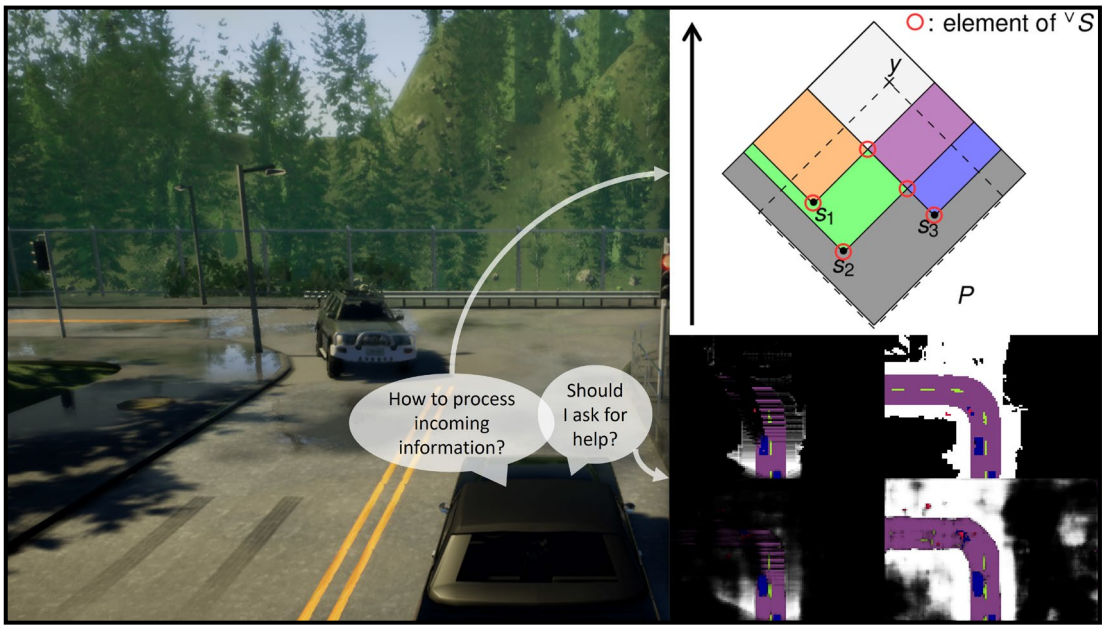
Efficient decentralized collaborative perception for autonomous vehicles

Maxime Chaveroche

Par **Maxime CHAVEROCHE**

Efficient decentralized collaborative perception for autonomous vehicles

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 29 septembre 2021
Spécialité : Sciences et Technologies de l'Information et des Systèmes : Unité de recherche Heudyasic (UMR-7253)

D2633

THESE DE DOCTORAT de l'Université de technologie de Compiègne

Préparée à Heudiasyc, UMR CNRS 7253

Efficient decentralized collaborative perception for autonomous vehicles

Soutenue par

Maxime CHAVEROCHE

le 29 septembre 2021

Ecole doctorale n°71

Sciences pour l'ingénieur

Spécialité

Sciences et technologies de l'information et des systèmes

Composition du jury

Jérémie MARY Maître de conférence – HDR, 27ème section CNU, Université de Lille, Criteo AI Lab.	*Rapporteur*
Frédéric PICHON Professeur, 61ème section CNU, Université d'Artois	*Rapporteur*
Thierry DENOEUX Professeur, 61ème section CNU, Université de technologie de Compiègne	*Examinateur*
Stéphane CANU Professeur, 61ème section CNU, INSA Rouen	*Examinateur*
Véronique CHERFAOUI Professeure, 61ème section CNU, Université de technologie de Compiègne	*Directrice de thèse*
Franck DAVOINE Chargé de recherche – HDR, section 07 CNRS, Université de technologie de Compiègne	*Directeur de thèse*

Perception collaborative décentralisée et efficace pour le véhicule autonome

Maxime Chaveroche

Soutenue le 29 septembre 2021

Abstract

Recently, we have been witnesses of accidents involving autonomous vehicles and their lack of sufficient information at the right time. One way to tackle this issue is to benefit from the perception of different view points, namely collaborative perception. We propose here a decentralized collaboration, i.e. peer-to-peer, in which the agents are active in their quest for full perception by asking for specific areas in their surroundings on which they would like to know more. Ultimately, we want to optimize a trade-off between the maximization of knowledge about moving objects and the minimization of the total information received from others, to limit communication costs and message processing time.

To this end, we chose to use Dempster-Shafer Theory (DST) in order to identify different types of uncertainties. In particular, DST allows us to distinguish what has never been perceived (out of range or occluded area) — which is mainly what collaborative perception tries to reduce — from what is debated among different sources (conflict arising from fusion of sensors or other vehicles perceptions). More generally, DST takes into account the specificity of evidence, meaning that it provides information about the reliability of an agent's belief, which is crucial for safety. DST also features the advantage of easily dealing with data incest with its *Cautious* fusion rule, which is a problem inherent to the decentralized approach. However, DST comes with high spatial and computational complexities, especially for dealing with data incest in fusion, which limits its usage to random experiments with few possible outcomes. Thus, we first proposed an efficient exact method to compute the decompositions needed for this Cautious fusion, exploiting what we called focal points. Then, we generalized this method to any Möbius transform in any partially ordered set (including all transformations in DST), we found ways to efficiently compute these focal points and we proposed a generalization of the decomposition required by the Cautious fusion. This generalized decomposition allows one to use this Cautious fusion in more cases, in particular cases where an agent has gathered very specific evidence. This enhances both accuracy and computational stability in consecutive fusions. However, algorithms naively based on our formulas would have a higher worst-case complexity than the complexity of the optimal general algorithms commonly employed in DST — which is already more than exponential. Therefore, we later proposed algorithms with complexities always better than the state of the art, and more general, leveraging properties of distributive lattices.

After this work on the fusion process itself, we tackled the issue of redundancy and irrelevance in decentralized collaborative perception. For this, we proposed a way to learn a communication policy that reverses the usual communication paradigm by only requesting from other vehicles what is unknown to the ego-vehicle, instead of filtering on the sender side. We tested three different models to be taken as base for a Deep Reinforcement Learning (DRL) algorithm and compared them to a broadcasting policy and a random policy. More precisely, we slightly modified a state-of-the-art generative model named Temporal Difference VAE (TD-VAE) to make it sequential. We named this variant Sequential TD-VAE (STD-VAE). We also proposed Locally Predictable VAE (LP-VAE), inspired by STD-VAE, designed to enhance its prediction capabilities. We showed that LP-VAE produced better belief states for prediction than STD-VAE, both as a standalone model and in the context of DRL. The last model we tested was a simple state-less model (Convolutional VAE). Experiments were conducted in the driving simulator CARLA, with vehicles exchanging parts of semantic grid maps. Policies learned based on LP-VAE featured the best trade-off, as long as future rewards were taken into account. Our best models reached on average about 25% of the maximum information gain while requesting only about 5% of the space around the ego-vehicle to others. We also provided interpretable hyperparameters controlling the reward function, which makes this trade-off adjustable (e.g. allowing greater communication costs).

Résumé

Récemment, nous avons été témoins d'accidents impliquant des véhicules autonomes et leur manque momentané d'information pertinente. Une manière d'adresser ce problème est d'avoir recours à la perception collaborative, c'est-à-dire de bénéficier de la perception d'une même scène sous différents points de vue. Nous proposons ici une collaboration décentralisée, i.e. pair-à-pair, dans laquelle les agents sont actifs dans leur quête pour la perception complète en demandant des zones spécifiques dans leur voisinage sur lesquelles ils voudraient en savoir plus. In fine, nous voulons optimiser un compromis entre la maximisation du savoir à propos des usagers de la route et la minimisation du volume total d'information reçu des autres, dans le but de limiter les coûts en communications et le temps de traitement des messages.

Dans cette optique, nous avons choisi d'utiliser la Théorie de Dempster-Shafer (DST) afin d'identifier différents types d'incertitude. En particulier, la DST distingue ce qui n'a jamais été perçu (zone hors de vue ou occultée) — ce qui est principalement ce que la perception collaborative essaie de réduire — de ce qui est débattu parmi des sources différentes (conflit provenant de la fusion de capteurs ou de perceptions d'autres véhicules). Plus généralement, la DST prend en compte la spécificité des observations, c'est-à-dire qu'elle fournit des informations sur la fiabilité des croyances d'un agent, ce qui est crucial pour la sécurité routière. La DST a aussi pour avantage, avec sa règle de fusion *Cautious*, d'éviter facilement la consanguinité des données, un problème inhérent à l'approche décentralisée. Toutefois, la DST vient avec de fortes complexités en temps et en espace, particulièrement dans le calcul de la fusion *Cautious*, ce qui limite son usage à des expériences aléatoires comportant peu d'événements atomiques. Ainsi, notre première contribution fut de proposer une méthode exacte et efficace pour le calcul des décompositions nécessaires à cette fusion Cautious, en exploitant ce que nous avons appelé points focaux. Nous avons ensuite généralisé cette méthode à toute transformée de Möbius dans tout ensemble partiellement ordonné (incluant toutes les transformations en DST), nous avons trouvé des moyens de calculer efficacement ces points focaux et nous avons proposé une généralisation de la décomposition requise par la fusion Cautious. Cette décomposition généralisée permet d'employer cette fusion Cautious dans plus de cas, en particulier ceux où un agent a reporté des observations très spécifiques. Nous montrons que ceci améliore à la fois la précision et la stabilité calculatoire de fusions successives. Cependant, des algorithmes basés naïvement sur nos formules auraient une plus haute complexité de pire cas que celle des algorithmes généraux optimaux communément utilisés en DST — qui est déjà plus qu'exponentielle. De fait, nous avons proposé plus tard des algorithmes ayant des complexités toujours meilleures que celles de l'état de l'art, et étant plus généraux, tirant partie des propriétés des treillis distributifs.

Après ce travail sur le processus de fusion en lui-même, nous nous sommes attaqués aux problèmes de redondance et de non-pertinence dans la perception collaborative décentralisée. Pour cela, nous avons proposé un moyen d'apprendre une politique de communication qui renverse le paradigme usuel de communication en ne demandant des autres véhicules que ce qui est inconnu de l'ego-véhicule, au lieu de filtrer du côté émetteur. Nous avons testé trois modèles différents pour servir de base à un algorithme d'apprentissage profond par renforcement (DRL) et les avons comparés à une politique de broadcast et à une politique aléatoire. Plus précisément, nous avons légèrement modifié un modèle génératif de l'état-de-l'art nommé *Temporal Difference* VAE (TD-VAE) pour le rendre séquentiel. Nous avons nommé cette variante *Sequential* TD-VAE (STD-VAE). Nous avons aussi proposé *Locally Predictable* VAE (LP-VAE), inspiré de STD-VAE, conçu pour améliorer ses capacités de prédiction. Nous avons montré que LP-VAE produisait de meilleurs *belief states* pour la prédiction que STD-VAE, à la fois en tant que modèle seul et dans le contexte du DRL. Le dernier modèle testé fut un simple modèle statique (*Convolutional* VAE). Nos expériences ont été faites dans le simulateur de conduite CARLA, avec des véhicules échangeant des parties de grilles sémantiques. Les politiques apprises sur LP-

VAE ont produit le meilleur compromis, tant que les récompenses futures étaient considérées. Nos meilleurs modèles ont atteint en moyenne environ 25% du gain maximum en information, tout en ne demandant qu'environ 5% de l'espace autour de l'ego-véhicule aux autres. Nous avons aussi fourni des hyperparamètres interprétables contrôlant la fonction de récompense, ce qui rend ce compromis ajustable (e.g. en autorisant de plus grands coûts en communication).

Contents

Symbol table

$\wedge$	*Infimum* operator. Also known as the *meet* operator. For two elements x and y in a partially ordered set P, the meet $x \wedge y$ represents the greatest element in P that is less than both x and y. If $x \leq y$, then $x \wedge y = x$.
$\vee$	*Supremum* operator. Also known as the *join* operator. For two elements x and y in a partially ordered set P, the join $x \vee y$ represents the least element in P that is greater than both x and y. If $x \leq y$, then $x \vee y = y$.
$\leq$	*less than or equal to.*
$\not\leq$	*Not less than nor equal to.*
$<$	*strictly less than (not equal to).*
$\geq$	*greater than or equal to.*
$\not\geq$	*Not greater than nor equal to.*
$>$	*strictly greater than (not equal to).*
$(P, \leq)$	Partially ordered set, i.e. the elements of P are partially ordered by the operator $\leq$. For any elements $x, y \in P$, if $x \leq y$, then $\leq$ orders x and y. If $x \not\leq y$ and $x \not\geq y$, then $\leq$ does not order x and y.
$\lvert\Omega\rvert$	Cardinality of a set Ω. If $\Omega = \{a, b, c\}$, then $\lvert\Omega\rvert = 3$.
$\subseteq$	Set inclusion operator. $A \subseteq B$ means that the set A is contained in the set B.
$\subset$	Strict set inclusion operator. $A \subset B$ means that the set A is contained in the set B, but $A \neq B$.
$\supseteq$	Dual set inclusion operator. $A \supseteq B$ means that the set A contains the set B.
$\subset$	Dual strict set inclusion operator. $A \supset B$ means that the set A contains the set B, but $A \neq B$.
2^Ω	Powerset of a set Ω, i.e. set containing all subsets of Ω. There are $2^{\lvert\Omega\rvert}$ of them.
$\cap$	Set intersection operator. It is the meet operator of $(2^\Omega, \subseteq)$.
$\cup$	Set union operator. It is the join operator of $(2^\Omega, \subseteq)$.
$A \backslash B$	Set exclusion operation. For two sets A and B, $A \backslash B$ is the set containing the elements of A that are not in B.
$\uparrow x$	Upper closure of x, i.e. the subset of P containing all elements greater than x. It is also known as an upset. When x is itself a set, $\uparrow x$ is the set containing all elements greater than at least one element of x.
$\downarrow x$	Lower closure of x, i.e. the subset of P containing all elements less than x. It is also known as an down set. When x is itself a set, $\downarrow x$ is the set containing all elements less than at least one element of x.
$x^{\uparrow S}$ or ${}^{S}_{\uparrow}x$	Upper closure of x in S, i.e. $(\uparrow x) \cap S$, if $S \subseteq P$.
$x^{\downarrow S}$ or ${}^{S}_{\downarrow}x$	Lower closure of x in S, i.e. $(\downarrow x) \cap S$, if $S \subseteq P$.
${}^{\wedge}S$	meet-closure of a set S, i.e. $\{\bigwedge A \ / \ \emptyset \subset A \subseteq S\}$.
${}^{\vee}S$	join-closure of a set S, i.e. $\{\bigvee A \ / \ \emptyset \subset A \subseteq S\}$.

