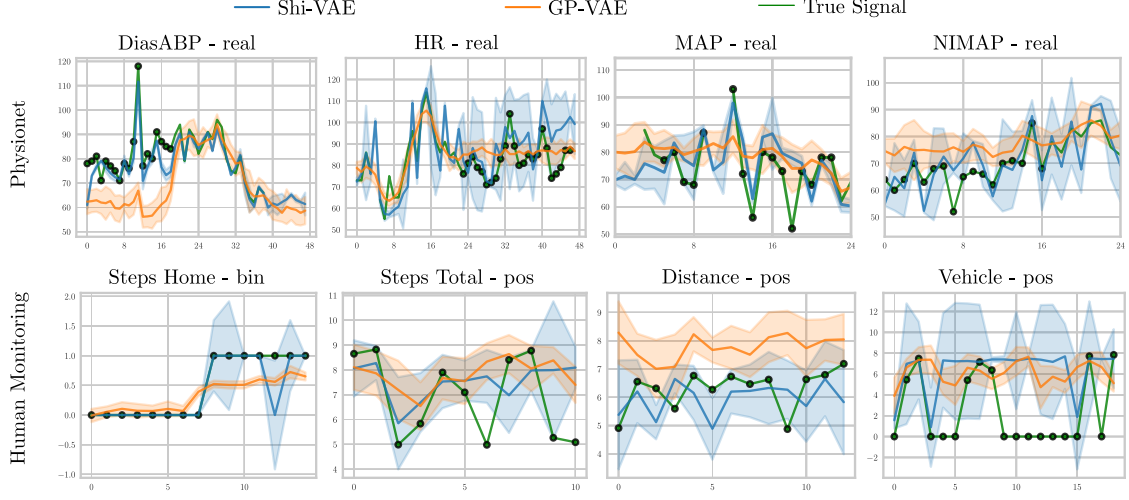


Figure 3.8: *Shi-VAE* and *GP-VAE* : Shi-VAE and GP-VAE example reconstruction for different attributes over the Physionet dataset (upper row) and the human monitoring database (bottom row). Missing values are indicated by black markers in the true signal.

3.4.3 Physionet

In this section, we compare both GP-VAE and Shi-VAE over the Physionet database (Silva et al., 2012). The data set contains 12,000 patients which were monitored on the intensive care unit (ICU) for 48 hours each. Each signal is sampled once an hour, hence their length is $T = 48$. At each hour, there is a measurement of 35 different variables⁵ (heart rate, blood pressure, etc.), any number of which might be missing. Plus, we further introduce artificial bursts of missing data up to an overall fraction of 10%. Note that the dataset already contains a large fraction of missing values (see Figure 3.5b, where most variables are almost completely missing).

In Table 3.3 we report GP-VAE and Shi-VAE average reconstruction error and average cross correlation. Observe that, as in the previous case, GP-VAE slightly improves the Shi-VAE in terms of average imputation error. However, Shi-VAE achieves a larger cross-correlation with respect to the ground-truth. To illustrate why reconstruction error can be a misleading metric when it comes to missing bursts, in the first row of Figure 3.8 we display the imputation of both methods for different missing bursts located at different Physionet attributes. Missing values are indicated by markers in the true signal. Observe that, while GP-VAE tends to impute missing burst with smooth solutions, Shi-VAE imputations certainly follow the true dynamics of the signal. But this discrepancy is not reflected in the average reconstruction error. In addition, observe that the Shi-VAE uncertainty (shaded area around the imputed signal) is informative and varies along time, allowing to identify regions of large and small uncertainty. On the other hand, the GP-VAE uncertainty does not show such a desired behaviour.

Model	Avg. Error	Cross. Corr
Shi-VAE	0.064 ± 0.003	38.061 ± 5.000
GP-VAE	0.060 ± 0.002	31.414 ± 1.016

Table 3.3: Physionet database results on the test set. For average error, lower is better. For cross correlation, larger is better.

⁵The list and definition of the attributes can be found in Silva et al. (2012).

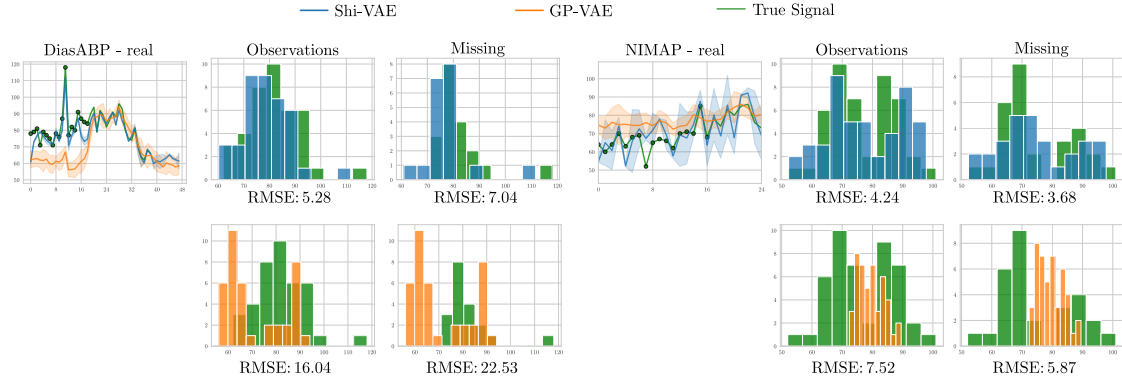


Figure 3.9: *Shi-VAE and GP-VAE histogram:* Comparison of Shi-VAE (blue) and GP-VAE (orange) using histograms and the evaluation metrics on missing and observed data. For RMSE (lower is better) and for cross-correlation metric (larger is better). On both signals DiasABP and NIMAP, our model is able to better capture the distribution of the real data (green), even multimodalities as depicted on the histograms for NIMAP variable. While empirically demonstrate how RMSE can be a misleading metric when assessing the performance of temporal models.

Similar conclusions can be drawn from the next related experiment. In Figure 3.9 we show one signal from the Physionet dataset, the real signal in green, the imputation from the Shi-VAE in blue and the imputation from the GP-VAE in orange. The first column on the right of each signal shows the distribution of the data with a histogram for the observed values of the signal, and the second column the distribution for the missing values (shown with black markers on the plot). The first row corresponds to the histogram that Shi-VAE produces, obtained by sampling from the model at each point. In the second row we do the same for the GP-VAE. We use the average of 10 samples produced by the models for the results. Below each histogram we show the corresponding average RMSE between the real samples and the imputed samples for each model. Observe that, while GP-VAE struggles to fit the real distribution even in the observed values, Shi-VAE provides a reasonably better result, being able to fit the two modes of the real distribution. This issue is not clearly reflected in the RMSE metric, which is not indeed very different between both models. On the contrary, the temporal correlation metric clearly shows the superior performance of Shi-VAE.

3.4.4 Human Monitoring Database

Finally, we reproduce the experiment for the human monitoring database described in Section 3.2. The average fraction of artificially introduced missing rate per attribute is 15%. In this case the length of the temporal sequences for each patient is different. For Shi-VAE and GP-VAE, we pad with zeros to the right those sequences with a length smaller than the maximum sequence length in a batch. As described in Section 3.3.5, the GP-VAE complexity badly scales with the sequence length. To run GP-VAE in reasonably time, any sequence larger than 50 time steps is subsampled to fit this maximum length. Note that Shi-VAE does not suffer from such penalization with respect to sequence length.

In Table 3.4, we report the error and cross correlation per attribute (seven of them, as described in Table 3.1), and the overall average values. Observe that, systematically, Shi-VAE achieves the largest correlation per attribute. In the second row of Figure 3.8 we show the imputation of both methods for different missing bursts located at different attributes. The robustness of the Shi-VAE can be observed in terms of the correlation between the imputed signal and the true one and in

terms of the uncertainty along time, which tends to be larger for those points in time for which the Shi-VAE mode is far from the true value. Again, such a behavior is not provided by GP-VAE, which produces more average imputations not correlated to the real dynamics of the data.

Variable	Model	Error	Cross Correlation
Average	Shi-VAE	0.200 ± 0.038	0.369 ± 0.140
	GP-VAE	0.184 ± 0.022	0.157 ± 0.031
Distance	Shi-VAE	0.201 ± 0.012	0.783 ± 0.249
	GP-VAE	0.205 ± 0.014	0.389 ± 0.092
Steps home	Shi-VAE	0.170 ± 0.054	0.010 ± 0.009
	GP-VAE	0.151 ± 0.016	0.011 ± 0.009
Steps total	Shi-VAE	0.269 ± 0.046	0.444 ± 0.181
	GP-VAE	0.268 ± 0.044	0.205 ± 0.038
App usage	Shi-VAE	0.113 ± 0.014	0.088 ± 0.045
	GP-VAE	0.115 ± 0.013	0.039 ± 0.008
Sport	Shi-VAE	0.216 ± 0.086	0.013 ± 0.005
	GP-VAE	0.121 ± 0.030	0.009 ± 0.004
Sleep	Shi-VAE	0.063 ± 0.010	0.034 ± 0.016
	GP-VAE	0.059 ± 0.010	0.013 ± 0.003
Vehicle	Shi-VAE	0.372 ± 0.043	1.215 ± 0.477
	GP-VAE	0.370 ± 0.028	0.436 ± 0.064

Table 3.4: Results for each variable for the human monitoring data set.

3.5 Discussion

In this work we propose Shi-VAE, a deep generative model that handles temporal and heterogeneous streams of data in the presence of missing data. While GP-VAE badly scales with long time series, Shi-VAE handles long term dependencies by encapsulating the temporal information into the continuous latent code \mathbf{z} by using RNN architectures. Having a hierarchical latent model with an additional discrete latent embedding \mathbf{s} provides a more flexible understanding of the data and benefits the latter process of modeling the heterogeneous distributions.

We have shown with a synthetic data set and two real-world medical data sets that standard error metrics are not completely informative to fully assess the performance of temporal models. We remark the importance of analyzing the temporal correlation in these type of studies by using sequences of missing data along time instead of fully random missing masks as it is normally done in similar works. In this scenario, Shi-VAE emerges as a robust solution to impute missing data bursts and perform dimensionality reduction.

Part II

Learning to Defer to Multiple Experts

Saudade, saudade
 Nothing more that I can say
 says it in a better way

saudade, saudade — MARO ▶

4

Learning to Defer

Contents

4.1	General Classification Problem	55
4.1.1	Binary Classification	57
4.1.2	Multiclass Classification: Cross-entropy loss	58
4.1.3	Multiclass-to-binary reduction: Code Based Surrogates	59
4.1.4	Can we abstain to predict? A motivating example towards L2D	60
4.2	Learning to Defer Background and Related Work	61
4.2.1	Learning to Defer within Rejection Learning	62
4.2.2	Learning to Defer in the Context of Human-Machine Collaboration	63
4.3	Learning to Defer	64
4.3.1	Preliminaries	65
4.3.2	Softmax Surrogate Loss: Single Expert	67
4.3.3	One-vs-All Surrogate Loss: Single Expert	68
4.3.4	Realizable-Surrogate Loss: Complement when deferring	68
4.3.5	Toy example for Learning to Defer surrogate losses	69
4.4	Confidence Calibration in Learning to Defer	70
4.4.1	Our Notion of Confidence Calibration	71
4.4.2	Softmax Parametrization: Single Expert	72
4.4.3	One-vs-All Parameterization: Single Expert	73
4.5	Summary of the Chapter	75

IN the first part of the thesis we focused on the development of *unsupervised* machine learning models (in our case VAEs) able to find hidden correlations in temporal data for recovering or imputing missing data. After carefully validating these temporal models, or any machine learning model in general, these models can be deployed in real-world scenarios. For example, imagine a classifier that has been trained on chest x-ray images to detect whether a person has lung cancer or not. If we used this system in practice, the first thing we would like to account is *uncertainty quantification*: we would like to know how certain the classifier is about its prediction.

Machine learning models are being deployed in a wide range of different fields, including healthcare (Zoabi et al., 2021; Codella et al., 2018), criminal justice (Zhong et al., 2018), ethical concerns in minority groups (Birhane et al., 2022; Tomasev et al., 2021), climate change (Lam et al.,

2022), and autonomous driving (Grigorescu et al., 2020). However, in these scenarios, it is essential to have trustworthy and safe systems (Hendrycks and Dietterich, 2019; Nguyen et al., 2015).

It is clear that technology has attained an unprecedented level of advancement, demonstrating its utmost capabilities and potential. Related to the chest x-ray example, Irvin et al. (2019) showed how a machine learning model was able to outperform 2 out of 3 expert radiologists. But this read is a bit simplistic of course: in a skin cancer experiment, Tschandl et al. (2020) wondered how can we leverage from human-machine collaboration. The findings revealed that even when they were confident, less experienced physicians showed a tendency to accept AI-based assistance that contradicted their initial diagnosis. The presumed benefits of integrating human and AI cooperation are also questioned by Jacobs et al. (2021). In a scenario involving antidepressant selection, they discover that AI recommendations can potentially *negatively* affect clinician treatment decisions. Also Schemmer et al. (2022) corroborate in a similar study that humans over-rely on AI advice and struggle to ignore incorrect advice. This raises an important question on *how should we design algorithms that effectively enable human-machine collaboration?* This question has fallen under the umbrella of *hybrid-intelligence* (Kamar, 2016a; Dellermann et al., 2019; Akata et al., 2020) approaches.

One option we can consider is allowing the model to “abstain” from making certain predictions. The decision not to predict can be based on the model’s confidence level. For example, a self-driving car might feel uncertain about stopping at a poorly visible traffic light. In a reliable system, this would lead to the human driver taking control in such situations and then returning control to the car after passing the traffic light. In such cases, we want the system to act *responsibly* (Madras et al., 2018) by referring the decision to the human, *not just* because the model’s confidence is low, *but also* because the system can assess the human’s confidence in that situation. This concept of not only abstaining from prediction (rejection learning) but also understanding the human’s knowledge has been explored in a recent theory known as *Learning to defer* (L2D). In L2D we are interested on both the machine’s and human’s confidence, and we will *defer when the human is more likely to be correct than the machine*. While achieving high accuracy in classification is crucial, it is essential to emphasize that these systems will be employed in critical scenarios. Hence, it is imperative that these systems offer consistent estimations that effectively convey the uncertainty and associated risks involved in decision-making based on their predictions. In simpler terms, we desire our systems to be appropriately *calibrated*, meaning that the system’s output should align with the true uncertainty for both the model and the human involved.

Outline In the following chapter we will try to go through the learning to defer framework: In Section 4.1, we will first outline a general classification problem – both binary and multiclass – and provide a motivating example of why using *only* the model is not a right option in some problems. We continue with Section 4.2 setting the context and background of L2D, followed by Section 4.3 which presents the L2D framework and the main contributions from the literature so far, together with a practical example. Finally, we introduce the concept of *confidence calibration* in Section 4.4, specially the expert’s confidence calibration, which is our prime interest, and comment on how different L2D approaches deal with this issue. Mainly, how the Mozannar and Sontag (2020)’s formulation is not well calibrated and how Verma and Nalisnick (2022)’s formulation tries to solve this issue.

Note on contributions The contributions presented in this chapter are attributed to their respective authors. Namely, the softmax surrogate loss by [Mozannar and Sontag \(2020\)](#), the OvA surrogate loss by [Verma and Nalisnick \(2022\)](#) and the realizable surrogate loss by [Mozannar et al. \(2023\)](#). However, some of the writing was withdrawn from our work ([Verma, Barrejón, and Nalisnick, 2023](#)) presented in the International Conference on Artificial Intelligence and Statistics (AISTATS) 2023 in Valencia. The toy examples for Sections 4.1.4 and Section 4.3.5 were designed by us and can be found in <https://github.com/dbarrejon/thesis-l2d>.

Acknowledgments We would also like to thank the authors of the following amazing works, which were of great help during the writing of this chapter: Rajeev Verma for his Msc. Thesis ([Verma, 2022](#)), Hussein Mozannar for the [slides](#)¹ of the course Machine Learning for Healthcare at MIT 2023 and Eric Nalisnick for his [slides](#)² on the calibration of learning to defer systems.

4.1 General Classification Problem

This section presents the common notation that will be assumed for the rest of this thesis, and remind the reader that, while in Chapters 2 and 3 we were facing a fully unsupervised problem, now the landscape completely drifts to fully *supervised learning*. We continue with the derivations of standard surrogate losses for general classification problems, linking the common equations used in such problems with the common probabilistic machine learning equations (maximum-likelihood estimation). Once the classification problem is outlined, we describe binary classification and multi-class classification, with their appropriate surrogate losses and common parametrizations and briefly outline how multi-class classification can be decomposed in multiple binary problems in a one-vs-all (a.k.a one-vs-rest) scheme. We conclude with a toy example where an AI system *alone* fails at fully describing a data distribution, and how *rejection learning* might be an interesting solution.

Data Consider the feature space denoted as $\mathcal{X} \subseteq \mathbb{R}^d$ and the label space denoted as \mathcal{Y} . We assume that \mathcal{Y} represents a categorical encoding of multiple classes (K classes), *i.e.* $\mathcal{Y} = \{1, 2, \dots, K\}$. In this context, $\mathbf{x}_n \in \mathcal{X}$ represents a feature vector, *e.g.* an image or text encoded as a vector, and $y_n \in \mathcal{Y}$ represents the associated class, defined within the label space \mathcal{Y} (one out of K classes), *e.g.* whether a patient is sick or not ($\mathcal{Y} = \{\text{sick}, \text{not sick}\}$). The training sample, consisting of N elements, is represented as $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$, where $\mathbf{x}_n \in \mathcal{X}$, $y_n \in \mathcal{Y}$, and both \mathbf{x}_i and \mathbf{y}_i are sampled from the dataset \mathcal{D} . In a machine learning classification problem, the objective is to learn a mapping $h : \mathcal{X} \rightarrow \hat{\mathcal{Y}}$. We refer to h as a prediction function, and we assess its performance using a loss function $\ell : \mathcal{Y} \times \hat{\mathcal{Y}} \rightarrow \mathbb{R}^+$.

Learning Denote the *classifier* as $h : \mathcal{X} \rightarrow \mathcal{Y}$, and we will denote as θ_h the parameters of the classifier h . Therefore, our goal is find the optimal parameters θ_h to fit the model to the data distribution \mathcal{D} . We will follow the derivations from [Murphy \(2022\)](#). Parameter estimation in statistical learning often relies on maximum likelihood estimation (MLE), which involves selecting the parameter values that maximize the probability of the observed training data. To formally define MLE, we denote it as $\hat{\theta}_{\text{MLE}}$ and express it as:

$$\hat{\theta}_{\text{MLE}} = \arg \max_{\theta} p(\mathcal{D}|\theta). \quad (4.1)$$

¹<https://mlhcmiit.github.io/2023/lectures/Human-AI%20interaction.pdf>

²https://enalisnick.github.io/Calibrated_L2D_talk.pdf

It is common to assume that the training samples are independently and identically distributed (iid) from the same underlying distribution. Under this assumption, the likelihood function becomes:

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{n=1}^N p(y_n|\mathbf{x}_n;\boldsymbol{\theta}). \quad (4.2)$$

To simplify calculations, it is often more convenient to work with the logarithm of the likelihood, denoted as $\mathcal{L}(\boldsymbol{\theta})$, given by:

$$\mathcal{L}(\mathcal{D};\boldsymbol{\theta}) = \sum_{n=1}^N \log p(y_n|\mathbf{x}_n;\boldsymbol{\theta}). \quad (4.3)$$

The MLE can then be expressed as:

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} = \arg \max_{\boldsymbol{\theta}} \sum_{n=1}^N \log p(y_n|\mathbf{x}_n;\boldsymbol{\theta}). \quad (4.4)$$

Since many optimization algorithms aim to minimize cost functions, it is common to redefine the objective function as the negative log-likelihood (NLL):

$$\text{NLL}(\mathcal{D}, \boldsymbol{\theta}) = - \sum_{n=1}^N \log p(y_n|\mathbf{x}_n;\boldsymbol{\theta}). \quad (4.5)$$

In a more general framework called *empirical risk minimization* (ERM), the MLE can be extended by replacing the (conditional) log loss term with any other loss function. Again, following [Murphy \(2022\)](#), the **empirical** risk, denoted as $L(\boldsymbol{\theta})$, is then defined as the average loss over the training samples:

$$L(\mathcal{D}, \boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(y_n, \mathbf{x}_n; \boldsymbol{\theta}), \quad (4.6)$$

where $\mathcal{L}(y_n; \mathbf{x}_n; \boldsymbol{\theta})$ represents the chosen loss function. ERM aims to minimize this empirical risk by adjusting the parameters $\boldsymbol{\theta}$. In a classification problem, the 0-1 loss function can be employed to measure misclassification rate. This loss function assigns a value of 0 when the predicted label matches the true label, and a value of 1 otherwise.

$$\ell_{0-1}(y_n, \boldsymbol{\theta}; \mathbf{x}_n) = \begin{cases} 0 & \text{if } y_n = h(\mathbf{x}_n; \boldsymbol{\theta}) \\ 1 & \text{if } y_n \neq h(\mathbf{x}_n; \boldsymbol{\theta}) \end{cases} \quad (4.7)$$

The empirical risk from Equation 4.6 then corresponds to the empirical misclassification rate on the training set

$$\mathcal{L}_{0-1}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^N \ell_{0-1}(y_n, \boldsymbol{\theta}; \mathbf{x}_n). \quad (4.8)$$

Surrogate Losses Regrettably, the 0-1 loss from Equation 4.7 exhibits non-smooth behavior characterized by a step function, as depicted in Figure 4.1, thereby posing challenges for optimization. In fact, it is NP-hard, as demonstrated by [Ben-David et al. \(2003\)](#). To overcome this limitation, we explore the adoption of a surrogate loss function ([Bartlett et al., 2006](#)). Typically, the surrogate is carefully chosen to be a convex upper bound that tightly approximates the 0-1 loss, facilitating the optimization process. The choice and design of surrogate losses depends on the nature of the problem, and it is not a straight-forward task.

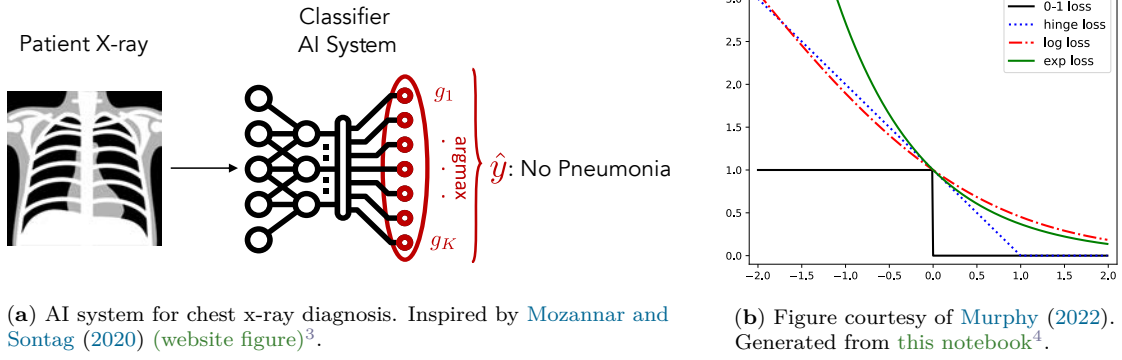


Figure 4.1: Example of AI system predicting pneumonia using softmax. Canonical 0-1 loss and associated binary surrogate losses: Given a patient's x-ray, an AI system decides whether the patient has pneumonia by selecting the output with higher probability (Figure (a)). Figure (b) depicts ℓ_{0-1} (Equation 4.7) and binary surrogate losses that are feasible to optimize. In this thesis, we will focus on the log loss.

Consistent surrogate losses The majority of machine learning practices involve optimizing surrogate loss functions instead of the true loss functions of interest. Surrogate loss functions are selected to ensure that optimizing them leads to the optimization of the true loss functions. Additionally, surrogate loss functions are preferred to be differentiable, allowing for efficient optimization. The concept of *consistency* encompasses the first property mentioned above. Throughout this chapter we will present consistent surrogate losses that have been proposed in the literature for the learning to defer framework, and this will set the ground for our contribution of extending these consistent surrogate losses to the multiple expert scenario in learning to defer. However, it must be noted that in practice one can relax the strong assumptions commonly made for consistent surrogate losses and still achieve great performance ([Mozannar et al., 2023](#)).

In the following Subsections we will briefly introduce which surrogate losses can be utilized for binary and multiclass classification problems in machine learning, and also outline how one can translate a multiclass problem to a multi-binary problems using code based surrogates.

4.1.1 Binary Classification

In the context of binary problems, an alternative notation can be used to express the misclassification rate. Let $y \in \mathcal{Y} = \{-1, +1\}$ represent the true label, and $\hat{y} \in \hat{\mathcal{Y}} = \{-1, +1\}$ denote our prediction $h(\mathbf{x}; \boldsymbol{\theta})$, the 0-1 loss is defined as:

$$\ell_{0-1}(y; \hat{y}) = \begin{cases} 0, & \text{if } y = \hat{y} \\ 1, & \text{if } y \neq \hat{y} \end{cases} = \mathbb{I}[y \neq \hat{y}] \quad (4.9)$$

The corresponding empirical risk, in terms of the misclassification rate, is given by:

$$L(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^N \ell_{0-1}(y_n; \hat{y}_n) = \frac{1}{N} \sum_{n=1}^N \mathbb{I}[y \neq \hat{y}] \quad (4.10)$$

In this formulation, the dependence on \mathbf{x}_n and $\boldsymbol{\theta}$ is implicit. However, the above equation is not a convex loss and we need a proper surrogate loss.

³<https://husseinmozannar.github.io/publication/mozannar-2020-consistent/>

⁴https://github.com/probml/pyprobml/blob/auto_notebooks_md/notebooks.md#hinge_loss_plot.ipynb

Logistic Loss We need a binary surrogate loss $\phi : \{\pm 1\} \times \mathbb{R} \mapsto \mathbb{R}_+$. Some suitable surrogate binary loss we may use is the *logistic loss* (also known as log loss). Let's consider a probabilistic binary classifier that generates a label distribution as follows:

$$p(y|\mathbf{x}; \boldsymbol{\theta}) = \sigma(yg(\mathbf{x})) = \frac{1}{1 + e^{-yg(\mathbf{x})}}, \quad (4.11)$$

where $g(\cdot)$ defines a prediction function $g : \mathcal{X} \mapsto \mathbb{R}$ for the log odds and $\sigma(\cdot)$ is the *sigmoid* or *logistic* function which maps the real line to values between 0 and 1, *i.e.* $\sigma : \mathbb{R} \rightarrow (0, 1)$, hence providing values that can be interpreted as probabilities. We can further view these functions as $g(\mathbf{x})$ as the log probabilities, often called *logits*, output by a machine learning model. For the remainder of this thesis, we relax notation and think that the parameters we want to fit $\boldsymbol{\theta}$ come from the predictor functions $g(\mathbf{x})$, or in other words, the optimization problem will be focused on finding the Bayesian optimal functions $g(\mathbf{x})$. Following with the derivations, if we apply the negative log likelihood to the above equation we obtain the logistic loss

$$\phi_{\log}(g; \mathbf{x}, y) = -\log p(y|g(\mathbf{x})) = \log(1 + e^{-yg(\mathbf{x})}), \quad (4.12)$$

where y denotes the *true* label⁵. Equation 4.12 can be simplified as $\phi(y, u) = \log(1 + \exp\{-yu\})$, where u are the predictions from the model $u = g(\mathbf{x})$. Figure 4.1(b) demonstrates that the proposed function provides a smooth upper bound to the 0-1 loss. It represents the relationship between the loss and the margin, denoted as $yg(\mathbf{x})$, which measures the "margin of safety" away from the threshold value of 0. Minimizing the negative log likelihood is therefore equivalent to minimizing a tight upper bound on the empirical 0-1 loss. For a more in depth analysis of surrogate losses in the context of binary classification we recommend the following works (Bartlett et al., 2006; Reid and Williamson, 2010).

Prediction The logistic loss function is a strictly proper composite loss (Reid and Williamson, 2009, 2010; Buja et al., 2005) with a well-defined inverse link function γ^{-1} which maps the predictions $g(\mathbf{x})$ to the label space \mathcal{Y} . Such inverse function is the *sigmoid* function presented before, *i.e.* $\gamma^{-1}(g(\mathbf{x})) = \sigma(g(\mathbf{x})) = 1/(1 + \exp\{-g(\mathbf{x})\})$. As it has been shown in the literature, we know that $\gamma^{-1}(g(\mathbf{x}))$ can be used as an estimator of the class probability $\gamma^{-1}(g(\mathbf{x})) = p(y = y|\mathbf{x})$. Hence, in practice the final predictor can be calculated as follows⁶:

$$\hat{y} = h(\mathbf{x}) = \text{sign}(p(y = y|\mathbf{x})) = \text{sign}\left(\gamma^{-1}(g(\mathbf{x})) - \frac{1}{2}\right) \quad (4.13)$$

4.1.2 Multiclass Classification: Cross-entropy loss

Now, our label space can have multiple classes $\mathcal{Y} = 1, \dots, K$. We wish to find a surrogate loss that we can use, and such loss is the *cross-entropy* (CE) loss, which is actually the logistic loss extended to the multiclass problem $\psi : \mathcal{Y} \times \mathbb{R} \mapsto \mathbb{R}_+$.

⁵We could also formulate the problem using an alternative binary notation $y \in \mathcal{Y} = \{0, 1\}$. In this case: $\phi_{\log}(g; \mathbf{x}, y) = y \log(\sigma(g(\mathbf{x}))) + (1 - y) \log(1 - \sigma(g(\mathbf{x})))$.

⁶We need the $\text{sign}(\cdot)$ operator because $y \in \mathcal{Y}$

Cross-entropy Loss Let $g_k : \mathcal{X} \mapsto \mathbb{R}$ for $k \in [1, K]$ where k denotes the class index. Again, one may think of g_k as the *logits* or log probabilities output by a machine learning model. Now we will consider a classifier assuming a *softmax* parameterization producing the resulting probability distribution on the label space

$$p_{\theta}(y|\mathbf{x}) = p(y|g(\mathbf{x})) = \text{softmax}(g_y(\mathbf{x})) = \frac{\exp\{g_y(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}} \exp\{g_{y'}(\mathbf{x})\}}. \quad (4.14)$$

$p_{\theta}(y|\mathbf{x})$ is a confidence estimate of the probability of the true label, but the *true* probability is often denoted as $\mathbb{P}(y = y|\mathbf{x})$. This will be extremely important along the rest of the thesis. However, we can still interpret $p_{\theta}(y|\mathbf{x})$ as some kind of probability measure. We will see that for our problem of interest related to learning to defer (Section 4.3) this no longer applies. Similarly as in the binary problem, θ come from the functions $g(\mathbf{x}) = \{g_1(\mathbf{x}), \dots, g_K(\mathbf{x})\}$ that define the classifier $h(\mathbf{x})$. Combining these K functions in the following softmax-base we obtain point-wise surrogate loss for multiclass classification:

$$\psi_{\text{CE}}(g_1, \dots, g_K; \mathbf{x}, y) = -\log p(y|g(\mathbf{x})) = -\log \left(\frac{\exp\{g_y(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}} \exp\{g_{y'}(\mathbf{x})\}} \right), \quad (4.15)$$

where y is the true class. By minimizing the cross entropy loss, the model learns to assign higher probabilities to the correct class and lower probabilities to the incorrect classes, effectively driving the model towards better classification performance.

Prediction Since the $\text{softmax}(\cdot)$ function described in Equation 4.14 provides valid probability estimates, it is trivial to see that the decoding function to obtain the final prediction \hat{y} will be the $\arg \max$ function

$$\hat{y} = h(\mathbf{x}) = \arg \max_{y \in \mathcal{Y}} p(y = y|\mathbf{x}) = \arg \max_{y \in \mathcal{Y}} g_y(\mathbf{x}) \quad (4.16)$$

In Figure 4.1(a) we included an intuitive example where a classifier using a softmax parametrization might predict for a given chest x-ray if a patient has pneumonia or not. This AI prediction is obtained applying Equation 4.16.

4.1.3 Multiclass-to-binary reduction: Code Based Surrogates

One reasonable approach one could think would be apply a divide-and-conquer strategy to the multi-class problem: instead tackling the multi-class problem altogether, we could decompose it into multiple binary classification problems. One possible solution for this approach would be applying error-correcting coding algorithms (Dietterich and Bakiri, 1995; Langford et al., 2005; Allwein et al., 2001). But we will follow the idea proposed by Ramaswamy et al. (2014), and refer the reader to the original paper for full details. Further more, we follow the explanation from Verma and Nalisnick (2022).

The objective of a code-based mechanism is to decompose a classification problem with \hat{K} classes into \hat{K} binary classification problems using a code matrix $\mathbf{M} = \{-1, 1, 0\}^{K \times \hat{K}}$ (notice that the original label space \mathcal{L} can be different to the prediction space $\hat{\mathcal{Y}}$). In this approach, the training sample $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ is split into \hat{K} training subsets \tilde{S}_j (for $j \in [\hat{K}]$) by replacing the original class labels with binary labels based on 4.1.2. Each subset \tilde{S}_j is used to train a binary classifier $g_k : \mathcal{X} \rightarrow \mathbb{R}^{\hat{K}}$. Consequently, for any input $\mathbf{x} \in \mathcal{X}$, the predictions $g(\mathbf{x}) = [g_1(\mathbf{x}), \dots, g_K(\mathbf{x})]$ are obtained, the same way as in Section 4.1.2.