Introduction

The concept of an automated car dates back to 1921, when the first public demonstration of a remote-controlled car took place in the USA [1]. At the time, this *phantom auto* was steered by a human in another vehicle behind it, via radio. Some started to fantasize about its potential for the future. In 1930 already, the writer Werner Illing was publishing *Utopolis*, an utopia in which vehicles were safely controlled by traffic lights. Five years later, in *The Living Machine* by writer David H. Keller, the self-steering car was credited with a sharp decline in the number of road accidents, allowing parents to safely send their children to school as well as enabling old and blind people to be transported across the continent. In the leading fifty years, technology, including microelectronics, rose to the point that some advanced driver-assistance systems such as Cruise control and Anti-lock Braking System (ABS) emerged. Then, in 1994 in France, the PROMETHEUS european project demonstrated the results of their work on autonomous vehicles. Their two vehicles autonomously drove more than 1000 km up to 130 km/h on three-lane highways around Paris, in the middle of heavy traffic. It was the first system based only on vision, while control by road infrastructure was privileged until then. One of the reasons of this change was the absence of infrastructure cost [2]. Since then, autonomous vehicles have increasingly become a serious topic of academic research, and the industry started issuing partially automated vehicles. Progress had been made, but as of 2020, almost 100 years since the introduction of the concept, there was still no self-driving car. Indeed, the Society of Automotive Engineers (SAE) established, in 2014, 6 levels of automation, 0 being fully manual to 5 being fully autonomous [3]. At the exception of the first vehicle officially recognized as a SAE level 3 system by the Japanese government[1], launched by Honda in 2021, the most advanced technologies such as Tesla's Autopilot are in fact only SAE level 2, i.e. technologies that require the constant attention of the driver who must be prepared to intervene at any time. That fact, dimmed by marketing campaigns, was reminded to us through the loss of human lives [4].

In January 2016 in China, a Tesla vehicle with Autopilot activated on highway did not avoid an unexpected obstacle in the form of a truck partially parked on its lane. It was following a leading vehicle, which changed lane to avoid the truck. The Tesla did not, probably due to a late detection caused by the occluding presence of the leading vehicle in its perceptive field. In May 2016 in the USA, another Tesla in Autopilot crashed into an 18-wheel tractor-trailer. As the tractor-trailer was crossing an intersection in front of the vehicle, the latter did not apply the brakes, rushing at 119 km/h under the trailer. It appeared that the vehicle did not distinguish the white trailer from the brightly lit sky. In March 2017 also in the USA, the first pedestrian was killed by an Uber vehicle in autopilot mode. Elaine Herzberg was crossing outside of a crosswalk, when the self-driving car arrived, detecting her first as an unknown object, then as a vehicle, and finally as a bicycle, each of which having different predicted paths. It was too late to begin emergency collision avoidance maneuvers once the pedestrian had been detected as such.

[1] https://asia.nikkei.com/Business/Automobiles/Honda-launches-world-s-first-level-3-self-driving-car

In March 2018 still in the USA, a Tesla with Autopilot enabled crashed into a concrete barrier separating an exit lane from highway, likely due to a misdetection of road surface marking. The list is not exhaustive, but already we see a pattern emerge: misdetections.

It is not surprising, then, that the academic community started to put an emphasis on the need to enhance the perception of these vehicles, beyond multi-sensor fusion. While different kinds of sensors (e.g. camera + LIDAR) catch different features and may already avoid to confuse the sky with a truck trailer, for example, there are cases where this is not enough. For instance, road markings cannot be easily sensed otherwise than with a camera. Also, a pedestrian partially hiding in bushes cannot be better detected with lasers, nor can be an obstacle in front of a leading vehicle. In these cases, it might be necessary to get the perception of the same scene from another viewpoint entirely. Of course, this cannot be achieved by the vehicle alone. It requires other agents, whether they are integrated as part of the road infrastructure or other vehicles. The safety benefits of implementing this *cooperative perception* have been demonstrated in [5] on public road and in the driving simulator CARLA [6].

It is now a hot topic. In November 2020, SAE International announced[2] the formation of the Cooperative Automation Driving System (CADS) committee, focusing on developing standards improving communications between autonomous vehicles and their surroundings. In parallel, the European Union, through its *Horizon Europe* and *H2020* programs, issued calls for projects about Connected, Cooperative and Automated Mobility (CCAM)[3].

Some works focus on communication protocols for this cooperation between autonomous vehicles [7]. For example, [8] proposes CarSpeak, a content-centric communication system in which a vehicle can request pieces of information localized in space, instead of requesting a particular peer. But, we are more interested by works on cooperative systems themselves. Some, remaining true to the original vision of autonomous vehicles, consider cooperations centralized and controlled by some piece of road infrastructure. In [9], a centralized cooperative system is proposed, relying on communications to the road infrastructure, in order to control vehicles at an intersection without traffic lights. In [10], another centralized cooperative system relying on communications to the road infrastructure is presented, this time to control vehicles at interchanges. Even cooperations for the control of pieces of road infrastructure are considered. For instance, in [11], they used Cooperative Deep Reinforcement Learning (CDRL) to simultaneously coordinate traffic lights at multiple intersections, given the state of nearby vehicles. More precisely, they used a variation of Deep Q-Learning (DQN). More generally, all cooperation with road infrastructures is described as a Vehicle-to-Infrastructure (V2I) cooperation. F-Cooper [12] is another example of V2I system. It is an improvement over Cooper [13], compressing 3D point clouds into lightweight features learned by a Convolutional Neural Network (CNN). While this alleviates the burden on wireless transmissions, it increases the computational load, which might be too much for individual vehicles. For this reason, their method is intended to be employed by vehicles on-edge, i.e. vehicles communicating with computing servers, as a V2I system. Another V2I system is described in [14], helping autonomous vehicles in estimating their own state (position and velocity). It is designed to work even in case of temporary infrastructure failure.

When vehicles cooperate directly with each other, in a decentralized way, it is a Vehicle-to-Vehicle (V2V) cooperation. Sometimes, Vehicle-to-Pedestrian are considered, where pedestrians hold wireless devices. A hybrid system that would work under any of these modes is classified as a Vehicle-to-everything (V2X) system [15]. In [16], a V2V system is proposed, in which vehicles predict the probable trajectory of all other nearby vehicles and issue warnings to vehicles that are likely to collide. In [17], the authors propose a V2V cooperative perception system

[2]https://www.sae.org/news/press-room/2020/11/sae-international-announces-the-formation-of-the-cooperative-automation-driving-system-committee

[3]https://nextmove.fr/appels/mobilite-autonome-et-connectee-les-appels-ccam-dhorizon-europe/

for two vehicles using perception grid maps. They introduce a novel method in the spirit of genetic algorithms to match and merge two grid maps. By doing so, they indirectly solve a relative pose estimation problem. Combined with a local Simultaneous Localization And Mapping (SLAM) algorithm to estimate the pose of each vehicle, it enables them to perform multivehicle perception association. This system allows a vehicle to see through a leading vehicle, but not much more. It requires proximity and overlap between the two vehicles perceptions. In [18], they presented a similar V2V cooperative perception system, providing see-through views to vehicles following other vehicles. They employed Correlative Scan Matching (CSM) to merge overlapping perception maps and derive from this fused perception several functionalities to assist the driver. In a follow-up article [19], they demonstrated the positive impact of the aforementioned cooperative perception system on autonomous driving. In [20], the authors exploited V2V cooperative perception to anticipate driving behaviors of other vehicles. Inspired by mirror neurons, they implemented intention awareness. Cooperative perception allowed them to get a see-through perception, seeing through a leading vehicle, so that emergency maneuvers could be predicted if an object suddenly appears in front of the leading vehicle. The cooperative perception system Cooper, proposed in [13], is a V2V system that shares raw sensor outputs. These outputs are 3D point clouds obtained through laser scans (LIDAR). This method avoids the accumulation of individual recognition errors and data incest (by not sharing fused data), but is heavy on the means of communication and processing. Also, not sharing fused data limits the reach of this system, unless all neighbors of nearby vehicles also send their data, which would be redundant and computationally heavy. A few works explicitly address the phenomenon of data incest. The authors of [21] propose a V2V cooperative perception system in which vehicles perform multi-object tracking, with Gaussian Mixture Probability Hypothesis Density (GM-PHD) filters, and share their hypotheses together. The fusion process is aware of data incest and uses a General Covariance Intersection (GCI) algorithm in consequence. In [22], the authors propose a V2V system which takes into account data incest, avoiding it by using the Cautious fusion rule [23] from Dempster-Shafer Theory (DST) [24]. There are also works that treat security issues. For instance, in [25], they propose to integrate trust concerns in cooperative perception. To evaluate trust between two vehicles, they try to match objects detections in the shared areas of their respective perceptive fields. In [26], they present a system for detecting sybil attacks in V2V cooperative perception. It is based on the exhange of opinions about other agents in the network and direct signal strength measures to be compared to the announced positions. It also takes into account data incest and avoids it with the Cautious fusion rule from DST.

In this thesis, we focus on the proposition of a V2V cooperative perception system. Indeed, while setting a multitude of sensors in the road infrastructure can be imagined for V2I systems, this implies an infrastructure cost and the impossibility to share information with other agents when there is no server available nearby. Moreover, even if the whole world is covered by these servers, this system would still not be robust, as it would locally display a single point of failure. A failing server means that an area of the world cannot benefit from V2I cooperative perception. It also makes the agents broadcast their entire perception, which can be heavy on the means of communication and computation, giving rise to delays. In contrast, the decentralized V2V setting means that agents directly exchange pieces of information between them, without the need of any available infrastructure in their vicinity. Nevertheless, it also comes with new problems such as data incest, which appears when agents can unknowingly retrieve their own piece of information from another vehicle. This phenomenon is often neglected in the literature of V2V cooperative perception. Furthermore, autonomous vehicles do not have the computing power of dedicated servers nor do they have a lot of time to process, which often leads to the choice of V2I interactions when such supplementary task is taken into account. For instance, in [25], they propose to evaluate trust between two vehicles, but they argue that doing these

computations on a server (V2I) would be better as vehicles are only in proximity for short periods of time. Notice that in Dempster-Shafer Theory (DST) [24], using the disjunctive fusion rule [27] makes malicious sources less important without needing supplementary computations, as these sources can only add misplaced uncertainty, instead of misplaced certainty. Moreover, the Cautious and Bold fusion rules [23] from DST ensures that there is no data incest in the vehicular network, as demonstrated in [22]. This is what motivates the work presented in chapters 1, 2 and 3. Each chapter corresponds to a published article. The first chapter proposes an efficient exact method to compute the decompositions needed by DST fusion operators, chosen for dealing with data incest. The second chapter generalizes the ideas of Chapter 1 to any Möbius transform in any partially ordered set, which in particular transformed our approach into a general framework for more efficient algorithms computing most transformations in DST. This also enabled us to propose a generalization of the decomposition required by the fusion operators we chose. This generalization allows one to use these fusion operators in more cases, notably in cases where an agent has gathered specific evidence with absolute certainty. This enhances both accuracy and computational stability in consecutive fusions. The third chapter proposes a complementary method for these same computations that guarantees better worst-case complexities, for any Möbius transform in any distributive lattice, including most transformations exploited in DST. The last chapter of this thesis, chapter 4, tackles another great problem of V2V cooperative perception systems, which is redundancy, coupled with irrelevance. Since there is no server to centralize all information, the same piece of information may travel several times in the vehicular network, which may burden the means of communication, even if this piece of information is completely useless for the receiving vehicles. A receiver may receive information about an object that it already directly perceives or that has no impact on its path planning. While there are works that consider this problem, they do not consider the possibility of filtering on the receiver side, i.e. valuing missing pieces of information and requesting the most valued ones from the vehicular network. In this chapter, we leveraged generative models and model-free Deep Reinforcement Learning (DRL) to learn a communication policy that requests specific areas around the ego-vehicle to the vehicular network. More precisely, we proposed generative models for temporal sequences, derived from the Variational AutoEncoder (VAE) [28]. This reduces the state space, while keeping the essential, easing the convergence of the RL algorithm. On top, we chose the now classic actor-critic DRL algorithm Proximal Policy Optimization (PPO) [29]. The work presented in this last chapter corresponds to an article in preparation.

The work contained in this thesis has been done in the context of Heudiasyc, in the team SyRI (Robotic Systems in Interaction), which concentrates on improving perception and localization in autonomous vehicles. It was financed by the Hauts-de-France region and the Laboratory of Excellence on Systems-of-Systems (Labex MS2T).

Chapter 1

Efficient computation of the Cautious and Bold fusion rules

1.1 Motivations

1.1.1 Communication content and knowledge representation

We want to characterize the space around the ego-vehicle, based on its own local sensors and external ones. But, rather than exchanging raw measurements, which would be heavy on the means of communication and computation, we want to share high level information. For this, we chose to represent the ego-vehicle's perception as a discretization of its surroundings in which each cell is a probability distribution on the class of its content, i.e. a local semantic grid map. The choice of semantic grids over e.g. object lists is explained by the fact that grids represent the whole space around the ego-vehicle, with what is known and what is not. Doing so, grid maps allow us to project the ego-vehicle perception in space and to deduce from it where external information is needed the most. Grid maps are also more flexible to represent arbitrary object shapes and avoid to explicitly perform a computationally heavy data association.

In addition, we chose to use Dempster-Shafer Theory (DST) [24, 30] in order to identify different types of uncertainties at the cell level. In particular, DST allows us to distinguish what has never been perceived (out of range or occluded area), which is mainly what collaborative perception tries to reduce, from what is debated among different sources (conflict arising from fusion of sensors or other vehicles perceptions). More generally, DST takes into account the specificity of evidence, meaning that it provides information about the reliability of an agent's belief, which is crucial for safety.

1.1.2 Data incest

Data incest is a situation in which the same evidence is taken into account more than once in information fusion. For instance, a vehicle might send its perception to another vehicle that would fuse it with its own perception and broadcast the result. Then, the first vehicle would receive the resulting information and fuse it with its current perception, as did the second vehicle. Doing so, the information contained in the perception of the first vehicle would have been taken twice by it, making the first vehicle overconfident in the information brought by its sensors. Repeating this process several times would make the whole vehicular network diverge, which may lead to accidents.

This is not a new topic and several works have already addressed this issue, but with limitations on the communication network topology. Unfortunately, vehicular networks have arbitrary and ever changing topologies [31], which makes works such as [32, 33] impracticable. Moreover, we are interested in sharing probability distributions over a classification, i.e. over a qualitative random variable with nominal values. This means that all methods based on Kalman filters or continuous random models in general, such as [31, 34], are not applicable. On the other hand, Dempster-Shafer Theory offers a way to avoid data incest in the fusion of probability distributions over a qualitative random variable with nominal values, independently from the network topology. Indeed, its Cautious fusion rule [23] introduces an operator that prevents the reinforcement of an hypothesis by design, in the same spirit as the Covariance Intersection Filter [35]. This provides a guaranteed consistency in the vehicular network, though sub-optimally. This use of the Cautious rule for distributed data fusion has been successfully implemented in [22, 36, 37].

1.1.3 Blocking point

It seems DST would be useful for both data incest management and efficient collaborative perception. Its Cautious fusion rule addresses the former, while its inherent distinction between different levels of specificity allows the agent to give its information needs some hierarchy. However, DST comes with high spatial and computational complexities, especially in the computation of the Cautious fusion, which limits the number of classes that can be considered. This is all the more limiting since autonomous vehicles have less computational power than a server in a centralized setting. In addition, the Cautious fusion operator cannot be applied to *dogmatic* belief distributions. This means that there always must be some mass assigned to all possible outcomes, which prevents consecutive fusions from truly filtering hypotheses. As a consequence, the number of sets of outcomes to consider can only grow and make subsequent fusions more difficult.

1.2 Introduction to Dempster-Shafer Theory

Let Ω be the set containing all the possible outcomes of a random experiment. Let us recall that a discrete Bayesian probability distribution $p : \Omega \to [0, 1]$ is defined in a probability space $(\Omega, 2^\Omega, \mathbb{P})$ such that for all $A \in 2^\Omega$,

$$\mathbb{P}(A) = \sum_{\omega \in A} p(\omega),$$

where 2^Ω is the power set of Ω, i.e. the set of all the subsets of Ω. In addition, we have $\mathbb{P}(\Omega) = 1$.

A belief assignment in Dempster-Shafer Theory (DST) [24, 30] is a mass function $m : 2^\Omega \to [0, 1]$, defined in a probability space $(2^\Omega, 2^{2^\Omega}, \mathbb{P})$ such that for all $A \subseteq 2^\Omega$,

$$\mathbb{P}(A) = \sum_{B \in A} m(B)$$

In addition, we have $\mathbb{P}(2^\Omega) = 1$. As we can see, this mass function assigns probabilities to sets of outcomes, instead of direct outcomes. The probability mass $m(B)$, for $B \in 2^\Omega$, can be interpreted as the probability that the observer has no evidence specific enough to determine which of the outcomes in B is actually occurring, while having enough to eliminate all other outcomes, i.e. the elements of $\overline{B}$. So, when no observation has yet been made, i.e. in case of total ignorance about a phenomenon, an observer would describe its knowledge as $m(\Omega) = 1$. Conversely, when an observer has gathered enough evidence $\mathcal{E}$ to eliminate with total certainty

all possible outcomes but one, one would describe its knowledge as $m(\{\omega\}|\mathcal{E}) = 1$, for some $\omega \in \Omega$. In particular, when $\mathbb{P}(\{\{\omega\} \ / \ \omega \in \Omega\}) = 1$, only direct observations are possible, which means that m represents probabilities of occurrence for the outcomes of Ω. Thus, m degenerates to the Bayesian probability distribution p in this case. This is why DST can be viewed as a generalization of Bayesian probability theory. This special case is referred to as a Bayesian belief assignment.

Example 1.2.0.1. For instance, think of the classic fair coin toss experiment, but with a slight modification. Say $\Omega = \{heads, tails, side\}$ and say we have a perfect laser sensor placed horizontally on a table that is only able to detect objects at a height that is just superior to the thickness of a coin. If we toss a coin so that it lands on this table, the sensor would only be able to tell us if it detects something or not, i.e. if the coin is on its side or not. Thus, assuming that we can only observe the result of this experiment through this sensor, we have $m(\{heads, tails\}|\overline{detection}) = 1$ and $m(\{side\}|detection) = 1$. This means that we have strong evidence that the coin is on its side if the sensor detects something, but weaker evidence of *heads* and *tails* if there is no detection. In Bayesian probability theory, the same conditions give us $p(heads|\overline{detection}) = 0.5$, $p(tails|\overline{detection}) = 0.5$ and $p(side|detection) = 1$. In other words, Bayesian probabilities describe the likelihood of an outcome, while belief assignments describe the likelihood that the observer is in some configuration of discernment regarding the outcome.

Example 1.2.0.2. Now, let us say that this sensor is not perfect. Let the probability that this sensor does not detect the coin on its side be $p(side|\ \overline{detection}) = 0.2$, the rest being equally distributed on *heads* and *tails* as before, i.e. $p(heads|\ \overline{detection}) = 0.4$, $p(tails|\ \overline{detection}) = 0.4$. Also, let the probability that this sensor does correctly detect the coin on its side be $p(side|\ detection) = 0.9$, the rest being equally distributed on *heads* and *tails* as before, i.e. $p(heads|\ detection) = 0.05$, $p(tails|\ detection) = 0.05$. This is translated in DST in terms of probabilities that the evidence we gathered does not help in discerning any outcome:
$m(\Omega|\ \overline{detection}) = 0.2$, $m(\{heads, tails\}|\ \overline{detection}) = 0.8$ and $m(\{side\}|\ detection) = 0.9$, $m(\Omega|\ detection) = 0.1$.

1.2.1 Information fusion

The upset $\uparrow C = \{B \in 2^\Omega \ / \ B \supseteq C\}$ contains all sets that contain C, for some $C \in 2^\Omega$. Thus, $\mathbb{P}(\uparrow C)$ is the probability that the observer has no evidence specific enough to determine which of the outcomes in C is actually occurring, no matter what can be said of the other outcomes. This probability is known as a *commonality* in DST. The function $q : 2^\Omega \to [0, 1]$ is known as the *commonality function*, defined for any set $C \in 2^\Omega$ as:

$$q(C) = \mathbb{P}(\uparrow C) = \sum_{B \supseteq C} m(B) \tag{1.1}$$

Dempster's fusion rule [30] combines two belief assignments into one by operating a Bayesian fusion on commonalities. This means that the resulting belief assignment reflects the posterior specificity of evidence when both sources accurately report the truth, i.e. without them incorrectly evaluating their observations and uncertainties. Directly applying a Bayesian fusion on the mass function m would only give us the resulting probability of each piece of evidence, which is useless and has little meaning. Indeed, a belief assignment represents the discernment of an observer. Doing so, m_1 represents the discernment of source 1, while m_2 represents the discernment of source 2. We do not want to know which common configurations of discernment are the most probable between these two sources. We want instead to create a third observer that would observe sources 1 and 2, combining their pieces of evidence to eventually eliminate some possible outcomes and so deducing the actual resulting probable configurations of discernment.

So, for two sources 1 and 2, Dempster's rule can be defined as:

$$\begin{aligned}\mathbb{P}_{12}(\uparrow C) &\propto \mathbb{P}_1(\uparrow C).\mathbb{P}_2(\uparrow C)\\ &\propto q_1(C).q_2(C)\\ &\propto \sum_{A\supseteq C} m_1(A).\sum_{B\supseteq C} m_2(B)\\ &\propto \sum_{A\cap B\supseteq C} m_1(A).m_2(B)\end{aligned}$$

The commonality function is reversible. As we will see in the next chapter, it is actually the zeta transform of m in $(2^\Omega, \supseteq)$. We have:

$$m_{12}(C) \propto \sum_{A\cap B=C} m_1(A).m_2(B)$$

When $\propto$ is replaced by $=$, i.e. when no normalization is done, we get the conjunctive fusion rule [38]. In Dempster's fusion rule however, we consider that we cannot assign any mass to the elimination of all possible outcomes, i.e. no mass on $\emptyset$. So, provided $m_{12}(\emptyset) < 1$, Dempster's fusion rule is:

$$m_{12}(C) = \frac{\sum_{A\cap B=C} m_1(A).m_2(B)}{1 - m_{12}(\emptyset)} \tag{1.2}$$

In particular, since all singletons are always mutually exclusive, it is easy to see that when $\mathbb{P}(\{\{\omega\} \,/\, \omega \in \Omega\}) = 1$, i.e. for two Bayesian belief assignments, the only sets A and B verifying $A \cap B = C$ are both equal to C itself. Thus, for two Bayesian belief assignments, Dempster's fusion rule is a pointwise multiplication, followed by a renormalization, i.e. a classic Bayesian fusion. Therefore, Dempster's fusion can be viewed as a generalization of Bayesian fusion.

Example 1.2.1.1. For example, let us say instead that $\Omega = \{heads, tails\}$ and that we have two independent sensors observing this coin laying flat on the table. Let both these sensors have a probability of 0.7 of correctly detecting a particular face. Let sensor 1 detect *heads*, while sensor 2 detects *tails*. In Bayesian probabilities, we would have $p(heads|S_1 = heads) = 0.7$, $p(tails|S_1 = heads) = 0.3$ and $p(heads|S_2 = tails) = 0.3$, $p(tails|S_2 = tails) = 0.7$, where S_i indicates the evidence gathered by sensor i. Notice that in this case, probabilities are more specific since they are not equiprobable. Yet, fusing these two measurements would give us $p(heads|S_1 = heads, S_2 = tails) = 0.5$ and $p(tails|S_1 = heads, S_2 = tails) = 0.5$, which makes sense but is indistinguishable from the probabilities we had when no observation regarding *heads* or *tails* was made, i.e. $p(heads)$ and $p(tails)$.

On the other hand, in DST, we would have $m(\{heads\}|S_1 = heads) = 0.7$, $m(\Omega|S_1 = heads) = 0.3$ and $m(\{tails\}|S_2 = tails) = 0.7$, $m(\Omega|S_2 = tails) = 0.3$. The conjunctive fusion of these mass functions in DST gives us $m(\Omega|S_1 = heads, S_2 = tails) = 0.09$, $m(\{heads\}|S_1 = heads, S_2 = tails) = 0.12$, $m(\{tails\}|S_1 = heads, S_2 = tails) = 0.12$ and $m(\emptyset|S_1 = heads, S_2 = tails) = 0.67$. This last mass is the probability that the observer has enough evidence to eliminate all possible outcomes of Ω. It is interpreted as a measure of conflict among sources and an indicator that some sources tend to rule out more possible outcomes than they should (overconfidence). It may also be a sign that Ω does not contain all possible outcomes. Normalizing this resulting mass function to ignore this conflict as in Dempster's rule, we would nonetheless get $m(\Omega|S_1 = heads, S_2 = tails) = 0.272$, $m(\{heads\}|S_1 = heads, S_2 = tails) = 0.364$ and $m(\{tails\}|S_1 = heads, S_2 = tails) = 0.364$, which still accounts for the fact that we gathered direct evidence of *heads* and *tails*. This information may be used in subsequent information fusions and is useful in

itself for decision making. For example, in our collaborative perception setting, it is important to know if we are uncertain about the outcome *because* we have conflicting pieces of evidence or *because* we do not have any evidence at all. Indeed, having at least an idea of the outcome is better than nothing, meaning that it is more urgent / valuable to acquire information about unobserved areas than to refine uncertainties on observed ones. This is all the more true that the ego-vehicle will probably itself observe these conflicting areas later in time, without needing external help.

1.2.2 Information fusion of unreliable sources

The down set $\downarrow C = \{B \in 2^\Omega \ / \ B \subseteq C\}$ contains all sets that C contains, for some $C \in 2^\Omega$. Thus, $\mathbb{P}(\downarrow C)$ is the probability that the observer has no evidence supporting other outcomes than the ones in C. This probability is known as an *implicability* in DST (See [39]). The function $b : 2^\Omega \to [0, 1]$ is known as the *implicability function*, defined for any set $C \in 2^\Omega$ as:

$$b(C) = \mathbb{P}(\downarrow C) = \sum_{B \subseteq C} m(B) \tag{1.3}$$

The disjunctive fusion rule [27] combines two belief assignments into one by operating a Bayesian fusion on implicabilities. This means that the resulting belief assignment reflects the maximum specificity of evidence on which both sources agree. Therefore, it provides a way to fuse two belief assignments that may not be entirely accurate, i.e. belief assignments that might eliminate more possible outcomes than the corresponding observations actually allow to. Indeed, in this case, we want to be careful with pieces of evidence that eliminate some possible outcomes. We do not want to set any possible outcome aside unless both sources agree on it. Formally, for two sources 1 and 2, the disjunctive fusion rule is defined as:

$$\begin{aligned}
\mathbb{P}_{12}(\downarrow C) &= \mathbb{P}_1(\downarrow C).\mathbb{P}_2(\downarrow C) \\
&= b_1(C).b_2(C) \\
&= \sum_{A \subseteq C} m_1(A). \sum_{B \subseteq C} m_2(B) \\
&= \sum_{A \cup B \subseteq C} m_1(A).m_2(B)
\end{aligned}$$

The implicability function is reversible. As we will see in the next chapter, it is actually the zeta transform of m in $(2^\Omega, \subseteq)$. We have:

$$m_{12}(C) = \sum_{A \cup B = C} m_1(A).m_2(B)$$

While the conjunctive fusion corresponds to the intersection of the statements made by each source, this disjunctive fusion corresponds to the union of the statements made by each source.