OvA loss Then, the learning process involves minimizing a surrogate multiclass classification loss $\psi : \mathcal{Y} \times \mathbb{R} \rightarrow \mathbb{R}_+$ making use of individual binary surrogate losses $\phi : \{\pm 1\} \times \mathbb{R} \rightarrow \mathbb{R}_+$ using the binary labels obtained from M . If we choose the logistic loss ϕ (Equation 4.12) we can write the loss function as

$$\psi(g_1, \dots, g_K; \mathbf{x}, y) = \sum_{k=1}^K \mathbb{I}[\mathbf{M}_{y,k} = 1] \phi[g_k(\mathbf{x})] + \mathbb{I}[\mathbf{M}_{y,k} = -1] \phi[-g_k(\mathbf{x})], \quad (4.17)$$

where we express $\phi[g_y(\mathbf{x})] = \phi(1, g_y(\mathbf{x}))$ and similarly $\phi[-g_y(\mathbf{x})] = \phi(-1, g_y(\mathbf{x}))$. For the specific case that $\mathcal{Y} = \hat{\mathcal{Y}}$ and $\mathbf{M}_{yk} = 1$ and otherwise $\mathbf{M}_{yk} = -1$ then

$$\psi(g_1, \dots, g_K; \mathbf{x}, y) = \phi[g_y(\mathbf{x})] + \sum_{\substack{y' \in \mathcal{Y}' \\ y' \neq y}} \phi[-g_{y'}(\mathbf{x})], \quad (4.18)$$

It is easy to see that Equation 4.18 can be rewritten as the well-known one-vs-all (OvA) loss for multiclass classification

$$\begin{aligned} \psi_{\text{OvA}}(g_1, \dots, g_K; \mathbf{x}, y) &= \phi[g_y(\mathbf{x})] + \sum_{\substack{y' \in \mathcal{Y}' \\ y' \neq y}} \phi[-g_{y'}(\mathbf{x})] \\ &= -\log \left(\frac{1}{1 + \exp\{-g_y(\mathbf{x})\}} \right) - \sum_{\substack{y' \in \mathcal{Y}' \\ y' \neq y}} \log \left(\frac{1}{1 + \exp\{g_{y'}(\mathbf{x})\}} \right) \\ &= -\log(\sigma(g_y(\mathbf{x}))) - \sum_{\substack{y' \in \mathcal{Y}' \\ y' \neq y}} \log(\sigma(-g_{y'}(\mathbf{x}))), \end{aligned} \quad (4.19)$$

Prediction To map these predictions $g(\mathbf{x})$ back to the original class labels \mathcal{Y} , we can follow the same procedure as described in Section 4.1.1 since for each class $y \in \mathcal{Y}$ we are solving a binary classification problem. However, in practice one may also apply the arg max function to g_1, \dots, g_K , *i.e.* $\hat{y} = \arg \max_{y \in \mathcal{Y}} g_y(\mathbf{x})$ as in the softmax parametrization.

This multi-class to binary reduction approach has been shown to be rather interesting in terms of calibration (Gupta and Ramdas, 2022) and will be of great relevance along the thesis, specially for our reference work (Verma and Nalisnick, 2022). This section just served as a short introduction to the multiclass-to-binary reduction problem. We refer the reader to some studies on binary proper composite losses (Reid and Williamson, 2009, 2010; Buja et al., 2005).

4.1.4 Can we abstain to predict? A motivating example towards L2D

So far we have presented two commonly used parameterization for solving multi-class classification problems: the softmax parametrization for the cross-entropy surrogate loss, and the OvA parametrization for the OvA surrogate loss. In Figure 4.2 we show with a toy dataset of Mixture of Gaussians (MoG) how the OvA formulation (softmax results are similar) behaves when trying to fit a binary classification task. With this example we aim to motivate the reader how a machine learning model can fail to fit the data distribution and how this could be passed to an external source, such as an auxiliary model with additional information, or our problem of interest, to a human expert.

First we check under the simplest setup: two class-independent and well-separated clusters. As depicted in Figure 4.2a the separation of the orange and blue classes is simple and can be

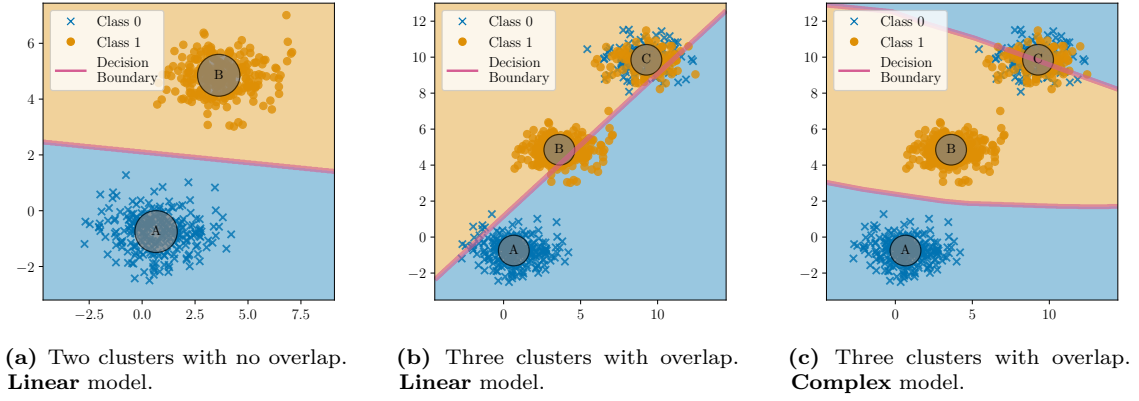


Figure 4.2: Toy example for binary classification with and without overlapping between classes: Binary classification example with two classes: orange (o) and blue (+). Figure (a) presents two non-overlapping clusters, while Figures (b) and (c) add a new extra cluster with overlap between the classes. Figure (a) shows an easy example with evident separation between classes. Figures (b) and (c) show how neither a linear classifier nor a complex model (*MLP classifier*) are able to data distribution with the new overlapping cluster. This motivates the idea of *rejecting* the samples for the overlapping cluster. Example available [here](#)⁷.

easily done using a linear model minimizing ϕ_{OVA} . However, when we include a new cluster where 50% of the samples are orange, and 50% are blue, the problem becomes trickier. When a linear model (Figure 4.2b) tries to solve the problem, we see that not only we fail to separate the two previous clusters, but also fail to separate the overlapping samples from the third cluster. With a complex model we recover the separation of the two previous clusters, but again, we are not able to separate the samples in the third cluster. By **complex**, we refer to a model with non-linear activations, *e.g.* a MLP classifier with three layers with a non-linear activation function.

The reader might be driven to think that instead of *giving up* into the design of an optimal model able to fit difficult distributions and pass the job to an additional source of information, one could obviously lucubrate about possible improvements of the base model. That is, one could incorporate Bayesian reasoning and include informative priors which favor a more spread or separation of the classes; one could also apply kernel methods such as SVM and try to solve the problem in the dual space, *etc.* However, we remind the reader that the aim of the thesis is to pursue the optimal human-machine interaction, and we want to design frameworks which are trust-worthy and safety in critical applications. Therefore, we suggest the reader to pause and think about the following suggestion: *what if, instead of relying on our model to fit the whole data distribution, we can have the option to abstain when needed?* This will be presented in the next section under the research field of *rejection learning* and will serve as common thread for the rest of the thesis.

4.2 Learning to Defer Background and Related Work

The concept of learning to defer (L2D) represents a significant advancement beyond the traditional learning to reject paradigm in machine learning. *Learning to reject* (L2R), or rejection learning, enables classifiers to abstain from making predictions and incurs a fixed cost, whereas learning to defer, known as L2D, takes the notion further by considering the interaction between the classifier and downstream decision-makers. Therefore, we can picture learning to defer as a more generalized framework of rejection learning.

⁷<https://github.com/dbarrejon/thesis-l2d>

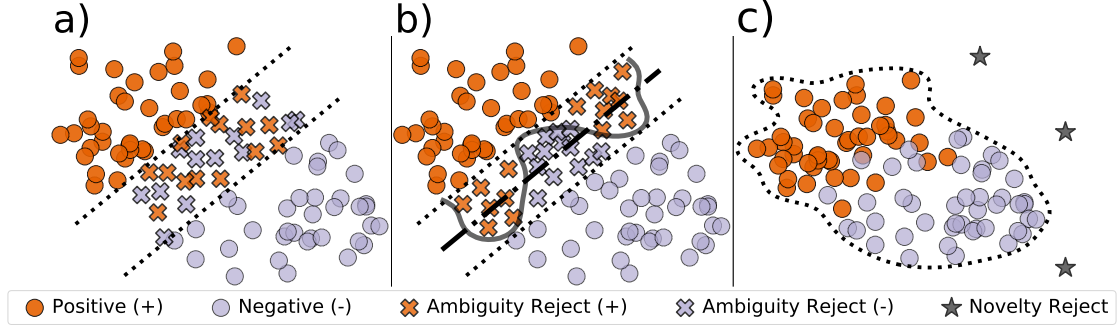


Figure 4.3: Example of rejection in a binary classification scenario with overlapping: The dotted lines represent the rejector region. Figure (a) depicts the ambiguity region, (b) compares the solution of an the ideal (solid line) vs. a linear rejector (dashed-dotted) and Figure (c) illustrates rejection for novel samples. Figures (a),(b) depict the case where samples *inside* the rejection region are rejected, and (c) the case where samples *outside* the rejector (stars) are rejected. Figure extracted from Hendrickx et al. (2021).

4.2.1 Learning to Defer within Rejection Learning

Rejection Learning Rejection Learning is a learning framework in which a system can choose not to predict for some samples. The system can choose not to make a prediction for samples that are far away from the training data (distance rejection), or the samples on which the model is not confident enough (ambiguity rejection). These two intuitions of rejection are exemplified in Figure 4.3 (Hendrickx et al., 2021)⁸, where the samples which are in the overlapping region could be rejected, as well as those samples very far from the data distribution. The origins of this problem can be traced back to the work developed by Chow (1957), where he studied the optimal rejection rule for a *fixed* cost τ , commonly referred to as the Chow rule:

$$h^*(x) = \begin{cases} \text{reject if } \max_{y \in Y} \mathbb{P}(y = y|x) \leq \tau \\ \arg \max_{y \in Y} \mathbb{P}(y = y|x) \text{ otherwise.} \end{cases} \quad (4.20)$$

From this equation we can read that we will reject if the model is not confident enough (with respect to the cost τ). In following works, Chow (1970) investigated the trade-off between accuracy and the rejection rate, highlighting the benefits of rejection learning in consequential decision-making applications with high costs for incorrect predictions. This set the path to new follow-up approaches which can be categorized in two types: 1) **confidence-based** (Bartlett and Wegkamp, 2008; Yuan and Wegkamp, 2010; Jiang et al., 2018; Grandvalet et al., 2009; Ramaswamy et al., 2018; Ni et al., 2019) and 2) **classifier-rejector** (Cortes et al., 2016a,b).

Confidence-based Algorithms Confidence-based methods are mainly based on taking into account the uncertainty in predictions and abstaining from making a prediction when the classifier's confidence falls below a certain threshold. In this case the design of the rejector becomes simpler: finding the optimal threshold. Regarding binary classification, Yuan and Wegkamp (2010) demonstrate that standard binary classification algorithms, which rely on strictly proper composite losses like logistic loss, exponential loss, and least squares loss, provide consistent algorithms for rejection learning. However, designing such algorithms for general multiclass classification problems is not a straightforward task and requires additional considerations.

⁸The authors highlight the work by Hendrickx et al. (2021) as a starting reference read for rejection learning.

Ramaswamy et al. (2018) further expand upon these findings and offer consistent algorithms for a general classification scenario.

The first intuitive approach would be, from a standard classifier (Section 4.1), use the uncertainty captured by the model to defer to an expert. This idea is used by Raghu et al. (2019), where they present a method which involves training a classifier model and using its predictive uncertainty to defer to an expert. This classifier is trained in a standard manner without any additional procedures for deferral. The next step would be incorporating in some way the expert’s uncertainty. This is the path that Bansal et al. (2021) followed: a confidence method that begins by estimating the probability of the expert being correct, $p(y = m)$. However, this estimate is independent of the input sample \mathbf{x} , meaning that $p(m = y|\mathbf{x}) = p(m = y)$.

Classifier-Rejector Algorithms Instead of learning one function which provides uncertainty estimates and an optimal threshold that determines whether a sample is *dangerous* to be predicted by the system, we could simultaneously learn two distinct functions: a classifier and a rejector altogether. This intuition has been coined in the rejection learning literature as *classifier-rejector* algorithms. In Cortes et al. (2016a) and Cortes et al. (2016b), the authors link the two presented rejection learning approaches by arguing that classifier-rejector methods offer a broader scope than confidence-based methods, resulting in more robust algorithms; and also theoretically prove the viability of these approaches in binary classification. However, extending this theory to the more general case of multi-class classification has proven challenging, as discovered by Ni et al. (2019). This motivates the search of new surrogate losses for general classification that can be used in rejection learning for Charoenphakdee et al. (2021).

4.2.2 Learning to Defer in the Context of Human-Machine Collaboration

It should be a *de facto* belief that the supreme goal of machine learning is not to be an independent agent but rather a complimentary agent to humans. So the dichotomy of either human or machine should not be considered. Hopefully, several works have explored *human-in-the-loop* learning with decision makers in mind, e.g. (De et al., 2021, 2020; Bansal et al., 2021). Raghu et al. (2019) show that algorithms designed for both prediction and triage can outperform systems composed of just machine or just humans. This pursuit of human-AI complementarity has been also considered as an objective itself, and several works investigated on the difficulties of these problems. In De et al. (2020) the authors prove that ridge regression under human-assistance is NP-hard and then derive a new objective function as a difference of nondecreasing submodular functions. In their following-up, work De et al. (2021) prove NP-hardness on margin-based classifiers. In general, the primary goal of all these works is to promote human-machine collaboration. Bansal et al. (2021) also followed this line by optimizing a the expected utility of the machine and the expert working as a team. using a confidence-based method. From all these works we can tell that having a well-defined notion of uncertainty in the predictions of machine learning systems is a critical factor in establishing trust in their accuracy and reliability. But these systems can further leverage from human uncertainty to be more robust (Peterson et al., 2019). Kerrigan et al. (2021) worry about how combining the model probabilities with the human predictions affects calibration, and propose a solution using confusion matrices. Furthermore, Steyvers et al. (2022) follow a Bayesian approach to investigate human-AI collaboration.

Human-AI complementarity can be viewed under different definitions. Similar to the L2D spirit, [Meresht et al. \(2020\)](#) investigate on how we can learn to switch between human and machines to allow existing reinforcement learning agents to operate under different automation levels. Also [Okati et al. \(2021\)](#) studied the problem of L2D in more general settings than classification, and derived the optimal deferral policy. But designing algorithms that enable human and machine collaboration is a hard task. [Donahue et al. \(2022\)](#) investigated the theoretical conditions for the achievement of complementarity in human-AI systems. They presented impossibility results that demonstrate scenarios where complementarity cannot be achieved by a human-AI system. Notably, their findings indicate that achieving complementarity is more attainable when there is substantial variation in errors between the human and algorithm across different samples. Also [Liu et al. \(2022\)](#) introduce a new extension of learning to defer, so called Learning to Defer with Uncertainty (LDU), which is based on a two-stage algorithm that involves training an ensemble of classifiers in the first stage and using an L2D system in the second stage, which incorporates the predictions and *uncertainties* of the classifiers to determine whether to rely on the classifier’s prediction or defer to the expert.

These findings shed light on how the learning to defer framework facilitates the natural emergence of complementarity and the division of labor. With knowledge of the human’s expertise, the model can adapt accordingly to complement the human’s abilities. This allows the model to focus on providing predictions for samples with lower error rates than the human, thereby fostering complementarity in the human-AI system.

4.3 Learning to Defer

Rejection learning has emerged as a promising paradigm for safety-critical applications. However, traditional rejection learning approaches tend to overlook the crucial role of downstream experts, who ultimately bear the responsibility of making decisions for the rejected samples. Furthermore, these experts might also fail in the decision task. In addressing this limitation, [Madras et al. \(2018\)](#) introduced a novel adaptive rejection learning framework known as *learning to defer* (L2D).

Learning to Defer is a promising learning system for high-stake decision making as both experts and the machine learning systems can work together based on their strengths and weaknesses, thereby alleviating some trust and predictability issues in decision making. Specifically, building upon the prior work of [Cortes et al. \(2016a\)](#), [Madras et al. \(2018\)](#) incorporate a rejector function $r(\mathbf{x})$ to model the decision of whether to defer to the expert $r(\mathbf{x}) = 1$ or not ($r(\mathbf{x}) = 0$). The elegance of the L2D framework is in its ability to model the expert’s *knowledge* without having to replicate the expert’s predictive performance. Instead of having a fixed cost for deferring as the confidence-based approaches presented before, L2D allows the system not just reject but also learns to adapt itself to some downstream expert. L2D falls under the umbrella of *classifier-rejector* approaches we commented before: we have a single system that can both behave as a classifier or a rejector, which can also be seen as a *meta-classifier* deciding which samples should be passed to the expert.

However, the learning objective for learning to defer is a difficult optimization problem. The first work to provide a consistent surrogate loss in the L2D literature was [Mozannar and Sontag \(2020\)](#). [Mozannar and Sontag \(2020\)](#) study the multi-class classification problem and theoretically prove that the loss proposed by [Madras et al. \(2018\)](#) is inconsistent. Other L2D approaches did

⁸<https://husseinmozannar.github.io/publication/mozannar-2020-consistent/>

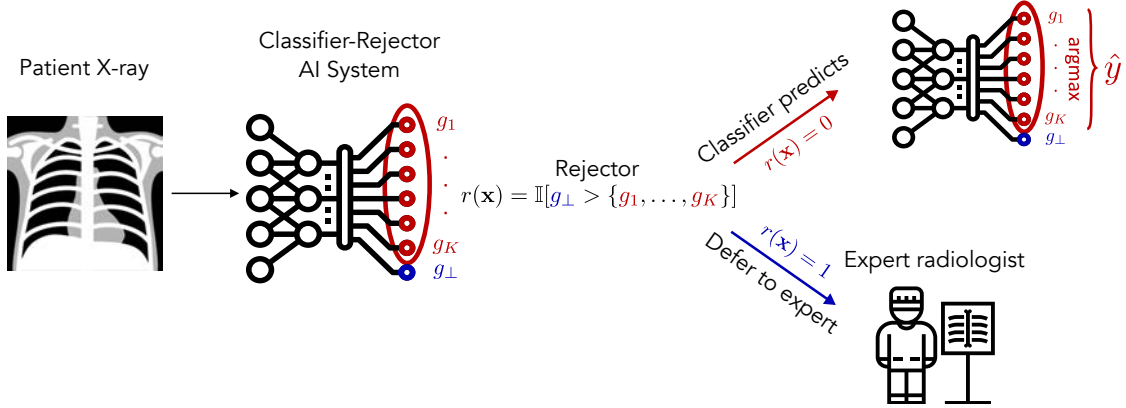


Figure 4.4: *L2D diagram.* We augment the label space $\mathcal{Y}^\perp = \mathcal{Y} \cup \{\perp\}$, where \perp is the deferral dimension. In practice, g_\perp can serve as a proxy estimate of the expert’s decision probability. If the deferral dimension \perp is greater than the class dimensions g_1, \dots, g_K , then we defer to the expert; in the opposite case, the decision is taken by the classifier. Figure adapted from Mozannar and Sontag (2020) website figure⁹.

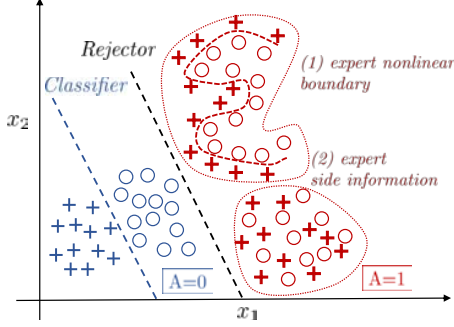
not come with consistency guarantees (Raghu et al., 2019; Wilder et al., 2020; Pradier et al., 2021; Okati et al., 2021; Liu et al., 2022). The next work to propose a proper consistent surrogate loss for L2D is the work done by Verma and Nalisnick (2022). Motivated by the importance of calibration in hybrid intelligence systems, Verma and Nalisnick (2022) analyze the validity of the loss presented by Mozannar and Sontag (2020) and find out that their proposal is not guaranteed to produce valid probabilities due to the use of softmax parametrization. Hence, they propose a new consistent surrogate loss for L2D based on one-vs-all classifiers.

In the following sections we will provide the technical background of L2D and comment how the two parametrizations: 1) *softmax* parametrization (Mozannar and Sontag, 2020) and 2) *one-vs-all* (OvA) parametrization (Verma and Nalisnick, 2022) behave with respect to expert’s confidence calibration.

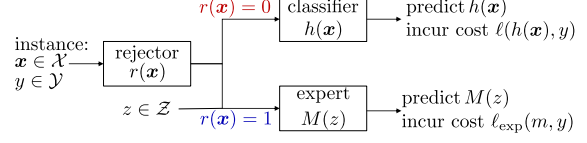
4.3.1 Preliminaries

Data We first define the data for multiclass L2D with one expert. Let \mathcal{X} denote the feature space, and let \mathcal{Y} denote the label space, which we assume to be a categorical encoding of multiple (K) classes. $\mathbf{x}_n \in \mathcal{X}$ denotes a feature vector, and $y_n \in \mathcal{Y}$ denotes the associated class defined by \mathcal{Y} (1 of K). The L2D problem also assumes that we have access to (human) expert demonstrations. Denote the expert’s prediction space as \mathcal{M} , which is usually taken to be equal to the label space: $\mathcal{M} = \mathcal{Y}$. Expert demonstrations are denoted $m_n \in \mathcal{M}$ for the associated features \mathbf{x}_n . The combined N -element training sample is $\mathcal{D} = \{\mathbf{x}_n, y_n, m_n\}_{n=1}^N$, where $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$ and $x_i, y_i \sim \mathcal{D}$, and $m_i \in \mathcal{M}, m_i \sim \mathcal{P}$. It is often assumed that the human has prior training or access to additional information not available to the classifier (Z) apart from input \mathcal{X} , i.e. $m_i \sim M|X, Z$. Compared to the standard classification setup we presented in Section 4.1, notice \mathcal{D} is the same with the addition of the expert’s *predictions*.

⁹<https://husseinmozannar.github.io/publication/mozannar-2020-consistent/>



(a) Binary problem with labels $\{+, o\}$. Extracted from Mozannar and Sontag (2020).



(b) Description of a L2D system. Adapted from Mozannar et al. (2023).

Figure 4.5: *Intuitive example of binary classification using L2D and description of an L2D system:* Figure (a) shows a binary problem with labels $\{+, o\}$. The expert’s region is colored in red, so the classifier is expected to fit the data in the blue region, which is linearly separable. The dashed-black line represents the rejector separating both groups. Figure (b) depicts the standard L2D pipeline: the rejector $r(\mathbf{x})$ decides who should predict, the classifier $h(\mathbf{x})$ or the expert $M(z)$. Notice that the expert predicts using side information z .

Intuition In Section 4.1.4 we saw that both a linear and a complex classifier alone could not fit some data distribution. In Figure 4.5a we show how L2D would tackle that problem. Assume we have two populations: the blue ($A = 0$) and the red ($A = 1$); and also two classes: $\{+, o\} = \mathcal{Y}$. The data \mathcal{X} is generated conditional on the target y and the population A : $\mathbf{x}|(y = y, A = 0)$ is normally distributed $\mathcal{N}(\mu_{y,0}, \Sigma)$, and $\mathbf{x}|(y = y, A = 1)$ consists of two clusters: cluster (1) normally distributed, but with non-separable means and (2) only separable by complex non-linear boundaries. We assume that the expert can find the nonlinear boundary for cluster (1) and has side information \mathcal{Z} to separate cluster (2). The optimal behavior for our L2D framework would be to learn a rejector r that can effectively separate populations $A = 1$ and $A = 0$, hence allowing the expert to predict at region $A = 1$ where the expert is more likely to be correct and allowing the classifier predict at region $A = 0$. However, we must say that in practice there are trade-offs as we will show later.

Models and Learning Mozannar and Sontag (2020)’s and Verma and Nalisnick (2022)’s L2D frameworks compose two models: a classifier and a rejector (Cortes et al., 2016a,b). Denote the *classifier* as $h : \mathcal{X} \rightarrow \mathcal{Y}$ and the *rejector* as $r : \mathcal{X} \rightarrow \{0, 1\}$. When $r(\mathbf{x}) = 0$, the classifier makes the decision. When $r(\mathbf{x}) = 1$, the system defers the decision to the human. When the classifier makes the prediction, then the system incurs a loss $\ell(h(\mathbf{x}), y)$. When the human makes the prediction (i.e. $r(\mathbf{x}) = 1$), the system incurs a loss $\ell_{\text{exp}}(m, y)$. Using the rejector to combine these losses, we have the overall classifier-rejector loss:

$$L(h, r) = \mathbb{E}_{\mathbf{x}, y, m} [(1 - r(\mathbf{x})) \ell(h(\mathbf{x}), y) + r(\mathbf{x}) \ell_{\text{exp}}(m, y)], \quad (4.21)$$

where the rejector is acting as an indicator function that controls which loss to use. The rejector can be interpreted as a *meta-classifier*, determining which inputs are appropriate to pass to $h(\mathbf{x})$. While this formulation is valid for general losses, the canonical 0-1 loss is of special interest for classification tasks:

$$L_{0-1}(h, r) = \mathbb{E}_{\mathbf{x}, y, m} [(1 - r(\mathbf{x})) \mathbb{I}[h(\mathbf{x}) \neq y] + r(\mathbf{x}) \mathbb{I}[m \neq y]], \quad (4.22)$$

where \mathbb{I} denotes an indicator function that checks if the prediction (either from the classifier h or the expert m) and label are equal. Upon minimization, the resulting Bayes optimal classifier and rejector satisfy (Mozannar and Sontag, 2020; Verma and Nalisnick, 2022):

$$h^*(\mathbf{x}) = \arg \max_{y \in \mathcal{Y}} \mathbb{P}(y = y|\mathbf{x}), \quad (4.23)$$

$$r^*(\mathbf{x}) = \mathbb{I} \left[\mathbb{P}(m = y|\mathbf{x}) \geq \max_{y \in \mathcal{Y}} \mathbb{P}(y = y|\mathbf{x}) \right], \quad (4.24)$$

where $\mathbb{P}(y|\mathbf{x})$ is the probability of the label under the data generating process, and $\mathbb{P}(m = y|\mathbf{x})$ is the probability that the expert is correct. The expert likely will have additional knowledge not available to the classifier, which possibly allows the expert to outperform the Bayes optimal classifier. When the aforementioned conditions are satisfied, the rejector and classifier can be considered Bayes optimal, indicating that they are the classifiers that exhibit the lowest expected error. Essentially, the classifier selects the label that has the highest posterior probability of being accurate, while the rejector chooses between the classifier and the expert based on the one with the highest posterior probability of being correct.

4.3.2 Softmax Surrogate Loss: Single Expert

Mozannar and Sontag (2020) proposed the first consistent surrogate loss for L_{0-1} , meaning that its minimizers agree with the Bayes optimal solutions in Equation 4.23. They accomplish this by first unifying the classifier and rejector via an augmented label space that includes the rejection option. Formally, this label space is defined as $\mathcal{Y}^\perp = \mathcal{Y} \cup \{\perp\}$ where \perp denotes the rejection option. Secondly, Mozannar and Sontag (2020) use a reduction to cost sensitive learning that ultimately resembles the cross-entropy loss for a softmax parameterization. Let $g_k : \mathcal{X} \mapsto \mathbb{R}$ for $k \in [1, K]$ where k denotes the class index, and let $g_\perp : \mathcal{X} \mapsto \mathbb{R}$ denote the rejection (\perp) option. These $K + 1$ functions are then combined in the following softmax-based, point-wise surrogate loss:

$$\begin{aligned} \Phi_{\text{SM}}(g_1, \dots, g_K, g_\perp; \mathbf{x}, y, m) = & -\log \left(\frac{\exp\{g_y(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}^\perp} \exp\{g_{y'}(\mathbf{x})\}} \right) \\ & - \mathbb{I}[m = y] \log \left(\frac{\exp\{g_\perp(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}^\perp} \exp\{g_{y'}(\mathbf{x})\}} \right). \end{aligned} \quad (4.25)$$

The intuition is that the first term aims to maximize the function g_k associated with the true label. Then, the second term maximizes the rejection function g_\perp but only if the expert is correct. When the expert makes a mistake ($\mathbb{I}[m = y] = 0$), the loss is based on the cross-entropy (CE) loss with the target. In such cases, if the predicted probability of the defer option $g_\perp(\mathbf{x})$ is high, the predicted probability of the correct option $g_y(\mathbf{x})$ must be lower, resulting in a higher loss. On the other hand, when the expert agrees with the target ($\mathbb{I}[m = y] = 1$), the learner faces a dilemma of whether to trust the expert's judgment or make its own prediction.

At test time, the classifier is determined by selecting the maximum value among k within the range $[1, K]$; and the rejector decides to defer if the deferral dimension g_\perp is greater than any other label dimension g_k

$$\hat{y} = h(\mathbf{x}) = \arg \max_{k \in [1, K]} g_k(\mathbf{x}) \quad (4.26)$$

$$r(\mathbf{x}) = \mathbb{I}[g_\perp(\mathbf{x}) \geq \max_k g_k(\mathbf{x})]. \quad (4.27)$$

These Equations are similar to Equations 4.23, but notice now that we use the confidence estimates g_k output by the system, not the true underlying probabilities $\mathbb{P}(y|\mathbf{x})$, since these latter ones are attained at the true Bayesian optimal point.

Comments on the α parameter In the original paper, [Mozannar and Sontag \(2020\)](#) propose the incorporation of a hyperparameter $\alpha \in \mathbb{R}^+$ to modify the weighting of the classifier loss when the expert is correct so that the classifier can focus on the samples when the expert makes a mistake. For instance, in a clinical scenario where clinicians are pressed for time, we would like that our model focuses on those regions where the clinicians are more prone to fail. We can view the α parameter as a modulator of the degree of *responsibility* we want to assign to the expert, *e.g.* by choosing $\alpha < 1$. However, the desired consistency property for the surrogate loss is lost for all $\alpha \neq 1$. In practice this hyperparameter is searched to maximize system accuracy on a validation set.

For the rest of the thesis, we will always assume that $\alpha = 1$ to hold consistency and we refer the reader to the original softmax formulation paper ([Mozannar and Sontag, 2020](#)) or other works who have studied the influence of the α parameter ([Gantzert, 2021](#)).

4.3.3 One-vs-All Surrogate Loss: Single Expert

With the information we presented in Section 4.1.3, we can now turn our attention to our next surrogate loss of interest. [Verma and Nalisnick \(2022\)](#) proposed an alternative consistent surrogate for multiclass L2D based on a one-vs-all (OvA) formulation. Again assume we have $K + 1$ functions $g_1(\mathbf{x}), \dots, g_K(\mathbf{x}), g_\perp(\mathbf{x})$ such that $g : \mathcal{X} \mapsto \mathbb{R}$. Their one-vs-all (OvA) surrogate loss has the point-wise form:

$$\begin{aligned} \psi_{\text{OvA}}(g_1, \dots, g_K, g_\perp; \mathbf{x}, y, m) = & \phi[g_y(\mathbf{x})] + \sum_{y' \in \mathcal{Y}, y' \neq y} \phi[-g_{y'}(\mathbf{x})] \\ & + \phi[-g_\perp(\mathbf{x})] + \mathbb{I}[m = y] (\phi[g_\perp(\mathbf{x})] - \phi[-g_\perp(\mathbf{x})]), \end{aligned} \quad (4.28)$$

where $\phi : \{\pm 1\} \times \mathbb{R} \mapsto \mathbb{R}_+$ is a binary surrogate loss. For instance, when ϕ is the logistic loss, we have $\phi[f(\mathbf{x})] = \log(1 + \exp\{-f(\mathbf{x})\})$ (as comment in Section 4.1.1). The classifier and rejector can be computed following the same scheme as the softmax formulation, namely Equations 4.26 and 4.27 respectively. The motivation for this loss is that it produces better calibrated systems than those produced by the softmax-based loss. The softmax loss has a degenerate parameterization that causes it, in practice, to overestimate the experts probability of correctness ([Verma and Nalisnick, 2022](#)). This is an important issue, specially in sensitive scenarios where confidence estimates must be trust-worthy. We will address this in Section 4.4.

4.3.4 Realizable-Surrogate Loss: Complement when deferring

The surrogates ϕ_{SM} and ψ_{OvA} have been proven to satisfy consistency. However, recently a new extension of ϕ_{SM} has been proposed by [Mozannar et al. \(2023\)](#). In this work the authors show how previous works fall short in achieving a human-AI system with low misclassification error, even when a linear classifier and rejector with zero error exist (the realizable setting). This motivates the proposal of a new surrogate loss for learning to defer, namely the *realizable surrogate* loss. This formulation is a realizable-consistent surrogate loss and not fully consistent because it is differentiable but non-convex for \mathbf{g} , where $\mathbf{g} = [g_1, \dots, g_K]$. However, it is convex in $g_k \forall k = [1, \dots, K]$. The surrogate is defined as

$$\phi_{\text{RS}}(g_1, \dots, g_K, g_\perp; \mathbf{x}, y, m) = -2 \log \left(\frac{\exp(g_y(\mathbf{x})) + \mathbb{I}[m = y] \exp(g_\perp(\mathbf{x}))}{\sum_{y' \in \mathcal{Y}^\perp} \exp\{g_{y'}(\mathbf{x})\}} \right) \quad (4.29)$$

When the expert is incorrect, *i.e.* $\mathbb{I}[m = y] = 0$, the loss incentivizes the classifier to be correct (loss is the same as ϕ_{SM} in Equation 4.25). However, now if the expert is *i.e.* $\mathbb{I}[m = y] = 0$, the learner can choose between fitting the target or deferring, with no penalty for choosing one or the other. The classifier can still learn on those regions where the expert is correct, which was before penalized in the softmax surrogate loss ϕ_{SM} . As stated by Mozannar et al. (2023), this allows the classifier to *complement* the expert, rather than penalizing for not fitting the target even when deferring.