Example 1.2.2.1. Taking back Example 1.2.1.1, where $\Omega = \{heads, tails\}$ and where we have two independent sensors observing a coin laying flat on a table. Both these sensors have a probability of 0.7 of correctly detecting a particular face. Sensor 1 detects *heads*, while sensor 2 detects *tails*. In Bayesian probabilities, we have $p(heads|S_1 = heads) = 0.7$, $p(tails|S_1 = heads) = 0.3$ and $p(heads|S_2 = tails) = 0.3$, $p(tails|S_2 = tails) = 0.7$, where S_i indicates the evidence gathered by sensor i. Fusing these two measurements gives us $p(heads|S_1 = heads, S_2 = tails) = 0.5$ and $p(tails|S_1 = heads, S_2 = tails) = 0.5$.

On the other hand, in DST, we have $m(\{heads\}|S_1 = heads) = 0.7$, $m(\Omega|S_1 = heads) = 0.3$ and $m(\{tails\}|S_2 = tails) = 0.7$, $m(\Omega|S_2 = tails) = 0.3$. The disjunctive fusion of these mass functions in DST gives us $m(\Omega|S_1 = heads, S_2 = tails) = 1$.

1.3 Link between Dempster-Shafer Theory and the Möbius inversion formula

Let $(P, \leq)$ be a finite set P partially ordered by some binary operator noted $\leq$.

Example 1.3.0.1. The partially ordered set $(2^\Omega, \subseteq)$ is the set containing all subsets of a set Ω ordered by inclusion. Let $\Omega = \{a, b, c\}$. We have, for instance, $\emptyset \subseteq \{a\} \subseteq \{a, b\} \subseteq \Omega$. However, we have $\{a\} \not\subseteq \{b\}$ and $\{b\} \not\subseteq \{a\}$. The operator $\subseteq$ does not apply to the pairs $(\{a\}, \{b\})$ and $(\{b\}, \{a\})$. There is no such order between these two elements of 2^Ω.

Example 1.3.0.2. The partially ordered set $(\mathbb{N}^*, |)$ is the set containing all positive integers ordered by divisibility. For instance, as 3 divides 6, we have $3|6$. However, as 3 does not divide 5, the operator $|$ does not apply to the pair $(3, 5)$.

Example 1.3.0.3. The partially ordered set $(\mathbb{Z}, \leq)$ is the set containing all integers ordered by magnitude. In this trivial case, all pairs of elements of $\mathbb{Z}$ are ordered by $\leq$. It is said that $(\mathbb{Z}, \leq)$ is a *totally* ordered set. Totally ordered sets are also called *chains*.

1.3.1 Zeta transform ("Discrete integral")

The zeta transform $g : P \to \mathbb{R}$ of a function $f : P \to \mathbb{R}$ is defined as follows:

$$\forall y \in P, \quad g(y) = \sum_{x \leq y} f(x) \tag{1.4}$$

It is **analogous to integration** in a discrete domain. Its name comes from incidence algebra [40], in which it corresponds to the multiplication of $f : P^2 \to \mathbb{R}$ with the *zeta function* $\zeta : P^2 \to \{0, 1\}$, such that $\zeta(x, y) = 1$ if $x \leq y$, i.e.

$$\forall a, b \in P, \quad (f * \zeta)(a, b) = \sum_{a \leq x \leq b} f(a, x).\zeta(x, b) = \sum_{a \leq x \leq b} f(a, x) = g(a, b)$$

Example 1.3.1.1. In DST, the implicability function b is defined as the zeta transform of the mass function m in $(2^\Omega, \subseteq)$, i.e. $\forall y \in 2^\Omega, \quad b(y) = \sum_{x \subseteq y} m(x)$.

Example 1.3.1.2. In DST, the commonality function q is defined as the zeta transform of the mass function m in $(2^\Omega, \supseteq)$, i.e. $\forall y \in 2^\Omega, \quad q(y) = \sum_{x \supseteq y} m(x)$.

1.3.2 Möbius transform ("Discrete derivative")

The Möbius *transform* must not be confused with the Möbius *transformations* that exist in the field of Geometry. Here, we work in the context of Order theory. The Möbius transform of g is f. It is **analogous to differentiation** in a discrete domain and is defined by the *Möbius inversion formula*:

$$\forall y \in P, \quad f(y) = \sum_{x \leq y} g(x).\mu_{P,\leq}(x, y) \tag{1.5}$$

where $\mu_{P,\leq}$ is the Möbius function of $(P, \leq)$, defined in its general form in [40] as follows:

$$\forall x, y \in P, \quad \sum_{x \leq z \leq y} \mu_{P,\leq}(x, z) = \sum_{x \leq z \leq y} \mu_{P,\leq}(z, y) = 0, \tag{1.6}$$

with $\mu_{P,\leq}(x,x) = 1$. This can be rewritten in the following recursive form:

$$\forall x, y \in P, \quad \mu_{P,\leq}(x,y) = \begin{cases} 1 & \text{if } x = y \\ -\sum_{x<z\leq y} \mu_{P,\leq}(z,y) & \text{otherwise} \end{cases} \tag{1.7}$$

In an incidence algebra, the Möbius transform of g corresponds to the multiplication of $g : P^2 \to \mathbb{R}$ with the Möbius function μ, i.e.

$$\forall a, b \in P, \quad (g * \mu_{P,\leq})(a,b) = \sum_{a\leq x\leq b} g(a,x).\mu_{P,\leq}(x,b) = f(a,b)$$

Example 1.3.2.1. Taking back the functions m and b from Example 1.3.1.1, m is the Möbius transform of b in $(2^\Omega, \subseteq)$, i.e. $\forall y \in 2^\Omega, \quad m(y) = \sum_{x\subseteq y} b(x) \ . \ (-1)^{|y|-|x|}$.

Example 1.3.2.2. Taking back the functions m and q from Example 1.3.1.2, m is the Möbius transform of q in $(2^\Omega, \supseteq)$, i.e. $\forall y \in 2^\Omega, \quad m(y) = \sum_{x\supseteq y} q(x) \ . \ (-1)^{|y|-|x|}$.

1.3.3 Multiplicative Möbius inversion theorem

There is also a multiplicative version of the zeta and Möbius transforms, with the same properties, in which the sum is replaced by a product:

$$\forall y \in P, \quad g(y) = \prod_{x\leq y} f(x) \quad \Leftrightarrow \quad f(y) = \prod_{x\leq y} g(x)^{\mu_{P,\leq}(x,y)}$$

Example 1.3.3.1. In DST, the disjunctive weight function v (introduced thereafter in section 1.4.2) is defined as the inverse of the multiplicative Möbius transform of b from Example 1.3.1.1 in $(2^\Omega, \subseteq)$, i.e. $\forall y \in 2^\Omega, \quad v(y) = \prod_{x\subseteq y} b(x)^{-(-1)^{|y|-|x|}}$.

Example 1.3.3.2. In DST, the conjunctive weight function w (introduced thereafter in section 1.4.1) is defined as the inverse of the multiplicative Möbius transform of q from Example 1.3.1.2 in $(2^\Omega, \supseteq)$, i.e. $\forall y \in 2^\Omega, \quad w(y) = \prod_{x\supseteq y} q(x)^{-(-1)^{|y|-|x|}}$.

1.4 Two fusion rules avoiding data incest in Dempster-Shafer Theory

1.4.1 The Cautious fusion rule

The Cautious fusion rule [23] is a heuristic for combining two belief assignments in a conjunctive way when these assignments may not be independent from each other. It is reminiscent of the Covariance Intersection filter [35] in that they both avoid data incest in a sub-optimal but guaranteed way, ignoring the actual dependency existing between two sources. It relies on a decomposition of belief assignments into atomic belief assignments (one for each piece of evidence) called *simple belief assignments*, introduced in its general form in [41].

The conjunctive decomposition of evidence

Let us say that we want to decompose m into the conjunctive fusion of $2^{|\Omega|}-1$ observers that each accounts for a single piece of evidence $A \subset \Omega$, i.e. a single configuration of discernment regarding the outcome that eliminates at least one possible outcome. This means that we would have:

$$\mathbb{P}(\uparrow B) = \prod_{A \subset \Omega} \mathbb{P}_A(\uparrow B) = q(B) = \prod_{A \subset \Omega} q_A(B)$$

In particular, this means that for $B = \Omega$, we get:

$$q(\Omega) = \prod_{A \subset \Omega} q_A(\Omega) = \prod_{A \subset \Omega} m_A(\Omega)$$

Since we may have $q(\Omega) > 0$, this means that for any $A \subset \Omega$, we may have $m_A(\Omega) > 0$. This makes sense in this decomposition of evidence since the observer may not be certain to even have a piece of evidence A, as seen in Example 1.2.0.2. Thus, each belief assignment m_A is defined for any $B \in 2^\Omega$ as:

$$m_A(B) = \begin{cases} w(A) & \text{if } B = \Omega \\ 1 - w(A) & \text{if } B = A \\ 0 & \text{otherwise} \end{cases}$$

where $w : 2^\Omega \to \mathbb{R}^+$, with $w(A) \in [0,1]$ for all $A \subset \Omega$, represents the level of uncertainty each observer has regarding its piece of evidence A. The function w is called the *conjunctive weight function*. Consequently, we have:

$$\mathbb{P}_A(\uparrow B) = q_A(B) = \begin{cases} w(A) & \text{if } B \not\subseteq A \\ 1 & \text{if } B \subseteq A \end{cases}$$

which means that for any $B \in 2^\Omega$:

$$\begin{aligned} q(B) &= \prod_{A \subset \Omega} q_A(B) \\ &= \prod_{A \not\supseteq B} w(A) \\ &= \prod_{A \subset \Omega} w(A). \prod_{\substack{A \subset \Omega \\ A \supseteq B}} w(A)^{-1} \\ &= \prod_{A \subset \Omega} m_A(\Omega). \prod_{\substack{A \subset \Omega \\ A \supseteq B}} w(A)^{-1} \\ &= q(\Omega). \prod_{\substack{A \subset \Omega \\ A \supseteq B}} w(A)^{-1} \end{aligned}$$

Let us assume that $q(\Omega) > 0$ and set $w(\Omega) = q(\Omega)^{-1}$, we finally get:

$$q(B) = \prod_{A \supseteq B} w(A)^{-1} \tag{1.8}$$

As we have seen in the previous subsection about information fusion, the commonality function q is the zeta transform of m in $(2^\Omega, \supseteq)$. In fact, there is a very similar transform that uses

the multiplication operator instead of the addition operator, namely the multiplicative zeta transform. The multiplicative zeta transform is a reversible transformation in the same way that q is relatively to m. Thus, there exists a function $u : 2^\Omega \to \mathbb{R}^+$ such that for any set $B \in 2^\Omega$,

$$q(B) = \prod_{A \supseteq B} u(A)$$

Therefore, when $q(\Omega) > 0$, the commonality function q is the inverse multiplicative zeta transform of w in $(2^\Omega, \supseteq)$. This means that w is unique and so does this decomposition of evidence. When $w(A) > 1$ for some $A \subset \Omega$, the function m_A is no longer a belief assignment if we still consider that $\mathbb{P}_A(\uparrow B)$ is fused with the rest of the decomposition, for any $B \in 2^\Omega$. However, if we consider that $\mathbb{P}_A(\uparrow B)$ is "un-fused", i.e. that it divides the rest of the decomposition, then m_A is defined by $w(A)^{-1}$. Since $w(A) > 1$, its inverse is in $[0, 1]$, which means that m_A is a belief assignment. Then, this belief assignment represents counter-evidence for A and is called an *inverse belief assignment* [41]. Thus, the general form of this decomposition of evidence actually corresponds to a sequence of conjunctive combinations and de-combinations. The Möbius inversion theorem [40] gives us for all $B \in 2^\Omega$:

$$w(B) = \prod_{A \supseteq B} q(A)^{(-1)^{|A|-|B|+1}} \tag{1.9}$$

Fusion of two conjunctive decompositions

As presented in an earlier subsection, the conjunctive fusion of two belief assignments m_1 and m_2 is defined by:

$$\begin{aligned}
\mathbb{P}_{12}(\uparrow B) &\propto \mathbb{P}_1(\uparrow B).\mathbb{P}_2(\uparrow B) \\
&\propto q_1(B).q_2(B) \\
&\propto \prod_{A \supseteq B} w_1(A)^{-1}. \prod_{A \supseteq B} w_2(A)^{-1} \\
&\propto \prod_{A \supseteq B} [w_1(A).w_2(A)]^{-1}
\end{aligned}$$

Thus, both the conjunctive fusion and Dempster's fusion are defined in the conjunctive weight space by a simple pointwise product, as in the commonality space.

The Cautious rule

The idea behind this heuristic fusion rule is quite simple: acknowledge the probable existence of pieces of evidence but do not give them more probability of occurrence than your most affirmative source does. For this, the conjunctive decomposition is key. Indeed, it gives a level of uncertainty $w(A)$ for each probable piece of evidence $A \subset \Omega$ associated with a belief assignment. Thus, when cautiously fusing two belief assignments m_1 and m_2, all we have to do is to compute their weight functions w_1 and w_2 with Eq. (1.9), to take the pointwise minimum and finally to reverse the transformation with Eq. (1.8). Formally, for two weight functions w_1 and w_2, the cautious fusion rule yields a weight function w_{12} as follows, for any set $B \in 2^\Omega$:

$$w_{12}(B) = w_1(B) \wedge w_2(B) \tag{1.10}$$

where $\wedge$ is the infimum operator, i.e. the minimum in $\mathbb{R}$. As we can see, if source 1 is more certain about some piece of evidence than source 2, then their fusion will be more certain about

this piece of evidence than source 2 but no more than source 1. Doing so, if we try to fuse source 1 with itself, we get the exact same belief assignment. This yields the same result, however, if we fuse source 1 with another independent source than has the exact same belief assignment, though it should instead yield of a more certain belief assignment. It ignores actual dependencies by always considering them maximal. This is why it is sub-optimal, but guarantees consistency.

It is obvious that w_{12} is a valid weight function corresponding to a belief assignment m_{12} when both w_1 and w_2 range in $[0, 1]$. It has the intuitive meaning given so far. Yet, this interpretation collapses when w_1 or w_2 range in $(1, +\infty]$. When one ranges in $[0, 1]$ but not the other, the cautious rule favorites positive evidence over counter evidence and yields a weight that is less than the one yielded by the usual conjunctive fusion. In this case, it is conservative as it does not let probable counter evidence increase the level of uncertainty on a probable piece of evidence. When both range in $(1, +\infty]$, the cautious rule takes the greatest level of uncertainty on counter evidence.