Allowing the classifier to complement the expert comes at a cost: the classifier could predict wrongly for the samples deferred to the human (recall the red region in Figure 4.5a where the classifier would fail in the cluster t(2) for region $A = 1$) and hence resulting in high errors for certain regions of the data domain. We can alleviate this by incorporating a cross-entropy term for the classifier in the following way:

$$\begin{aligned} \phi_{\text{RS}}^\alpha(g_1, \dots, g_K, g_\perp; \mathbf{x}, y, m) = & -\alpha \log \left(\frac{\exp(g_y(\mathbf{x})) + \mathbb{I}[m = y] \exp(g_\perp(\mathbf{x}))}{\sum_{y' \in \mathcal{Y}^\perp} \exp\{g_{y'}(\mathbf{x})\}} \right) \\ & - (1 - \alpha) \log \left(\frac{\exp\{g_y(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}} \exp\{g_{y'}(\mathbf{x})\}} \right). \end{aligned} \quad (4.30)$$

This new proposal with the new hyperparameter $\alpha \in [0, 1]$ is a convex combination of ϕ_{RS} and the cross-entropy of the classifier (notice the denominator only involves the label space \mathcal{Y} , rather than \mathcal{Y}^\perp). This new loss, a part from the commented behavior above, also permits using linear models and still be able to accommodate complex classification tasks. We just presented this new surrogate loss here for the sake of completeness and literature review of consistent surrogate losses in L2D; for further details and theoretical proofs we refer to the original paper.

4.3.5 Toy example for Learning to Defer surrogate losses

Reusing the MoG dataset from Section 4.1.4, we showcase in Figure 4.6 how softmax, OvA and Realizable surrogate losses perform under this task. Again, we have a binary classification problem with labels $\{\times, \circ\}$ with three clusters: two clusters with no class overlap (lower left and centered cluster) and one third cluster with class overlap (upper right cluster). For the experiment, we design an oracle is an oracle, *i.e.* predicts perfectly, on the third cluster, and randomly in clusters 0 and 1. The intuition is that, given this complex binary task, the expert has side information that allows him to predict perfectly the region of the space where the classifier would fail. For the OvA and softmax surrogates we used a complex model consisting on a linear model with a ReLU layer before the last linear layer, and for the Realizable surrogate a linear model consisting of the same model as OvA and softmax without the ReLU layer. We show in blue the regions of the space that are not deferred, or in other words, the regions where the classifier will be responsible for predicting, and in orange the deferred regions where the expert will predict. We see that OvA (Figure 4.6a) and Realizable (Figure 4.6c) surrogate losses behave similarly, effectively deferring to the expert the correct samples, while softmax (Figure 4.6b) also defers those regions of the space that will be considered as *ambiguous* or novel.

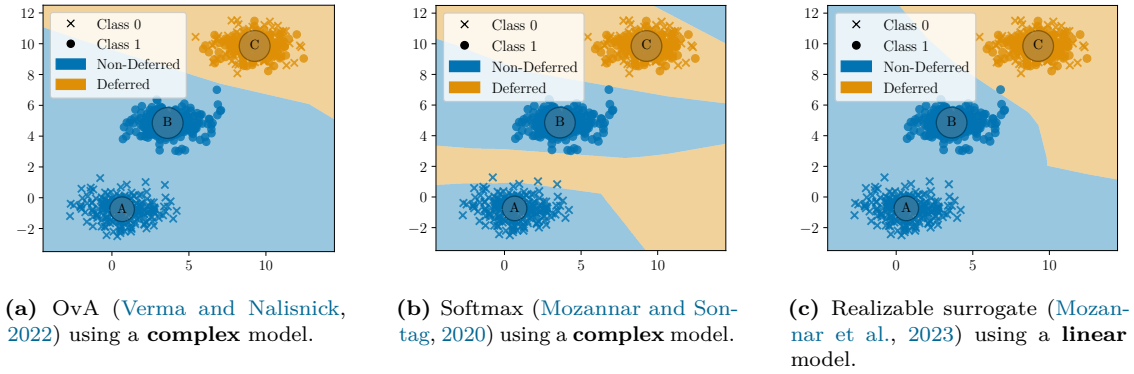


Figure 4.6: Comparison of L2D surrogate losses in a MoG example: For a binary task with labels $\{x, o\}$ and samples distributed in three clusters: clusters A and B for each class and linearly separable; and cluster C with class overlap between the samples. The expert predicts perfectly on cluster C and random elsewhere. We show how OvA (a), softmax (b) and Realizable surrogate (c) solve the task. While OvA and softmax use complex models (with a linear model they do not succeed), the new Realizable surrogate can separate the expert region with a linear model. Example available [here](#)¹⁰.

Which formulation to use? Not that obvious In terms of system accuracy we see that these three losses behave similarly: we end up deferring to the expert the samples that we would like the expert to predict. However, it is very interesting the difference between OvA and Realizable surrogate with respect to softmax. First, OvA and Realizable permit the classifier to account for responsibility in novel regions of the space, while softmax surrogate exhibits a more cautious behavior and also defers samples that approximate the intersection between the two linearly separable classes. In this case, the optimal choice of *which* L2D we should use is not evident, and will be constrained to the kind of problem we are solving. For example, we could pose two situations in medical scenarios:

1. If the problem we are facing is *sensible*, meaning that false negatives could lead to egregious mistakes, such as misdiagnosing if a given patient has cancer, then one could argue that the softmax formulation (Figure 4.6b) would be the right choice, since the gap between the two linearly-separable clusters could be also deferred.
2. If deferring to an expert shall be also associated to a cost of deferring, due to the availability of a doctor to diagnose a patient for example, then we could opt for the OvA or Realizable formulations, and trust the classifier in those novel regions.

Of course, this is just an example with the sole purpose of motivating the reader to think that the performance of each formulation is completely tied to the nature of the problem.

4.4 Confidence Calibration in Learning to Defer

When we want to assess the performance of a machine learning model on multi-class classification problems, the most usual metrics to report are accuracy, precision, recall, confusion matrix, just to name a few. But when the problem of interest is to be deployed in critical applications, such as autonomous driving or medical scenarios, we require not just a point estimate but we are also interested on quantifying the uncertainty of such estimate. For example, if an L2D system is being

¹⁰<https://github.com/dbarrejon/thesis-12d>

used for medical diagnosis, then a doctor will want to inspect the system’s probabilities, at least for purposes of sanity checking. Even more interesting, recent studies (Benz and Rodriguez, 2023) show that humans struggle to trust predictions based on confidence values. Therefore, if these confidence values, a part from being misjudged a priori by the experts, do not really reflect *good* probability estimates, then the whole human-machine collaboration pipeline becomes degenerate.

To help prevent such scenarios, we want our systems to be well *calibrated*. The output probabilities should reflect the true uncertainties of the model and human. In other words, the L2D system should be a good forecaster. If the system says the expert has a 70% chance of being correct $\mathbb{P}(m = y|\mathbf{x}_0) = 0.7$, then the expert should indeed be correct in about 70 out of 100 cases for samples similar to \mathbf{x}_0 . In order to achieve optimal decision making in scenarios with varying class distributions and misclassification costs, it is crucial for a classifier to generate accurately calibrated posterior probability estimates Kull et al. (2017). For a proper human-machine collaboration, certain factors such as transparency, trust, and fairness are essential (Cramer et al., 2008; Madras et al., 2018; Schmidt and Biessmann, 2020; Zhang et al., 2020). Bhatt et al. (2021) argue that we should pay special attention to uncertainty for fairness, transparency, decision-making, and trust in automated-AI systems. Calibration should be a must-have property in critical scenarios, even more important than accuracy (Sayin et al., 2022). In this study, (Tschandl et al., 2020) found that AI systems can mislead physicians into incorrect diagnoses, even when the doctor is originally confident. This further highlights the need of calibrated estimates.

Furthermore, calibration has gained significant attention in recent machine learning literature (Guo et al., 2017; Kull et al., 2019; Vaicenavicius et al., 2019; Nixon et al., 2019; Gupta and Ramdas, 2022; Minderer et al., 2021). The dominant methodology is to apply *post-hoc calibration*: fitting additional parameters on validation data to re-calibrate the formerly mis-calibrated model (Guo et al., 2017; Yu et al., 2022). These methods could potentially be applied in the L2D framework—such as, by adding a temperature parameter to the expert terms in our surrogate losses—but we are primarily interested in the native, ‘out-of-the-box’ calibration properties of the losses.

4.4.1 Our Notion of Confidence Calibration

For this thesis we adopt the notion of confidence calibration presented by Guo et al. (2017) a well-studied measure of the quality of confidence estimates in machine learning literature. However, given that calibration is a very hot topic in the machine learning community, one may find different definitions of calibration, such as the one proposed by Gupta and Ramdas (2022), where they propose using top-label calibration instead of confidence calibration in multi-class problems.

L2D literature has mainly focused on the overall performance of the system. However, we have made clear that calibration should also be studied in L2D framework. In the following section we present how the two consistent surrogate losses – softmax and OvA– find confidence estimates, with more emphasis on the estimates for the expert’s correctness probability $\mathbb{P}(m = y|\mathbf{x})$ and we use calibration as a measurement of *goodness* of such estimates. We also present the degenerate behavior that the softmax formulation exhibits (Mozannar and Sontag, 2020) and how it results in expert’s confidence estimates greater than one, as pointed out by Verma and Nalisnick (2022). We do the same for the OvA formulation Verma and Nalisnick (2022).

Calibration We next define the relevant notion of calibration. Throughout this thesis we will use *confidence calibration* (Dawid, 1982). This applies to any confidence estimate, but in the thesis we will mainly focus on the expert’s correctness probability. For an estimator of expert correctness $t_{\perp}(\mathbf{x}) : \mathcal{X} \mapsto (0, 1)$, we call t *calibrated* if, for any confidence level $c \in (0, 1)$, the actual proportion of times the expert is correct is equal to c :

$$\mathbb{P}(m = y \mid t_{\perp}(\mathbf{x}) = c) = c. \quad (4.31)$$

This statement should hold for all possible instances \mathbf{x} with confidence c . We can denote as t_y the confidence estimate of our true probability $t_y(\mathbf{x}) \approx \mathbb{P}(y = y|\mathbf{x})$ and t_{\perp} as the confidence estimate of the expert’s correctness probability $t_{\perp}(\mathbf{x}) \approx \mathbb{P}(m = y|\mathbf{x})$. However, we will see next if this actually holds for softmax and OvA formulations. Since expert correctness is a binary classification problem, distribution calibration, confidence calibration, and classwise calibration all coincide (Vaicenavicius et al., 2019). We can measure the degree of calibration using *expected calibration error* (ECE). In this case, the relevant ECE is defined as

$$\text{ECE}(t_{\perp}) = \mathbb{E}_{\mathbf{x}} |\mathbb{P}(m = y \mid t_{\perp}(\mathbf{x}) = c) - c|, \quad (4.32)$$

where $\mathbb{E}_{\mathbf{x}}$ is usually approximated with samples. Typically, the ECE is calculated by dividing predictions into bins of equal width based on their confidence levels. This measure is commonly represented graphically as reliability diagrams. For each bin, we examine whether the calibration equation (Equation 4.32) holds true. The disparity between the bin’s confidence and the observed accuracy is plotted, and the average of these differences is referred to as ECE (see Figure 4.7).

4.4.2 Softmax Parametrization: Single Expert

As commented above, we want that our confidence are well calibrated so that they reflect the true probabilities of our distribution. Therefore, we need to answer two questions: 1) how to get those confidence estimates and 2) are these confidence estimates well calibrated?.

How do we obtain the confidence estimates t_{\perp} and t_y for ϕ_{SM} ? For our calibration study we need to find the confidence estimates t_y and t_{\perp} . One could initially think that these estimates are just the softmax outputs of the model

$$t_{\perp}(\mathbf{x}) = \frac{\exp\{g_{\perp}(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}^{\perp}} \exp\{g_{y'}(\mathbf{x})\}} \quad (4.33)$$

$$t_y(\mathbf{x}) = \frac{\exp\{g_y(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}^{\perp}} \exp\{g_{y'}(\mathbf{x})\}}. \quad (4.34)$$

However, as pointed out by Verma and Nalisnick (2022), and following (Mozannar and Sontag, 2020)’s Theorem 1, we know the **Bayes optimal functions** $g_1^*, \dots, g_{\perp, J}^*$ relate to the true probabilities in the following way. For $\mathbb{P}(m = y|\mathbf{x})$ we get

$$\frac{\mathbb{P}(m = y|\mathbf{x})}{1 + \mathbb{P}(m = y|\mathbf{x})} = \frac{\exp\{g_{\perp}^*(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}^{\perp}} \exp\{g_{y'}^*(\mathbf{x})\}} = t_{\perp}^*(\mathbf{x}), \quad (4.35)$$

and for $\mathbb{P}(y = y|\mathbf{x})$ we get

$$\frac{\mathbb{P}(y = y|\mathbf{x})}{1 + \mathbb{P}(m = y|\mathbf{x})} = \frac{\exp\{g_y^*(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}^{\perp}} \exp\{g_{y'}^*(\mathbf{x})\}} = t_y^*(\mathbf{x}). \quad (4.36)$$

Rearranging the above equations obtain

$$\mathbb{P}(m = y|\mathbf{x}) = \frac{t_{\perp}^*(\mathbf{x})}{1 - t_{\perp}^*(\mathbf{x})} = \frac{\exp\{g_{\perp}^*(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}} \exp\{g_{y'}^*(\mathbf{x})\}} \quad (4.37)$$

$$\mathbb{P}(y = y|\mathbf{x}) = \frac{1}{1 - t_{\perp}^*(\mathbf{x})} \frac{\exp\{g_y^*(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}^{\perp}} \exp\{g_{y'}^*(\mathbf{x})\}} = \frac{\exp\{g_y^*(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}} \exp\{g_{y'}^*(\mathbf{x})\}}. \quad (4.38)$$

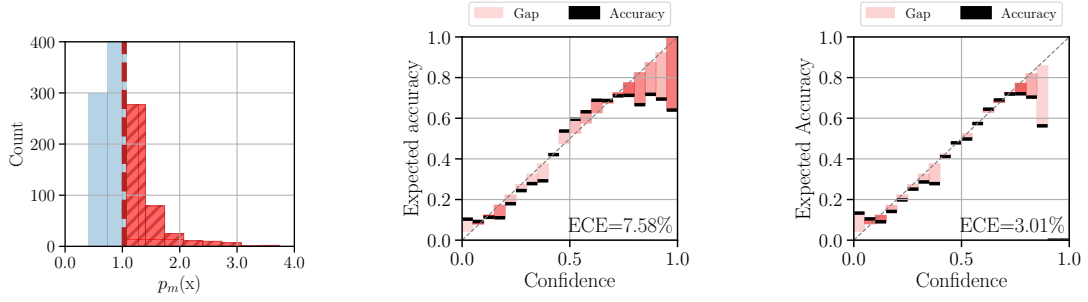
Notice that, in the denominator in Equations 4.37 and 4.38 and , we don't have \mathcal{Y}^{\perp} , but \mathcal{Y} , *i.e.* we use the label space without the deferral dimension. It is easy to check that $t_y(\mathbf{x}) \approx \mathbb{P}(y = y|\mathbf{x})$ provides valid estimates since $\sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y'|\mathbf{x}) = 1$ and also $\mathbb{P}(y = y|\mathbf{x}) \in [0, 1]$. However, as $t_{\perp}(\mathbf{x})$ approaches 1, $\mathbb{P}(m = y|\mathbf{x})$ goes to infinity, *i.e.* the expert's correct probability gets unbounded $\mathbb{P}(m = y|\mathbf{x}) \in [0, \infty)$. Of course, the fact that this probability estimate goes to infinity will have consequences with respect to calibration.

How good are the confidence estimates for expert's correctness t_{\perp} ? Now we are concerned about how *good* our expert's correctness estimates are under the softmax parametrization. The goal is to empirically prove that effectively $\mathbb{P}(m = y|\mathbf{x})$ can go to infinity. Luckily, [Verma and Nalisnick \(2022\)](#) already proved this. We decided to include Figure 3.2. a),b) from the original paper in this thesis under Figure 4.7 and we refer the reader to their work for more details. From Figure 4.7a we can see how the estimate of the expert's correctness probability can be bigger than one $\mathbb{P}(m = y|\mathbf{x}) > 1$ and Figure 4.7b validates the hypothesis: the calibration results are very poor for the softmax parametrization. The average ECE for the expert's correctness estimate is very high and the red-shaded region shows how, for confidence values from 0.9 to 1 the expected accuracy is smaller.

A part from the intrinsic limitation of the softmax parametrization in the context of L2D, many other works worry about the generalized and default usage of the softmax parametrization in multi-class classification problems. In [Pearce et al. \(2021\)](#) the authors investigate the contradiction between neural networks' inability to effectively increase uncertainty for out-of-distribution data and the limited success of using softmax confidence as a proxy for uncertainty. [Wang et al. \(2021\)](#) show that simply using the largest model output under a softmax parametrization can be limiting, specially in OOD applications. In [Sensoy et al. \(2018\)](#) authors show that softmax provides unreliable uncertainty estimation and propose using a dirichlet distributions to quantify uncertainty instead.

4.4.3 One-vs-All Parameterization: Single Expert

The softmax parameterization has been shown to exhibit poor calibration properties, which are not only desirable in learning to defer algorithms, but in any machine learning framework. The unbounded estimate for the expert's correctness probability was effectively found by [Verma and Nalisnick \(2022\)](#), who proposed the OvA parameterization to account for the miscalibration problem of the softmax. Next, we follow the same strategy as before, we show 1) how are the confidence estimates t_{\perp} and t_y found under this parameterization and 2) check if these estimates, specially the expert's correctness estimate t_{\perp} are calibrated.



(a) Empirical distribution of $\mathbb{P}(m = y|\mathbf{x})$, denoted as $p_m(\mathbf{x})$.

(b) Reliability diagram and ECE for **softmax** formulation.

(c) Reliability diagram and ECE for **OvA** formulation.

Figure 4.7: Calibration of Softmax and OvA parametrization Subfigure (a) reports the empirical values for $\mathbb{P}(m = y|\mathbf{x})$ for the CIFAR-10 dataset, where we corroborate that for certain samples the expert probability is bigger than 1 (denoted in red). Subfigures (b) and (c) show the reliability diagram and the ECE for softmax and OvA respectively, when $\mathbb{P}(m = y|\mathbf{x})$ is capped to $(0, 1]$. The red-shaded region represents the proportion of samples on the bin (the darker, the more samples it has). We see that the degenerate behavior of softmax effectively results in worse calibration results (bigger ECE is worse). Results from original paper (Verma and Nalisnick, 2022).

How do we obtain confidence estimates from ϕ_{OvA} ? As effectively proved by Verma and Nalisnick (2022) (Appendix C.2 from the original paper), we know that the OvA parametrization provides valid probability estimates for $\mathbb{P}(m = y|\mathbf{x})$ and for $\mathbb{P}(y = y|\mathbf{x})$ for all labels $y \in \mathcal{Y}$

$$\mathbb{P}(m = y|\mathbf{x}) = t_{\perp}^*(\mathbf{x}) = \sigma(g_{\perp}^*(\mathbf{x})) = \frac{1}{1 + \exp\{-g_{\perp}^*(\mathbf{x})\}} \quad (4.39)$$

$$\mathbb{P}(y = y|\mathbf{x}) = t_y^*(\mathbf{x}) = \sigma(g_y^*(\mathbf{x})) = \frac{1}{1 + \exp\{-g_y^*(\mathbf{x})\}} \quad (4.40)$$

This holds true because the sigmoid function $\sigma(\cdot)$ is the inverse-link function for the logistic loss ϕ (Reid and Williamson, 2010) and because now both estimates t_y and t_{\perp} have valid ranges between $[0, 1]$. Therefore, $\mathbb{P}(m = y|\mathbf{x}) \approx t_{\perp}(\mathbf{x})$ and $\mathbb{P}(y = y|\mathbf{x}) \approx t_y(\mathbf{x})$.

How good are these confidence estimates? In Figure 4.7c we included the results from Verma and Nalisnick (2022) using the new OvA parametrization for L2D. Now the expected accuracy is much closer to the real confidence estimate and the ECE drops about 50% compared to the softmax parametrization (Figure 4.7b), 7.58% for softmax versus 3.01% for OvA.

However, one significant drawback of the suggested one-vs-all formulation is that it restricts us from computing normalized probabilities for all classes. Instead, we can only estimate the probability of each output label in isolation. Consequently, while we can assess the confidence calibration of the OvA classifier, we are unable to evaluate its distribution calibration. However, as pointed out in the original paper (Verma and Nalisnick, 2022), in real-world scenarios, it's nearly impossible to achieve perfect distribution calibration Zhao et al. (2021). Hence using the One-vs-All (OvA) formulation can be a practical trade-off. It allows us to have a reliable estimator for the expert's correctness probability $\mathbb{P}(m = y|\mathbf{x})$ and helps achieve confidence calibration.

4.5 Summary of the Chapter

This chapter introduced the Learning to Defer framework under a multi class classification problem

1. We presented the notation for a **general multiclass classification** problem, together with a motivating example where *only* a classifier fails on the classification task.
2. We presented the **background** and related work for learning to defer.
3. We introduced **learning to defer** (L2D) and the two main consistent surrogate losses yet proposed: the *softmax* formulation (Mozannar and Sontag, 2020) and the *OvA* (Verma and Nalisnick, 2022) formulation.
4. We introduced the concept of confidence calibration, particularly focusing on **expert's confidence calibration**. We discuss how the *softmax* formulation can lead to miscalibrated solutions and explain how the *OvA* approach helps overcome this problem

But on a bright fall morning, I'm with it
 I stood a little while within it
 Man, you have to know
 Know the way

iMi — Bon Iver ▶

5

Learning to Defer to Multiple Experts

Contents

5.1	L2D To Multiple Experts	79
5.1.1	Softmax Surrogate Loss: Multiple Experts	80
5.1.2	One-vs-All Surrogate Loss: Multiple Expert	80
5.1.3	Toy example with Multiple Experts L2D	81
5.1.4	Inconsistency of Mixture of Experts	82
5.2	Confidence Calibration of Expert Confidence	82
5.2.1	Softmax Parameterization: Multi-Expert	83
5.2.2	One-vs-All Parameterization: Multi-Expert	84
5.3	Ensembling Expert with Conformal Inference	84
5.3.1	Conformal Inference	84
5.3.2	Conformal Inference on Sets of Experts	85
5.3.3	Choice of Hyperparameters for Regularized Conformal Ensembles	86
5.4	Related Work	87
5.5	Experiments	88
5.5.1	Overall System Accuracy	88
5.5.2	Confidence Calibration	90
5.5.3	Conformal Ensembles	92
5.6	Conclusions	94

SOLVING complex problems often requires the involvement of multiple experts (Fay et al., 2006). For example, in healthcare, serious illnesses require the patient be treated by *multiple* specialist and such task requires multiple opinions, as when a team of doctors consults on a difficult medical diagnosis. These experts may have non-overlapping specialties, such as in a large construction project that requires the advice of engineers, architects, geologists, lawyers, etc. Synthesizing information from these various experts creates an additional challenge—even for humans—as it can be unclear how to combine the information from the disparate sources.

In Chapter 4 we presented the *Learning to defer* (L2D) framework where a *rejector* model acts as a meta-classifier, predicting whether the downstream classifier or human is more likely to make the correct decision for a given input. Yet existing L2D frameworks do not obviously accommodate additional experts. For instance, the rejector’s job becomes more challenging when there are multiple experts. In human-machine collaboration, the primary challenge is

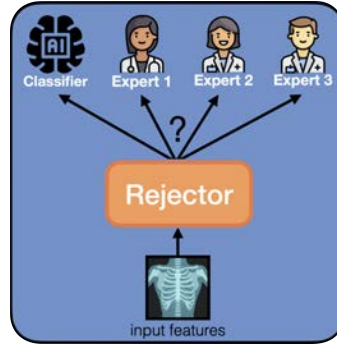


Figure 5.1: *Learning to Defer to Multiple Experts diagram:* For a given input, now we have the option to defer not only to either the classifier or one expert, but to a set of experts. Figure from Verma, Barrejón, and Nalisnick (2023) AISTAT’s poster¹.

often thought to be when to rely on the machine vs the human. Yet when there are multiple experts, there are two decisions to be made: *when to defer* and *to whom to defer*. Some experts may perform better than the model but perhaps others will not. Thus assessing and monitoring expert quality is an important sub-task.

Contribution In our work (Verma, Barrejón, and Nalisnick, 2023) we develop the statistical foundations of **multiclass L2D with multiple experts**. Specifically, we address the following open problems:

1. Deriving a consistent surrogate loss under the multiple expert setting.
2. Studying whether these systems are confidence calibrated.
3. Developing a principled technique for ensembling expert decisions.

The first and second contributions ensure the soundness of the optimization problem and resulting downstream decision making. For the first contribution, we derive two consistent surrogates—one based on a softmax parameterization, the other on a one-vs-all (OvA) parameterization—that are analogous to the single expert losses proposed by Mozannar and Sontag (2020) and Verma and Nalisnick (2022), respectively. We then study the frameworks’ ability to estimate $\mathbb{P}(m_j = y|\mathbf{x})$, the probability that the j th expert will correctly predict the label for \mathbf{x} . Theory shows the softmax-based loss causes mis-calibration to propagate between the estimates while the OvA-based loss does not (though in practice, we find there are trade offs). Our third contribution, expert ensembles, follows from our study of calibration, as we propose a conformal inference procedure for selecting a subset of experts. We empirically validate our methods on the tasks of image classification, galaxy categorization, skin lesion diagnosis, and hate speech detection. We find that our consistent losses result in superior accuracy and calibration when compared to existing systems based on (inconsistent) mixtures of experts (Hemmer et al., 2022).

Notes on the contributions: This chapter is based on two contributions: the first contribution (Verma, Barrejón, and Nalisnick, 2022) [ICML 2022 Workshop on Human-Machine Collaboration and Teaming], where we extended softmax and OvA surrogate losses to multiple experts and analyze their calibration properties, and the second (Verma, Barrejón, and Nalisnick, 2023) [Artificial Intelligence and Statistics (AISTATS), 2023] where expanded the work to more real-world dataset, and incorporated the conformal ensemble of experts. However, the toy example for multiple experts described in Section 5.1.3 did not appear in the mentioned contributions.

¹<https://virtual.aistats.org/media/PosterPDFs/AISTATS%202023/5689.png?t=1681813237.7885134>

5.1 L2D To Multiple Experts

In the previous chapter we present the two consistent surrogate losses for L2D that haven't been proposed so far in the literature. We now turn to the multi-expert setting, deriving two consistent surrogate losses that are analogous to [Mozannar and Sontag \(2020\)](#)'s and [Verma and Nalisnick \(2022\)](#)'s single-expert loss functions.

Data We first define the data for multiclass, multi-expert *learning to defer* (L2D). Following the same reasoning as in the single expert setup (4.3.1), let $\mathbf{x}_n \in \mathcal{X}$ and $y_n \in \mathcal{Y}$ be the feature and label respectively. The L2D problem assumes that we have access to (usually human) expert demonstrations. Now let there be J experts, and denote each expert's prediction space as \mathcal{M}_j (which again we will assume is equal to the label space: $\mathcal{M}_j = \mathcal{Y} \ \forall j$). The expert demonstrations are denoted $m_{n,j} \in \mathcal{M}_j$ for the associated features \mathbf{x}_n . The combined N -element training sample is then denoted $\mathcal{D} = \{\mathbf{x}_n, y_n, m_{n,1}, \dots, m_{n,J}\}_{n=1}^N$.

Models Again we use the classifier-rejector formulation ([Cortes et al., 2016a,b](#)). Remember that the goal is to learn two functions: the *classifier*, $h : \mathcal{X} \rightarrow \mathcal{Y}$, and the *rejector*. The classifier (h) is unchanged from the single-expert setting. The rejector, on the other hand, must be modified. In L2D with one expert, the rejector makes a binary decision—to defer or not. In multi-expert L2D, the rejector also must choose *to which* expert to assign the instance. Hence let the rejector be denoted $r : \mathcal{X} \rightarrow \{0, 1, \dots, J\}$. When $r(\mathbf{x}) = 0$, the classifier makes the decision. When $r(\mathbf{x}) = j$, the classifier abstains, hence resulting in the system deferring the decision to the j th expert.

Learning Again the learning objective is the 0 – 1 loss. We can re-write Equation 4.22 for the multi-expert setting as:

$$L_{0-1}(h, r) = \mathbb{E}_{\mathbf{x}, y, \{m_j\}_{j=1}^J} \left[\mathbb{I}[r(\mathbf{x}) = 0] \mathbb{I}[h(\mathbf{x}) \neq y] + \sum_{j=1}^J \mathbb{I}[r(\mathbf{x}) = j] \mathbb{I}[m_j \neq y] \right]. \quad (5.1)$$

The corresponding Bayes optimal classifier and rejector are:

$$h^*(\mathbf{x}) = \arg \max_{y \in \mathcal{Y}} \mathbb{P}(y = y | \mathbf{x}), \quad (5.2)$$

$$r^*(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbb{P}(y = h^*(\mathbf{x}) | \mathbf{x}) > \mathbb{P}(m_{j'} = y | \mathbf{x}) \ \forall j' \\ \arg \max_{j \in [1, J]} \mathbb{P}(m_j = y | \mathbf{x}) & \text{otherwise,} \end{cases} \quad (5.3)$$

where $\mathbb{P}(y | \mathbf{x})$ is again the probability of the label under the data generating process and $\mathbb{P}(m_j = y | \mathbf{x})$ is the true probability that the j th expert is correct. We provide the derivation of this rule in Section B.1. Recall that, by assumption, the expert likely will have additional knowledge not available to the classifier. This assumption is what allows the expert to possibly outperform the Bayes optimal classifier.

5.1.1 Softmax Surrogate Loss: Multiple Experts

Given the preceding definitions, we can now define the multi-expert analog of the softmax-based surrogate loss (Mozannar and Sontag, 2020). Define the augmented label space as $\mathcal{Y}^\perp = \mathcal{Y} \cup \{\perp_1, \dots, \perp_J\}$ where \perp_j denotes the decision to defer to the j th expert. Let the classifier be composed of K functions: $g_k : \mathcal{X} \mapsto \mathbb{R}$ for $k \in [1, K]$ where k denotes the class index. Then let the rejector be implemented with J functions: $g_{\perp,j} : \mathcal{X} \mapsto \mathbb{R}$ for $j \in [1, J]$ where j is the expert index. We propose to combine these $K+J$ functions via the following softmax-parameterized surrogate loss:

$$\begin{aligned} \phi_{\text{SM}}^J(g_1, \dots, g_K, g_{\perp,1}, \dots, g_{\perp,J}; \mathbf{x}, y, m_1, \dots, m_J) = \\ -\log \left(\frac{\exp\{g_y(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}^\perp} \exp\{g_{y'}(\mathbf{x})\}} \right) \\ - \sum_{j=1}^J \mathbb{I}[m_j = y] \log \left(\frac{\exp\{g_{\perp,j}(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}^\perp} \exp\{g_{y'}(\mathbf{x})\}} \right). \end{aligned} \quad (5.4)$$

The intuition is that the first term maximizes the function g_k associated with the true label. The second term maximizes the rejection function $g_{\perp,j}$ but only if the j th expert's prediction is correct. At test time, the classifier is obtained by taking the maximum over $k \in [1, K]$: $\hat{y} = h(\mathbf{x}) = \arg \max_{k \in [1, K]} g_k(\mathbf{x})$. The rejection function is similarly formulated as

$$r(\mathbf{x}) = \begin{cases} 0 & \text{if } g_{h(\mathbf{x})} > g_{\perp,j'} \quad \forall j' \in [1, J] \\ \arg \max_{j \in [1, J]} g_{\perp,j}(\mathbf{x}) & \text{otherwise.} \end{cases} \quad (5.5)$$

Our proof of the soundness of Equation 5.4 follows the same approach that Mozannar and Sontag (2020) employed—specifically, a reduction to cost-sensitive learning that ultimately resembles the cross-entropy loss for a softmax parameterization.

Theorem 5.1.1. ψ_{SM}^J (Equation 5.4) is a convex (in g), calibrated surrogate loss for the 0 – 1 multi-expert learning to defer loss (Equation 5.1).

The complete proof can be found in Appendix B.2. The result guarantees that the minimizers $g_1^*, \dots, g_K^*, g_{\perp,1}^*, \dots, g_{\perp,J}^*$ correspond to the Bayes optimal classifier and rejector given in Equation 5.2 and Equation 5.3 respectively.

5.1.2 One-vs-All Surrogate Loss: Multiple Expert

We next turn to the OvA surrogate loss. Let the label space \mathcal{Y}^\perp and the functions $g_1, \dots, g_K, g_{\perp,1}, \dots, g_{\perp,J}$ be defined just as above for the softmax case. The OvA-based multi-expert L2D surrogate is then:

$$\begin{aligned} \psi_{\text{OVA}}^J(g_1, \dots, g_K, g_{\perp,1}, \dots, g_{\perp,J}; \mathbf{x}, y, m_1, \dots, m_J) = \\ \phi[g_y(\mathbf{x})] + \sum_{y' \in \mathcal{Y}, y' \neq y} \phi[-g_{y'}(\mathbf{x})] + \sum_{j=1}^J \phi[-g_{\perp,j}(\mathbf{x})] \\ + \sum_{j=1}^J \mathbb{I}[m_j = y] (\phi[g_{\perp,j}(\mathbf{x})] - \phi[-g_{\perp,j}(\mathbf{x})]) \end{aligned} \quad (5.6)$$

where $\phi : \{\pm 1\} \times \mathbb{R} \mapsto \mathbb{R}_+$ is again a binary surrogate loss. For instance, when ϕ is the logistic loss, we have $\phi[f(\mathbf{x})] = \log(1 + \exp\{-f(\mathbf{x})\})$. The g -functions are entirely the same, and the classifier and rejector are computed exactly as in the softmax case.

We cannot construct our consistency proof in the same direct manner used in the softmax case. Like [Verma and Nalisnick \(2022\)](#), we proceed by the method of *error correcting output codes* ([Dietterich and Bakiri, 1995](#); [Langford et al., 2005](#); [Allwein et al., 2001](#); [Ramaswamy et al., 2014](#)), a general technique for reducing multiclass problems to multiple binary problems. We prove the consistency of ψ_{OvA}^J by way of the following two results.

Theorem 5.1.2. *For a strictly proper binary composite loss ϕ with a well-defined continuous inverse link function γ^{-1} , ψ_{OvA}^J (Equation 5.6) is a calibrated surrogate for the 0–1 multi-expert learning to defer loss (Equation 5.1).*

The complete proof is in Appendix B.3. Assuming *minimizability* ([Steinwart, 2007](#))—i.e. that our hypothesis class is sufficiently large (all measurable functions)—the calibration result from Theorem 5.1.2 implies consistency.

Corollary 5.1.3. *Assume that $g \in \mathcal{F}$, where \mathcal{F} is the hypothesis class of all measurable functions. Minimizability ([Steinwart, 2007](#)) is then satisfied for ψ_{OvA}^J , and it follows that ψ_{OvA}^J is a consistent surrogate for the 0 – 1 multi-expert learning to defer loss (Equation 5.1).*

Thus, the minimizers of ψ_{OvA}^J (over all measurable functions) agree with the Bayes optimal classifier and rejector (Equation 5.2 and Equation 5.3 respectively).

5.1.3 Toy example with Multiple Experts L2D

For the sake of completeness, and expanding the single expert example from Section 4.6 to the multiple expert setup, we showcase the two propose surrogate losses presented in [Verma, Barrejón, and Nalisnick \(2023\)](#), and also add the Realizable surrogate loss from [Mozannar et al. \(2023\)](#) in this new multiple expert setup.²

Multi-expert Realizable Surrogate We can intuitively extend the realizable surrogate loss presented in Section 4.3.4 to the multiple expert setup as

$$\phi_{\text{multi-RS}}(g_1, \dots, g_K, g_{\perp}; \mathbf{x}, y, m) = -2 \log \left(\frac{\exp(g_y(\mathbf{x})) + \sum_{j=1}^J \mathbb{I}[m_j = y] \exp(g_{\perp,j}(\mathbf{x}))}{\sum_{y' \in \mathcal{Y}^{\perp}} \exp\{g_{y'}(\mathbf{x})\}} \right), \quad (5.7)$$

where the reader may notice that we just included the the summation over the J experts on the contributions for each expert. The evaluation of the validity of this new surrogate loss under the same theoretical premises as the single expert surrogate loss derived in [Mozannar et al. \(2023\)](#) is left as an exercise for the reader.

Proof-of-concept with MoG dataset In Figure 5.2 we retrieve the same MoG dataset we had before, but now we add a new cluster D where a second expert is an oracle. We showcase the performance the multi-expert formulation of OvA (Figure 5.2a), softmax (5.2b) and realizable surrogate (5.2c). We see that the behavior for every surrogate the intuitive result one would expect. Again, we see the behavior for OvA and realizable surrogate is rather similar, while softmax still delegates more responsibility in the experts.

²The derivations for Realizable Surrogate loss from [Mozannar et al. \(2023\)](#) were not included in the original paper [Verma, Barrejón, and Nalisnick \(2023\)](#) since they were presented concurrently in the same conference.

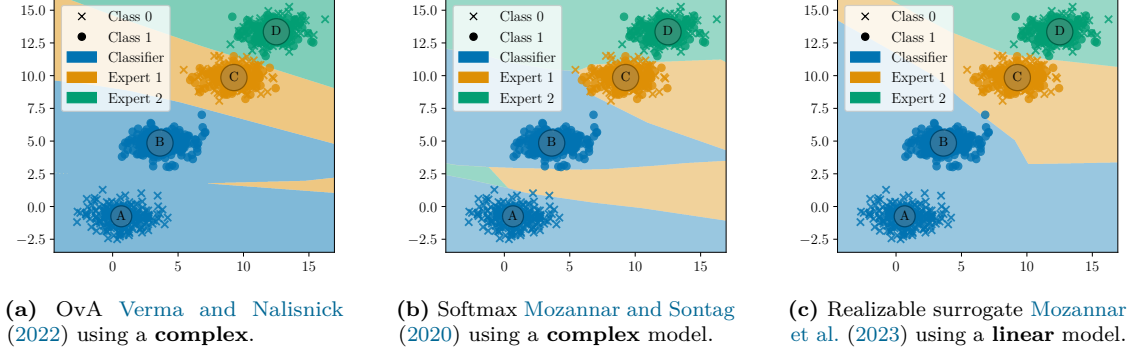


Figure 5.2: Comparison of Multiple Expert L2D surrogate losses in a MoG example: For a binary task with labels $\{\times, \circ\}$, we have four clusters: clusters A and B which are linearly separable; and clusters B and C where two experts predict correctly in one of the clusters respectively. As expected, the classifier fits those regions where the expert are not oracle (blue) and the rejector is able to discriminate the regions for expert 1 (orange) and for expert 2 (green) accordingly. Complex model differs from the linear model just by a non-linear activation at the output. Example available [here](#)³.

5.1.4 Inconsistency of Mixture of Experts

While we are the first to propose a consistent surrogate loss, previous work has proposed a *mixture of experts* (MoE) approach to multi-expert L2D. Hemmer et al. (2022) formulated the following model of the probability of the label under the whole team (of J experts and one classifier):

$$p(y|\mathbf{x}, m_1, \dots, m_J; \theta_w, \theta_h) = w_0(\mathbf{x}; \theta_w) \cdot p(y|\mathbf{x}; \theta_h) + \sum_{j=1}^J \mathbb{I}[m_j = y] \cdot w_j(\mathbf{x}; \theta_w), \quad (5.8)$$

where $p(y|\mathbf{x}; \theta_h)$ denotes the classifier’s probability. The function $\mathbf{w}(\mathbf{x}; \theta_w) \in \Delta^{J+1}$ —where Δ^{J+1} is the $(J+1)$ -dimensional simplex—defines the mixture weights. w_0 assigns weight to the classifier, and w_j for $j \in [1, J]$ denotes the weight given to the j th expert. At test time, the index of the maximum weight determines to which downstream decision maker to assign responsibility. Hemmer et al. (2022) fit this MoE model using the negative log-likelihood of $p(y|\mathbf{x}, m_1, \dots, m_J; \theta_w, \theta_h)$; denote their loss $L_{\text{MoE}}(\theta_w, \theta_h)$. In Appendix B.4.1, we show that $L_{\text{MoE}}(\theta_w, \theta_h)$ is inconsistent. We provide a full discussion of related work in Section 5.4.

5.2 Confidence Calibration of Expert Confidence

We next turn to the *calibration* (Dawid, 1982) properties of multi-expert L2D. While training with a consistent loss should produce models that are well-calibrated, previous work on the single-expert setting found that the underlying parameterizations can strongly influence calibration in practice. Specifically, Verma and Nalisnick (2022) show that the softmax formulation’s estimators can be unbounded, resulting in ‘probability’ estimates above one. As for the calibration of the classifier, Verma and Nalisnick (2022) found that there is a systemic issue and can be improved with standard post-hoc techniques like temperature scaling (Kull et al., 2019), if necessary. Their findings also apply to the multi-expert scenario, and thus we consider only the rejector going forward.

We are particularly interested in the rejector’s ability to estimate $\mathbb{P}(m_j = y|\mathbf{x})$, the conditional probability that the j th expert is correct. If the L2D system says that $\mathbb{P}(m_j = y|\mathbf{x}_0) = 0.8$, then

³<https://github.com/dbarrejon/thesis-12d>

the j th expert should be correct 80% of the time for inputs very similar to \mathbf{x}_0 . This quantity is crucial not only for the system’s ability to correctly defer but is also useful for interpretability and safety—to quantify what the model thinks that the human knows.

Therefore, we will extend the notion of calibration defined for the single expert setup in Section 4.4 to the multi-expert setup. Now, for an estimator of expert correctness $t_{\perp,j}(\mathbf{x}) : \mathcal{X} \mapsto (0, 1)$, we call $t_{\perp,j}$ *calibrated* if, for any confidence level $c \in (0, 1)$, the actual proportion of times the expert is correct is equal to c :

$$\mathbb{P}(m_j = y \mid t_{\perp,j}(\mathbf{x}) = c) = c. \quad (5.9)$$

Analogously to the formulation of the ECE for the single expert case (as represented by Equation 4.32), we can now define the expression for the multi-expert ECE as follows:

$$\text{ECE}(t_{\perp,j}) = \mathbb{E}_{\mathbf{x}} |\mathbb{P}(m_j = y \mid t_{\perp,j}(\mathbf{x}) = c) - c|. \quad (5.10)$$

where $\mathbb{E}_{\mathbf{x}}$ is usually approximated with samples.

5.2.1 Softmax Parameterization: Multi-Expert

For the softmax formulation, the estimator of the probability that the j th expert is correct can be derived as follows; see Appendix B.2 (Equation B.11). The Bayes optimal functions $g_1^*, \dots, g_{\perp,J}^*$ have the following relationship with the underlying probability of expert correctness:

$$\frac{\mathbb{P}(m_j = y | \mathbf{x})}{1 + \sum_{j'=1}^J \mathbb{P}(m_{j'} = y | \mathbf{x})} = \frac{\exp\{g_{\perp,j}^*(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}^{\perp}} \exp\{g_{y'}^*(\mathbf{x})\}} = t_{\perp,j}^*(\mathbf{x}). \quad (5.11)$$

Denote the RHS of Equation 5.11 as $t_{\perp,j}^*(\mathbf{x})$. Since we have J equations, one for each expert, we can uniquely solve for $\mathbb{P}(m_j = y | \mathbf{x})$ as:

$$\mathbb{P}(m_j = y | \mathbf{x}) = \frac{t_{\perp,j}^*(\mathbf{x})}{1 - \sum_{j'=1}^J t_{\perp,j'}^*(\mathbf{x})} = \frac{\exp\{g_{\perp,j}^*(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}} \exp\{g_{y'}^*(\mathbf{x})\}}. \quad (5.12)$$

Equation 5.12 exhibits the same pathology as the single expert setting: it is unbounded from above. For $t_{\perp,j}(\mathbf{x}) > 0$, as $\sum_{j'=1}^J t_{\perp,j'}(\mathbf{x})$ approaches one, the estimate of $\mathbb{P}(m_j = y | \mathbf{x})$ will go to infinity. Moreover, the estimator for the j th expert depends on the estimators for the other experts due to the denominator involving the quantity $t_{\perp,j}^*(\mathbf{x})$ for all experts (Equation 5.11). Thus, if one $t_{\perp,j}(\mathbf{x})$ is mis-calibrated, this error will likely propagate to the other estimators. In the experimental section we will test if this degeneracy can occur in practice.

Following the same intuition as the rejector — fully detailed in Appendix B.2 (Equation B.10) — the estimates of the classifier can be described as follows:

$$\frac{\mathbb{P}(y = y | \mathbf{x} = \mathbf{x})}{1 + \sum_{j=1}^J \mathbb{P}(m_j = y | \mathbf{x} = \mathbf{x})} = \frac{\exp\{g_y(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}^{\perp}} \exp\{g_{y'}(\mathbf{x})\}}. \quad (5.13)$$

Rearranging the equations we finally get

$$\mathbb{P}(y = y | \mathbf{x}) = \frac{1}{1 - \sum_{j'=1}^J t_{\perp,j'}^*(\mathbf{x})} \frac{\exp\{g_y^*(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}^{\perp}} \exp\{g_{y'}^*(\mathbf{x})\}} = \frac{\exp\{g_y^*(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}} \exp\{g_{y'}^*(\mathbf{x})\}}. \quad (5.14)$$

Similarly as for the single expert case for the softmax formulation, it is easy to see that $\sum_{y' \in \mathcal{Y}} \mathbb{P}(y = y' | \mathbf{x}) = 1$ and also that the classifier probability is bounded $\mathbb{P}(y = y | \mathbf{x}) \in [0, 1]$.

5.2.2 One-vs-All Parameterization: Multi-Expert

For the OvA formulation (Verma and Nalisnick, 2022), the probability that the j -th expert is correct is directly modeled by the j th deferral function. In the same spirit as for the single expert case, and assuming we are using the logistic binary loss ϕ , we have:

$$\mathbb{P}(m_j = y|\mathbf{x}) = \sigma(g_{\perp,j}^*(\mathbf{x})) = \frac{1}{1 + \exp\{-g_{\perp,j}^*(\mathbf{x})\}} \quad (5.15)$$

$$\mathbb{P}(y = y|\mathbf{x}) = \sigma(g_y^*(\mathbf{x})) = \frac{1}{1 + \exp\{-g_y^*(\mathbf{x})\}}. \quad (5.16)$$

The complete proof is described in Appendix B.3. The expert’s correctness estimator has the correct range of $[0, 1]$ for any setting of $g_{\perp,j} \in \mathbb{R}$. Moreover, there is no dependence across expert deferral functions $g_{\perp,1}, \dots, g_{\perp,J}$, unlike the softmax case. In the experimental section (5.5) we will test these properties to result in better calibration in practice.

5.3 Ensembling Expert with Conformal Inference

Multi-expert L2D, as defined above, operates by selecting just one expert upon deferral. This approach is sensible if querying each expert results in an independent expense (such as a consulting fee). However, in other settings, the cost incurred by deferring may just be that of time and efficiency (i.e. a lack of automation). In this case, the cost of querying additional experts would be negligible; for example, we could send multiple experts simultaneous messages asking for their decisions. Given the estimators of $\mathbb{P}(m_j = y|\mathbf{x})$ presented in the previous section, it is then natural to ask how we might ensemble experts according to these estimates of correctness. Below we present a methodology based on *conformal inference* for obtaining dynamic, minimal ensembles of experts.

5.3.1 Conformal Inference

Conformal Inference *Conformal inference* (CI) (Shafer and Vovk, 2008)⁴ constructs a confidence interval (or set) for predictive inference. In the traditional multiclass classification setting, given a new observation \mathbf{x}_{n+1} , we wish to determine the correct associated label $y_{n+1} = y_{n+1}^*$, where y_{n+1}^* denotes the true class label. CI allows us to construct a distribution-free confidence set $\mathcal{C}(\mathbf{x}_{n+1})$ that will cover the true label with *marginal* probability $1 - \alpha$:

$$\mathbb{P}(y_{n+1}^* \notin \mathcal{C}(\mathbf{x}_{n+1})) \leq \alpha \quad \forall \mathbb{P} \in \mathfrak{P}$$

where \mathfrak{P} represents the space of all distributions—hence the ‘distribution-free’ quality. Denote the test statistic as $S(\mathbf{x}, y; \mathcal{D})$. It is known as a *non-conformity* function: a higher value of S means that (\mathbf{x}, y) is less conforming to the distribution represented (empirically) by \mathcal{D} . Despite this guarantee, CI is only as good as its test statistic in practice. For instance, the marginal coverage is naively satisfied if we construct the set randomly by setting $\mathcal{C}(\mathbf{x}) = \mathcal{Y}$ with probability $1 - \alpha$ and returning the empty set otherwise. CI is implemented by calculating the non-conformity function on a validation set and computing the empirical $1 - \alpha$ quantile \hat{q}_α (with a finite sample correction). At test time, elements are added to the set until the non-conformity function passes the previously-computed quantile. Conformal inference has gain much attention in the recent years, both in regression (Romano et al., 2019) and classification tasks (Romano et al., 2020; Angelopoulos et al., 2020; Sadinle et al., 2019).

⁴We also suggest to newcomers in conformal inference the introduction to conformal inference by Angelopoulos and Bates (2021).

5.3.2 Conformal Inference on Sets of Experts

Conformal Sets of Experts We propose applying CI to perform uncertainty quantification for the experts. Thus, here, $\mathbf{C}(\mathbf{x})$ represents a set of experts. Firstly, we assume there is a best expert: for a new observation \mathbf{x}_{n+1} , let j_{n+1}^* denote the best expert such that

$$\mathbb{P}(m_{j_{n+1}^*} = y | \mathbf{x}_{n+1}) > \mathbb{P}(m_e = y | \mathbf{x}_{n+1}) \quad \forall e \neq j_{n+1}^*. \quad (5.17)$$

We would then like to construct a set such that j_{n+1}^* is covered with marginal probability $1 - \alpha$:

$$\mathbb{P}(j_{n+1}^* \notin \mathbf{C}(\mathbf{x}_{n+1})) \leq \alpha \quad \forall \mathbb{P} \in \mathfrak{P} \quad (5.18)$$

where $\mathbf{C}(\mathbf{x}_{n+1})$ again denotes the conformal set and \mathfrak{P} is the same as above. The set will have a dynamic size that changes with \mathbf{x} , ensuring our ensemble makes efficient use of expert queries. Unlike in most applications of CI, we can use the procedure to form an ensemble by aggregating the predictions of all experts in the set.

Naive Statistic We start by adapting a score function from multiclass classification. Let $s_j(\mathbf{x})$ denote the estimator that the j th expert is correct. For the softmax case, $s_j(\mathbf{x}) = t_{\perp,j}(\mathbf{x}) / (1 - \sum_{j'} t_{\perp,j'}(\mathbf{x}))$ (Equation 5.12), and for OvA, $s_j(\mathbf{x}) = \phi(g_{\perp,j}(\mathbf{x}))$ (Equation 5.15). Let π_1, \dots, π_J denote the indices for a descending ordering of the estimators $s_j(\mathbf{x})$, i.e. s_{π_1} is the expert who has the best chance of being correct (according to the rejector). The resulting non-conformity function and test statistic are:

$$S(\mathbf{x}, y, m_1, \dots, m_J; \mathcal{D}) = \sum_{e=1}^E s_{\pi_e}(\mathbf{x}) \quad (5.19)$$

where π_E is the index of the expert who has the lowest score s_{π_E} of all *correct experts* ($m = y$). This expression means that we will keep adding the correctness scores s in descending order until we include all experts who correctly predict the given instance. Hence $E = J$ only when all experts are correct and $E < J$ otherwise.

Regularized Statistic A problem with the statistic above is that multiple experts can be correct, resulting in noise that obscures the identity of the best expert. In this case, since we are using the estimates for the experts' correctness probability as our *pseudo* soft labels for the conformity scores calculations, one might view this similar to a multilabel classification problem, where more than one expert can be correct. As shown by [Cauchois et al. \(2021\)](#), these scenarios result in bigger predicted sets. Also, since the experts predictions are not fixed, but the experts have a certain probability of being correct, we can also see that these predictions will have *noise*, and this also results in bigger sets ([Einbinder et al., 2022](#)). In the experiments, we effectively show that this 'naive' statistic is not robust to noise, resulting in inflated set sizes (which are sometimes vacuous). Similar problems are discussed by [Angelopoulos et al. \(2020\)](#). To address this issue, we employ *conformal risk control* ([Angelopoulos et al., 2022](#)) to directly control the false negative rate. We create regularized prediction sets as follows:

$$C_{\lambda_\alpha}(\mathbf{x}) = \{j : s_j(\mathbf{x}) + \beta(s_j(\mathbf{x}) - \kappa) > 1 - \lambda_\alpha\} \quad (5.20)$$

where β and κ are the parameters of the regularization and λ_α is chosen to have $1 - \alpha$ coverage guarantees. In the following subsection, we describe λ_α and how we choose the regularization parameters. The general idea is to choose κ so that confidences lower than this threshold can happen with probability at most α . We choose β to optimize the size of the sets.

5.3.3 Choice of Hyperparameters for Regularized Conformal Ensembles

We begin by giving a brief introduction to the procedure of conformal risk control. For detailed exposition, we refer the reader to the original paper (Angelopoulos et al., 2022).

Conformal risk control (Angelopoulos et al., 2022) is a generalized form of conformal prediction which aims to control any bounded monotone loss function $\ell(\cdot)$ in expectation. In our work, we are interested in False Negative Rate (FNR) as a specific loss function which satisfies the monotonicity property as a function of λ (Equation 5.20). Given access to the calibration data $\{\mathbf{x}_n, y_n\}_{n=1}^N$, the goal in conformal risk control is to find $\hat{\lambda}$ so that the following coverage guarantee holds:

$$\mathbb{E} [\ell(C_{\hat{\lambda}}(\mathbf{x}_{n+1}))] \leq \alpha.$$

Without loss of generality, we consider ℓ to be a non-increasing function of λ and bounded by a constant B . Procedurally, it works by defining $S(\mathbf{x}_{1:N}; \lambda) = \frac{1}{N} \sum_{i=1}^N \ell(C_{\lambda}(\mathbf{x}_i))$. For $\alpha \in (-\infty, B]$, $\hat{\lambda}$ is then defined as

$$\hat{\lambda} = \inf \left\{ \lambda : \frac{n}{n+1} S(\mathbf{x}_{1:N}; \lambda) + \frac{B}{n+1} \leq \alpha \right\}.$$

Assuming exchangeability on $\ell(C_{\lambda}(\mathbf{x}_i))$, this results in the desired coverage guarantees for $C_{\hat{\lambda}}(\mathbf{x}_{n+1})$. In our work, we use a grid of equally spaced 1500 values in $[0, 1]$ to pick $\hat{\lambda}$. We have two hyperparameters κ and β in the regularized conformal ensemble procedure discussed in Section 5. In Algorithm 1 we detail the procedure to choose κ .

Algorithm 1 Choice of κ

Data: Error rate: α , # Experts: J , Data: $\{\mathbf{s}^i(\mathbf{x}), \mathbf{e}^i(\mathbf{x})\}_{i=1}^N$ with expert's confidence $\mathbf{s}^i(\mathbf{x}) \in [0, 1]^J$ and true expert predictions $\mathbf{e}^i(\mathbf{x}) \in \{0, 1\}^J$

$B \leftarrow \{\cdot\}$

for $i \in [1, N]$ **do**

for $j \in [1, J]$ **do**

if $\mathbb{I}\{e_j^i(\mathbf{x}) == 1\}$ **then**

$B \leftarrow B \cup \{s_j^i(\mathbf{x})\}$

end

end

$S \leftarrow \text{sort}(B)$ s.t. $u_i, u_j \in S, i \leq j$, then $u_i \geq u_j$

$\kappa^* \leftarrow 1 - \alpha$ quantile of S

end

We can employ corrections to account for finite sample size N on line 9. Given this procedure to choose κ^* , one may argue that our choice of κ^* can give us meaningful prediction sets by designing a prediction set as:

$$C_2(\mathbf{x}) = \{j : s_j(\mathbf{x}) \geq \kappa^*\}.$$

However, our next proposition establishes that $C_{\lambda}(\mathbf{x})$ results in prediction sets at most as large as $C_2(\mathbf{x})$.

Proposition 5.3.1. *Define the prediction sets $C_{\lambda}(\mathbf{x}) = \{j : s_j(\mathbf{x}) + \beta(s_j(\mathbf{x}) - \kappa^*) > 1 - \lambda\}$ and $C_2(\mathbf{x}) = \{j : s_j(\mathbf{x}) \geq \kappa^*\}$, where κ^* is defined as above, $\beta \geq 0$, $0 \leq \lambda \leq 1$, then it trivially holds that*

$$C_{\lambda}(\mathbf{x}) \subseteq C_2(\mathbf{x}).$$

We tune β in a grid-search manner. The grid size for β is $[3.5, 1e^{-3}]$ with steps of 50 samples. We split the total number of deferred samples into two portions: one for tuning hyperparameters β and κ , another for the regular conformal procedure. 30% of the deferred samples are used to tune the ensembling hyperparameters.

5.4 Related Work

Multi-Expert Models There have been several works that use models to improve the decision making of multiple experts (Benz and Rodriguez, 2022; Straitouri et al., 2023) and to fuse decisions from models and humans (Keswani et al., 2021; Kerrigan et al., 2021). In the context of L2D, we already presented in Section 5.1.4 the work by Hemmer et al. (2022), who proposed the only existing model for multi-expert L2D. Yet their approach does not have any supporting theoretical guarantees, such as consistency (like ours). Another MoE model related to L2D was proposed by Pradier et al. (2021). They propose a novel MoE model called Preferential MoE which augments human expertise in decision making only when necessary — hence similar to the learning to defer motivation. Again, this model does not account for consistency. Keswani et al. (2021) also proposed an MoE-based model for the multi-expert scenario but not for the standard L2D setting that we consider. Rather they allow for responsibility to be passed to multiple downstream sources—specifically, to any of the 2^{J+1} possible sets involving the experts and/or model. An important difference with our work is that here each expert has its own domain of expertise. In a similar flavor to our work, Benz and Rodriguez (2022) present a counterfactual inference mechanism to infer *when* and *to whom* we should ask for second opinions to improve allocation of resources in automated decision support systems.

Learning to Defer Previous L2D extensions did not come with consistency guarantees (Raghu et al., 2019; Wilder et al., 2020; Pradier et al., 2021; Okati et al., 2021; Liu et al., 2022). Verma and Nalisnick (2022) proposed the second provably consistent surrogate for multiclass L2D based on a one-vs-all formulation. Charusaie et al. (2022) further studied the L2D optimization problem, proving results for complementarity and active learning. Our work extends Mozannar and Sontag (2020)’s and Verma and Nalisnick (2022)’s results to the multi-expert setting—for which no one has yet to propose a consistent surrogate loss.

Calibration in L2D Verma and Nalisnick (2022) motivate their OvA surrogate from the standpoint of calibration and thus is the only other work that studies the confidence calibration of L2D systems. We extend their work to the multi-expert setting. Calibration has received much attention of late in the wider machine learning literature (Guo et al., 2017; Kull et al., 2019; Vaicenavicius et al., 2019; Gupta and Ramdas, 2022). The dominant methodology is to apply post-hoc calibration: fitting additional parameters on validation data to re-calibrate the formerly mis-calibrated model. These methods could potentially be applied here—such as, by adding a temperature parameter to the per-expert terms in the OvA loss—but we are primarily interested in the native, ‘out-of-the-box’ calibration properties of the losses.

Conformal Inference for Human-AI Collaboration In the last years, more interest is appearing on the adoption of conformal inference as an uncertainty quantification tool in the context of human-AI collaboration. As theoretically and empirically proved by Babbar et al. (2022), it may be more beneficial to recommend a subset of options rather than a single option. Babbar et al. (2022) study a similar work flow (apply CI then pass to a human) and also propose applying CI only to non-deferred samples, which results in smaller set sizes. Straitouri et al. (2023) apply CI to a classifier and then pass the prediction set to a human to make the final decision. To the best of our knowledge, no previous work has applied CI to obtain sets of experts.

5.5 Experiments

Our experimental setup closely follows that of Verma and Nalisnick (2022)—but extended to multiple experts. For all runs, we report the mean and standard error across 3 random seeds. We perform three types of experiments. In the first, we check the system accuracy of the derived consistent surrogate losses in three consequential tasks (Subsection 5.5.1): galaxy classification, skin lesion diagnosis, and hate speech detection. We find that the OvA-trained model often outperforms both the softmax variant and MoE baseline. Secondly, we investigate the confidence calibration properties of the surrogates (Subsection 5.5.2). As hypothesized, the OvA loss results in less calibration error on both simulated and real data (possibly explaining its superior accuracy). Lastly, we investigate the efficacy of our conformal ensembling procedure (Subsection 5.5.3). For the naive statistic, the OvA loss’ superior calibration results in appropriately smaller sets. For the regularized statistic, both losses perform equally well. Our implementations are publicly available at https://github.com/rajev/Multi_L2D.

5.5.1 Overall System Accuracy

Data Sets We report the overall system accuracy for three real-world data sets: HAM10000 (Tschandl et al., 2018) for skin lesions diagnosis, Galaxy-Zoo (Bamford et al., 2009) for galaxy classification, and HateSpeech (Davidson et al., 2017) for hate speech detection. The train-validation-test split is 60% – 20% – 20%. Following Verma and Nalisnick (2022), we down-sample Galaxy-Zoo to 10,000 instances.

Models We use a 34-layer residual network (ResNet34) and a 50-layer residual network (ResNet50) as base models for HAM10000 and Galaxy-Zoo respectively. For HateSpeech, we use a 100-dimensional *fasttext* (Joulin et al., 2016) representation of each input and a ConvNet (Kim, 2014) as the base model. We refer the reader to Verma and Nalisnick (2022) for more details on the training and hyperparameter selection, as we follow their setup.

Experimental Setup We train the systems with an increasing number of experts, ranging from 2 to 10. We comment the details of how we simulate the experts below. For each run, we enlarge the pool by adding increasingly accurate experts, and this process is repeated 3 times with different random seeds. We keep the base model fixed across these runs except for the additional output dimensions required by the expanded expert pool. Below you can find the description of the experts’ configurations for the studied datasets.

Galaxy-Zoo consists of a large number of galaxy images that were classified into three main morphological classes: spiral, elliptical, and irregular. The HateSpeech dataset is a benchmark

dataset for hate speech classification, consisting of tweets labeled as hate speech, offensive language, or neither, with hate speech involving offensive language and discriminatory expressions, offensive language including profanity, and neither containing non-offensive content. For **Galaxy-Zoo** and **HateSpeech**, we define the following experts using the human annotations available in the datasets and using various perturbations of these predictions:

1. **Human expert**: we sample predictions from the provided human annotations.
2. **Flipping human expert**: Expert who flips the given prediction with some probability p_{flip} .
3. **Probabilistic expert**: Expert who makes use of the annotations with some probability $p_{\text{annotator}}$, or predicts randomly otherwise.

The whole expert configuration is described in Table 5.1.

Table 5.1: Hate Speech and Galaxy-Zoo experts configuration.

	Expert configuration	$p_{\text{flip}}[\%]$	$p_{\text{annotator}}[\%]$
1	Random Expert	-	-
2	Probabilistic Expert	-	10
3	Flipping Human Expert	50	-
4	Probabilistic Expert	-	75
5	Flipping Human Expert	30	-
6	Flipping Human Expert	20	-
7	Probabilistic Expert	-	85
8	Human Expert	-	-
9	Probabilistic Expert	-	50
10	Human Expert	-	-

The HAM10000 dataset (Tschandl et al., 2018) is composed of dermatoscopic images corresponding to 7 diagnostic categories in the realm of pigmented lesions. These 7 categories can be further decomposed into **benign**: melanocytic nevi (**nv**), benign keratinocytic lesions (**bk1**), dermatofibromas (**df**) and vascular lesions (**vasc**); and **malign**: melanomas (**mel**), basal cell carcinomas (**bcc**) and actinic keratoses and intraepithelial carcinomas (**akiec**).

Table 5.2: HAM10000 experts configuration.

	Expert configuration	$p_{\text{in}}[\%]$	$p_{\text{out}}[\%]$	Diagnostic Category [in]
1	Random Expert	-	-	[nv , bk1 , df , vasc , mel , bcc , akiec]
2	Dermatologist for malign	25	15	[mel , bcc , akiec]
3	Dermatologist for benign	25	15	[nv , bk1 , df , vasc]
4	Specialized dermatologist in nv	50	15	[nv]
5	Specialized dermatologist in vasc	70	15	[vasc]
6	Specialized dermatologist in mel	75	15	[mel]
7	Dermatologist for benign	75	25	[nv , bk1 , df , vasc]
8	MLP Mixer	-	-	[nv , bk1 , df , vasc , mel , bcc , akiec]
9	Experienced dermatologist	80	50	[nv , bk1 , df , vasc , mel , bcc , akiec]
10	Experienced dermatologist	80	60	[nv , bk1 , df , vasc , mel , bcc , akiec]

In contrast to the Galaxy-Zoo and Hatespeech dataset, for HAM10000 we do not have individual annotators predictions, but just the ground truth label. Further information can be found in the original dataset description (Tschandl et al., 2018). In order to recreate a setup comparable to a real-world scenario, we create different experts configurations:

1. **Random expert:** This expert predicts randomly among all classes.
2. **Dermatologist expert:** These experts will be specialized in a set of categories, and will predict with probability p_{in} . Out of that set, they will predict with probability p_{out} .
3. **MLPMixer:** We derive HAM10000’s expert predictions from the predictions of an 8-layer MLP Mixer (Tolstikhin et al., 2021), which has access to additional metadata such as age, gender, and diagnosis type.

As it can be seen in Table 5.2, we gradually add experts from a random expert to a final expert which simulates an experienced dermatologist. From Kittler et al. (2002) we know that clinical diagnosis of cutaneous melanoma with the unaided eye is only about 60% accuracy, and that dermatologists equipped with dermatoscope can achieve accuracies of 75%–84%. That is the reason why we chose for the simulated dermatologist experts to have those probabilities p_{in} and p_{out} .