Overall, this behavior guarantees that w_{12} corresponds to a belief assignment m_{12}. Indeed, let us rewrite Eq. (1.10) as follows:

$$w_{12}(B) = w_1(B).\frac{w_1(B) \wedge w_2(B)}{w_1(B)}$$

We have $w_1(B) \wedge w_2(B) \leq w_1(B)$ for any $B \in 2^\Omega$. This means that $\frac{w_1(B) \wedge w_2(B)}{w_1(B)} \in [0, 1]$, which implies that w_{12} represents the conjunctive fusion of w_1 with the sequence of conjunctive combinations defined by the weight function $\frac{w_1 \wedge w_2}{w_1}$. As this necessarily yields a weight function that corresponds to a valid belief assignment, we have that the cautious rule is always valid, even with weights greater than 1.

1.4.2 The Bold fusion rule

The Bold fusion rule [23] is a heuristic for combining two belief assignments in a disjunctive way when these assignments may not be independent from each other. It is the dual of the Cautious fusion rule, sub-optimally guaranteeing consistency with the disjunctive fusion rule instead of the conjunctive fusion rule. It relies on the dual of the conjunctive decomposition of evidence, introduced in [23].

The disjunctive decomposition of evidence

Let us say that we want to decompose m into the disjunctive fusion of $2^{|\Omega|} - 1$ unreliable observers that each accounts for a single piece of evidence $A \supset \emptyset$, i.e. a single configuration of discernment that supports at least one possible outcome. This means that we would have:

$$\mathbb{P}(\downarrow B) = \prod_{A \supset \emptyset} \mathbb{P}_A(\downarrow B) = b(B) = \prod_{A \supset \emptyset} b_A(B)$$

In particular, this means that for $B = \emptyset$, we get:

$$b(\emptyset) = \prod_{A \supset \emptyset} b_A(\emptyset) = \prod_{A \supset \emptyset} m_A(\emptyset)$$

Since we may have $b(\emptyset) > 0$, this means that for any $A \supset \emptyset$, we may have $m_A(\emptyset) > 0$. This makes sense in this decomposition of evidence since the observer may not be confident enough to commit to any possible configuration of discernment. Thus, each belief assignment m_A is

defined for any $B \in 2^\Omega$ as:

$$m_A(B) = \begin{cases} v(A) & \text{if } B = \emptyset \\ 1 - v(A) & \text{if } B = A \\ 0 & \text{otherwise} \end{cases}$$

where $v : 2^\Omega \to \mathbb{R}^+$, with $v(A) \in [0, 1]$ for all $A \supset \emptyset$, represents the level of uncertainty each observer has regarding its own capacity to correctly evaluating the specificity of its observations as the piece of evidence A. Here each observer is uncertain about itself, about its mapping between observations and belief assignment. This is different from the conjunctive decomposition, in which each observer is uncertain about the reliability of the observations it receives (e.g. risk of false detection). The function v is called the *disjunctive weight function*. Consequently, we have:

$$\mathbb{P}_A(\downarrow B) = b_A(B) = \begin{cases} v(A) & \text{if } B \not\supseteq A \\ 1 & \text{if } B \supseteq A \end{cases}$$

which means that for any $B \in 2^\Omega$:

$$\begin{aligned} b(B) &= \prod_{A \supset \emptyset} b_A(B) \\ &= \prod_{A \not\subseteq B} v(A) \\ &= \prod_{A \supset \emptyset} v(A). \prod_{\substack{A \supset \emptyset \\ A \subseteq B}} v(A)^{-1} \\ &= \prod_{A \supset \emptyset} m_A(\emptyset). \prod_{\substack{A \supset \emptyset \\ A \subseteq B}} v(A)^{-1} \\ &= b(\emptyset). \prod_{\substack{A \supset \emptyset \\ A \subseteq B}} v(A)^{-1} \end{aligned}$$

Let us assume that $b(\emptyset) > 0$ and set $v(\emptyset) = b(\emptyset)^{-1}$, we finally get:

$$b(B) = \prod_{A \subseteq B} v(A)^{-1} \tag{1.11}$$

There exists a function $u : 2^\Omega \to \mathbb{R}^+$ such that for any set $B \in 2^\Omega$,

$$b(B) = \prod_{A \subseteq B} u(A)$$

Therefore, when $b(\emptyset) > 0$, the implicability function b is the inverse multiplicative zeta transform of v in $(2^\Omega, \subseteq)$. This means that v is unique and so does this decomposition of evidence. When $v(A) > 1$ for some $A \supset \emptyset$, the function m_A is no longer a belief assignment if we still consider that $\mathbb{P}_A(\downarrow B)$ is fused with the rest of the decomposition, for any $B \in 2^\Omega$. Thus, the general form of this decomposition of evidence actually corresponds to a sequence of disjunctive combinations and de-combinations. The Möbius inversion theorem [40] gives us for all $B \in 2^\Omega$:

$$v(B) = \prod_{A \subseteq B} b(A)^{(-1)^{|A|-|B|+1}} \tag{1.12}$$

Fusion of two disjunctive decompositions

As presented in an earlier subsection, the disjunctive fusion of two belief assignments m_1 and m_2 is defined by:

$$\begin{aligned}\mathbb{P}_{12}(\downarrow B) &= \mathbb{P}_1(\downarrow B).\mathbb{P}_2(\downarrow B) \\ &= b_1(B).b_2(B) \\ &= \prod_{A\subseteq B} v_1(A)^{-1}. \prod_{A\subseteq B} v_2(A)^{-1} \\ &= \prod_{A\subseteq B} [v_1(A).v_2(A)]^{-1}\end{aligned}$$

Thus, the disjunctive fusion is defined in the disjunctive weight space by a simple pointwise product, as in the implicability space.

The Bold rule

The idea behind this heuristic fusion rule is very similar to the Cautious rule: acknowledge the probable existence of common eliminations of some outcomes but do not give these configurations more probability of occurrence than your most confident source does. For this, the disjunctive decomposition is key. Indeed, for each probable piece of evidence $A \supset \emptyset$ associated with a belief assignment, it gives a level of uncertainty $v(A)$ that the observation indeed eliminates the outcomes in $\overline{A}$. Thus, when fusing two belief assignments m_1 and m_2 with the Bold rule, all we have to do is to compute their weight functions v_1 and v_2 with Eq. (1.12), to take the pointwise minimum and finally to reverse the transformation with Eq. (1.11). Formally, for two weight functions v_1 and v_2, the Bold fusion rule yields a weight function v_{12} as follows, for any set $B \in 2^\Omega$:

$$v_{12}(B) = v_1(B) \wedge v_2(B) \tag{1.13}$$

where $\wedge$ is the infimum operator, i.e. the minimum in $\mathbb{R}$. As we can see, if source 1 is more confident about some piece of evidence than source 2, then their fusion will be more confident about this piece of evidence than source 2 but no more than source 1. Doing so, if we try to fuse source 1 with itself, we get the exact same belief assignment. This yields the same result, however, if we fuse source 1 with another independent source than has the exact same belief assignment, though it should instead yield of a more confident belief assignment. It ignores actual dependencies by always considering them maximal. This is why it is sub-optimal, but guarantees consistency, as the Cautious rule does.

Treating uncertainties in the same way, the same behavior as the Cautious rule is displayed by the Bold rule regarding counter-evidence. Consequently, rewriting Eq. (1.13), we get the same property:

$$v_{12}(B) = v_1(B).\frac{v_1(B) \wedge v_2(B)}{v_1(B)}$$

We have $v_1(B) \wedge v_2(B) \leq v_1(B)$ for any $B \in 2^\Omega$. This means that $\frac{v_1(B)\wedge v_2(B)}{v_1(B)} \in [0, 1]$, which implies that v_{12} represents the disjunctive fusion of v_1 with the sequence of disjunctive combinations defined by the weight function $\frac{v_1\wedge v_2}{v_1}$. As this necessarily yields a weight function that corresponds to a valid belief assignment, we have that the bold rule is always valid, even with weights greater than 1.

1.4.3 Limitations

The Cautious and Bold fusion rules feature interesting properties. However, it should be noted that they only apply to respectively conjunctive and disjunctive weight functions. To obtain these weight functions from belief assignments, one has to pass by Eq. (1.1) or (1.3) and Eq. (1.9) or (1.12), which all have exponential complexities in time. While there are ways to alleviate this burden for Eq. (1.1) and (1.3), there was none to alleviate the ones on Eq. (1.9) and (1.12) in general. Furthermore, if all outcomes of Ω may actually occur, it was impossible to compute the conjunctive weight function of a belief assignment m that would not place belief on Ω, i.e. $m(\Omega) = 0$. Conversely, it was impossible to compute the disjunctive weight function of a belief assignment m that would not place belief on $\emptyset$, i.e. $m(\emptyset) = 0$. Therefore, unless an approximation of these belief assignments was done to make $m(\Omega) > 0$ or $m(\emptyset) > 0$, fusing them with another belief assignment with the Cautious or Bold rule was not possible. This approximation, aside the small bias it adds, worsens the situation with computations in Eq. (1.1) and (1.9). Indeed, to alleviate the computational burden on these equations, one has to rely on *focal sets*, i.e. sets B verifying $m(B) > 0$. The issue is that the conjunctive fusion of belief assignments m_1 and m_2 generates new focal sets $A \cap B$, where $m_1(A) > 0$ and $m_2(B) > 0$. While other focal sets may disappear if there is no superset to support them in the other belief assignment, this is impossible if both m_1 and m_2 have Ω as focal set, since Ω is of course a superset for all of its subsets. Doing so, fusing in a conjunctive manner, as does the Cautious rule, can only create more and more focal sets to consider in Eq. (1.1) and (1.9) if Ω is a focal set for all belief assignments. The same phenomenon occurs with the Bold fusion rule when $m(\emptyset) > 0$, since $\emptyset$ is obviously contained in all sets.

1.5 Efficiently computing the conjunctive and disjunctive decompositions

This section contains the translation (and augmentation) of our first publication [42]. It concerns the computation of Eq. (1.9) and (1.12) in the general case.

1.5.1 Introduction

Dempster-Shafer Theory (DST) [24] is an elegant formalism that generalizes Bayesian probability theory. It is more expressive than the latter because it allows an observer to represent its beliefs in the state of a variable of interest not only by assigning credit directly to a specific state (strong evidence) but also by assigning credit to any set of possible states (weaker evidence). More precisely, for Ω the set of all possible outcomes, belief may be assigned to any set of 2^Ω in DST. This belief assignment is called a *mass function* and provides meta-information quantifying the level of uncertainty of the observer itself, which is crucial for decision making.

Nevertheless, this information comes with a cost: considering $2^{|\Omega|}$ potential values may lead to temporally and spatially heavy algorithms. They can become difficult to use for more than a dozen possible states (e.g. 20 states in Ω generates about a million subsets), though one may need to consider more possible states (e.g. in classification or identification tasks). This complexity is even more limiting for real time applications. To tackle this issue, a lot of work has been done to reduce the complexity of computations used in information fusion with Dempster's rule [30]. We distinguish two main approaches that we refer to as *powerset based* and *evidence based*.

The *powerset based* approach encompasses all algorithm based on the structure of the lattice 2^Ω. They have a complexity dependent on $|\Omega|$. In the general case, the family of optimal

algorithms of this kind is based on the *Fast Möbius Transform* (FMT) [43]. Their complexity is $O(|\Omega|.2^{|\Omega|})$ in time and $O(2^{|\Omega|})$ in space.

The *evidence based* approach encompasses all algorithm that seeks to reduce computations to sets that carry information, namely *focal sets*, which are in general far less numerous than $2^{|\Omega|}$. This approach is often more efficient than the powerset based one since it only depends on the information contained in sources with a quadratic complexity at most. Doing so, it enables one to exploit the full potential of DST by letting one choose any set of outcomes Ω, no matter its size.

Furthermore, even if it is possible that this approach leads to situations in which the FMT is more efficient [44], these situations become less and less likely to occur as $|\Omega|$ grows. In addition, the evidence based approach directly benefits from the use of information approximation methods, some of which being very efficient [45]. Thus, this approach seems superior to the FMT in most cases, above all when $|\Omega|$ is large, where a method with exponential complexity like the FMT is simply intractable.

The conjunctive decomposition, also known as *canonical decomposition* [41], is an important belief representation in DST. In particular, it allowed for the definition of the *Cautious fusion rule* [23] and its generalizations [46]. This rule consistently combines belief assignments that may not be independent, which is often the case in real applications such as information sharing in a vehicular network [22]. Its dual, the disjunctive decomposition, has been proposed in [23] where it has been used to define the *Bold fusion rule*. This latter rule is useful when sources may not be independent and are not entirely reliable. More generally, the conjunctive and disjunctive decompositions are used to define infinite families of t-norm and uninorm based fusion rules [47]. They are also exploited in conflict analysis [48], clustering [49] and belief reinforcement/weakening [50].

Yet, few options are available regarding the computation of the conjunctive decomposition. [23] proposed linear evidence based methods that only works in two particular cases, namely the *consonant* and *quasi-Bayesian* cases. However, in the general case, the conjunctive and disjunctive decompositions cannot be simplified into mathematical expressions that would only feature focal sets. To the best of our knowledge, the only algorithms to have been proposed for this general computation are the FMT and matrix calculus [51], as suggested in [23, 41], the latter being less efficient both in time and space than the former. Thus, until now, all method based on these decompositions in the general case had to use the FMT, and so had a complexity at least exponential in time and space.

Here, we provide an evidence based method for the computation of the conjunctive and disjunctive decompositions in the general case. This paper is organized as follows : Section 1.5.2 introduces elements of DST on which we build our proposition. Section 1.5.3 presents our method for the conjunctive decomposition and mathematically demonstrates why its complexity scales with information instead of $|\Omega|$. Section 1.5.4 transposes this method to the computation of the disjunctive decomposition. We then conclude in section 1.5.5.

1.5.2 Preliminary definitions

Before diving into the details of our method, it is necessary to recall some notions around the conjunctive decomposition of evidence. The disjunctive decomposition simply being its dual, what is said here for the conjunctive decomposition can easily be transposed to the former. We assume the reader is already familiar with the notions of frame of discernment (i.e. Ω) and mass function.