Models We use a 34-layer residual network (ResNet34) and a 50-layer residual network (ResNet50) as base models for HAM10000 and Galaxy-Zoo respectively. For HateSpeech, we use a 100-dimensional *fasttext* (Joulin et al., 2016) representation of each input and a ConvNet (Kim, 2014) as the base model. We refer the reader to Verma and Nalisnick (2022) for more details on the training and hyperparameter selection, as we follow their setup. Ideally, the L2D systems should exhibit strictly increasing accuracy due to adding experts of increasing quality. We compare our models against three baselines: one classifier consisting on the same base model used by the other methods; the best expert from the pool of experts and Hemmer et al. (2022)’s MoE described before.

Results The top row of Figure 5.3 reports the mean and standard error of the system accuracy as the number of experts increases. While the OvA, softmax, and MoE models perform comparably on HateSpeech (left), OvA’s performance (blue) is notably better on HAM10000 (center) and Galaxy-Zoo (right) as its accuracy never falls below the one classifier baseline (red), while the others’ accuracies do. From this results we validate one hypothesis usually claimed during this thesis: human-AI collaborating systems outperform only humans or only AI models, as it can be seen on the average behavior of the all human-AI models compared to the best expert (purple) or only classifier (red) baselines.

5.5.2 Confidence Calibration

In Section 5.2, we found that the two surrogates have very different estimators of $\mathbb{P}(m_j = y_i | \mathbf{x}_i)$, the probability that the j th expert is correct. We now test if these theoretical differences have consequences for practice. To ensure ECE is well-defined for the softmax loss, we cap any confidences greater than 1 at 1. In addition to reporting calibration for the preceding experiment (system accuracy), we also perform simulations using the standard splits of CIFAR-10 (Krizhevsky, 2009). We use a 28-layer wide residual network (Zagoruyko and Komodakis, 2016), following Verma and Nalisnick (2022). Our results suggest that systems trained with the softmax surrogate exhibit degradation in calibration as the number of experts increases. Furthermore, other experts in the committee significantly affect the calibration of other experts.

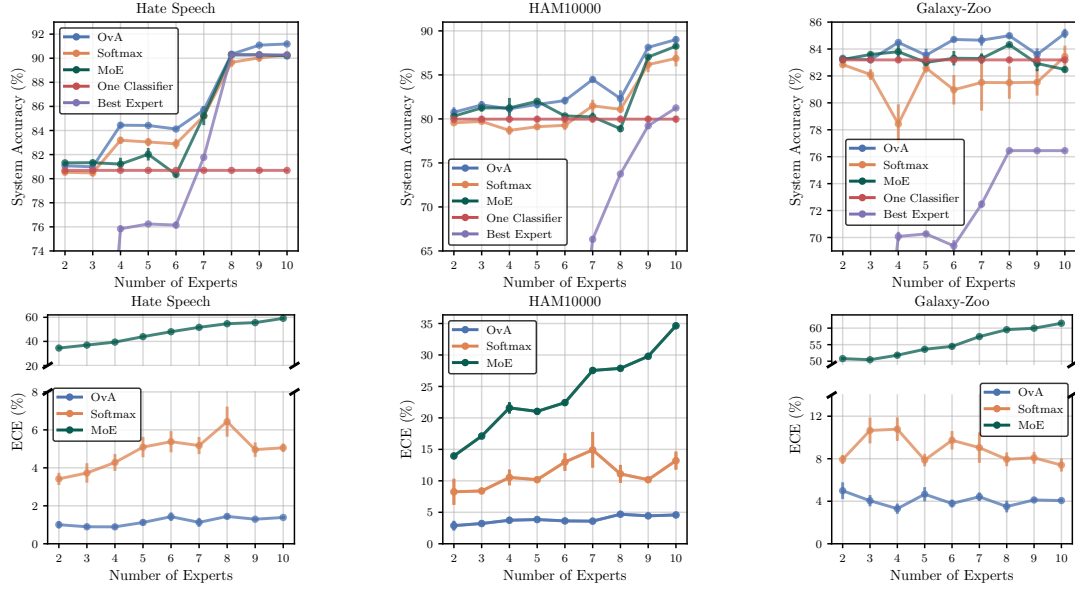


Figure 5.3: System Accuracy and Calibration. The figures above report the system accuracy (top row) and calibration error (bottom row) as experts of increasing ability are added (from 2 to 10). One classifier (red), best expert (purple), and a mixture of experts (green) (Hemmer et al., 2022) serve as baselines. We see that the OvA-trained model (blue) performs well in every case. On the other hand, the softmax-trained model (orange) falls below the one classifier baseline for both HAM10000 and GalaxyZoo.

Simulation #1: Increasing Experts We perform a simulation to see how the methods perform under an increasing number of experts. We generate a synthetic expert with a correctness probability of 70% over the first five classes and random across all other classes. We then replicate that expert and add it to the expert pool, ranging from 4 to 20 total experts. Figure 5.4a reports the average ECE across experts as the pool increases. The OvA method (blue) is roughly stable at about 2% ECE as experts are added. The softmax method (orange) has roughly double the ECE ($\sim 4.5\%$). In Figure 5.4b, we report the overall system accuracy to see if these calibration differences have an effect. We see some positive effects, with OvA (blue) having a better accuracy for 12 and fewer experts. However, the softmax (orange) has the best accuracy at 20 experts, despite its calibration still being worse.

Simulation #2: Expert Dependence We next perform a simulation to see how calibration error can propagate across the estimators. We simulate four experts with one always being random and the other three having a probability of correctness that increases from 20% to 95% on the first five classes (random for others). We hypothesize that the softmax’s ECE for the random expert will *increase* when the probability of correctness for the other three experts increases due to the tied parameterization. Figures 5.4c and 5.4d report the results, with the former reporting average ECE and the latter the ECE of just the random expert. Firstly, from Figure 5.4c, we see that again the OvA method is better calibrated across all experimental settings. Then from Figure 5.4d, we see that our hypothesis is confirmed: OvA (blue) is able to model the random expert well no matter the other experts’ abilities, but the softmax (orange) is not. The softmax’s ECE increases almost in-step with the expert correctness, except for some cancellation effect happening at 80%. This is clearly an undesirable behavior from the standpoint of safety since any ECE above zero means that the system is reporting that the expert is better than random and thus misleading the user.

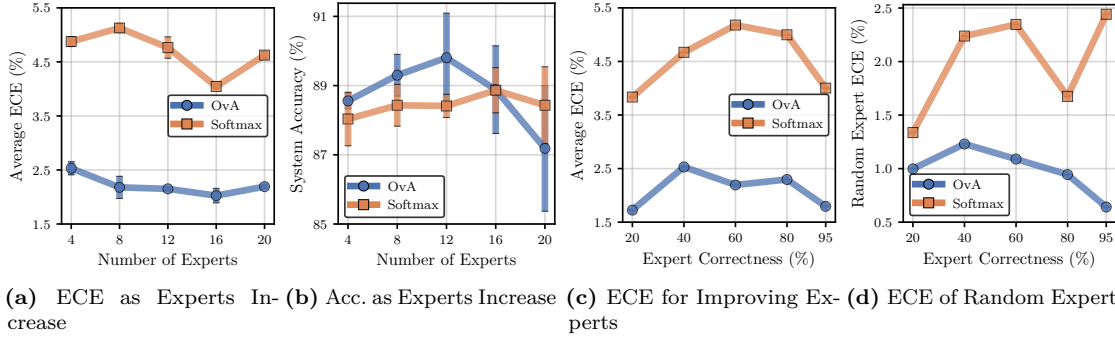


Figure 5.4: Confidence Calibration Simulations. The figures above report the results for confidence calibration simulations performed on CIFAR-10. The first and second subfigures show calibration error (a) and system accuracy (b) as the number of experts increases from 4 to 20. We see that the OvA formulation (blue) has better calibration across all runs, but this translate to better accuracy only for 16 or fewer experts. The third and forth subfigures show calibration error as experts’ abilities increase (a) and when one expert is kept at random chance (b). OvA (blue) shows better calibration in both metrics.

Hate Speech, HAM10000, and Galaxy-Zoo Lastly we report the calibration error of the models trained for the experiments reported in Section 5.5.1. The results are in the bottom row of Figure 5.3. The trend we observe in the CIFAR-10 simulations is also observed here, with OvA (blue) having the best calibration. This may explain why OvA has the best system accuracy. Unsurprisingly, the MoE has extremely poor calibration, which is likely due to its inconsistent optimization objective which allows for sub-optimal models (as we prove in Proposition B.4.1).

5.5.3 Conformal Ensembles

Lastly, we study our proposal of using CI to ensemble multiple experts. We first analyze the two proposed statistics, demonstrating the regularized version’s superior ability to recover the experts who are oracles. We then report the downstream effect on the overall system accuracy, comparing performance to that of a fixed-size ensemble of experts. We finish with an additional experiment where we test each CI formulation on on a pool of experts with increasing overlapping expertise.

Experts and Setup We experiment with two settings on CIFAR-10, each with 10 total experts. In the first (*no noise*), we synthesize experts such that they are an oracle on an increasing subset of the classes and guaranteed to be wrong on the classes not in that set. In the second (*with noise*), the experts are oracles in the same way but now have a (uniformly) random chance of being correct for the non-oracle classes. The theory of CI guarantees that the sets *marginally* cover all oracle experts. Yet, ideally, we wish the sets to contain *only* the experts who are oracles. We use $\alpha = 0.1$ for Equation 5.18 in all experiments.

Expert Identification The CI results are reported in Figures 5.5a and 5.5b. The number of oracles is on the y-axis, and the average set size is on the x-axis. Optimal performance would be the $y = x$ line. The results for the no-noise setting are reported in Figure 5.5a. The naive statistic (solid lines) considerably inflates the set size for both softmax and OvA. Yet OvA is much closer to $y = x$, which suggests superior calibration leads to better CI. The performance of the regularized statistic is shown by the dashed lines. Both softmax and OvA perform nearly perfectly. Figure 5.5b reports the with-noise setting, and we find that the naive statistic performs terribly for both losses. The regularized statistic, on the other hand, performs well for both softmax and OvA. Softmax demonstrates slight superiority for 2 – 5 oracles.

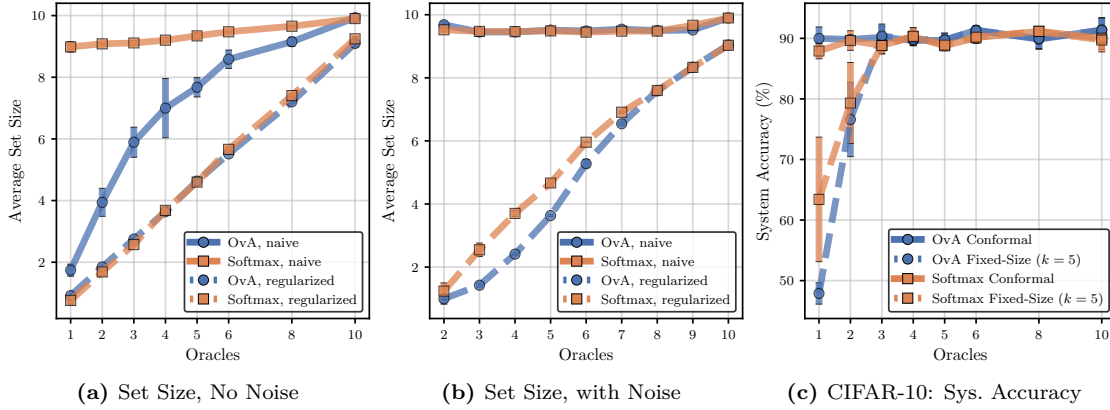


Figure 5.5: *Conformal Sets of Experts.* The figures above report our analysis of the two statistics proposed in Section 5. Subfigures (a) and (b) show the ability of the two statistics to select the correct number of experts as the number of oracle experts increases—so optimal performance is the $y = x$ line. Subfigure (a) reports the no-noise setting (so experts are either perfectly incorrect or correct), and we see that the naive statistic (solid lines) overestimates the set size. The problem is even worse in Subfigure (b). However, the regularized statistic (dashed lines) is able to do well in both cases. Subfigure (c) shows how ensembling the set affects system accuracy. The conformal approach is able to out-perform a fixed size of 5 experts for a small number of oracles and is equivalent at higher numbers.

Overall System Accuracy We next investigate using the conformal set as an ensembling strategy. Upon deferral, we use majority voting across the set to generate the final prediction. We compare this with the baseline of using a fixed ensemble size—specifically, the top five ranked experts. We show the results for CIFAR-10 in Figure 5.5c. The crucial settings are for one and two oracles since using a fixed ensemble size is guaranteed to fail—as is confirmed by the plot ($< 80\%$ accuracy). We see that the conformal ensembles are clearly superior here, achieving around 90% accuracy. For three or more oracles, both methods have equal performance. This is expected since only three oracles are needed to form a correct majority. We emphasize that CI’s *adaptivity* is highly desirable so that the best experts can be identified with transparency and queried efficiently.

Overlapping expertise among experts For this experiment we have 10 experts for the CIFAR-10 dataset, each of them being an oracle on a specific class out of the 10 classes from the dataset, and we will increase the overlapping probability of these experts being correct on the other classes where the experts are not oracle from 10% to 95% overlap. That is, we will vary from specialized experts to fully overlapped experts. We hope to see that the average set size for specialized experts is close to 1 and for fully overlapped experts close to the total number of experts. We report the results in Figure 5.6.

First of all, we look at the **average set size** provided by each CI statistic. It is worth noticing from Figure 5.6a that the average set size for the naive conformal method, both for softmax and OvA, is always close to the total number of experts, for specialized and overlapped experts. If we remember, the naive test statistic is calculated among *all correct experts*. This is a very important point, because if an expert happens to be correct outside of their expertise domain, this results in a very big non-conformity score because of the low confidence of such expert. That is, imagine for certain sample \mathbf{x} and class $y = 3$, for low overlapping probabilities, we might have $E = 3$, where $e = 1$ could be the oracle for class $y = 3$ and $e = 2, e = 3$ two experts that were correct by chance. From Equation 14 in the manuscript, we can expect that, best-case

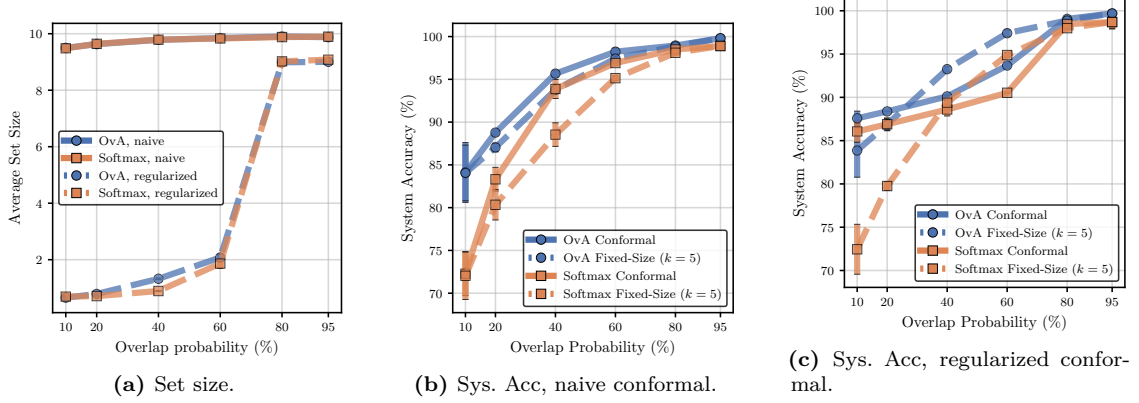


Figure 5.6: *Gradual overlapping expertise results.* The figures above report the average set size (a) and system accuracies under the naive conformal method (b) and regularized conformal method (c) for increasing expertise overlap for CIFAR-10. From Figure (a) we see that the regularized conformal method is more dynamic than the naive method for both OvA and softmax. System accuracies for the naive conformal method (b) are slightly better than fixed-size ensemble because we ensemble all experts, whereas for the regularized conformal method (c) we see a slightly drop due to the adaptivity of the conformal sets

scenario $s_{\pi_1} > s_{\pi_2} > s_{\pi_3}$, or even worse $s_{\pi_1} > \dots > s_{\pi_8} > s_{\pi_2} > s_{\pi_3}$, because other experts' confidences could be also greater than confidences from correct experts by chance. Therefore, we will obtain bigger test-statistics that result in very larger conformal sets. This problem has already been addressed in Angelopoulos et al. (2022). However, notice how the regularized test statistic is capable of producing smaller sets. The idea is that now we optimize additional parameters (described in Section 5 in the manuscript) to ensure that confidences lower than a certain threshold are filtered out for the calculation of the test statistic.

Finally, we report in Figures 5.6b and 5.6c the **system accuracies** for the naive conformal method and the regularized conformal method respectively. For the naive conformal method we obtain better results than using a fixed-size ensemble of experts of size 5. Because set sizes are almost always close to the total number of experts, and we do majority voting, then as long as there is a correct expert plus an expert correct by chance, we will predict correctly. However, for the regularized conformal method we notice a drop in the system accuracy for lower overlapping probabilities. Since for such cases now the set sizes are smaller, we have smaller number of experts in the set and therefore less chance of having correct experts by chance in the ensemble. Despite this drop in the accuracy, we clearly have a more dynamic and less conservative ensembling method.

5.6 Conclusions

We have extended the L2D framework to support multiple experts. We proposed two optimization objectives and proved that they are both consistent. Our proposed optimization objectives are simple to use in practice and could be embedded into any empirical risk minimization framework. Additionally, we also studied their potential to be confidence calibrated, showing that the softmax-based objective can result in mis-calibrated models in practice. Lastly, we considered a principled procedure for selecting *minimal* sets of experts to ensemble. For future work, we aim to improve the data efficiency of our method by extending the active learning results of Charusaie et al. (2022) to the multi-expert setting.

*AI [alone] would have the capacity to write a good song.
But not a great one.
It lacks the nerve.*

— Nick Cave 

6

Conclusions and Future Lines of Work

Contents

6.1 Summary of Methods and Contributions	95
6.1.1 Part I: Medical Data Wrangling using VAEs	95
6.1.2 Part II: Learning to Defer to Multiple Experts	96
6.2 Suggestions for Future Research	97
6.2.1 Next Steps in VAEs for Irregular Data	97
6.2.2 Next Steps in Learning to Defer	99

THE doctoral manuscript is structured into two primary sections, focusing on medical data wrangling using variational autoencoders and learning to defer to multiple experts. Chapter 2 and Chapter 4 provide an overview of the problem, motivation, background information, and relevant references concerning variational autoencoders and learning to defer, respectively. In the current chapter, we present a summary of our key contributions discussed in Chapter 3 and Chapter 5, and propose potential directions for future research.

6.1 Summary of Methods and Contributions

6.1.1 Part I: Medical Data Wrangling using VAEs

This first part opens with an introductory chapter, Chapter 2, which does not include any original contribution. We begin by explaining the VAE learning framework and then proceed to discuss three significant challenges associated with handling irregular observations: missing data, diverse data types, and temporal data. For each challenge, we provide the rationale behind it and refer to relevant reference works that influenced our contribution outlined in Chapter 3.

Firstly, we present our proposed model, the Sequential Heterogeneous Incomplete Variational Autoencoder (Shi-VAE), which extends the capabilities of Variational Autoencoders (VAEs) to address the aforementioned challenges. Our evaluation focuses on missing data in sequences, as real-world scenarios often exhibit bursts of missing data, such as sensor failures in a hospital setting. Additionally, we identify that standard error metrics like Root Mean Square Error (RMSE)

are insufficient for assessing temporal models. Therefore, we incorporate the cross-correlation between the ground truth and the imputed signal as part of our analysis.

We conducted a comparative analysis of our method against several common baselines for missing data imputation, as well as the GP-VAE model proposed by (Fortuin et al., 2020) which is regarded as a state-of-the-art approach for handling missing data in temporal series. The GP-VAE model utilizes a Gaussian Process (GP) in the latent space to capture the temporal dynamics of the sequences. To evaluate the performance of our model, we utilized two distinct databases. The first is a human monitoring database comprising data collected from a mobile app used by psychiatric patients from two hospitals: Hospital Universitario Fundación Jiménez Díaz and Hospital Universitario Rey Juan Carlos. This database encompasses heterogeneous data sources, including variables such as steps, app usage, distance, and more. The data exhibits temporal characteristics, and the rate of missing values is significant. Additionally, we evaluated our method on a common intensive care unit (ICU) database, specifically the Physionet database (Silva et al., 2012). This database contains measurements of 35 electrophysiological signals for 12,000 patients who were monitored for 48 hours in the intensive care unit. Through our empirical analysis, we demonstrate how our Shi-VAE model achieves competitive results with the GP-VAE method. Moreover, we observe a higher correlation between the imputed signals and the ground truth. These findings emphasize the effectiveness and suitability of our model in effectively addressing the challenges presented by real-world scenarios involving diverse and missing data.

This thesis contribution demonstrates two key aspects. Firstly, it showcases the viability of using deep generative models for addressing the data wrangling challenges in medical scenarios. Secondly, it outlines a pathway for aggregating diverse data streams into a shared latent space. This latent space serves a dual purpose: it allows for the propagation of information across time and enables the generation or imputation of missing data by leveraging a deeper and more generalized conceptual space.

6.1.2 Part II: Learning to Defer to Multiple Experts

The structure of the second part of the thesis mirrors that of the first part. Chapter 4 acts as an introductory chapter to the theory of learning to defer, without contributing anything novel. Initially, we introduce the notation for a general classification problem and illustrate with an example that human-machine collaboration can be a viable solution in certain scenarios. Subsequently, we provide the background on learning to defer and proceed to present two main surrogate losses referenced in this thesis: the *softmax* (Mozannar and Sontag, 2020) loss and the *OvA* (Verma and Nalisnick, 2022) loss. We also discuss the issue of softmax generating invalid probability estimates for expert confidence, as identified by (Verma and Nalisnick, 2022), and how the *OvA* formulation circumvents this issue. This chapter establishes the foundation for our original contribution in Chapter 5.

Chapter 5 introduces the second contribution of this thesis, which involves extending the concept of learning to defer to multiple experts. Initially, we propose two novel surrogate losses derived from the *softmax* and *OvA* formulations, taking into account the scenario where there are multiple experts to defer to, as opposed to just one. Theoretical analysis demonstrates the consistency of these surrogate losses. Furthermore, we investigate whether the issue of confidence calibration observed in the single expert setting also persists in the multi-expert setting. Our findings confirm that this problem indeed persists: the *softmax* surrogate loss generates unreliable

confidence estimates for expert confidences, leading to miscalibration issues, whereas the *OvA* formulation addresses this issue to some extent, albeit with practical trade-offs.

Additionally, we contribute to the effective performance of expert ensembling by proposing a conformal inference technique applied to sets of experts during the deferring process. Initially, we present a naive conformal inference technique, which, under certain expert settings, fails to identify all the correct experts and results in larger conformal sets. We discover that this is attributed to experts not consistently predicting the same output due to the inherent probability of their correctness. To address this issue, we propose a regularized version of the technique that successfully identifies all the correct experts. To validate our approach, we conduct empirical experiments on various classification tasks, including standard image classification, galaxy classification, skin lesion classification, and hate speech classification. Our regularized conformal ensemble procedure demonstrates its capability to accurately identify the experts who have a higher likelihood of making correct predictions, even when considering variations in the individual experts' probabilities of being correct.

This final contribution holds twofold significance. Firstly, it enriches the learning to defer literature by providing a valuable reference for dealing with multiple experts, which is a common situation in real-world contexts. The theoretical and empirical validation across diverse datasets and classification tasks enhances its practical relevance. Secondly, the contribution highlights the importance of a conformal inference procedure for effectively identifying the most accurate experts and filtering out those with lower probabilities, thus ensuring reliable predictions and mitigating potential misleading or adversarial outcomes.

6.2 Suggestions for Future Research

Given the comprehensive coverage of the two main parts discussed throughout the manuscript, we now delve into new research ideas within these areas. By presenting these future research directions, our goal is to stimulate further exploration, foster innovation, and address critical concerns within the domains encompassed by this thesis.

6.2.1 Next Steps in VAEs for Irregular Data

Firstly, we explore future research ideas for variational autoencoders that address challenges such as handling heterogeneous likelihoods, missing data, and temporal data. These areas offer opportunities for developing innovative approaches to enhance the capabilities and effectiveness of variational autoencoders.

Heterogeneous Likelihoods

More flexible distributions In this thesis we tackle the problem of likelihood penalization when working with likelihood from different domains, *i.e.* continuous and discrete likelihoods. For the discrete case, we worked with binary and categorical distributions. One idea we have in mind is to use more flexible discrete distributions, such as the continuous Bernoulli [Loaiza-Ganem and Cunningham \(2019\)](#) or the categorical distribution [Gordon-Rodriguez et al. \(2020\)](#). The support for the Bernoulli distribution is $\{0, 1\}$, but, for example, pixel data is $[0, 1]$. The continuous Bernoulli ([Loaiza-Ganem and Cunningham, 2019](#)) is a new parameter distribution with $[0, 1]$ -support. The continuous categorical distribution ([Gordon-Rodriguez et al., 2020](#)) represents the nontrivial multivariate generalization of the continuous Bernoulli. These two distributions could help to prevent potential issues and alleviate the heterogeneous likelihood problem.

Exponential families Another next step in handling heterogeneous likelihoods is to develop a framework that enables the mapping of diverse domain-specific likelihoods onto a common space, allowing for fair and balanced comparison between them. One way is to express well-used statistical distributions (*e.g.* normal, gamma, beta, Bernoulli, Categorical, *etc*) as *exponential families*. With this approach we are able fit different distributions using the same parametrization *formula*, where only the fitted parameters change (*natural parameters*). Also, the likelihood imbalance problem is circumvented this way. The concept of expressing distributions as exponential families and the related challenge of likelihood balancing have been addressed in a study by [Javaloy and Valera \(2020\)](#). They propose a novel data preprocessing technique known as Lipschitz *standardization*, which aims to balance the likelihoods across variables. By applying this preprocessing step, the likelihood imbalance problem can be effectively circumvented. Also the theory of energy based models (EBM) ([LeCun et al., 2006](#)) could be another interesting path to follow. In a recent work, [Wu et al. \(2021\)](#) propose a conjugate energy based models (CEBM), where among others proposals, we find interesting the idea of using these methods for modeling the data not at pixel level (for the case of images) but at the level of latent representations.

Modeling data-relationship, rather than data itself

The concept of not directly modeling the data itself, but instead focusing on the latent representation, introduces a promising paradigm where the objectives or training methods need not be solely based on the data level. Instead, we can concentrate on modeling the relationships or correlations between data samples that consist of various likelihoods or modalities, aiming to learn these relationships directly. This approach involves designing a framework that captures the dependencies between the samples.

This novel idea aligns well with the principles found in Gaussian process literature and kernel methods more broadly. In particular, we draw inspiration from the emerging field of Deep Kernel Learning ([Wilson et al., 2016](#); [Heaukulani and van der Wilk, 2019](#); [Aitchison et al., 2021](#)). Deep Kernel Learning (DKL) is a framework that merges the power of deep learning with the flexibility of kernel methods. It involves learning a data-dependent kernel function using a neural network, known as the kernel network, which replaces the traditional fixed kernel functions used in kernel methods. By doing so, DKL enables the modeling of complex relationships and the capture of non-linear dependencies within the data. We believe, as modeling data is important, properly modeling the relationship itself is important too. We believe that discovering a principled framework for generating *data relationships* rather than just data could potentially address challenges in various domains, including healthcare.

Advanced Techniques for Temporal Data

The Shi-VAE model effectively handled temporal data streams by utilizing a continuous latent space that captured the temporal dynamics of the sequences. In our approach, we opted to employ the LSTM architecture, a simplex yet effective recurrent neural network (RNN) architecture, for handling the temporal data. It is worth noting that there exist more recent or enhanced versions of RNN architectures that could have been utilized. For instance, bidirectional LSTM ([Schuster and Paliwal, 1997](#)) could have been employed to leverage information from both the past and future. However, due to the autoregressive nature of our VAE model, incorporating the bidirectional scheme presents challenges in terms of derivation. Also the architectures proposed in other works

dealing with missing data and temporal data could also be explored (Lipton et al., 2016b; Che et al., 2018; Cao et al., 2018). For next steps we could also think on incorporating attention mechanisms using Transformers (Vaswani et al., 2017) or similar architectures. However, it should be noted that integrating such architectures may not be straightforward, similar to the bidirectional case.

In the context of temporal sequences, our sequences had missing data and varying lengths, which added complexity to the problem. However, it is worth exploring how irregular time intervals can be incorporated and addressed in future works. Recent studies, such as the work by Schirmer et al. (2022), have tackled this issue and may provide interesting insights and approaches to solving the problem of irregular time intervals.

Active Learning

Imagine a situation in which we are tasked with making a drug prescription based on a patient’s clinical history. However, the clinical records may contain missing values due to irregular patient visits or other factors. Those missing values could be retrieved from different labs or hospitals, but with an associated cost. This example sparked our interest in utilizing our Shi-VAE model to determine *which* variables to retrieve and *when*, aiming to optimize decision-making. This concept aligns with the principles of *active learning*, which is a machine learning approach that involves strategically selecting informative data points to label and incorporate into the training process, thereby improving model performance with fewer labeled examples.

We found inspiration in the work of Ma et al. (2019), who proposed the EDDI framework, a VAE-based model with an acquisition function that maximizes expected information gain for a set of target variables. Our aim was to extend the EDDI model within our Shi-VAE framework, allowing us to not only identify the most valuable information-providing variable but also determine the optimal timing for retrieval. Although this idea was under development during the writing of the thesis, conclusive results were not yet sufficiently robust to draw firm conclusions about the proposal.

6.2.2 Next Steps in Learning to Defer

We explore new directions for learning to defer, including novel ideas, practical applications, and addressing concerns related to human-machine collaboration and reliance on AI systems.

Application of Learning to Defer to Multiple Experts

Our contribution (Verma, Barrejón, and Nalisnick, 2023) was tested across various tasks, including galaxy classification, skin lesion identification, and hate speech classification. This work not only strengthens existing L2D approaches to handle multiple experts but also introduces new possibilities for its application in different fields, where it can be effectively deployed in real-life scenarios. In our manuscript we mainly motivated the L2D framework within the medical context, where an L2D system could be employed to assign ‘easy’ tasks to doctors while seeking their expertise for more challenging and high-risk scenarios. On the other hand, the conformal ensembling procedure described in our work allows us to have the flexibility to identify and filter experts based on their expected reliability. Once we obtain the pool of experts, we can make decisions on how many experts to query, considering that this process can be costly.

Furthermore, this system could be incorporated into forums or chat platforms to address situations where individuals who promote hatred and are challenging to identify could be referred

to an team of administrators. This feature would also be particularly valuable in social media platforms, where algorithms often operate without human intervention, and striking the right balance between preventing harmful content and preserving freedom of speech is a sensitive issue. Therefore, incorporating human control becomes essential.

The application of conformal ensembling of experts as a *filtering* procedure extends to various scenarios. For instance, in a law firm, it can be implemented to delegate different legal processes to the most reliable lawyers specialized in their respective fields. This approach effectively mitigates the biases that may naturally be present in other lawyers. In summary, there is a wide range of possibilities where the multi-expert setting of L2D could be applied.

Temporal L2D with Conformal Guarantees

Another interesting venue of research is extending L2D to deal with temporal sequences, as studied by Joshi et al. (2023), but incorporating the application of conformal procedures. However, when dealing with sequential data, the application of conformal inference is not straightforward due to the violation of the exchangeability assumption. Therefore, it is worth exploring new approaches proposed by Barber et al. (2023) or Stankeviciute et al. (2021) to address this issue.

New Forms of Calibration

In this thesis, we worried about calibration following the definition of *confidence calibration* proposed by Guo et al. (2017). But for future works, it would be interesting applying the notion of calibration proposed by Gupta and Ramdas (2022), where they propose using *top-label calibration* instead of confidence calibration in multi-class problems. The concept of top-label calibration aligns perfectly with our goal of generating more trust-worthy outputs from machine learning ML models to facilitate decision-making. Additionally, the authors of the original work suggest a framework for reducing multiclass problems to binary ones, making the comparison with the One-vs-All (OvA) approach a promising initial step.

Conformalize L2D

An additional idea is to extend the application of conformal inference to L2D by not only ensuring reliable ensembles of experts, as demonstrated in our previous work (Verma et al., 2023), but also by applying it to the model’s output, as demonstrated by Babbar et al. (2022). In a real case study, Babbar et al. (2022) validated that conformal predictions provide superior benefits to humans compared to top-1 predictions. Likewise, (Straitouri et al., 2023) introduced a novel method that uses an adaptive conformal inference algorithm to propose a subset of output labels from a classifier to a human expert, and they demonstrated that experts achieve greater prediction accuracy under this approach as well. These findings clearly indicate that employing conformal techniques is an effective approach for incorporating reliability and trust in algorithms facilitating human-machine collaboration, besides the additional benefit of quantifying uncertainty.

Access to Real Experts' Correctness Probabilities

So far, most works on learning to defer tasks have used different datasets from different domains, where the experts' probabilities of correctness are manually designed (in most cases). It would be beneficial to the field having benchmark datasets that not only provide access to the experts' predictions as *hard* ground-truth labels but also as *soft* labels. However, this presents challenges as curating such datasets is time-consuming, requires a substantial statistical population to ensure fairness, and, importantly, requires a substantial financial cost. Nonetheless, some authors have started releasing datasets or making expert probabilities available. For instance, [Collins et al. \(2022\)](#) released CIFAR10-S, a version of CIFAR10 with soft labels, and [\(Tschandl et al., 2018\)](#) made available the skin lesion dataset we used in our experiments, which now includes access to both human and machine learning probabilities.

Balancing Reliance on AI

In the context of L2D, it is imperative to enhance the collaboration between artificial intelligence and humans in decision-making by developing a deep comprehension of how humans perceive machines. While machine learning outputs like saliency maps, LIME (Local Interpretable Model-agnostic Explanations) or t-SNE plots can offer valuable insights for data understanding, it is worth noting that there are cases where these interpretable outputs can have the opposite effect ([Lipton, 2018](#)). Furthermore, in practice, these interpretable outputs can potentially mislead professionals in their decision-making process [Arun et al. \(2021\)](#).

Hence, it is imperative to investigate how humans comprehend the functioning of AI and the impact of their actions on it. This concept is commonly referred to as the *mental model*, and exploring this area holds great promise for research. In a study conducted by [Gaube et al. \(2021\)](#), radiologists and physicians were presented with eight instances of advice, which could either originate from humans or AI. Their task was to evaluate the quality of the advice. However, the trick was that all the advice came from humans, and only four cases were correct. The findings revealed that experts were able to distinguish between good and poor advice, but they rated human advice significantly higher in quality, thus showing a clear underreliance on AI.

Consider now the scenario where a clinician is informed that the AI assistant occasionally outperforms humans. In such a situation, there is an incentive to rely more on the AI system. Consequently, it becomes necessary to develop algorithms that can address these inherent biases. As pointed out by [Bućinca et al. \(2021\)](#), if individuals can be encouraged to engage in more analytical thinking when processing AI recommendations and explanations, the tendency to overly rely on AI will diminish.

Closing Statement

We started the thesis with the question: *How can humans leverage machine learning?* Throughout this thesis, our focus revolved around exploring the interaction between humans and machine learning methods to achieve better results. Firstly, we regarded AI as a valuable tool for alleviating the data processing burden. Secondly, we recognized AI as an automated system capable of discerning situations where human involvement is necessary. From our findings, we can conclude that *humans can leverage machine learning*. And also that AI systems can achieve great goals and produce valuable information. But one thing is certain: *not alone*.

Appendices



ELBO Derivation for Shi-VAE

A.1 Shi-VAE *without* discrete latent space \mathbf{s}_t

The joint probability is described by the following equation.

$$p(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T}) = \prod_{t=1}^T p(\mathbf{z}_t | \mathbf{z}_{< t}) p(\mathbf{x}_t | \mathbf{z}_{\leq t}) \quad (\text{A.1})$$

We consider that data \mathbf{x}_t can be divided into observed data and missing data, such that

$$p(\mathbf{x}_t | \mathbf{z}_{\leq t}) = \prod_{d \in \mathcal{O}_t} p(x_{td} | \mathbf{z}_{\leq t}) \prod_{d \in \mathcal{M}_t} p(x_{td} | \mathbf{z}_{\leq t}) = p(\mathbf{x}_t^o | \mathbf{z}_{\leq t}) p(\mathbf{x}_t^m | \mathbf{z}_{\leq t}). \quad (\text{A.2})$$

The variational distribution can be defined as

$$q(\mathbf{x}_{\leq T}^m, \mathbf{z}_{\leq T} | \mathbf{x}_{\leq T}^o) = \prod_{t=1}^T q(\mathbf{z}_t | \mathbf{x}_{\leq t}^o, \mathbf{z}_{< t}) p(\mathbf{x}_t^m | \mathbf{z}_{\leq t}) \quad (\text{A.3})$$

Generative

$$p(\mathbf{x}_t | \mathbf{z}_{< t}) = \prod_{d=1}^D p(x_{td} | \mathbf{z}_t) = \prod_{d=1}^D p(x_{td} | \gamma_{td} = h_d(\mathbf{y}_{t,d}, \mathbf{h}_{t-1})) \quad (\text{A.4})$$

Notice we have one DNN per dimension of the input sample \mathbf{x}_t , *i.e.*, $\eta_d(\cdot)$ for each dimension d and $\mathbf{Y}_t = [\mathbf{y}_{t,1}, \dots, \mathbf{y}_{t,d}] = \varphi_{\omega}^{\mathbf{z}}(\mathbf{z}_t)$ This allows to cope with heterogeneous data.

Inference

$$q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{z}_{< t}) = \mathcal{N}(\boldsymbol{\mu}_q(\tilde{\mathbf{x}}_t), \boldsymbol{\Sigma}_q(\tilde{\mathbf{x}}_t)), \quad (\text{A.5})$$

where $\tilde{\mathbf{x}}_t$ denotes a D -dimensional vector where the missing dimensions have been replaced by zeros, and $\boldsymbol{\mu}_q(\tilde{\mathbf{x}}_t)$ and $\boldsymbol{\Sigma}_q(\tilde{\mathbf{x}}_t)$ are parametrized as

$$[\boldsymbol{\mu}_q(\tilde{\mathbf{x}}_t), \boldsymbol{\Sigma}_q(\tilde{\mathbf{x}}_t)] = \varphi_{\omega}^{\text{enc}}(\varphi_{\omega}^{\mathbf{x}}(\tilde{\mathbf{x}}_t), \mathbf{h}_{t-1}) \quad (\text{A.6})$$

Prior

$$p(\mathbf{z}_t | \mathbf{z}_{< t}) = \mathcal{N}(\boldsymbol{\mu}_{o,t}, \boldsymbol{\Sigma}_{o,t}^2), \text{ where } [\boldsymbol{\mu}_{o,t}, \boldsymbol{\Sigma}_{o,t}^2] = \varphi_{\omega}^{\text{prior}}(\mathbf{h}_{t-1}) \quad (\text{A.7})$$

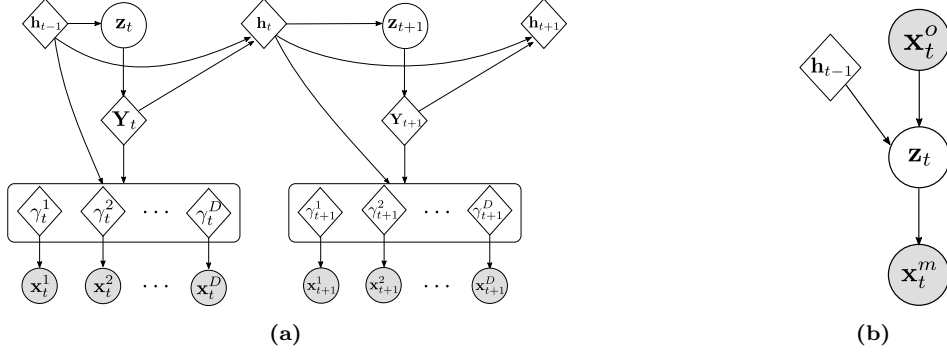


Figure A.1: Generative model (a) and inference model (b) for Shi-VAE *without* discrete latent space \mathbf{s}_t .

RNN state

$$\mathbf{h}_t = f_{\theta}(\mathbf{Y}_t, \mathbf{h}_{t-1}), \quad (\text{A.8})$$

where f_{θ} is a RNN such as LSTM or GRU.

ELBO

$$\begin{aligned} \log p(\mathbf{X}_{\leq T}^o) &\geq \int \int q(\mathbf{x}_{\leq T}^m, \mathbf{z}_{\leq T} | \mathbf{x}_{\leq T}^o) \log \left(\frac{p(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T})}{q(\mathbf{x}_{\leq T}^m, \mathbf{z}_{\leq T} | \mathbf{x}_{\leq T}^o)} \right) d\mathbf{z}_{\leq T}, d\mathbf{x}_t^m \\ &= \int \int \sum_{t=1}^T q(\mathbf{x}_{\leq T}^m, \mathbf{z}_{\leq T} | \mathbf{x}_{\leq T}^o) \log \left(\frac{p(\mathbf{z}_t | \mathbf{z}_{< t}) p(\mathbf{x}_t^o, \mathbf{x}_t^m | \mathbf{z}_{\leq t})}{q(\mathbf{x}_t^m, \mathbf{z}_{\leq t} | \mathbf{x}_{\leq t}^o)} \right) d\mathbf{z}_{\leq t}, d\mathbf{x}_t^m \\ &= \sum_{t=1}^T \int \int q(\mathbf{x}_{\leq t}^m, \mathbf{z}_{\leq t} | \mathbf{x}_{\leq t}^o) \log \left(\frac{p(\mathbf{z}_t | \mathbf{z}_{< t}) p(\mathbf{x}_t^o | \mathbf{z}_{\leq t}) p(\mathbf{x}_t^m | \mathbf{z}_{\leq t})}{p(\mathbf{x}_t^m | \mathbf{z}_{\leq t}) q(\mathbf{z}_t | \mathbf{x}_{\leq t}^o, \mathbf{z}_{< t})} \right) d\mathbf{z}_{\leq t}, d\mathbf{x}_t^m \\ &= \sum_{t=1}^T \int \int q(\mathbf{x}_{\leq t}^m, \mathbf{z}_{\leq t} | \mathbf{x}_{\leq t}^o) \log p(\mathbf{x}_t^o | \mathbf{z}_{\leq t}) d\mathbf{z}_{\leq t}, d\mathbf{x}_t^m \quad \textcircled{\text{I}} \\ &\quad - \int \int q(\mathbf{x}_{\leq t}^m, \mathbf{z}_{\leq t} | \mathbf{x}_{\leq t}^o) \log \left(\frac{q(\mathbf{z}_t | \mathbf{x}_{\leq t}^o, \mathbf{z}_{< t})}{p(\mathbf{z}_t | \mathbf{z}_{< t})} \right) d\mathbf{z}_{\leq t}, d\mathbf{x}_t^m \quad \textcircled{\text{II}} \end{aligned}$$

For the first term $\textcircled{\text{I}}$ we can derive the following expression

$$\begin{aligned} \int \int q(\mathbf{x}_{\leq t}^m, \mathbf{z}_{\leq t} | \mathbf{x}_{\leq t}^o) \log p(\mathbf{x}_t^o | \mathbf{z}_{\leq t}) d\mathbf{z}_{\leq t}, d\mathbf{x}_t^m &= \int \int p(\mathbf{x}_{\leq t}^m | \mathbf{z}_{\leq t}) q(\mathbf{z}_{\leq t} | \mathbf{x}_{\leq t}^o) \log p(\mathbf{x}_t^o | \mathbf{z}_{\leq t}) d\mathbf{z}_{\leq t}, d\mathbf{x}_t^m \\ &= \underbrace{\int p(\mathbf{x}_{\leq t}^m | \mathbf{z}_{\leq t}) d\mathbf{x}_t^m}_1 \int q(\mathbf{z}_{\leq t} | \mathbf{x}_{\leq t}^o) \log p(\mathbf{x}_t^o | \mathbf{z}_{\leq t}) d\mathbf{z}_{\leq t} \\ &= \int q(\mathbf{z}_{\leq t} | \mathbf{x}_{\leq t}^o) \log p(\mathbf{x}_t^o | \mathbf{z}_{\leq t}) d\mathbf{z}_{\leq t}. \end{aligned}$$

The second expression $\textcircled{\text{II}}$ can be rewritten in the following form:

$$\begin{aligned}
& \int \int q(\mathbf{x}_{\leq T}^m, \mathbf{z}_{\leq T} | \mathbf{x}_{\leq T}^o) \log \left(\frac{q(\mathbf{z}_t | \mathbf{x}_{\leq t}^o, \mathbf{z}_{< t})}{p(\mathbf{z}_t | \mathbf{z}_{< t})} \right) d\mathbf{z}_{\leq t}, d\mathbf{x}_t^m \\
&= \int \int p(\mathbf{x}_{\leq t}^m | \mathbf{z}_{\leq t}) q(\mathbf{z}_{\leq t} | \mathbf{x}_{\leq t}^o) \log \left(\frac{q(\mathbf{z}_t | \mathbf{x}_{\leq t}^o, \mathbf{z}_{< t})}{p(\mathbf{z}_t | \mathbf{z}_{< t})} \right) d\mathbf{z}_{\leq t}, d\mathbf{x}_t^m \\
&= \underbrace{\int p(\mathbf{x}_{\leq t}^m | \mathbf{z}_{\leq t}) d\mathbf{x}_t^m}_1 \int q(\mathbf{z}_{< t} | \mathbf{x}_{< t}^o) q(\mathbf{z}_t | \mathbf{x}_{\leq t}^o, \mathbf{z}_{< t}) \log \left(\frac{q(\mathbf{z}_t | \mathbf{x}_{\leq t}^o, \mathbf{z}_{< t})}{p(\mathbf{z}_t | \mathbf{z}_{< t})} \right) d\mathbf{z}_{< t} \\
&= \int q(\mathbf{z}_{< t} | \mathbf{x}_{< t}^o) \text{KL}(q(\mathbf{z}_t | \mathbf{x}_{\leq t}^o, \mathbf{z}_{< t}) || p(\mathbf{z}_t | \mathbf{z}_{< t})) d\mathbf{z}_{< t}.
\end{aligned}$$

The final ELBO expression will be

$$\begin{aligned}
\text{ELBO} &= \sum_{t=1}^T \left(\int q(\mathbf{z}_{\leq t} | \mathbf{x}_{\leq t}^o) \log p(\mathbf{x}_t^o | \mathbf{z}_{\leq t}) d\mathbf{z}_{\leq t} - \int q(\mathbf{z}_{< t} | \mathbf{x}_{< t}^o) \text{KL}(q(\mathbf{z}_t | \mathbf{x}_{\leq t}^o, \mathbf{z}_{< t}) || p(\mathbf{z}_t | \mathbf{z}_{< t})) d\mathbf{z}_{< t} \right) \\
&= \mathbb{E}_{\mathbf{z}_{\leq T} \sim q(\mathbf{z}_{\leq T} | \mathbf{x}_{\leq T}^o)} \left[\sum_{t=1}^T \log p(\mathbf{x}_t^o | \mathbf{z}_{\leq t}) - \text{KL}(q(\mathbf{z}_t | \mathbf{x}_{\leq t}^o, \mathbf{z}_{< t}) || p(\mathbf{z}_t | \mathbf{z}_{< t})) \right]
\end{aligned}$$

A.2 Shi-VAE *with* discrete latent space \mathbf{s}_t

The approach involves a mixture of Gaussian prior on \mathbf{z}_t , temporal dependencies solely on \mathbf{z}_t , and updating the state of the RNN \mathbf{h}_t using $(\mathbf{z}_t, \mathbf{s}_t)$.

Generative

$$p(\mathbf{X}_{\leq T}, \mathbf{Z}_{\leq T}, \mathbf{S}_{\leq T}) = \prod_{t=1}^T p(\mathbf{z}_t | \mathbf{h}_{t-1}, \mathbf{s}_t) p(\mathbf{x}_t | \mathbf{z}_t, \mathbf{h}_{t-1}, \mathbf{s}_t) \quad (\text{A.9})$$

$$= \prod_{t=1}^T p(\mathbf{z}_t | \mathbf{h}_{t-1}, \mathbf{s}_t) p(\mathbf{x}_t | \gamma_{t,d} = h_d(\mathbf{y}_{t,d}, \mathbf{s}_t, \mathbf{h}_{t-1})) p(\mathbf{s}_t), \quad (\text{A.10})$$

where

$$\text{Likelihood : } p(\mathbf{x}_t | \mathbf{z}_t, \mathbf{h}_{t-1}, \mathbf{s}_t) = p(\mathbf{x}_t^m | \mathbf{z}_t, \mathbf{h}_{t-1}, \mathbf{s}_t) p(\mathbf{x}_t^o | \mathbf{z}_t, \mathbf{h}_{t-1}, \mathbf{s}_t) \quad (\text{A.11})$$

$$\text{Transformation : } \mathbf{Y}_t = [\mathbf{y}_{t,1}, \dots, \mathbf{y}_{t,d}] = \varphi_{\omega}^{\mathbf{z}}(\mathbf{z}_t) \quad (\text{A.12})$$

$$\text{Prior } \mathbf{s}_t : p(\mathbf{s}_t) \sim \text{Categorical}(K^{-1}) \quad (\text{A.13})$$

$$\text{Prior } \mathbf{z}_t : p(\mathbf{z}_t | \mathbf{h}_{t-1}, \mathbf{s}_t) = \mathcal{N}(\mu_{o,t}, \Sigma_{o,t}^2), \quad [\mu_{o,t}, \Sigma_{o,t}^2] = \varphi_{\omega}^{\text{prior}}(\mathbf{h}_{t-1}, \mathbf{s}_t) \quad (\text{A.14})$$

$$\text{RNN State : } \mathbf{h}_t = f_{\theta}(\mathbf{y}_t, \mathbf{h}_{t-1}) \quad (\text{A.15})$$

Recognition

$$q(\mathbf{Z}_{\leq T}, \mathbf{S}_{\leq T}, \mathbf{X}_{\leq T}^m | \mathbf{X}_{\leq T}^o) = \prod_{t=1}^T q(\mathbf{s}_t | \mathbf{x}_t^o) q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{h}_{t-1}, \mathbf{s}_t) p(\mathbf{x}_t^m | \mathbf{z}_t, \mathbf{s}_t) \quad (\text{A.16})$$

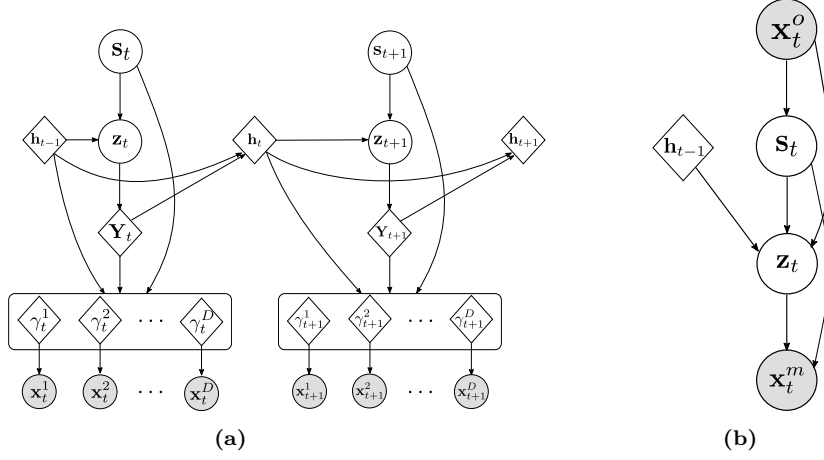


Figure A.2: Generative model (Fig. A.2a) and recognition model (Fig. A.2b) for T-missVAE with Gaussian Mixture Prior.

where

$$q(\mathbf{s}_t | \mathbf{x}_t^o) = \text{Categorical}(\boldsymbol{\pi}(\tilde{\mathbf{x}}_t^\varphi)) \quad (\text{A.17})$$

$$q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{h}_{t-1}, \mathbf{s}_t) = \mathcal{N}(\boldsymbol{\mu}_q(\tilde{\mathbf{x}}_t^\varphi, \mathbf{s}_t, \mathbf{h}_{t-1}), \boldsymbol{\Sigma}_q(\tilde{\mathbf{x}}_t^\varphi, \mathbf{s}_t, \mathbf{h}_{t-1})) \quad (\text{A.18})$$

$$\tilde{\mathbf{x}}_t^\varphi = \varphi_\omega^\mathbf{x}(\tilde{\mathbf{x}}_t) \quad (\text{A.19})$$

$$(\text{A.20})$$

ELBO

$$\begin{aligned} \log p(\mathbf{X}_{\leq T}^o) &\geq \int \int \int q(\mathbf{X}_{\leq T}^m, \mathbf{Z}_{\leq T}, \mathbf{S}_{\leq T} | \mathbf{X}_{\leq T}^o) \log \left(\frac{p(\mathbf{X}_{\leq T}, \mathbf{S}_{\leq T}, \mathbf{Z}_{\leq T})}{q(\mathbf{X}_{\leq T}^m, \mathbf{Z}_{\leq T}, \mathbf{S}_{\leq T} | \mathbf{X}_{\leq T}^o)} \right) d\mathbf{Z}_{\leq T} d\mathbf{S}_{\leq T} d\mathbf{X}_{\leq T}^m \\ &= \sum_{t=1}^T \int \int \int q(\mathbf{x}_t^m, \mathbf{z}_t, \mathbf{s}_t | \mathbf{x}_t^o) \log \left(\frac{p(\mathbf{x}_t^o | \mathbf{z}_t, \mathbf{s}_t) p(\mathbf{x}_t^m | \mathbf{z}_t, \mathbf{s}_t) p(\mathbf{z}_t | \mathbf{h}_{t-1}, \mathbf{s}_t) p(\mathbf{s}_t)}{q(\mathbf{s}_t | \mathbf{x}_t^o) q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{h}_{t-1}, \mathbf{s}_t) p(\mathbf{x}_t^m | \mathbf{z}_t, \mathbf{s}_t)} \right) d\mathbf{z}_t d\mathbf{s}_t d\mathbf{x}_t^m \\ &= \sum_{t=1}^T \left[\int \int \int q(\mathbf{x}_t^m, \mathbf{z}_t, \mathbf{s}_t | \mathbf{x}_t^o) \log p(\mathbf{x}_t^o | \mathbf{z}_t, \mathbf{s}_t) d\mathbf{z}_t d\mathbf{s}_t d\mathbf{x}_t^m \quad \textcircled{\text{I}} \right. \\ &\quad - \int \int \int q(\mathbf{x}_t^m, \mathbf{z}_t, \mathbf{s}_t | \mathbf{x}_t^o) \log \left(\frac{q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{h}_{t-1}, \mathbf{s}_t)}{p(\mathbf{z}_t | \mathbf{h}_{t-1}, \mathbf{s}_t)} \right) d\mathbf{z}_t d\mathbf{s}_t d\mathbf{x}_t^m \quad \textcircled{\text{II}} \\ &\quad \left. - \int \int \int q(\mathbf{x}_t^m, \mathbf{z}_t, \mathbf{s}_t | \mathbf{x}_t^o) \log \left(\frac{q(\mathbf{s}_t | \mathbf{x}_t^o)}{p(\mathbf{s}_t)} \right) d\mathbf{z}_t d\mathbf{s}_t d\mathbf{x}_t^m \right] \quad \textcircled{\text{III}} \end{aligned}$$

The first term $\textcircled{\text{I}}$ corresponds to the reconstruction:

$$\begin{aligned} &\int \int \int q(\mathbf{x}_t^m, \mathbf{z}_t, \mathbf{s}_t | \mathbf{x}_t^o) \log p(\mathbf{x}_t^o | \mathbf{z}_t, \mathbf{s}_t) d\mathbf{z}_t d\mathbf{s}_t d\mathbf{x}_t^m \\ &= \int \int \int q(\mathbf{s}_t | \mathbf{x}_t^o) q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{h}_{t-1}, \mathbf{s}_t) p(\mathbf{x}_t^m | \mathbf{z}_t, \mathbf{s}_t) \log p(\mathbf{x}_t^o | \mathbf{z}_t, \mathbf{s}_t) d\mathbf{z}_t d\mathbf{s}_t d\mathbf{x}_t^m \\ &= \int \int q(\mathbf{s}_t | \mathbf{x}_t^o) q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{h}_{t-1}, \mathbf{s}_t) \log p(\mathbf{x}_t^o | \mathbf{z}_t, \mathbf{s}_t) d\mathbf{z}_t d\mathbf{s}_t \\ &= \mathbb{E}_{q(\mathbf{s}_t | \mathbf{x}_t^o)} \mathbb{E}_{q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{h}_{t-1}, \mathbf{s}_t)} [\log p(\mathbf{x}_t^o | \mathbf{z}_t, \mathbf{s}_t)] \end{aligned}$$

The second term **(II)** :

$$\begin{aligned}
& \int \int \int q(\mathbf{x}_t^m, \mathbf{z}_t, \mathbf{s}_t | \mathbf{x}_t^o) \log \left(\frac{q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{h}_{t-1}, \mathbf{s}_t)}{p(\mathbf{z}_t | \mathbf{h}_{t-1}, \mathbf{s}_t)} \right) d\mathbf{z}_t d\mathbf{s}_t d\mathbf{x}_t^m \\
&= \int \int q(\mathbf{s}_t | \mathbf{x}_t^o) q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{h}_{t-1}, \mathbf{s}_t) \log \left(\frac{q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{h}_{t-1}, \mathbf{s}_t)}{p(\mathbf{z}_t | \mathbf{h}_{t-1}, \mathbf{s}_t)} \right) d\mathbf{z}_t d\mathbf{s}_t \\
&= \mathbb{E}_{q(\mathbf{s}_t | \mathbf{x}_t^o)} [\text{KL}(q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{h}_{t-1}, \mathbf{s}_t) || p(\mathbf{z}_t | \mathbf{h}_{t-1}, \mathbf{s}_t))]
\end{aligned}$$

The third term **(III)** :

$$\begin{aligned}
& \int \int \int q(\mathbf{x}_t^m, \mathbf{z}_t, \mathbf{s}_t | \mathbf{x}_t^o) \log \left(\frac{q(\mathbf{s}_t | \mathbf{x}_t^o)}{p(\mathbf{s}_t)} \right) d\mathbf{z}_t d\mathbf{s}_t d\mathbf{x}_t^m \\
&= \int \int q(\mathbf{s}_t | \mathbf{x}_t^o) q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{h}_{t-1}, \mathbf{s}_t) \log \left(\frac{q(\mathbf{s}_t | \mathbf{x}_t^o)}{p(\mathbf{s}_t)} \right) d\mathbf{z}_t d\mathbf{s}_t \\
&= \text{KL}(q(\mathbf{s}_t | \mathbf{x}_t^o) || p(\mathbf{s}_t)) \int q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{h}_{t-1}, \mathbf{s}_t) d\mathbf{z}_t \\
&= \text{KL}(q(\mathbf{s}_t | \mathbf{x}_t^o) || p(\mathbf{s}_t))
\end{aligned}$$

The final ELBO has the following form:

$$\begin{aligned}
\log p(\mathbf{X}_{\leq T}^o) &\geq \sum_{t=1}^T \left[\mathbb{E}_{q(\mathbf{s}_t | \mathbf{x}_t^o)} \mathbb{E}_{q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{h}_{t-1}, \mathbf{s}_t)} [\log p(\mathbf{x}_t^o | \mathbf{z}_t, \mathbf{s}_t)] \right. \\
&\quad \left. - \mathbb{E}_{q(\mathbf{s}_t | \mathbf{x}_t^o)} [\text{KL}(q(\mathbf{z}_t | \mathbf{x}_t^o, \mathbf{h}_{t-1}, \mathbf{s}_t) || p(\mathbf{z}_t | \mathbf{h}_{t-1}, \mathbf{s}_t))] \right. \\
&\quad \left. - \text{KL}(q(\mathbf{s}_t | \mathbf{x}_t^o) || p(\mathbf{s}_t)) \right]
\end{aligned}$$

B

Proofs and Derivations for Multiple Expert L2D

In this section, we provide proofs for the main results in the paper. We derive the Bayes optimal rule for L2D to multiple experts, and show that the surrogate losses proposed in the paper are consistent. Next, we show that the mixture of experts formulation (Hemmer et al., 2022) is not consistent. We continue the notation from the main paper. For simplicity, we do not worry about measure-theoretic considerations and assume that appropriate conditions hold that allow us to interchange summations and integrals, for example. We begin by giving a formal definition of what it means for a surrogate loss to be consistent.

Definition B.0.1. (Consistent Loss Function). A surrogate loss function $\psi : \mathcal{C} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ operating in the surrogate space $\mathcal{C} \subseteq \mathbb{R}^K$ along with some suitable decoding function $g : \mathcal{C} \rightarrow \mathcal{Y}$ is said to be consistent if for all distributions \mathcal{D} , $\forall \epsilon > 0$, $\exists \delta > 0$ such that if

$$|\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [\psi(y, \mathbf{c}(\mathbf{x}))] - \inf_{\mathbf{u} \in \mathcal{C}} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [\psi(y, \mathbf{u})]| < \delta, \quad (\text{B.1})$$

holds for a prediction function $h : \mathcal{X} \rightarrow \mathcal{C}$, $h(\mathbf{x}) := \mathbf{c}(\mathbf{x}) \in \mathbb{R}^K$, then it must hold that

$$\mathbb{P}_{(\mathbf{x}, y) \sim \mathcal{D}} [g \circ h(\mathbf{x}) \neq y] \leq \mathbb{P}_{(\mathbf{x}, y) \sim \mathcal{D}} [h^*(\mathbf{x}) \neq y] + \epsilon, \quad (\text{B.2})$$

where $h^* : \mathcal{X} \rightarrow \mathcal{Y}$ is the Bayes optimal predictor.

A common example of one decoding function in machine learning is the arg max function. Intuitively, consistency implies that the minimization of a surrogate loss $\psi(\cdot)$ results in a prediction function $h(\cdot)$ whose expected error converges to the Bayes risk.

B.1 Bayes Rule for Learning to Defer with Multiple Experts

We have J experts and a classifier, where the system either allows the classifier to make the final prediction or defers to one of the J experts. When the classifier makes the prediction, the system incurs loss $\ell_{\text{clf}}(\hat{y}, y)$ where $\hat{y} = h(\mathbf{x})$. When the system defers to the j^{th} expert, it incurs a loss $\ell_{\text{exp}}(m_j, y)$. In what follows, we frame the learning to defer problem as a general classification problem, and aim to find a function $g : \mathcal{X} \rightarrow \hat{\mathcal{Y}} := \mathcal{Y} \cup \{\perp_1, \perp_2, \dots, \perp_J\}$ and $|\hat{\mathcal{Y}}| = K + J$ with

the minimum expected loss (also known as *risk*). We consider g as modeling a probabilistic decision rule $\delta(\hat{y}|\mathbf{x}) := [\delta(\hat{y} = 1|\mathbf{x}), \delta(\hat{y} = 2|\mathbf{x}), \dots, \delta(\hat{y} = K + J|\mathbf{x})]$ where $\delta(\hat{y} = i|\mathbf{x})$ denotes the confidence in making the i^{th} decision for $\mathbf{x} \sim \mathbf{x}$. We write *risk* as follows:

$$\mathcal{R}_{\mathcal{D}}[\delta(\hat{y}|\mathbf{x})] = \sum_{i=1}^K \sum_{j=1}^{K+J} \int_{\mathbf{x}} \delta(\hat{y} = j|\mathbf{x}) \ell(\hat{y} = j, y = i) \mathbb{P}(y = i) \mathbb{P}(\mathbf{x}|y = i) d\mathbf{x}, \quad (\text{B.3})$$

where $\ell : (\hat{y}, y) \mapsto \mathbb{R}_+$ is a general loss function, i runs over the input label space, and j runs over the output prediction space (classifier and all the experts). We further expand the risk in Equation B.3 based on the definition of the loss function in learning to defer

$$\begin{aligned} \mathcal{R}_{\mathcal{D}}[\delta(\hat{y}|\mathbf{x})] &= \sum_{i=1}^K \int_{\mathbf{x}} \left(\sum_{j=1}^K \delta(\hat{y}_j|\mathbf{x}) \ell_{\text{clf}}(j, i) \right. \\ &\quad \left. + \sum_{j=K+1}^{K+J} \left(\sum_{m=1}^K \delta(\hat{y}_j|\mathbf{x}) \ell_{\text{exp}}(m_j, y) \mathbb{P}(m_j|\mathbf{x}, y = i) \right) \right) \mathbb{P}(y = i) \mathbb{P}(\mathbf{x}|y = i) d\mathbf{x}, \end{aligned}$$

where we have used shorthand $\delta(\hat{y}_j|\mathbf{x})$ to denote $\delta(\hat{y} = j|\mathbf{x})$, and $\mathbb{P}(m_j|\mathbf{x}, y = i) = \mathbb{P}(m_j = m|\mathbf{x}, y = i)$.