Definition 1.5.2.1 (*Focal element*)**.** For any mass function m, a focal element (a.k.a focal set)

is a set A such that $m(A) \neq 0$. Sets that are not focal sets do not carry any information about m. In the following, we note $\mathcal{F}$ the set containing all focal sets of a mass function m.

Representations for a conjunctive fusion

Definition 1.5.2.2 (*Commonality function*)**.** For any mass function m, a commonality function q is defined as follows:

$$\forall A \subseteq \Omega, \quad q(A) = \sum_{B \supseteq A} m(B) = \sum_{\substack{B \supseteq A \\ B \in \mathcal{F}}} m(B) \tag{1.14}$$

It is important to notice that all subset of Ω may be associated with a commonality in $[0, 1]$, whether it is a focal set or not.

Remark. A procedure directly based on this formula would have a complexity between $O(|\mathcal{F}|)$ and $O(|\mathcal{F}|^2)$. Then, the mass function m can simply be retrieved from q with the same complexity by reversing the computation of (1.14) on focal sets only. However, this reverse computation assumes that focal sets are known, either because no modification has been done to q or because of the use of well known operators like Dempster's fusion operator $\bigoplus$ that is known to only create focal sets at the intersection of the focal sets of the two previous mass functions. This also assumes that m is always kept in memory or re-computed from q to compute new values of q on-the-fly. Otherwise, the reverse operation has to be performed by a general algorithm such as the FMT.

Definition 1.5.2.3 (*Conjunctive fusion rule*)**.** To fuse the information brought by two different sources 1 et 2, it is necessary to define a combination rule. The conjunctive fusion rule has been introduced by Smets in the Transferable Belief Model (TBM) [39], through the binary operator $\textcircled{\cap}$. It is designed to fuse two sources when both are considered reliable. The result is a conjunction of their statements. In commonality space, the conjunctive fusion rule is defined as follows:

$$\forall A \subseteq \Omega, \quad (q_1 \textcircled{\cap} q_2)(A) = q_1(A) \ . \ q_2(A)$$

Definition 1.5.2.4 (*Conjunctive decomposition*)**.** The conjunctive decomposition has been introduced in its general form by Smets [41]. It decomposes any *non-dogmatic* mass function, i.e. any mass function such that $\Omega \in \mathcal{F}$, into the conjunctive fusion of simple mass functions noted A^w, where $A \subset \Omega$ and w is the conjunctive weight function defined thereafter.

In commonality space, we have:

$$\forall A \subset \Omega, \quad A^w_q : \begin{cases} q(B) = 1 & \forall B \subseteq A \\ q(B) = w(A) & \forall B \not\subseteq A \end{cases}$$

The conjunctive decomposition of q is:

$$q = \textcircled{\cap}_{A \subset \Omega} A^w_q$$

We get:

$$\forall B \subseteq \Omega, \quad q(B) = \prod_{A \not\supseteq B} w(A) = q(\Omega). \prod_{\substack{A \subset \Omega \\ A \supseteq B}} w(A)^{-1} \tag{1.15}$$

In consequence, according to the Möbius inversion theorem recalled in [43], we have that for any mass function on 2^Ω, if Ω is a focal element (enables in particular to pose $w(\Omega)^{-1} = q(\Omega)$), then its conjunctive decomposition is defined by the following weight function w:

$$\forall B \subseteq \Omega, \quad w(B) = \prod_{A \supseteq B} q(A)^{(-1)^{|A|-|B|+1}} \tag{1.16}$$

We see that this weight function w is not directly based on m but on q, which does not possess any neutral value for non-focal elements. Moreover, replacing these commonalities by their equivalent sum of masses does not simply well in the general case.

Remark. All in all, three points prevent an evidence based computation:

- the product on all supersets of A in Eq. (1.15), for all $B \in 2^\Omega$,
- same problem in Eq. (1.16),
- the ignorance about focal sets in m if we modify w or q (e.g. through any combination rule other than Dempster's) to reverse Eq. (1.14).

If we could find a link between the focal elements of m and the ones of $w - 1$, we would be able to reduce the complexity of these transformations.

Representations for a disjunctive fusion (same remarks)

Definition 1.5.2.5 (*Implicability function*)**.** For any mass function m, an implicability function b is defined as follows:

$$\forall A \subseteq \Omega, \quad b(A) = \sum_{B \subseteq A} m(B) = \sum_{\substack{B \subseteq A \\ B \in \mathcal{F}}} m(B) \tag{1.17}$$

It is important to notice that all subset of Ω may be associated with an implicability in $[0, 1]$, whether it is a focal set or not.

Definition 1.5.2.6 (*Disjunctive fusion rule*)**.** The disjunctive fusion rule [27] in an alternative fusion rule that is specific to DST. It exploits the existence of sets of different cardinalities in order to reflect a lack of trust in the sources to be fused. It corresponds to the disjunction of their statements, e.g. if one indicates an outcome $\{\omega_1\}$, while the other points to another outcome $\{\omega_2\}$, the resulting statement will be $\{\omega_1, \omega_2\}$ instead of the conjunctive result $\emptyset$. In implicability space, the disjunctive fusion rule is defined as follows:

$$\forall A \subseteq \Omega, \quad (b_1 \textcircled{\cup} b_2)(A) = b_1(A) \; . \; b_2(A)$$

Definition 1.5.2.7 (*Disjunctive decomposition*)**.** The disjunctive decomposition has been introduced in [23]. It decomposes any *subnormal* mass function, i.e. any mass function such that $\emptyset \in \mathcal{F}$, into the disjunctive fusion of simple mass functions noted A_v, where $A \supset \emptyset$ and v is the disjunctive weight function defined thereafter.

In implicability space, we have:

$$\forall A \supset \emptyset, \quad A_v^b : \begin{cases} b(B) = 1 & \forall B \supseteq A \\ b(B) = v(A) & \forall B \not\supseteq A \end{cases}$$

The disjunctive decomposition of b is:

$$b = \textcircled{\cup}_{A \supset \emptyset} A_v^b$$

We get:

$$\forall B \subseteq \Omega, \quad b(B) = \prod_{A \not\supseteq B} v(A) = b(\emptyset). \prod_{\substack{A \supseteq \emptyset \\ A \subseteq B}} v(A)^{-1} \tag{1.18}$$

In consequence, according to the Möbius inversion theorem, we have that for any mass function on 2^Ω, if $\emptyset$ is a focal element (enables in particular to pose $v(\emptyset)^{-1} = b(\emptyset)$), then its disjunctive decomposition is defined by the following weight function v:

$$\forall B \subseteq \Omega, \quad v(B) = \prod_{A \subseteq B} b(A)^{(-1)^{|A|-|B|+1}} \tag{1.19}$$

1.5.3 Evidence based computation of the conjunctive decomposition

In this section, we detail our evidence based method for the computation of the conjunctive decomposition. It exploits on the notion of *focal point* that we introduce here, a notion derived from the one of focal set and that contains it. The dual of our method computing the disjunctive decomposition is deduced in Section 1.5.4 from the one for the conjunctive one, since the lattice 2^Ω is symmetrical for any set Ω.

More specifically, we provide a proof for what we call the *proxy theorem* 1.5.3.2 and for its corollary 1.5.3.1, which enable us to conclude that focal points are sufficient to the definition of the weight function w. As a consequence, we propose a recursive formula using focal points that allows us to efficiently compute the conjunctive decomposition by reusing previously computed weights.

Definition 1.5.3.1 (*Focal point*)**.** We define a *focal point* as the set among all sets associated with a same commonality expression (i.e. a selection of focal supersets) that contains all the others. Formally, noting $\mathring{\mathcal{F}}$ the set containing all focal points, they are defined by:

$$\forall A \in \mathring{\mathcal{F}}, \ \forall B \subseteq \Omega, \quad \mathcal{F}_{\supseteq A} = \mathcal{F}_{\supseteq B} \Rightarrow A \supseteq B$$

with the intuitive notation $\mathcal{F}_{\supseteq A} = \{F \in \mathcal{F} \ / \ F \supseteq A\}$.

Thus, a focal point is the biggest set contained by all the sets in a selection of focal sets. In other words, it is the intersection of all these sets. So, any focal point A is defined by:

$$\exists \mathcal{E} \subseteq \mathcal{F}, \mathcal{E} \neq \emptyset, \quad A = \bigcap_{F \in \mathcal{E}} F$$

Property 1.5.3.1 (*Inclusion of focal sets*)**.** We have $\mathcal{F} \subseteq \mathring{\mathcal{F}}$ since for all focal set $A \in \mathcal{F}$, there is a selection of focal sets $\mathcal{E} \subseteq \mathcal{F}$ such that $|\mathcal{E}| = 1$ and $A = \bigcap_{F \in \mathcal{E}} F$.

Property 1.5.3.2 (*Decomposition into bigger focal points*)**.** The intersection of two focal points is also a focal point. It is equal to the union of their respective selections of focal sets:

$$\forall A, B \in \mathring{\mathcal{F}}, \quad A \cap B = \left(\bigcap_{F \in \mathcal{F}_{\supseteq A}} F \right) \cap \left(\bigcap_{F \in \mathcal{F}_{\supseteq B}} F \right) = \bigcap_{F \in \mathcal{F}_{\supseteq A} \cup \mathcal{F}_{\supseteq B}} F$$

Remark. Note that this means any focal point can be described as either F, where $F \in \mathcal{F}$, or $\mathring{F} \cap F$, where $\mathring{F} \in \mathring{\mathcal{F}}$. Doing so, we can find all focal points of a belief assignment in $O\left(|\mathring{\mathcal{F}}|.|\mathcal{F}|\right)$.

Lemma 1.5.3.1 (*Exponent equilibrium*)**.** *Following the binomial theorem, we have* $\forall C \subseteq \Omega$, $\forall A \subset C$*:*

$$\sum_{\substack{B \supseteq A \\ B \subseteq C}} (-1)^{|B|-|A|+1} = -\sum_{k=0}^{|C|-|A|} \binom{|C|-|A|}{k} (-1)^k . \, 1^{|C|-|A|-k} = -(1-1)^{|C|-|A|} = 0$$

Theorem 1.5.3.1 (*Focal point formula*)**.** *For any* $A \subseteq \Omega$*, the weight* $w(A)$ *can be decomposed into a product of commonalities on focal points supersets of* A *only. Formally, we have:*

$$w(A) = q(A)^{-1} . \prod_{F \in \mathring{\mathcal{F}}_{\supset A}} q(F)^{e_{A,F}} \tag{1.20}$$

with the intuitive notation $\mathring{\mathcal{F}}_{\supset A} = \{F \in \mathring{\mathcal{F}} \ / \ F \supset A\}$*, and*

$$\forall F \in \mathring{\mathcal{F}}_{\supset A}, \quad e_{A,F} = 1 - \sum_{\substack{B \in \mathring{\mathcal{F}}_{\supset A} \\ B \subset F}} e_{A,B}.$$

Proof. First, let us note $e_{A,B}$ the sum of all exponents associated with a same commonality expression, defined by $\mathcal{F}_{\supseteq B}$, among sets containing A, excepted A itself. More clearly, we pose $e_{A,B}$ as:

$$e_{A,B} = \sum_{\substack{X \supset A \\ \mathcal{F}_{\supseteq X} = \mathcal{F}_{\supseteq B}}} (-1)^{|X|-|A|+1}$$

Using Definition 1.5.3.1, we know that for any focal point $F \in \mathring{\mathcal{F}}_{\supset A}$, we have:

$$e_{A,F} = \sum_{\substack{X \supset A \\ X \subseteq F \\ \mathcal{F}_{\supseteq X} = \mathcal{F}_{\supseteq F}}} (-1)^{|X|-|A|+1}$$

Moreover, note that for any couple of focal points $B, F \in \mathring{\mathcal{F}}$, if $B \neq F$, then $\mathcal{F}_{\supseteq B} \neq \mathcal{F}_{\supseteq F}$. So, for any set $A \subseteq \Omega$ and for any focal point $F \in \mathring{\mathcal{F}}$, we can introduce the following decomposition:

$$\prod_{\substack{X \supseteq A \\ X \subseteq F}} q(X)^{(-1)^{|X|-|A|+1}} = q(A)^{-1} . \left[\prod_{\substack{B \in \mathring{\mathcal{F}}_{\supset A} \\ B \subset F}} q(B)^{e_{A,B}} \right] . \, q(F)^{e_{A,F}}$$

This translates in terms of exponents as:

$$\sum_{\substack{X \supseteq A \\ X \subseteq F}} (-1)^{|X|-|A|+1} = -1 + \left[\sum_{\substack{B \in \mathring{\mathcal{F}}_{\supset A} \\ B \subset F}} e_{A,B} \right] + e_{A,F},$$

which is equal to 0, in virtue of Lemma 1.5.3.1. This finally gives us:

$$e_{A,F} = 1 - \sum_{\substack{B \in \mathring{\mathcal{F}}_{\supset A} \\ B \subset F}} e_{A,B}$$

■

Example 1.5.3.1 (*Quasi-Bayesian case*)**.** Let us consider the case of a *quasi-Bayesian* mass function m, i.e. a mass function such that $\Omega \in \mathcal{F}$ and all pairs of focal sets (F_i, F_j), where Ω is neither of them, verify $F_i \cap F_j = \emptyset$. In this case, the only set that can be generated by the intersection of any number of focal points is $\emptyset$. Thus, we have $\mathring{\mathcal{F}} = \mathcal{F} \cup \{\emptyset\}$. Theorem 1.5.3.1 gives us for all set A in 2^Ω:

$$\forall A \subseteq \Omega, \quad w(A) = \begin{cases} q(\Omega)^{-1} & \text{if } A = \Omega \\ q(A)^{-1}.\ q(\Omega) & \text{if } A \in \mathcal{F} \backslash \{\Omega\} \\ q(A)^{-1}.\ q(\Omega)^{1-|\mathcal{F}\backslash\{\Omega\}|}.\ \displaystyle\prod_{F \in \mathcal{F}\backslash\{\Omega\}} q(F) & \text{if } A = \emptyset \\ 1 & \text{otherwise} \end{cases}, \tag{1.21}$$

where $q(\emptyset) = \sum_{B \in 2^\Omega} m(B) = 1$. This particular case was already known by [23] (Proposition 1).