Next, we denote:

$$\begin{aligned} w_{i,j} &= \ell_{\text{clf}}(j, i) \\ w_{i,\perp_j} &= \sum_{m=1}^K \delta(\hat{y}_j|\mathbf{x}) \ell_{\text{exp}}(m_j, y) \mathbb{P}(m_j|\mathbf{x}, y = i) \end{aligned}$$

Thus, $\mathcal{R}_{\mathcal{D}}[\delta(\hat{y}|\mathbf{x})]$ can be written as:

$$\mathcal{R}_{\mathcal{D}}[\delta(\hat{y}|\mathbf{x})] = \sum_{i=1}^K \int_{\mathbf{x}} \left(\sum_{j=1}^K \delta(\hat{y}_j|\mathbf{x}) w_{i,j} + \sum_{j=K+1}^{K+J} \delta(\hat{y}_j|\mathbf{x}) w_{i,\perp_j} \right) \mathbb{P}(y = i) \mathbb{P}(\mathbf{x}|y = i) d\mathbf{x}.$$

Denote $w_{i,\perp}^* := \min_{j \in [J]} \{w_{i,\perp_j}\}$, we have

$$\mathcal{R}_{\mathcal{D}}[\delta(\hat{y}|\mathbf{x})] \geq \sum_{i=1}^K \int_{\mathbf{x}} \left(\sum_{j=1}^K \delta(\hat{y}_j|\mathbf{x}) w_{i,j} + \sum_{j=K+1}^{K+J} \delta(\hat{y}_j|\mathbf{x}) w_{i,\perp}^* \right) \mathbb{P}(y = i) \mathbb{P}(\mathbf{x}|y = i) d\mathbf{x}.$$

We also denote $\sum_{j=K+1}^{K+J} \delta(\hat{y}_j|\mathbf{x})$ as $\delta(\hat{y}_{\perp}|\mathbf{x})$. Then the lower bound of $\mathcal{R}_{\mathcal{D}}[\delta(\hat{y}|\mathbf{x})]$, denoted as $\bar{\mathcal{R}}_{\mathcal{D}}[\delta(\hat{y}|\mathbf{x})]$, is

$$\bar{\mathcal{R}}_{\mathcal{D}}[\delta(\hat{y}|\mathbf{x})] = \sum_{i=1}^K \int_{\mathbf{x}} \left(\sum_{j=1}^K \delta(\hat{y}_j|\mathbf{x}) w_{i,j} + \delta(\hat{y}_{\perp}|\mathbf{x}) w_{i,\perp}^* \right) \mathbb{P}(y = i) \mathbb{P}(\mathbf{x}|y = i) d\mathbf{x}.$$

Since $\sum_{j=1}^K \delta(\hat{y}_j|\mathbf{x}) + \delta(\hat{y}_{\perp}|\mathbf{x}) = 1.0$ and $\int_{\mathbf{x}} \mathbb{P}(\mathbf{x}|y = i) d\mathbf{x} = 1.0$, we follow [Chow \(1957\)](#) to decompose $\bar{\mathcal{R}}_{\mathcal{D}}[\delta(\hat{y}|\mathbf{x})]$ in two terms:

$$\bar{\mathcal{R}}_{\mathcal{D}} = \bar{\mathcal{R}}_{\mathcal{D}}^{\perp} + \bar{\mathcal{R}}_{\mathcal{D}}^{\delta},$$

where

$$\begin{aligned}\bar{\mathcal{R}}_{\mathcal{D}}^{\perp} &= \sum_{i=1}^K \mathbb{P}(y = i) \cdot w_{i,\perp}^*, \\ \bar{\mathcal{R}}_{\mathcal{D}}^{\delta} &= \int_{\mathbf{x}} \sum_{j=1}^{[K] \cup \{\perp\}} \delta(\hat{y}_j | \mathbf{x}) Z_j(\mathbf{x}) d\mathbf{x}, \text{ and} \\ Z_j(\mathbf{x}) &= \sum_{i=1}^K (w_{i,j} - w_{i,\perp}^*) \mathbb{P}(\mathbf{x}) \mathbb{P}(y = i | \mathbf{x}), \quad j \in \{1, 2, \dots, K, \perp\}.\end{aligned}$$

To elaborate, we simplify the problem from deferring to multiple experts to deferring to just the one expert with the minimum $w_{i,\perp}$ in obtaining the lower bound $\bar{\mathcal{R}}_{\mathcal{D}}^{\perp}$. We also observe that we have no control over $\bar{\mathcal{R}}_{\mathcal{D}}^{\perp}$. However, we can control $\bar{\mathcal{R}}_{\mathcal{D}}^{\delta}$ by controlling the decision rule δ . We have $Z_{\perp}(\mathbf{x}) = 0$, and also it holds that

$$\bar{\mathcal{R}}_{\mathcal{D}}^{\delta} \geq \int_{\mathbf{x}} \min_j [Z_j(\mathbf{x})] d\mathbf{x},$$

where the equality holds iff $\delta(\hat{y}_k | \mathbf{x}) = 1.0$ for $k = \arg \min_j Z_j(\mathbf{x})$. Thus, the optimal rule is to deterministically (i.e. with confidence 1.0) choose $k \in \{1, 2, \dots, K, \perp\}$ with the minimum $Z_j(\mathbf{x})$. This means that choosing j for which the $Z_j(\mathbf{x})$ is the smallest minimizes the risk. Given that $Z_{\perp} = 0$, this means that the classifier predicts when the minimum $Z_j(\mathbf{x})$ is negative. Thus, deferral happens when $Z_j(\mathbf{x})$ is positive for all j , i.e., the optimal rejection rule $r^*(\mathbf{x})$ is:

$$r^*(\mathbf{x}) = \mathbb{I}[Z_j(\mathbf{x}) \geq 0; \forall j \in \{1, \dots, K\}]. \quad (\text{B.4})$$

This rejection rule is similar to the learning to defer to one expert. Given the definition of $Z_j(\mathbf{x})$, the optimal behavior to choose which expert to defer to is the one with minimum $w_{i,\perp,j}$. We further simplify the optimal rejection rule in the following proposition.

Proposition B.1.1. *The Bayes optimal rejection rule for L2D with multiple experts is given as:*

$$r^*(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbb{E}_{y|\mathbf{x}}[\ell_{\text{clf}}(\hat{y}, y)] \geq \min_{j \in J} \mathbb{E}_{y|\mathbf{x}}[\mathbb{E}_{\mathbf{m}|\mathbf{x}, y}[\ell_{\text{exp}}(\mathbf{m}_j, y)]] \quad \forall \hat{y} \in \mathcal{Y} \\ 0 & \text{otherwise.} \end{cases} \quad (\text{B.5})$$

Proof: The proof follows immediately from the definition of $r^*(\mathbf{x})$ in Equation B.4.

In our work, we use the canonical 0-1 loss for both ℓ_{clf} and ℓ_{exp} . In this case, the rejection rule can trivially be written as in the following corollary.

Corollary B.1.2. *For a misclassification 0-1 loss, the optimal rejection rule is:*

$$r^*(\mathbf{x}) = \mathbb{I}\left[\max_{j \in J} \mathbb{P}(\mathbf{m}_j = y | \mathbf{x} = \mathbf{x}) \geq \max_{y \in \mathcal{Y}} \mathbb{P}(y = y | \mathbf{x} = \mathbf{x})\right], \quad (\text{B.6})$$

where $\mathbb{P}(\mathbf{m}_j = y | \mathbf{x} = \mathbf{x})$ is the expert's correctness probability for the j^{th} expert, and $\mathbb{P}(y = y | \mathbf{x} = \mathbf{x})$ is the regular class probability.

To sum it up, the Bayes optimal rule is to compare the confidences of the experts and the classifier, and follow whosoever has the highest confidence. The rule is analogous to the single expert setting proved in [Mozannar and Sontag \(2020\)](#).

B.2 Proof of Theorem 3.1: Consistency of ψ_{SM}^J

Convexity of ψ_{SM}^J is immediately clear. We provide the proof for consistency below:

For simplicity, we denote

$$\begin{aligned} -\log \left(\frac{\exp\{g_y(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}^\perp} \exp\{g_{y'}(\mathbf{x})\}} \right) &= \zeta_y(\mathbf{x}), \\ -\log \left(\frac{\exp\{g_{\perp,j}(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}^\perp} \exp\{g_{y'}(\mathbf{x})\}} \right) &= \zeta_{\perp,j}(\mathbf{x}). \end{aligned} \quad (\text{B.7})$$

Then, ψ_{SM}^J can be written as:

$$\Phi_{SM}^J(g_1, \dots, g_K, g_{\perp,1}, \dots, g_{\perp,J}; \mathbf{x}, y, m_1, \dots, m_J) = \zeta_y(\mathbf{x}) + \sum_{j=1}^J \mathbb{I}[m_j = y] \cdot \zeta_{\perp,j}(\mathbf{x}). \quad (\text{B.8})$$

We consider the pointwise risk $\mathcal{C}[\psi_{SM}^J]$ defined as:

$$\mathcal{C}[\psi_{SM}^J] = \mathbb{E}_{y|\mathbf{x}=\mathbf{x}} \mathbb{E}_{m|\mathbf{x}=\mathbf{x}, y=y} [\psi_{SM}^J(g_1, \dots, g_K, g_{\perp,1}, \dots, g_{\perp,J}; \mathbf{x}, y, m_1, \dots, m_J)], \quad (\text{B.9})$$

where $m|\mathbf{x} = \mathbf{x}, y = y$ is a compact representation for each $m_j|\mathbf{x} = \mathbf{x}, y = y$. Our setup assumes that each m_j is independent. Denote $\eta_y(\mathbf{x}) = \mathbb{P}(y = y|\mathbf{x} = \mathbf{x})$, we expand the expectations:

$$\begin{aligned} \mathcal{C}[\psi_{SM}^J] &= \sum_{y \in \mathcal{Y}} \eta_y(\mathbf{x}) \cdot \zeta_y(\mathbf{x}) + \sum_{j=1}^J \left(\sum_{y \in \mathcal{Y}} \eta_y(\mathbf{x}) \sum_{m_j \in \mathcal{M}} \mathbb{P}(m_j = m_j|\mathbf{x} = \mathbf{x}, y = y) \mathbb{I}[m_j = y] \cdot \zeta_{\perp,j}(\mathbf{x}) \right). \\ \mathcal{C}[\psi_{SM}^J] &= \sum_{y \in \mathcal{Y}} \eta_y(\mathbf{x}) \cdot \zeta_y(\mathbf{x}) + \sum_{j=1}^J \left(\sum_{y \in \mathcal{Y}} \eta_y(\mathbf{x}) \sum_{m_j \in \mathcal{M}} \mathbb{P}(m_j = y|\mathbf{x} = \mathbf{x}, y = y) \cdot \zeta_{\perp,j}(\mathbf{x}) \right). \\ \mathcal{C}[\psi_{SM}^J] &= \sum_{y \in \mathcal{Y}} \eta_y(\mathbf{x}) \cdot \zeta_y(\mathbf{x}) + \sum_{j=1}^J \mathbb{P}(m_j = y|\mathbf{x} = \mathbf{x}) \cdot \zeta_{\perp,j}(\mathbf{x}). \end{aligned}$$

Next, we consider the minimizer of $\mathcal{C}[\psi_{SM}^J]$. Since we have established convexity, we can analyze the minimizers of $\mathcal{C}[\psi_{SM}^J]$ by taking the partial derivatives w.r.t. $g_y\{\mathbf{x}\}$ and $g_{\perp,j}\{\mathbf{x}\}$ respectively and set them to 0. Thus, w.r.t. $g_y\{\mathbf{x}\}$, we have

$$\frac{\partial \mathcal{C}[\psi_{SM}^J]}{\partial g_y\{\mathbf{x}\}} = 0 \implies \frac{\exp\{g_y(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}^\perp} \exp\{g_{y'}(\mathbf{x})\}} = \frac{\mathbb{P}(y = y|\mathbf{x} = \mathbf{x})}{1 + \sum_{j=1}^J \mathbb{P}(m_j = y|\mathbf{x} = \mathbf{x})}. \quad (\text{B.10})$$

Similarly, w.r.t. $g_{\perp,j}\{\mathbf{x}\}$ we have

$$\frac{\partial \mathcal{C}[\psi_{SM}^J]}{\partial g_{\perp,j}\{\mathbf{x}\}} = 0 \implies \frac{\exp\{g_{\perp,j}(\mathbf{x})\}}{\sum_{y' \in \mathcal{Y}^\perp} \exp\{g_{y'}(\mathbf{x})\}} = \frac{\mathbb{P}(m_j = y|\mathbf{x} = \mathbf{x})}{1 + \sum_{j=1}^J \mathbb{P}(m_j = y|\mathbf{x} = \mathbf{x})}. \quad (\text{B.11})$$

The above equations hold true for optimal classifier and the rejector. Thus, if we take the decision as in the main text, we are agreeing with the Bayes solution (considering that denominators are same in both the above conditions).

B.3 Proof of Theorem 3.2: Consistency of ψ_{OVA}^J

The proof follows directly from [Verma and Nalisnick \(2022\)](#). However, for completion we provide the full proof below.

For the surrogate prediction functions $g_1, \dots, g_K, g_{\perp,1}, \dots, g_{\perp,J}$, and the binary classification surrogate loss $\phi : \{-1, 1\} \times \mathbb{R} \rightarrow \mathbb{R}_+$, ψ_{OVA}^J takes the following pointwise-form:

$$\begin{aligned} \mathcal{C}[\psi_{\text{OVA}}^J] &= \psi_{\text{OVA}}^J(g_1, \dots, g_K, g_{\perp,1}, \dots, g_{\perp,J}; \mathbf{x}, y, m_1, \dots, m_J) \\ &= \phi[g_y(\mathbf{x})] + \sum_{y' \in \mathcal{Y}, y' \neq y} \phi[-g_{y'}(\mathbf{x})] \\ &\quad + \sum_{j=1}^J \phi[-g_{\perp,j}(\mathbf{x})] + \sum_{j=1}^J \mathbb{I}[m_j = y] (\phi[g_{\perp,j}(\mathbf{x})] - \phi[-g_{\perp,j}(\mathbf{x})]). \end{aligned}$$

We consider the pointwise *inner* ψ_{OVA} risk for some $\mathbf{x} = \mathbf{x}$ written as follows:

$$\mathcal{C}[\psi_{\text{OVA}}^J] = \mathbb{E}_{y|\mathbf{x}=\mathbf{x}} \mathbb{E}_{m|\mathbf{x}=\mathbf{x}, y=y} [\psi_{\text{OVA}}^J(g_1, \dots, g_K, g_{\perp,1}, \dots, g_{\perp,J}; \mathbf{x}, y, m_1, \dots, m_J)],$$

We expand both the expectations one-by-one below:

$$\begin{aligned} \mathcal{C}[\psi_{\text{OVA}}^J] &= \mathbb{E}_{y|\mathbf{x}=\mathbf{x}} \left[\phi[g_y(\mathbf{x})] + \sum_{y' \in \mathcal{Y}, y' \neq y} \phi[-g_{y'}(\mathbf{x})] + \sum_{j=1}^J \phi[-g_{\perp,j}(\mathbf{x})] \right. \\ &\quad \left. + \sum_{j=1}^J \left(\sum_{m_j \in \mathcal{M}} \mathbb{P}(m_j = m_j | \mathbf{x} = \mathbf{x}, y = y) \mathbb{I}[m_j = y] (\phi[g_{\perp,j}(\mathbf{x})] - \phi[-g_{\perp,j}(\mathbf{x})]) \right) \right]. \end{aligned}$$

Denote $\mathbb{P}(y = y | \mathbf{x} = \mathbf{x})$ as $\eta_y(\mathbf{x})$, then

$$\begin{aligned} \mathcal{C}[\psi_{\text{OVA}}^J] &= \mathbb{E}_{y|\mathbf{x}=\mathbf{x}} \sum_{y \in \mathcal{Y}} \left[\phi[g_y(\mathbf{x})] + \sum_{y' \in \mathcal{Y}, y' \neq y} \phi[-g_{y'}(\mathbf{x})] \right] \\ &\quad + \sum_{j=1}^J \left(\phi[-g_{\perp,j}(\mathbf{x})] \right. \\ &\quad \left. + \sum_{y \in \mathcal{Y}} \eta_y(\mathbf{x}) \left[\sum_{m_j \in \mathcal{M}} \mathbb{P}(m_j = m_j | \mathbf{x} = \mathbf{x}, y = y) \mathbb{I}[m_j = y] (\phi[g_{\perp,j}(\mathbf{x})] - \phi[-g_{\perp,j}(\mathbf{x})]) \right] \right) \\ \mathcal{C}[\psi_{\text{OVA}}^J] &= \sum_{y \in \mathcal{Y}} \eta_y(\mathbf{x}) \left[\phi[g_y(\mathbf{x})] + \sum_{y' \in \mathcal{Y}, y' \neq y} \phi[-g_{y'}(\mathbf{x})] \right] \\ &\quad + \sum_{j=1}^J \left(\phi[-g_{\perp,j}(\mathbf{x})] + \underbrace{\sum_{y \in \mathcal{Y}} \eta_y(\mathbf{x}) \sum_{m_j \in \mathcal{M}} \mathbb{P}(m_j = y | \mathbf{x} = \mathbf{x}, y = y) (\phi[g_{\perp,j}(\mathbf{x})] - \phi[-g_{\perp,j}(\mathbf{x})])}_{\mathbb{P}(m_j = y | \mathbf{x} = \mathbf{x})} \right) \\ \mathcal{C}[\psi_{\text{OVA}}^J] &= \sum_{y \in \mathcal{Y}} \eta_y(\mathbf{x}) \left[\phi[g_y(\mathbf{x})] + \sum_{y' \in \mathcal{Y}, y' \neq y} \phi[-g_{y'}(\mathbf{x})] \right] \\ &\quad + \sum_{j=1}^J \left[\phi[-g_{\perp,j}(\mathbf{x})] + \mathbb{P}(m_j = y | \mathbf{x} = \mathbf{x}) (\phi[g_{\perp,j}(\mathbf{x})] - \phi[-g_{\perp,j}(\mathbf{x})]) \right]. \end{aligned}$$

Denote $\mathbb{P}(m_j = y | \mathbf{x} = \mathbf{x})$ as p_{m_j} , then we have

$$\begin{aligned} \mathcal{C}[\psi_{\text{OVA}}^J] &= \sum_{y \in \mathcal{Y}} \eta_y(\mathbf{x}) \left[\phi[g_y(\mathbf{x})] + \sum_{y' \in \mathcal{Y}, y' \neq y} \phi[-g_{y'}(\mathbf{x})] \right] \\ &\quad + \sum_{j=1}^J [(1 - p_{m_j}) \phi[-g_{\perp,j}(\mathbf{x})] + p_{m_j} \phi[g_{\perp,j}(\mathbf{x})]] \end{aligned}$$

We further simplify the above equation as follows:

$$\begin{aligned} \mathcal{C}[\psi_{\text{OVA}}^J] &= \sum_{y \in \mathcal{Y}} [\eta_y(\mathbf{x}) \cdot \phi[g_y(\mathbf{x})] + (1 - \eta_y(\mathbf{x})) \cdot \phi[-g_y(\mathbf{x})]] \\ &\quad + \sum_{j=1}^J [(1 - p_{m_j}) \phi[-g_{\perp,j}(\mathbf{x})] + p_{m_j} \phi[g_{\perp,j}(\mathbf{x})]]. \end{aligned}$$

Thus, we conclude from the above expression that we $K+J$ binary classification problems where the pointwise risk (or inner risk) for the i^{th} binary classification problem is given as $\eta_y(\mathbf{x}) \phi(g_y(\mathbf{x})) + (1 - \eta_y(\mathbf{x})) \phi(-g_y(\mathbf{x}))$ for $i \in [K]$ and $p_{m_j}(\mathbf{x}) \phi(g_{\perp,j}(\mathbf{x})) + (1 - p_{m_j}(\mathbf{x})) \phi(-g_{\perp,j}(\mathbf{x}))$ when $i \in [J]$. Thus, minimizer of the inner ψ_{OVA} -risk can be analyzed in terms of the pointwise minimizer of the *inner ϕ -risk* for each of the $K+J$ sub binary classification problems. Denote the minimizer of pointwise *inner ψ_{OVA} -risk* as \mathbf{g}^* , then the above decomposition means g_i^* corresponds to the minimizer of the *inner ϕ -risk* for the i th binary classification problem.

We know that the Bayes solution for the binary classification problem is $\text{sign}(\eta(\mathbf{x}) - \frac{1}{2})$ where $\eta(\mathbf{x})$ denotes $p(y = 1 | \mathbf{x} = \mathbf{x})$. Now when the binary surrogate loss ϕ is a strictly proper composite loss for binary classification, by the property of strictly proper composite losses, we have $\text{sign}(g_y^*(\mathbf{x}))$ would agree with the Bayes solution of the Binary classification, i.e. $g_y^*(\mathbf{x}) > 0$ if $\eta_y(\mathbf{x}) > \frac{1}{2}$. And similarly $g_{\perp}^*(\mathbf{x}) > 0$ if $p_{m_j}(\mathbf{x}) > \frac{1}{2}$. Furthermore, we have the existence of a continuous and increasing inverse link function γ^{-1} for the binary surrogate ϕ with the property that $\gamma^{-1}(g_y^*(\mathbf{x}))$ would converge to $\eta_y(\mathbf{x})$. Similarly, $\gamma^{-1}(g_{\perp,j}^*(\mathbf{x}))$ would converge to $p_{m_j}(\mathbf{x})$. Thus, when the binary surrogate loss ϕ is a strictly proper composite loss, and the classifier and the rejector are defined as in the main text, the minimizer of the pointwise risk $\mathcal{C}[\psi_{\text{OVA}}^J]$ agree with the Bayes optimal solution. Thus, ψ_{OVA}^J is a calibrated loss function for L2D w.r.t. 0-1 misclassification loss.

B.4 Inconsistency of the Mixture of Experts Formulation (Hemmer et al., 2022)

Proposition B.4.1. L_{MoE} is inconsistent for learning to defer.

The proof works by construction. Specifically, we construct a distribution over $\mathcal{X} \times \mathcal{Y}$ for which the necessary condition for consistency does not hold true.

Consider we have $\mathcal{X} = \{\mathbf{x}\}$ and $\mathcal{Y} = \{0, 1\}$, i.e. the input space contains the singleton element \mathbf{x} with 2 output labels. We define the following distribution \mathcal{D} such that $\mathbb{P}(\mathbf{x}, 0) = \alpha_0$, $\mathbb{P}(\mathbf{x}, 1) = \alpha_1$. For completion, $\alpha_0 + \alpha_1 = 1$. For simplicity, we consider one expert who predicts correctly with perfect confidence, i.e. $m = y \forall y$. The mixture of experts method works by estimating the allocator scores $w_e(\mathbf{x})$ (for expert) and $w_c(\mathbf{x})$ (for classifier) such that $w_e(\mathbf{x}) + w_c(\mathbf{x}) = 1$, and the classifier scores $c_0(\mathbf{x}), c_1(\mathbf{x})$ with $\sum_{i=0}^1 c_i(\mathbf{x}) = 1$. In such a setting, we have

$$\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [L_{\text{MoE}}(F, A, \mathbf{x}, y, m)] = -\alpha_0 [\log(w_e + w_c \cdot c_0)] - \alpha_1 [\log(w_e + w_c \cdot c_1)].$$

It is an easy argument to see that the minimum value of the above expression is 0, *i.e.*,

$$\inf_{A, \bar{F}} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [L_{\text{MoE}}(F, A, \mathbf{x}, y, \mathbf{m})] = 0.$$

MoE system decides to defer to the expert if $w_e(\mathbf{x}) > w_c(\mathbf{x})$. For $\delta > 0$, choose $w_e(\mathbf{x}) = 0.5 - \delta$ and $w_c(\mathbf{x}) = 0.5 + \delta$. Note that $\forall \delta > 0$, the system would always decide not to defer to the expert. Also choose $c_0(\mathbf{x}) = 1, c_1(\mathbf{x}) = 0$. For such an allocator \bar{A} and the classifier \bar{F} ,

$$\begin{aligned} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [L_{\text{MoE}}(\bar{F}, \bar{A}, \mathbf{x}, y, \mathbf{m})] &= -\alpha_1 \cdot \log(0.5 - \delta) \\ &\leq -\alpha_1 \cdot (0.5 - \delta - 1) = \alpha_1 \cdot (0.5 + \delta), \end{aligned}$$

where the inequality comes from using $\log(x) \leq x - 1$. Next, choose α_1 such that $\alpha_1 = \frac{\delta}{0.5 + \delta}$ (why this is true is left as an exercise to the reader). Combining everything, we have shown that

$$|\mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [L_{\text{MoE}}(\bar{F}, \bar{A}, \mathbf{x}, y, \mathbf{m})] - \inf_{A, F} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}} [L_{\text{MoE}}(F, A, \mathbf{x}, y, \mathbf{m})]| \leq \delta.$$

Thus, our choice of \bar{A} and \bar{F} satisfy Equation B.1 for all $\delta > 0$. Since in our construction, we always allow the decision to be made by the classifier which can only predict class label $h(\mathbf{x}) \in \{0, 1\}$, we have $\mathbb{P}_{(\mathbf{x}, y) \sim \mathcal{D}} [h(\mathbf{x}) \neq y] = \alpha_1$. And the Bayes optimal rule $h^*(\mathbf{x})$ is to always let the expert make the prediction, thus, $\mathbb{P}_{(\mathbf{x}, y) \sim \mathcal{D}} [h^*(\mathbf{x}) \neq y] = 0$. Hence, we have

$$\mathbb{P}_{(\mathbf{x}, y) \sim \mathcal{D}} [h(\mathbf{x}) \neq y] = \mathbb{P}_{(\mathbf{x}, y) \sim \mathcal{D}} [h^*(\mathbf{x}) \neq y] + \eta,$$

where $\eta = \alpha_1$. Thus, $\forall \epsilon < \kappa, \epsilon > 0$, Equation B.2 fails to hold true. Hence, we have shown that the optimization of L_{MoE} allows faulty solutions that may not reach the Bayes optimal predictor.

Bibliography

- S. Abiteboul, M. Arenas, P. Barceló, M. Bienvenu, D. Calvanese, C. David, R. Hull, E. Hüllermeier, B. Kimelfeld, L. Libkin, W. Martens, T. Milo, F. Murlak, F. Neven, M. Ortiz, T. Schwentick, J. Stoyanovich, J. Su, D. Suciu, V. Vianu, and K. Yi. Research directions for principles of data management (dagstuhl perspectives workshop 16151), 2017. Cited on pages: 5.
- L. Aitchison, A. Yang, and S. W. Ober. Deep kernel processes. In *International Conference on Machine Learning*, pages 130–140. PMLR, 2021. Cited on pages: 98.
- Z. Akata, D. Balliet, M. De Rijke, F. Dignum, V. Dignum, G. Eiben, A. Fokkens, D. Grossi, K. Hindriks, H. Hoos, et al. A research agenda for hybrid intelligence: augmenting human intellect with collaborative, adaptive, responsible, and explainable artificial intelligence. *Computer*, 53(08):18–28, 2020. Cited on pages: 7, 54.
- A. Alemi, B. Poole, I. Fischer, J. Dillon, R. A. Saurous, and K. Murphy. Fixing a broken elbow. In *International conference on machine learning*, pages 159–168. PMLR, 2018. Available in: [\[5\]](#). Cited on pages: 18.
- P. D. Allison. *Missing data*. Sage publications, 2001. Cited on pages: 21.
- E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113141, 2001. Cited on pages: 59, 81.
- A. Angelopoulos, S. Bates, J. Malik, and M. I. Jordan. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations (ICLR)*, 2020. Cited on pages: 84, 85.
- A. N. Angelopoulos and S. Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification, 2021. Available in: [\[5\]](#). Cited on pages: 84.
- A. N. Angelopoulos, S. Bates, A. Fisch, L. Lei, and T. Schuster. Conformal risk control. *arXiv preprint arXiv:2208.02814*, 2022. Cited on pages: 85, 86, 94.
- N. Arun, N. Gaw, P. Singh, K. Chang, M. Aggarwal, B. Chen, K. Hoebel, S. Gupta, J. Patel, M. Gidwani, et al. Assessing the trustworthiness of saliency maps for localizing abnormalities in medical imaging. *Radiology: Artificial Intelligence*, 3(6):e200267, 2021. Cited on pages: 101.
- M. Ashman, J. So, W. Tebbutt, V. Fortuin, M. Pearce, and R. E. Turner. Sparse gaussian process variational autoencoders. *arXiv preprint arXiv:2010.10177*, 2020. Cited on pages: 43.
- V. Babbar, U. Bhatt, and A. Weller. On the utility of prediction sets in human-ai teams. In *International Joint Conference on Artificial Intelligence*, 2022. Available in: [\[5\]](#). Cited on pages: 88, 100.
- S. P. Bamford, R. C. Nichol, I. K. Baldry, K. Land, C. J. Lintott, K. Schawinski, A. Slosar, A. S. Szalay, D. Thomas, M. Torki, D. Andreescu, E. M. Edmondson, C. J. Miller, P. Murray, M. J. Raddick, and J. Vandenberg. Galaxy Zoo: the dependence of morphology and colour on environment*. *Monthly Notices of the Royal Astronomical Society*, 393(4):1324–1352, 2009. Cited on pages: 88.
- G. Bansal, B. Nushi, E. Kamar, E. Horvitz, and D. S. Weld. Is the most accurate ai the best teammate? optimizing ai for teamwork. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11405–11414, 2021. Cited on pages: 63.
- R. F. Barber, E. J. Candes, A. Ramdas, and R. J. Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816 – 845, 2023. Available in: [\[5\]](#). Cited on pages: 100.

- D. Barrejón, P. M. Olmos, and A. Artés-Rodríguez. Medical data wrangling with sequential variational autoencoders. *IEEE Journal of Biomedical and Health Informatics*, 26(6):2737–2745, 2021. Available in: [\[4\]](#). Cited on pages: 36, 43.
- P. L. Bartlett and M. H. Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(59):1823–1840, 2008. Cited on pages: 62.
- P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006. Cited on pages: 56, 58.
- F. Bashir and H.-L. Wei. Handling missing data in multivariate time series using a vector autoregressive model-imputation (var-im) algorithm. *Neurocomputing*, 276:23–30, 2018. Cited on pages: 22.
- E. Beede, E. Baylor, F. Hersch, A. Iurchenko, L. Wilcox, P. Ruamviboonsuk, and L. M. Vardoulakis. A human-centered evaluation of a deep learning system deployed in clinics for the detection of diabetic retinopathy. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–12, 2020. Cited on pages: 6.
- S. Ben-David, N. Eiron, and P. M. Long. On the difficulty of approximately maximizing agreements. *Journal of Computer and System Sciences*, 66(3):496–514, 2003. Cited on pages: 56.
- Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. Cited on pages: 29.
- N. C. Benz and M. G. Rodriguez. Counterfactual inference of second opinions. In *Conference on Uncertainty in Artificial Intelligence*, 2022. Cited on pages: 87.
- N. L. C. Benz and M. G. Rodriguez. Human-aligned calibration for ai-assisted decision making. *arXiv preprint arXiv:2306.00074*, 2023. Cited on pages: 71.
- J. Berrevoets, F. Imrie, T. Kyono, J. Jordon, and M. van der Schaar. To impute or not to impute? missing data in treatment effect estimation. In *International Conference on Artificial Intelligence and Statistics*, pages 3568–3590. PMLR, 2023. Available in: [\[4\]](#). Cited on pages: 21.
- U. Bhatt, J. Antorán, Y. Zhang, Q. V. Liao, P. Sattigeri, R. Fogliato, G. Melançon, R. Krishnan, J. Stanley, O. Tickoo, et al. Uncertainty as a form of transparency: Measuring, communicating, and using uncertainty. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 401–413, 2021. Cited on pages: 71.
- N. Bidargaddi, P. Musiat, V.-P. Makinen, M. Ermes, G. Schrader, and J. Licinio. Digital footprints: facilitating large-scale environmental psychiatric research in naturalistic settings through data from everyday technologies. *Molecular psychiatry*, 22(2):164–169, 2017. Cited on pages: 3.
- A. Birhane, W. Isaac, V. Prabhakaran, M. Diaz, M. C. Elish, I. Gabriel, and S. Mohamed. Power to the people? opportunities and challenges for participatory ai. *Equity and Access in Algorithms, Mechanisms, and Optimization*, pages 1–8, 2022. Cited on pages: 53.
- C. M. Bishop. *Pattern recognition and machine learning*, volume 4. Springer, 2006. Cited on pages: 15.
- D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017. Cited on pages: 14.
- N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *International conference on machine learning (ICML)*, 2012. Cited on pages: 32.
- S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015. Cited on pages: 20.
- Z. Buçinca, M. B. Malaya, and K. Z. Gajos. To trust or to think: cognitive forcing functions can reduce overreliance on ai in ai-assisted decision-making. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–21, 2021. Cited on pages: 101.
- A. Buja, W. Stuetzle, and Y. Shen. Loss functions for binary class probability estimation and classification: Structure and applications. *Technical Report*, 2005. Cited on pages: 58, 60.

- Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015. Cited on pages: 22.
- L. Canzian and M. Musolesi. Trajectories of depression: unobtrusive monitoring of depressive states by means of smartphone mobility traces analysis. In *Proceedings of the 2015 ACM International Joint conference on Pervasive and Ubiquitous Computing*, pages 1293–1304, 2015. Cited on pages: 38.
- W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li. Brits: Bidirectional recurrent imputation for time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. Available in: [\[1\]](#). Cited on pages: 31, 37, 99.
- M. Cauchois, S. Gupta, and J. C. Duchi. Knowing what you know: valid and validated confidence sets in multiclass and multilabel prediction. *J. Mach. Learn. Res.*, 22:81–1, 2021. Cited on pages: 85.
- N. Charoenphakdee, Z. Cui, Y. Zhang, and M. Sugiyama. Classification with rejection based on cost-sensitive classification. In *International Conference on Machine Learning*, 2021. Cited on pages: 63.
- M.-A. Charusaie, H. Mozannar, D. Sontag, and S. Samadi. Sample efficient learning of predictors that complement humans. In *International Conference on Machine Learning*, 2022. Cited on pages: 87, 94.
- Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018. Available in: [\[2\]](#). Cited on pages: 31, 37, 99.
- X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. Variational lossy autoencoder. In *International Conference on Learning Representations*, 2017. Available in: [\[3\]](#). Cited on pages: 19.
- K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. Cited on pages: 31, 41.
- C. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970. doi: 10.1109/TIT.1970.1054406. Cited on pages: 62.
- C. K. Chow. An optimum character recognition system using decision functions. *IRE Trans. Electron. Comput.*, 6:247–254, 1957. Cited on pages: 7, 9, 62, 112.
- X. Chu, I. F. Ilyas, S. Krishnan, and J. Wang. Data cleaning: Overview and emerging challenges. In *Proceedings of the 2016 international conference on management of data*, pages 2201–2206, 2016. Cited on pages: 5.
- J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio. A recurrent latent variable model for sequential data. *Advances in Neural Information Processing systems (NeurIPS)*, 2015. Cited on pages: 32, 33, 37.
- N. C. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pages 168–172. IEEE, 2018. Cited on pages: 53.
- M. Collier, A. Nazabal, and C. K. Williams. VAEs in the presence of missing data. In *Workshop Art of Learning with Missing Values (Artemiss)*, 2020. Cited on pages: 21, 37.
- K. M. Collins, U. Bhatt, and A. Weller. Eliciting and learning with soft labels from every annotator. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 10, pages 40–52, 2022. Available in: [\[4\]](#). Cited on pages: 101.
- C. Cortes, G. DeSalvo, and M. Mohri. Learning with rejection. In *International Conference on Algorithmic Learning Theory*, 2016a. Cited on pages: 62, 63, 64, 66, 79.

- C. Cortes, G. DeSalvo, and M. Mohri. Boosting with abstention. In *Advances in Neural Information Processing Systems*, 2016b. Cited on pages: 62, 63, 66, 79.
- H. Cramer, V. Evers, S. Ramlal, M. Van Someren, L. Rutledge, N. Stash, L. Aroyo, and B. Wielinga. The effects of transparency on trust in and acceptance of a content-based art recommender. *User Modeling and User-adapted interaction*, 18:455–496, 2008. Cited on pages: 71.
- C. Cremer, X. Li, and D. Duvenaud. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pages 1078–1086. PMLR, 2018. Available in: [\[5\]](#). Cited on pages: 18.
- A. Damianou, N. D. Lawrence, and C. H. Ek. Multi-view learning as a nonparametric nonlinear inter-battery factor analysis. *Journal of Machine Learning Research*, 22(86):1–51, 2021. Available in: [\[5\]](#). Cited on pages: 24.
- A. C. Damianou, C. H. Ek, M. K. Titsias, and N. D. Lawrence. Manifold relevance determination. In *Proceedings of the 29th International Conference on Machine Learning*, 2012. Available in: [\[5\]](#). Cited on pages: 24.
- T. Davidson, D. Warmesley, M. W. Macy, and I. Weber. Automated hate speech detection and the problem of offensive language. In *International AAAI Conference on Web and Social Media*, 2017. Cited on pages: 88.
- A. P. Dawid. The well-calibrated Bayesian. *Journal of the American Statistical Association*, 77(379):605–610, 1982. Cited on pages: 72, 82.
- A. De, P. Koley, N. Ganguly, and M. Gomez-Rodriguez. Regression under human assistance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2611–2620, 2020. Cited on pages: 63.
- A. De, N. Okati, A. Zarezade, and M. Gomez-Rodriguez. Classification under human assistance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5905–5913, 2021. Cited on pages: 63.
- M. De Choudhury, S. Counts, and E. Horvitz. Predicting postpartum changes in emotion and behavior via social media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3267–3276, 2013. Cited on pages: 38.
- D. Dellermann, P. Ebel, M. Söllner, and J. M. Leimeister. Hybrid intelligence. *Business & Information Systems Engineering*, 61:637–643, 2019. Cited on pages: 4, 7, 54.
- L. Devroye. Random variate generation in one line of code. In *Proceedings of the 28th conference on Winter simulation*, pages 265–272, 1996. Available in: [\[5\]](#). Cited on pages: 20.
- N. Dhir, D. Zilli, T. Rudny, and A. Tosi. Automatic type inference with a nested latent variable model. In *ICML Workshop on AutoML*, 2018. Cited on pages: 25.
- N. Dhir, T. Rudny, D. Zilli, and A. Tosi. An automatic type-inferential general latent feature model. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. Cited on pages: 25.
- T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2(1):263286, 1995. Cited on pages: 59, 81.
- N. Dilokthanakul, P. A. Mediano, M. Garnelo, M. C. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016. Cited on pages: 40.
- K. Donahue, A. Chouldechova, and K. Kenthapadi. Human-algorithm collaboration: Achieving complementarity and avoiding unfairness. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1639–1656, 2022. Available in: [\[5\]](#). Cited on pages: 64.
- B.-S. Einbinder, S. Bates, A. N. Angelopoulos, A. Gendler, and Y. Romano. Conformal prediction is robust to label noise. *arXiv preprint arXiv:2209.14295*, 2022. Cited on pages: 85.
- J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. Cited on pages: 29.

- O. Fabius and J. R. Van Amersfoort. Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581*, 2014. Cited on pages: 32.
- A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatin, A. Novikov, F. J. R Ruiz, J. Schrittwieser, G. Swirszcz, et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022. Cited on pages: 3.
- D. Fay, C. Borrill, Z. Amir, R. Haward, and M. A. West. Getting the most out of multidisciplinary teams: A multi-sample study of team innovation in health care. *Journal of Occupational and Organizational Psychology*, 79(4):553–567, 2006. Cited on pages: 77.
- V. Fortuin, D. Baranchuk, G. Rätsch, and S. Mandt. GP-VAE: Deep probabilistic time series imputation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1651–1661. PMLR, 2020. Available in: [\[5\]](#). Cited on pages: 37, 43, 96.
- M. Fraccaro, S. K. Sønderby, U. Paquet, and O. Winther. Sequential neural models with stochastic layers. *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. Available in: [\[5\]](#). Cited on pages: 32.
- J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001. Cited on pages: 44.
- T. Furche, G. Gottlob, L. Libkin, G. Orsi, and N. W. Paton. Data wrangling for big data: Challenges and opportunities. In *EDBT*, volume 16, pages 473–478, 2016. Cited on pages: 5.
- A. Gandomi and M. Haider. Beyond the hype: Big data concepts, methods, and analytics. *International journal of information management*, 35(2):137–144, 2015. Cited on pages: 2.
- D. J. Gantzert. Determining responsibility when learning to defer to an expert. Master’s thesis, University of Amsterdam, 2021. Cited on pages: 68.
- I. Gatopoulos and J. M. Tomczak. Self-supervised variational auto-encoders, 2021. Available in: [\[5\]](#). Cited on pages: 19.
- S. Gaube, H. Suresh, M. Raue, A. Merritt, S. J. Berkowitz, E. Lermer, J. F. Coughlin, J. V. Gutttag, E. Colak, and M. Ghassemi. Do as ai say: susceptibility in deployment of clinical decision-aids. *NPJ digital medicine*, 4(1):31, 2021. Cited on pages: 101.
- S. Gershman and N. Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the annual meeting of the cognitive science society*, volume 36, 2014. Available in: [\[5\]](#). Cited on pages: 17.
- Z. Ghahramani and M. I. Jordan. Supervised learning from incomplete data via an EM approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1994. Cited on pages: 22.
- Y. Gong, H. Hajimirsadeghi, J. He, T. Durand, and G. Mori. Variational selective autoencoder: Learning from partially-observed heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*, pages 2377–2385. PMLR, 2021. Cited on pages: 25.
- E. Gordon-Rodriguez, G. Loaiza-Ganem, and J. Cunningham. The continuous categorical: a novel simplex-valued exponential family. In *International Conference on Machine Learning (ICML)*. PMLR, 2020. Cited on pages: 97.
- Y. Grandvalet, A. Rakotomamonjy, J. Keshet, and S. Canu. Support vector machines with a reject option. In *Advances in Neural Information Processing Systems*, 2009. Cited on pages: 62.
- S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020. Cited on pages: 54.
- A. Guerrero-López, C. Sevilla-Salcedo, V. Gómez-Verdejo, and P. M. Olmos. Multi-view hierarchical variational autoencoders with factor analysis latent space. *arXiv preprint arXiv:2207.09185*, 2022. Cited on pages: 24.
- C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International conference on machine learning (ICML)*. PMLR, 2017. Cited on pages: 71, 87, 100.

- C. Gupta and A. Ramdas. Top-label calibration and multiclass-to-binary reductions. In *International Conference on Learning Representations (ICLR)*, 2022. Cited on pages: 60, 71, 87, 100.
- C. Heaukulani and M. van der Wilk. Scalable bayesian dynamic covariance modeling with variational wishart and inverse wishart processes. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32 (NeurIPS)*, volume 32. Curran Associates, Inc., 2019. Available in: [\[5\]](#). Cited on pages: 98.
- P. Hemmer, S. Schellhammer, M. Vössing, J. Jakubik, and G. Satzger. Forming Effective Human-AI Teams: Building Machine Learning Models that Complement the Capabilities of Multiple Experts. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2022. Cited on pages: xviii, 78, 82, 87, 90, 91, 111, 116.
- K. Hendrickx, L. Perini, D. Van der Plas, W. Meert, and J. Davis. Machine learning with a reject option: A survey. *arXiv preprint arXiv:2107.11277*, 2021. Available in: [\[5\]](#). Cited on pages: 62.
- D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. Available in: [\[5\]](#). Cited on pages: 54.
- S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. Cited on pages: 30, 31, 36, 41.
- S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001. Cited on pages: 29.
- M. D. Hoffman and M. J. Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1, 2016. Available in: [\[5\]](#). Cited on pages: 18.
- J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982. Cited on pages: 29.
- J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghighi, R. Ball, K. Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019. Cited on pages: 54.
- M. Jacobs, M. F. Pradier, T. H. McCoy Jr, R. H. Perlis, F. Doshi-Velez, and K. Z. Gajos. How machine-learning recommendations influence clinician treatment selections: the example of antidepressant selection. *Translational psychiatry*, 11(1):108, 2021. Cited on pages: 7, 54.
- S. H. Jain, B. W. Powers, J. B. Hawkins, and J. S. Brownstein. The digital phenotype. *Nature biotechnology*, 33(5):462–463, 2015. Cited on pages: 3.
- E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations (ICLR)*, 2017. Cited on pages: 20, 43.
- A. Javaloy and I. Valera. Lipschitz standardization for multivariate learning. *arXiv preprint arXiv:2002.11369*, 2020. Cited on pages: 98.
- A. Javaloy, M. Meghdadi, and I. Valera. Boosting heterogeneous vaes via multi-objective optimization. In *Workshop Your Model Is Wrong: Robustness and Misspecification in Probabilistic Modeling in Neural Information Processing Systems (NeurIPS)*, 2021. Available in: [\[5\]](#). Cited on pages: 26.
- A. Javaloy, M. Meghdadi, and I. Valera. Mitigating modality collapse in multimodal vaes via impartial optimization. In *International Conference on Machine Learning*, pages 9938–9964. PMLR, 2022. Cited on pages: 24, 26.
- M. Jazbec, M. Ashman, V. Fortuin, M. Pearce, S. Mandt, and G. Rätsch. Scalable gaussian process variational autoencoders. In *International Conference on Artificial Intelligence and Statistics*, pages 3511–3519. PMLR, 2021a. Available in: [\[5\]](#). Cited on pages: 43.
- M. Jazbec, M. A. L. Pearce, and V. Fortuin. Factorized gaussian process variational autoencoders. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2021b. Available in: [\[5\]](#). Cited on pages: 43.

- H. Jiang, B. Kim, M. Y. Guan, and M. Gupta. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems*, 2018. Cited on pages: 62.
- M. I. Jordan. Serial order: A parallel distributed processing approach. In *Technical Report 8604, Institute for Cognitive Science, University of California, San Diego*, 1986. Available in: [\[5\]](#). Cited on pages: 29.
- M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233, 1999. Cited on pages: 16.
- S. Joshi, S. Parbhoo, and F. Doshi-Velez. Learning-to-defer for sequential medical decision-making under uncertainty. *Transactions on Machine Learning Research*, 2023. Available in: [\[5\]](#). Cited on pages: 100.
- A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016. Cited on pages: 88, 90.
- J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Židek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. Cited on pages: 3.
- E. Kamar. Directions in hybrid intelligence: Complementing ai systems with human intelligence. In *International Joint Conference on Artificial Intelligence*, 2016a. Cited on pages: 7, 54.
- E. Kamar. Hybrid workplaces of the future. *XRDS: Crossroads, The ACM Magazine for Students*, 23(2): 22–25, 2016b. Cited on pages: 4.
- S. Kandel, J. Heer, C. Plaisant, J. Kennedy, F. Van Ham, N. H. Riche, C. Weaver, B. Lee, D. Brodbeck, and P. Buono. Research directions in data wrangling: Visualizations and transformations for usable and credible data. *Information Visualization*, 10(4):271–288, 2011. Cited on pages: 5.
- P. Keerin, W. Kurutach, and T. Boongoen. Cluster-based knn missing value imputation for dna microarray data. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 445–450. IEEE, 2012. Available in: [\[5\]](#). Cited on pages: 21.
- A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018. Cited on pages: 24.
- G. Kerrigan, P. Smyth, and M. Steyvers. Combining human predictions with model probabilities via confusion matrices and calibration. *Advances in Neural Information Processing Systems*, 34:4421–4434, 2021. Cited on pages: 63, 87.
- V. Keswani, M. Lease, and K. Kenthapadi. Towards unbiased and accurate deferral to multiple experts. In *AAAI/ACM Conference on AI, Ethics, and Society*, 2021. Cited on pages: 87.
- Y. Kim. Convolutional neural networks for sentence classification. In *Conference on Empirical Methods in Natural Language Processing*, 2014. Cited on pages: 88, 90.
- Y. Kim, S. Wiseman, A. Miller, D. Sontag, and A. Rush. Semi-amortized variational autoencoders. In *International Conference on Machine Learning*, pages 2678–2687. PMLR, 2018. Available in: [\[5\]](#). Cited on pages: 17.
- D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014. Available in: [\[5\]](#). Cited on pages: 18, 19, 20, 42.
- D. P. Kingma, M. Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. Cited on pages: 14.
- H. Kittler, H. Pehamberger, K. Wolff, and M. Binder. Diagnostic accuracy of dermoscopy. *The lancet oncology*, 3(3):159–165, 2002. Cited on pages: 90.
- A. Krizhevsky. Learning multiple layers of features from tiny images. *Technical Report*, 2009. Cited on pages: 90.
- M. Kull, T. Silva Filho, and P. Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *Artificial Intelligence and Statistics (AISTATS)*, pages 623–631. PMLR, 2017. Available in: [\[5\]](#). Cited on pages: 71.

- M. Kull, M. Perello-Nieto, M. Kängsepp, T. S. Filho, H. Song, and P. Flach. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with dirichlet calibration. In *Advances in Neural Information Processing Systems*, 2019. Cited on pages: 71, 82, 87.
- R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, A. Pritzel, S. Ravuri, T. Ewalds, F. Alet, Z. Eaton-Rosen, et al. Graphcast: Learning skillful medium-range global weather forecasting. *arXiv preprint arXiv:2212.12794*, 2022. Cited on pages: 53.
- J. Langford, Tti-Chicago, J. Net, and A. Beygelzimer. Sensitive error correcting output codes. In *Conference on Learning Theory*, 2005. Cited on pages: 59, 81.
- N. D. Lawrence. Data readiness levels, 2017. Available in: [\[1\]](#). Cited on pages: 4, 5.
- Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. Cited on pages: 98.
- J. D. Lee and K. A. See. Trust in automation: Designing for appropriate reliance. *Human factors*, 46(1): 50–80, 2004. Cited on pages: 8.
- J. M. Leimeister. Collective intelligence. *Business & Information Systems Engineering*, 2(4):245–248, 2010. doi: 10.1007/s12599-010-0114-8. Cited on pages: 6.
- S. C.-X. Li, B. Jiang, and B. Marlin. Learning from incomplete data with generative adversarial networks. In *International Conference on Learning Representations*, 2019a. Available in: [\[2\]](#). Cited on pages: 37.
- X. Li, Z. Chen, L. K. M. Poon, and N. L. Zhang. Learning latent superstructures in variational autoencoders for deep multidimensional clustering. In *International Conference on Learning Representations*, 2019b. Available in: [\[3\]](#). Cited on pages: 28.
- R. LiKamWa, Y. Liu, N. D. Lane, and L. Zhong. Moodscope: Building a mood sensor from smartphone usage patterns. In *International Conference on Mobile systems, Applications, and Services (MOBISYS)*, 2013. Cited on pages: 38.
- Z. C. Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018. Cited on pages: 101.
- Z. C. Lipton, J. Berkowitz, and C. Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015. Available in: [\[4\]](#). Cited on pages: 14, 30.
- Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel. Learning to diagnose with LSTM recurrent neural networks. In *International Conference on Learning Representations (ICLR)*, 2016a. Available in: [\[5\]](#). Cited on pages: 31, 36.
- Z. C. Lipton, D. C. Kale, R. Wetzel, et al. Modeling missing data in clinical time series with rnns. *Machine Learning for Healthcare (ML4HC)*, 56, 2016b. Available in: [\[6\]](#). Cited on pages: 31, 36, 99.
- J. Liu, B. Gallego, and S. Barbieri. Incorporating Uncertainty in Learning to Defer Algorithms for Safe Computer-Aided Diagnosis. *Scientific reports*, 12(1):1–9, 2022. Cited on pages: 64, 65, 87.
- G. Loaiza-Ganem and J. P. Cunningham. The continuous bernoulli: fixing a pervasive error in variational autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. Cited on pages: 97.
- H. Lu, D. Fraundorfer, M. Rabbi, M. S. Mast, G. T. Chittaranjan, A. T. Campbell, D. Gatica-Perez, and T. Choudhury. Stresssense: Detecting stress in unconstrained acoustic environments using smartphones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 351–360, 2012. Cited on pages: 38.
- Y. Luo, X. Cai, Y. Zhang, J. Xu, et al. Multivariate time series imputation with generative adversarial networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. Cited on pages: 31, 37.
- C. Ma. *Advances in Bayesian Machine Learning: From Uncertainty to Decision Making*. PhD thesis, University of Cambridge, 2022. Available in: [\[7\]](#). Cited on pages: 14.

- C. Ma and C. Zhang. Identifiable generative models for missing not at random data imputation. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. Available in: [\[1\]](#). Cited on pages: 21.
- C. Ma, S. Tschitschek, K. Palla, J. M. H. Lobato, S. Nowozin, and C. Zhang. EDDI: Efficient dynamic discovery of high-value information with partial VAE, 2019. Available in: [\[1\]](#). Cited on pages: 22, 37, 99.
- C. Ma, S. Tschitschek, Y. Li, R. Turner, J. M. Hernandez-Lobato, and C. Zhang. Hm-vaes: a deep generative model for real-valued data with heterogeneous marginals. In *Symposium on Advances in Approximate Bayesian Inference (AABI)*, pages 1–8, 2020a. Available in: [\[1\]](#). Cited on pages: 25.
- C. Ma, S. Tschitschek, R. Turner, J. M. Hernández-Lobato, and C. Zhang. Vaem: a deep generative model for heterogeneous mixed type data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020b. Available in: [\[1\]](#). Cited on pages: 25, 37.
- D. Madras, T. Pitassi, and R. Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems*, 2018. Cited on pages: 7, 54, 64, 71.
- H. W. Marsh. Pairwise deletion for missing data in structural equation models: Nonpositive definite matrices, parameter estimates, goodness of fit, and adjusted sample sizes. *Structural Equation Modeling: A Multidisciplinary Journal*, 5(1):22–36, 1998. Available in: [\[1\]](#). Cited on pages: 21.
- P.-A. Mattei and J. Frellsen. Leveraging the exact likelihood of deep latent variable models. *Advances in Neural Information Processing Systems*, 31, 2018. Available in: [\[1\]](#). Cited on pages: 18.
- P.-A. Mattei and J. Frellsen. MIWAE: Deep generative modelling and imputation of incomplete data sets. In *International Conference on Machine Learning (ICML)*. PMLR, 2019. Available in: [\[1\]](#). Cited on pages: 22, 37.
- A. Mehrotra, F. Tsapeli, R. Hendley, and M. Musolesi. Mytraces: Investigating correlation and causation between users emotional states and mobile phone interaction. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):1–21, 2017. Cited on pages: 38.
- V. B. Meresht, A. De, A. Singla, and M. Gomez-Rodriguez. Learning to switch between machines and humans. *arXiv preprint arXiv:2002.04258*, 2020. Cited on pages: 64.
- M. Minderer, J. Djolonga, R. Romijnders, F. Hubis, X. Zhai, N. Houlsby, D. Tran, and M. Lucic. Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems*, 34:15682–15694, 2021. Cited on pages: 71.
- P. Moreno-Muñoz, A. Artés, and M. Alvarez. Heterogeneous multi-output gaussian process prediction. *Advances in neural information processing systems*, 31, 2018. Available in: [\[1\]](#). Cited on pages: 25.
- H. Mozannar and D. A. Sontag. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020. Cited on pages: 8, 9, 54, 55, 57, 64, 65, 66, 67, 68, 70, 71, 72, 75, 78, 79, 80, 82, 87, 96, 113.
- H. Mozannar, H. Lang, D. Wei, P. Sattigeri, S. Das, and D. Sontag. Who should predict? exact algorithms for learning to defer to humans. In *AISTATS*, 2023. Cited on pages: 55, 57, 66, 68, 69, 70, 81, 82.
- K. P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. Available in: [\[1\]](#). Cited on pages: 55, 56, 57.
- E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Do deep generative models know what they don’t know? In *International Conference on Learning Representations*, 2019. Available in: [\[1\]](#). Cited on pages: 20.
- A. Nazabal, P. M. Olmos, Z. Ghahramani, and I. Valera. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501, 2020. Available in: [\[1\]](#). Cited on pages: 9, 22, 23, 25, 26, 27, 28, 33, 37, 40, 42.
- R. M. Neal and G. E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. *Learning in graphical models*, pages 355–368, 1998. Cited on pages: 16.
- J. Needham. *Science and civilisation in China*, volume 5. Cambridge University Press, 1974. Cited on pages: 2.

- A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015. Cited on pages: 54.
- C. Ni, N. Charoenphakdee, J. Honda, and M. Sugiyama. On the calibration of multiclass classification with rejection. In *Advances in Neural Information Processing Systems*, 2019. Available in: [\[1\]](#). Cited on pages: 62, 63.
- J. Nixon, M. W. Dusenberry, L. Zhang, G. Jerfel, and D. Tran. Measuring calibration in deep learning. In *CVPR Workshops*, volume 2, 2019. Cited on pages: 71.
- N. Okati, A. De, and M. Gomez-Rodriguez. Differentiable learning under triage. In *Advances in Neural Information Processing Systems, (NeurIPS)*, 2021. Cited on pages: 64, 65, 87.
- T. Pearce, A. Brintrup, and J. Zhu. Understanding softmax confidence and uncertainty. *arXiv preprint arXiv:2106.04972*, 2021. Cited on pages: 73.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011. Cited on pages: 44.
- I. Peis, C. Ma, and J. M. Hernández-Lobato. Missing data imputation and acquisition with deep hierarchical models and hamiltonian monte carlo. In *Advances in Neural Information Processing Systems*, volume 35, pages 35839–35851, 2022. Available in: [\[1\]](#). Cited on pages: 25.
- J. C. Peterson, R. M. Battleday, T. L. Griffiths, and O. Russakovsky. Human uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9617–9626, 2019. Cited on pages: 63.
- M. F. Pradier, J. Zazo, S. Parbhoo, R. H. Perlis, M. Zazzi, and F. Doshi-Velez. Preferential mixture-of-experts: Interpretable models that rely on human expertise as much as possible. *AMIA Summits on Translational Science Proceedings*, 2021:525, 2021. Cited on pages: 65, 87.
- Y. L. Qiu, H. Zheng, and O. Gevaert. Genomic data imputation with variational auto-encoders. *GigaScience*, 9(8), 2020. Cited on pages: 37.
- M. Raghu, K. Blumer, G. Corrado, J. Kleinberg, Z. Obermeyer, and S. Mullainathan. The algorithmic automation problem: Prediction, triage, and human effort. *arXiv preprint arXiv:1903.12220*, 2019. Cited on pages: 63, 65, 87.
- H. G. Ramaswamy, B. Srinivasan Babu, S. Agarwal, and R. C. Williamson. On the consistency of output code based learning algorithms for multiclass learning problems. In *Conference on Learning Theory*, 2014. Cited on pages: 59, 81.
- H. G. Ramaswamy, A. Tewari, and S. Agarwal. Consistent algorithms for multiclass classification with an abstain option. *Electronic Journal of Statistics*, 12:530–554, 2018. Cited on pages: 62, 63.
- A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. Cited on pages: 3.
- C. E. Rasmussen, C. K. Williams, et al. *Gaussian processes for machine learning*, volume 1. Springer, 2006. Cited on pages: 43.
- M. D. Reid and R. C. Williamson. Surrogate regret bounds for proper losses. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 897–904, 2009. Cited on pages: 58, 60.
- M. D. Reid and R. C. Williamson. Composite binary losses. *Journal of Machine Learning Research*, 11 (83):2387–2422, 2010. Available in: [\[1\]](#). Cited on pages: 58, 60, 74.
- D. J. Rezende and F. Viola. Taming vaes. *arXiv preprint arXiv:1810.00597*, 2018. Available in: [\[1\]](#). Cited on pages: 20.

- D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014. Cited on pages: 18, 20.
- Y. Romano, E. Patterson, and E. Candes. Conformalized quantile regression. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. Cited on pages: 84.
- Y. Romano, M. Sesia, and E. Candes. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:3581–3591, 2020. Cited on pages: 84.
- L. Romero-Medrano and A. Artés-Rodríguez. Multi-source change-point detection over local observation models. *Pattern Recognition*, 134:109116, 2023. Available in: [\[5\]](#). Cited on pages: 25.
- D. B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976. Available in: [\[5\]](#). Cited on pages: 21.
- M. Sadinle, J. Lei, and L. Wasserman. Least ambiguous set-valued classifiers with bounded error levels. *Journal of the American Statistical Association*, 114(525):223–234, 2019. Cited on pages: 84.
- B. Sayin, F. Casati, A. Passerini, J. Yang, and X. Chen. Rethinking and recomputing the value of ml models. *arXiv preprint arXiv:2209.15157*, 2022. Cited on pages: 71.
- J. L. Schafer and J. W. Graham. Missing data: our view of the state of the art. *Psychological methods*, 7(2):147, 2002. Available in: [\[5\]](#). Cited on pages: 21.
- M. Schemmer, P. Hemmer, N. Kühn, C. Benz, and G. Satzger. Should i follow ai-based advice? measuring appropriate reliance in human-ai decision-making. *CHI Conference on Human Factors in Computing System, Workshop on Trust and Reliance in AI-Human Teams (trAIIt)*, 2022. Cited on pages: 54.
- M. Schirmer, M. Eltayeb, S. Lessmann, and M. Rudolph. Modeling irregular time series with continuous recurrent units. In *International Conference on Machine Learning (ICML)*, volume 162, pages 19388–19405, 2022. Available in: [\[5\]](#). Cited on pages: 99.
- P. Schmidt and F. Biessmann. Calibrating human-ai collaboration: Impact of risk, ambiguity and transparency on algorithmic bias. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 431–449. Springer, 2020. Cited on pages: 71.
- M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997. Available in: [\[5\]](#). Cited on pages: 31, 98.
- M. Sensoy, L. Kaplan, and M. Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018. Cited on pages: 73.
- G. Shafer and V. Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(12):371–421, 2008. Cited on pages: 84.
- C. Shang, A. Palmer, J. Sun, K.-S. Chen, J. Lu, and J. Bi. Vigan: Missing view imputation with generative adversarial networks. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 766–775. IEEE, 2017. Cited on pages: 37.
- I. Silva, G. Moody, D. J. Scott, L. A. Celi, and R. G. Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 Computing in Cardiology*, pages 245–248. IEEE, 2012. Cited on pages: 37, 45, 48, 96.
- D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016. Cited on pages: 3.
- D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018. Cited on pages: 3.
- K. Stankeviciute, A. M Alaa, and M. van der Schaar. Conformal time-series forecasting. *Advances in neural information processing systems*, 34:6216–6228, 2021. Cited on pages: 100.

- I. Steinwart. How to compare different loss functions and their risks. *Constructive Approximation*, 26: 225–287, 2007. Cited on pages: 81.
- D. J. Stekhoven and P. Bühlmann. Missforestnon-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012. Available in: [\[5\]](#). Cited on pages: 21.
- M. Steyvers, H. Tejada, G. Kerrigan, and P. Smyth. Bayesian modeling of human–ai complementarity. *Proceedings of the National Academy of Sciences*, 119(11):e2111547119, 2022. Cited on pages: 63.
- N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano. Learning to summarize with human feedback. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc., 2020. Available in: [\[5\]](#). Cited on pages: 3.
- E. Straitouri, L. Wang, N. Okati, and M. G. Rodriguez. Improving expert predictions with conformal prediction. In *International Conference on Machine Learning (ICML)*, 2023. Available in: [\[5\]](#). Cited on pages: 87, 88, 100.
- J. H. Tigay. *The evolution of the Gilgamesh epic*. Bolchazy-Carducci Publishers, 2002. Cited on pages: 2.
- M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999. Cited on pages: 15.
- I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. In *Advances in Neural Information Processing Systems*, 2021. Cited on pages: 90.
- N. Tomasev, K. R. McKee, J. Kay, and S. Mohamed. Fairness for unobserved characteristics: Insights from technological impacts on queer communities. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 254–265, 2021. Cited on pages: 53.
- J. Tomczak and M. Welling. VAE with a vampprior. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1214–1223. PMLR, 2018. Cited on pages: 19, 28, 40.
- P. Tschandl, C. Rosendahl, and H. Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018. Cited on pages: 88, 89, 101.
- P. Tschandl, C. Rinner, Z. Apalla, G. Argenziano, N. C. F. Codella, A. C. Halpern, M. Janda, A. Lallas, C. Longo, J. Malvehy, J. Paoli, S. Puig, C. Rosendahl, H. P. Soyer, I. Zalaudek, and H. Kittler. Humancomputer collaboration for skin cancer recognition. *Nature Medicine*, pages 1–6, 2020. Cited on pages: 7, 54, 71.
- J. Vaicenavicius, D. Widmann, C. Andersson, F. Lindsten, J. Roll, and T. Schön. Evaluating model calibration in classification. In *Conference on Artificial Intelligence and Statistics*, 2019. Cited on pages: 71, 72, 87.
- I. Valera and Z. Ghahramani. Automatic discovery of the statistical types of variables in a dataset. In *International Conference on Machine Learning (ICML)*, pages 3521–3529, 2017. Cited on pages: 25.
- I. Valera, M. F. Pradier, M. Lomeli, and Z. Ghahramani. General latent feature models for heterogeneous datasets. *Journal of Machine Learning Research (JMLR)*, 21(100):1–49, 2020. Cited on pages: 25.
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. Cited on pages: 99.
- R. Vedantam, I. Fischer, J. Huang, and K. Murphy. Generative models of visually grounded imagination. In *International Conference on Learning Representations*, 2018. Available in: [\[5\]](#). Cited on pages: 22.
- R. Verma. On the calibration of learning to defer systems. Master’s thesis, University of Amsterdam, 2022. Cited on pages: 55.
- R. Verma and E. Nalisnick. Calibrated learning to defer with one-vs-all classifiers. In *International Conference on Machine Learning (ICML)*, 2022. Cited on pages: 8, 9, 54, 55, 59, 60, 65, 66, 67, 68, 70, 71, 72, 73, 74, 75, 78, 79, 81, 82, 84, 87, 88, 90, 96, 115.

- R. Verma, D. Barrejón, and E. Nalisnick. On the calibration of learning to defer to multiple experts. In *Workshop on Human-Machine Collaboration and Teaming in International Confere of Machine Learning*, 2022. Available in: [\[5\]](#). Cited on pages: 78.
- R. Verma, D. Barrejón, and E. Nalisnick. Learning to defer to multiple experts: Consistent surrogate losses, confidence calibration, and conformal ensembles. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, Proceedings of Machine Learning Research, pages 11415–11434. PMLR, 25–27 Apr 2023. Available in: [\[5\]](#). Cited on pages: 8, 55, 78, 81, 99, 100.
- H. Wang, W. Liu, A. Bocchieri, and Y. Li. Can multi-label classification networks know what they dont know? *Advances in Neural Information Processing Systems (NeurIPS)*, 34:29074–29087, 2021. Cited on pages: 73.
- L. Weng. From autoencoder to beta-vae. *lilianweng.github.io*, 2018. Available in: [\[5\]](#). Cited on pages: 19.
- I. R. White, P. Royston, and A. M. Wood. Multiple imputation using chained equations: issues and guidance for practice. *Statistics in medicine*, 30(4):377–399, 2011. Cited on pages: 22, 44.
- B. Wilder, E. Horvitz, and E. Kamar. Learning to complement humans. In *International Joint Conference on Artificial Intelligence*, 2020. Cited on pages: 65, 87.
- R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992. Cited on pages: 20.
- A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378. PMLR, 2016. Cited on pages: 98.
- H. Wu, B. Esmaeili, M. Wick, J.-B. Tristan, and J.-W. Van De Meent. Conjugate energy-based models. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11228–11239. PMLR, 18–24 Jul 2021. Available in: [\[5\]](#). Cited on pages: 98.
- J. Yoon, J. Jordon, and M. Schaar. GAIN: Missing data imputation using generative adversarial nets. In *International Conference on Machine Learning (ICML)*. PMLR, 2018. Cited on pages: 37.
- Y. Yu, S. Bates, Y. Ma, and M. I. Jordan. Robust calibration with multi-domain temperature scaling. *arXiv preprint arXiv:2206.02757*, 2022. Cited on pages: 71.
- M. Yuan and M. Wegkamp. Classification methods with reject option based on convex risk minimization. *Journal of Machine Learning Research*, 11(5):111–130, 2010. Cited on pages: 62.
- S. Zagoruyko and N. Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016. Cited on pages: 90.
- Y. Zhang, Q. V. Liao, and R. K. Bellamy. Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 295–305, 2020. Cited on pages: 71.
- S. Zhao, M. Kim, R. Sahoo, T. Ma, and S. Ermon. Calibrating predictions to decisions: A novel approach to multi-class calibration. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. Cited on pages: 74.
- H. Zhong, Z. Guo, C. Tu, C. Xiao, Z. Liu, and M. Sun. Legal judgment prediction via topological learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3540–3549, 2018. Cited on pages: 53.
- Y. Zoabi, S. Deri-Rozov, and N. Shomron. Machine learning-based prediction of covid-19 diagnosis based on symptoms. *npj digital medicine*, 4(1):3, 2021. Cited on pages: 53.