Theorem 1.5.3.2 (*Proxy theorem*)**.** *For any set $A \subseteq \Omega$, if there is a smallest focal point superset, i.e. if there exists a focal point $P \in \mathring{\mathcal{F}}_{\supset A}$ such that for all focal points $F \in \mathring{\mathcal{F}}_{\supset A}$, we have $P \subseteq F$, then:*

$$w(A) = q(A)^{-1}.\ q(P).$$

Proof. According to Theorem 1.5.3.1, for any set $A \subseteq \Omega$, if there exists a focal point $P \in \mathring{\mathcal{F}}_{\supset A}$ such that for all focal point $F \in \mathring{\mathcal{F}}_{\supset A}$, we have $P \subseteq F$, then $e_{A,P} = 1$. Thus, for all focal point $F \in \mathring{\mathcal{F}}_{\supset P}$ such that there is no focal point in the open interval (P, F), we have:

$$e_{A,F} = 1 - e_{A,P} = 0.$$

Similarly, for all focal point $F \in \mathring{\mathcal{F}}_{\supset P}$ such that, for all focal point B in the open interval (P, F), there is a no focal point in the open interval (P, B), we have:

$$e_{A,F} = 1 - (e_{A,P} + \sum_{\substack{B \in \mathring{\mathcal{F}}_{\supset P} \\ B \subset F}} e_{A,B}) = 1 - (1 + \sum_{\substack{B \in \mathring{\mathcal{F}}_{\supset P} \\ B \subset F}} 0) = 0.$$

It is easy to see that by recursion, for all focal point $F \in \mathring{\mathcal{F}}_{\supset P}$, we obtain: $e_{A,F} = 0$. So, still according to Theorem 1.5.3.1, we get:

$$w(A) = q(A)^{-1}.\ q(P)^1\ .\ \prod_{F \in \mathring{\mathcal{F}}_{\supset P}} q(F)^0 = q(A)^{-1}.\ q(P)$$

■

Corollary 1.5.3.1 (*Sufficiency of $\mathring{\mathcal{F}}$ to define w*)**.** *For all set A that is not a focal point, there is a proxy focal point $P \in \mathring{\mathcal{F}}_{\supset A}$ such that $q(A) = q(P)$ and for all focal point $F \in \mathring{\mathcal{F}}_{\supset A}$, we have $P \subseteq F$. Thus, according to Theorem 1.5.3.2, we have:*

$$\forall A \notin \mathring{\mathcal{F}}, \quad w(A) = 1.$$

Proof. By Definition 1.5.3.1 of a focal point, for any set $A \subseteq \Omega$, there is a focal point $P \in \mathring{\mathcal{F}}_{\supseteq A}$ such that $\mathcal{F}_{\supseteq P} = \mathcal{F}_{\supseteq A} \Rightarrow P \supseteq A$. This means that $q(A) = q(P)$. Also, if A is not a focal point, then $P \in \mathring{\mathcal{F}}_{\supset A}$ (since $\Omega \in \mathring{\mathcal{F}}$). In addition, if A is not a focal point, Property 1.5.3.2 tells us that

$$\bigcap \mathring{\mathcal{F}}_{\supset A} = \bigcap \left(\bigcup_{F \in \mathring{\mathcal{F}}_{\supset A}} \mathcal{F}_{\supseteq F} \right) = \bigcap \mathcal{F}_{\supset A} = P.$$

By definition of the operator $\cap$, we get that $\forall F \in \mathring{\mathcal{F}}_{\supset A}, \quad F \supseteq P$. ■

Example 1.5.3.2 (*Consonant case*)**.** Let us now examine the case of a *consonant* mass function, i.e. a mass function such that all its n focal sets F_i verify $F_1 \subset F_2 \subset \cdots \subset F_n \subset \Omega$. In this case, the proxy theorem 1.5.3.2 applies to all set A in $2^\Omega \backslash \{\Omega\}$. Furthermore, as the intersection of any number of these focal sets is necessarily one of them, we have $\mathring{\mathcal{F}} = \mathcal{F}$. We obtain:

$$\forall A \subseteq \Omega, \quad w(A) = \begin{cases} q(\Omega)^{-1} & \text{if } A = \Omega \\ q(F_i)^{-1}.\, q(F_{i+1}) & \text{if } \exists i \in [\![1, n]\!] \ / \ A = F_i \\ 1 & \text{otherwise} \end{cases} \tag{1.22}$$

This particular case was already known by [23] (Proposition 2).

Theorem 1.5.3.3 (*Recursive focal point formula*)**.** *Thanks to Corollary 1.5.3.1 and taking back Eq. (1.15), we obtain that for any set $A \subseteq \Omega$, the weight $w(A)$ can be decomposed into a product of weights associated with focal points supersets of A as follows:*

$$\forall A \subseteq \Omega, \quad w(A) = \begin{cases} q(A)^{-1}.\prod_{F \in \mathring{\mathcal{F}}_{\supset A}} w(F)^{-1} & \text{if } A \in \mathring{\mathcal{F}} \\ 1 & \text{otherwise} \end{cases} \tag{1.23}$$

Remark. A procedure directly based on this formula would have a complexity between $O(|\mathring{\mathcal{F}}|)$ and $O(|\mathring{\mathcal{F}}|^2)$. Its inverse (1.15) only on focal points leads to the same complexity.

Thus, our method varies from $O(|\mathring{\mathcal{F}}|)$ to $O(|\mathring{\mathcal{F}}|^2)$, where $|\mathring{\mathcal{F}}| \in [\ |\mathcal{F}|,\ 2^{|\Omega|}\ ]$, depending on the structure of $\mathcal{F}$, as seen in the two notable particular cases presented in Examples 1.5.3.1 and 1.5.3.2. In general, the more there are big focal sets that are not nested in others, the more $|\mathring{\mathcal{F}}|$ has chances to be great. Nevertheless, the most natural and interpretable structures are the ones close to either Example 1.5.3.1 or 1.5.3.2, which tend to have a number of focal points close to their number of focal sets. Moreover, our approach directly benefits from methods that approximate $\mathcal{F}$, which should maintain $|\mathring{\mathcal{F}}|$ close to $|\mathcal{F}|$. These approximation methods do not limit the expressiveness of belief assignments.

1.5.4 Transposition to the computation of the disjunctive decomposition

In this section, with simply provide the dual of the theorems and corollaries of section 1.5.3 that come from the symmetry of 2^Ω.

Definition 1.5.4.1 (*Dual focal point*)**.** We define a *dual focal point* as the set among all sets associated with a same implicability expression (i.e. a selection of focal subsets) that is contained in all the others. Formally, noting $\overline{\mathring{\mathcal{F}}}$ the set containing all dual focal points, they are defined by:

$$\forall A \in \overline{\mathring{\mathcal{F}}},\ \forall B \subseteq \Omega, \quad \mathcal{F}_{\subseteq A} = \mathcal{F}_{\subseteq B} \Rightarrow A \subseteq B$$

with the intuitive notation $\mathcal{F}_{\subseteq A} = \{F \in \mathcal{F} \ / \ F \subseteq A\}$.

Thus, a dual focal point is the smallest set containing all the sets in a selection of focal sets. In other words, it is the union of all these sets. So, any dual focal point A is defined by:

$$\exists \mathcal{E} \subseteq \mathcal{F}, \mathcal{E} \neq \emptyset, \quad A = \bigcup_{F \in \mathcal{E}} F$$

Property 1.5.4.1 (*Inclusion of focal sets*)**.** As for focal points, we have $\mathcal{F} \subseteq \overline{\mathring{\mathcal{F}}}$ since for all focal set $A \in \mathcal{F}$, there is a selection of focal sets $\mathcal{E} \subseteq \mathcal{F}$ such that $|\mathcal{E}| = 1$ and $A = \bigcup_{F \in \mathcal{E}} F$.

Property 1.5.4.2 (*Decomposition into smaller focal points*)**.** The union of two dual focal points is also a dual focal point. It is equal to the union of their respective selections of focal sets:

$$\forall A, B \in \overset{\circ}{\overline{\mathcal{F}}}, \quad A \cup B = \left(\bigcup_{F \in \mathcal{F}_{\supseteq A}} F \right) \cup \left(\bigcup_{F \in \mathcal{F}_{\supseteq B}} F \right) = \bigcup_{F \in \mathcal{F}_{\supseteq A} \cup \mathcal{F}_{\supseteq B}} F$$

Remark. Note that this means any dual focal point can be described as either F, where $F \in \mathcal{F}$, or $\overset{\circ}{F} \cup F$, where $\overset{\circ}{F} \in \overset{\circ}{\overline{\mathcal{F}}}$. Doing so, we can find all dual focal points of a belief assignment in $O\left(|\overset{\circ}{\overline{\mathcal{F}}}|.|\mathcal{F}|\right)$.

Corollary 1.5.4.1 (*Dual focal point formula*)**.** *For any $A \subseteq \Omega$, the weight $v(A)$ can be decomposed into a product of implicabilities on focal points subsets of A only. Formally, we have:*

$$v(A) = b(A)^{-1}. \prod_{F \in \overset{\circ}{\overline{\mathcal{F}}}_{\subset A}} b(F)^{e_{A,F}} \tag{1.24}$$

with the intuitive notation $\overset{\circ}{\overline{\mathcal{F}}}_{\subset A} = \{F \in \overset{\circ}{\overline{\mathcal{F}}} \ / \ F \subset A\}$, and

$$\forall F \in \overset{\circ}{\overline{\mathcal{F}}}_{\subset A}, \quad e_{A,F} = 1 - \sum_{\substack{B \in \overset{\circ}{\overline{\mathcal{F}}}_{\subset A} \\ B \supset F}} e_{A,B}.$$

Example 1.5.4.1 (*Quasi-Bayesian dual case*)**.** Let us consider the dual of the *quasi-Bayesian* case of Example 1.5.3.1. This means that we have a mass function m such that $\emptyset \in \mathcal{F}$ and all pairs of focal sets (F_i, F_j), where $\emptyset$ is neither of them, verify $F_i \cup F_j = \Omega$. In this case, the only set that can be generated by the union of any number of dual focal points is Ω. Thus, we have $\overset{\circ}{\overline{\mathcal{F}}} = \mathcal{F} \cup \{\Omega\}$. Theorem 1.5.4.1 gives us for all set A in 2^Ω:

$$\forall A \subseteq \Omega, \quad v(A) = \begin{cases} b(\emptyset)^{-1} & \text{if } A = \emptyset \\ b(A)^{-1}. \, b(\emptyset) & \text{if } A \in \mathcal{F} \backslash \{\emptyset\} \\ b(A)^{-1}. \, b(\emptyset)^{1-|\mathcal{F}\backslash\{\emptyset\}|}. \prod_{F \in \mathcal{F}\backslash\{\emptyset\}} b(F) & \text{if } A = \Omega \\ 1 & \text{otherwise} \end{cases}, \tag{1.25}$$

where $b(\Omega) = \sum_{B \in 2^\Omega} m(B) = 1$. Notice however that the quasi-Bayesian case of Example 1.5.3.1 is actually the worst case for the complexity tied to the computation of v. More precisely, the worst case is the one corresponding to $\mathcal{F}$ containing all singletons of 2^Ω. In the same way, this dual quasi-Bayesian case is the worst case complexity for the computation of w. More precisely, the worst for w is attained when the complement to Ω of each singleton is a focal set.

Theorem 1.5.4.1 (*Proxy theorem*)**.** *For any set $A \subseteq \Omega$, if there is a biggest focal point subset, i.e. if there exists a dual focal point $P \in \overset{\circ}{\overline{\mathcal{F}}}_{\subset A}$ such that for all dual focal points $F \in \overset{\circ}{\overline{\mathcal{F}}}_{\subset A}$, we have $P \supseteq F$, then:*

$$v(A) = b(A)^{-1}. \, b(P).$$

Corollary 1.5.4.2 (*Sufficiency of $\overset{\circ}{\overline{\mathcal{F}}}$ to define v*)**.** *For all set A that is not a dual focal point, there is a proxy dual focal point $P \in \overset{\circ}{\overline{\mathcal{F}}}_{\subset A}$ such that $b(A) = b(P)$ and for all dual focal point $F \in \overset{\circ}{\overline{\mathcal{F}}}_{\subset A}$, we have $P \supseteq F$. Thus, according to Theorem 1.5.4.1, we have:*

$$\forall A \notin \overset{\circ}{\overline{\mathcal{F}}}, \quad v(A) = 1.$$

Corollary 1.5.4.3 (*Recursive dual focal point formula*)**.** *Thanks to Corollary 1.5.4.2 and taking back Eq. (1.18), we obtain that for any set $A \subseteq \Omega$, the weight $v(A)$ can be decomposed into a product of weights associated with dual focal points subsets of A as follows:*

$$\forall A \subseteq \Omega, \quad v(A) = \begin{cases} b(A)^{-1}.\prod_{F \in \overline{\mathring{\mathcal{F}}} \subset A} v(F)^{-1} & \text{if } A \in \overline{\mathring{\mathcal{F}}} \\ 1 & \text{otherwise} \end{cases} \tag{1.26}$$

Remark. A procedure directly based on this formula would have a complexity between $O(|\overline{\mathring{\mathcal{F}}}|)$ and $O(|\overline{\mathring{\mathcal{F}}}|^2)$. Its inverse (1.15) only on focal points leads to the same complexity.

1.5.5 Conclusion and perspectives

We have presented here the difficulties tied to the design of evidence based algorithms for the transformation of a belief assignment into its conjunctive or disjunctive decomposition and its inverse. For this, we proposed a novel mathematical notion that we called *focal point*, derived from the notion of focal set. With these focal points, we exploit the properties of the function to be transformed in 2^Ω, while the state-of-the-art optimal general algorithms, based on the *Fast Möbius Transform* (FMT), ignore it. Doing so, we can now design new general algorithms with complexities inferior to the exponential one of these optimal algorithms. Outside the scope of this article, the notion of focal point can be used in DST for information analysis without having to compute the conjunctive (or disjunctive) decomposition beforehand. More generally, any method exploiting the conjunctive or disjunctive decomposition at some point, like the Cautious fusion rule or the Bold fusion rule, can benefit from a reduction in their complexity.

However, the computation of these focal points has a complexity in $O(|\mathring{\mathcal{F}}| \,.\, |\mathcal{F}|)$, where $\mathring{\mathcal{F}}$ is the set containing these focal points and $\mathcal{F}$ is the set containing the focal sets of some belief assignment. Depending on $\mathcal{F}$, the set $\mathring{\mathcal{F}}$ might be as big as 2^Ω, which means that $O(|\mathring{\mathcal{F}}| \,.\, |\mathcal{F}|)$ might be far greater than $O(|\Omega| \,.\, 2^{|\Omega|})$, i.e. the complexity of the FMT, in some cases. Thus, we plan to provide complementary methods guaranteeing better complexities than the FMT. Furthermore, the question of how to compute back the mass function from either the commonality or implicability function in the general case without the FMT remains. Ideally, we would like to exploit these already computed focal points once again, after modification of the conjunctive or disjunctive decomposition. This requires a broader view on our approach.

1.6 Conclusion

In this chapter, we set the scene in which this thesis plays its role. We motivated our choice for a distributed collaborative perception system and highlighted the challenges that have to be overcome for its practical implementation. We identified two main axes on which we chose to focus: data incest management and communication efficiency. For the latter, we opted for the use of semantic grid maps in order to know both what is perceived and what is not. This emphasizes the need for a distinction between several types of uncertainties, taking the imperfection of the observer into account. It appears that this distinction is also useful for data incest management in the fusion of probability distributions over a qualitative random variable in an unpredictable communication network topology. To make this distinction, we decided to employ Dempster-Shafer Theory (DST). In DST, the Cautious and Bold fusion rules are simple and guaranteed ways to avoid data incest. The issue is computational efficiency, on which directly depends the practicability of our system. Exponential time and space complexities are quite common in DST, and while there are ways to limit computations to informative components in some instances, there were no such method to compute the Cautious and Bold

fusion rules. The *Fast Möbius Transform* (FMT) corresponds to the family of algorithms to compute most DST transformations that is optimal for a given sample space. Given that this sample space grows exponentially with the number of outcomes to consider, this complexity is more than exponential in the number of outcomes. This was the most efficient way to compute the conjunctive and disjunctive decompositions of evidence on which the Cautious and Bold fusion rules rely. Thus, we tried and created a method exploiting informative components that we called *focal points*. This method allows us to design new general algorithms that exploit the structure of the informative components of a function, instead of the structure of the sample space alone. Doing so, less than exponential complexities *can* be obtained for the Cautious and Bold fusion rules in the general case.

The exponential complexity of the FMT being already prohibitive for more than a dozen possible outcomes to consider in a random experiment, we consider it as an absolute worst case complexity that must not be exceeded. Yet, the method we just proposed, while having a far lighter complexity in many cases, might feature a far worse complexity in other situations that are difficult to predict. Therefore, we need another, more stable, method that can guarantee a complexity always inferior to the one of the FMT. This is the objective of Chapter 3. Furthermore, the notion of focal point and our approach in general seems generalizable to other transformations in DST. This would be useful in particular for the computation of the mass function from the commonality function after application of the Cautious or Bold fusion rule. Studying our approach in this broader perspective might also enable us to alleviate the limitation on the applicability of the Cautious fusion rule and the Bold fusion rule. We might be able to use these rules on respectively dogmatic and normal belief assignments. Our approach even seems generalizable outside DST, to other spaces than the powerset of a set. All of this is the subject of the next chapter 2.

Chapter 2

Focal points and their implications for Möbius Transforms and Dempster-Shafer Theory

This chapter contains the adaptation of our fourth paper published in the international journal *Information Sciences* [52]. Though published in conferences before this one, our second [42] and third [53] papers correspond to Chapter 3. This is due to the fact that they require a proper generalization of the notion of focal point we introduced in the previous chapter, which is the purpose of the present chapter.

2.1 Introduction

Dempster-Shafer Theory (DST) [24] is an elegant formalism that generalizes Bayesian probability theory by considering the specificity of evidence. This means that it enables a source (e.g. some sensor model) to represent its belief in the state of a variable by assigning credit not only directly to a possible state, as in Bayesian probability theory, but also to groups of states, reflecting some model uncertainty or randomness. This assignment of belief is called a *mass function*. It generalizes the notion of probability distribution by providing meta-information that quantifies the level of uncertainty about one's believes considering the way one established them, which is critical for decision making.

Nevertheless, this information comes with a cost: let Ω be the set containing all possible states. This set Ω has $2^{|\Omega|}$ subsets. Thus, in DST, we consider $2^{|\Omega|}$ potential values instead of only $|\Omega|$ in Bayesian probability theory, which can lead to computationally and spatially very expensive algorithms. Computations can become difficult to perform for more than a dozen possible states (e.g. 20 states in Ω generate more than a million subsets), although we may need to consider much more of them (e.g. for classification or identification). This imposes a choice in DST between expressiveness of the model (i.e. $|\Omega|$) and fast computations (especially considering limited resources such as in embedded systems). To tackle this issue, a lot of work has been done to reduce the complexity of transformations used to combine belief sources with Dempster's rule [30]. We distinguish between two approaches that we call *powerset-based* and

evidence-based.

The *powerset-based* approach concerns all algorithms based on the structure of the powerset 2^Ω. They have a complexity dependent on $|\Omega|$. Early works [54–57] proposed optimizations by restricting the structure of evidence to only singletons and their negation, which greatly restrains the expressiveness of DST. Later, a family of optimal algorithms working in the general case, that is, those based on the *Fast Möbius Transform* (FMT) [43], was discovered. Their complexity is $O(|\Omega|.2^{|\Omega|})$ in time and $O(2^{|\Omega|})$ in space. The FMT has become the de facto standard for the computation of every transformation in DST. Consequently, efforts were made to reduce the size of Ω to benefit from the optimal algorithms of the FMT. More specifically, [44] refers to the process of conditioning by the *combined core* (intersection of the unions of all sets of nonnull mass of each belief source) and *lossless coarsening* (partitioning of Ω into super-elements that each represents a group of elements always appearing together in all sets of nonnull mass). There are also approximation methods such as Monte Carlo methods [44], which depend on a number of trials that must be large and grows with $|\Omega|$, and lossy coarsening [58], but we focus here on exact methods. More recently, DST has been generalized to any lattice (not just 2^Ω) in [59], which has been used to propose a method [60] limiting its expressiveness to only intervals in Ω in order to keep DST transformations tractable. If there is no order between the elements of Ω, this method comes down again to restricting the assignment of credit to only singletons.

The *evidence-based* approach concerns all algorithms that aim to restrict computations (not expressiveness) to the only subsets that have a nonnull mass, i.e. that contain information (*evidence*). These are called *focal sets* and are usually far less numerous than $2^{|\Omega|}$. This approach, also referred as the *obvious* one, implicitly originates from the seminal work of Shafer [24] and is often more efficient than the powerset-based one since it only depends on information contained in sources in a quadratic way. Doing so, it allows for the exploitation of the full potential of DST by enabling one to choose any frame of discernment, without concern about its size. Moreover, the evidence-based approach benefits directly from the use of approximation methods, some of which are very efficient [45]. Therefore, this approach seems superior to the FMT in most use cases, above all when $|\Omega|$ is large, where an algorithm with exponential complexity is just intractable.

But, unfortunately, focal sets are not sufficient to define the zeta and Möbius transforms. In particular, if one wishes to compute the multiplicative Möbius transform of an additive zeta transform (e.g. computing the conjunctive or disjunctive weight function from the commonality or implicability function), focal sets are not enough in the general case. For this, we already proposed in [42][1] the notion of *focal point*, which is sufficient to completely define the conjunctive and disjunctive decompositions.

One other argument against the evidence-based approach is the lack of knowledge about the structure of the set of focal sets due to its inconstancy. Indeed, the FMT draws its power from the knowledge of the structure of Boolean lattices. Doing so, all algorithm using only focal sets is forced to have a quadratic complexity in the number of these focal sets, which can be worse than the complexity of FMT algorithms when this number approaches the size of the powerset 2^Ω. Yet, *focal points* do have a specific structure that has been successfully exploited in [53], where we proposed methods with complexities that are variable but always inferior to the complexity of the FMT, its trimmed version [61] and more generally the Fast Zeta Transform [62, 63], and may be even lower than $O(|\Omega|.F)$ in some cases, where F is the number of focal sets.

This present chapter extends our previous works [42] and [53] (presented in the next chapter), focusing on the study of these focal points to provide their generalization to any partially ordered set and the reformulation of the Möbius inversion theorem based on them, which were missing.

[1] This short reference is written in french, but does not need to be read to follow this article.

This new contribution also applies to the multiplication of any function by the zeta or Möbius function in any incidence algebra [40]. The second part of this article proposes applications in DST exploiting these focal points. We limit this second part to the classical DST, for the sake of clarity, although our results are applicable to its generalization to any lattice [59].

This paper is organized as follows: Section 2.2 presents the elements on which our notions are built. Section 2.3 presents our contributions to the zeta and Möbius transforms. Section 2.4 discusses multiple applications to DST. Finally, we conclude this article in section 2.5.

2.2 Background of our method

Let $(P, \leq)$ be a semifinite (lower semifinite for $\leq$, upper semifinite for $\geq$) set partially ordered by some binary operator noted $\leq$, e.g. the powerset $(2^\Omega, \subseteq)$ or $(2^\Omega, \supseteq)$ containing all subsets of a set Ω ordered by inclusion, the set $(\mathbb{N}^*, |)$ of all positive integers ordered by divisibility, the set $(\mathbb{N}, \leq)$ of all nonnegative integers ordered by $\leq$, etc.

2.2.1 Support of a function in P

The support $\text{supp}(f)$ of a function $f : P \to \mathbb{R}$ is defined as $\text{supp}(f) = \{x \in P \ / \ f(x) \neq 0\}$. For example, in DST, the set of focal elements of a mass function m is $\text{supp}(m)$.

With this notion, it is obvious that Eq. 1.4 can be reduced to:

$$\forall y \in P, \quad g(y) = \sum_{\substack{x \in \text{supp}(f) \\ x \leq y}} f(x) \tag{2.1}$$

2.2.2 Order theory

Minimal/Maximal elements Minimal elements of a set S are its elements such that there is no element in S that is less than them. The set containing them is noted $\min(S)$. If there is only one minimal element in S, it is its *minimum*. For example, in a totally ordered set (i.e. a *chain*), there can only be one minimal element, i.e. its minimum. Dually, the same principle holds for maximal elements of a set S, noted $\max(S)$, and its *maximum*.

Supremum/Infimum The *supremum* (also known as *join*) of a set S of elements of P is its least upper bound, i.e. the least element in P that is greater than or equal to every element of S. It is noted $\bigvee S$, but only exists if the set of the upper bounds of S has only one minimal element. In particular, if S has a maximum, then it is $\bigvee S$. Dually, the *infimum* (also known as *meet*) of a set S of elements of P is its greatest lower bound, i.e. the greatest element in P that is less than or equal to every element of S. It is noted $\bigwedge S$.

Lattice and semilattice An *upper semilattice* S is a set such that every of its nonempty subsets has its supremum in S. A *lower semilattice* S is a set such that every of its nonempty subsets has its infimum in S. A *lattice* is both an upper and a lower semilattice. In particular, all element of a *finite* lattice L can be described as either $\bigwedge L$ or the supremum (join) of a nonempty subset of its *join-irreducible elements*.

2.3 Focal points and our *Efficient* Möbius inversion formula

The purpose of this section is to present our work on what we call the *Efficient Möbius inversion formula* and its developments. Section 2.3.1 gives an overview of the approach. Section 2.3.2 tries to simplify the Möbius inversion formula of Eq. 1.5, highlighting the emergence of our focal points. Section 2.3.3 properly defines these focal points and exposes the simplifications they allow on the Möbius inversion formula. Section 2.3.4 proposes ways to compute them. Finally, section 2.3.5 exploits one of these ways to uncover links between the additive and multiplicative Möbius transforms of a same function. In addition, section 2.3.6 discusses some aspects of our approach and section 2.3.7 bridges the gap between our theoretical results and their usage.

Let us note $(P, \leq)$ the set P partially ordered by some binary operator noted $\leq$, and let g and f be the functions of Eq. 1.4. For the sake of simplicity, we will consider in this section that P is finite. If P is semifinite or infinite, see section 2.3.6. We assume that f is defined in a compact way, simply through $\text{supp}(f)$. We also assume that g is defined in a compact way, through a partition of P, noted $\mathcal{G}$, into parts X such that all elements of P greater than at least one element of $\min(X)$ and less than at least one element of $\max(X)$ is in X and has the same image through g as any other element of X. Multiple partitions may fulfill these conditions. For instance, we could force these parts to be intervals (i.e. to force $|\min(X)| = |\max(X)| = 1$ for all part X) or not. This is of no importance for what follows, but the fewer parts there are, the fewer computations will be needed. An example of such an *image partition* is illustrated in Fig. 2.1. The image through g of all elements in P is thus determined by only minimal and maximal elements of parts. Also, note that all of our following results can be applied to any incidence algebra, simply considering $g(a, \cdot)$ and the support $\text{supp}(f(a, \cdot)) = \{x \in P \ / \ f(a, x) \neq 0\}$ for some $a \in P$ instead of $\text{supp}(f)$, since functions are then defined on intervals instead of single elements.

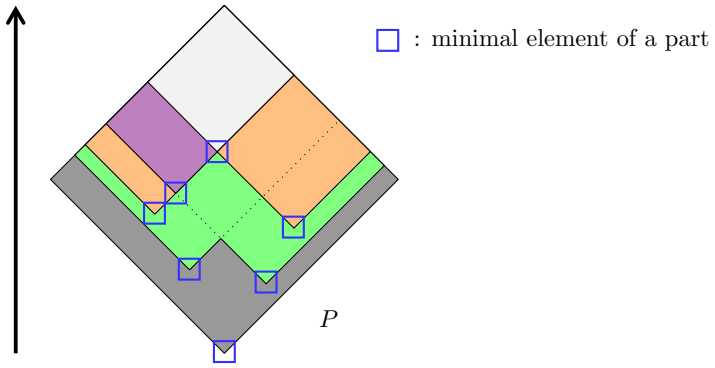


Figure 2.1: Example of image partition $\mathcal{G}$ of P corresponding to some function g. The (discretized) area of the diamond represents P, which has a minimum in this example. Each point of it corresponds to one of its elements. The arrow on the left represents the order $\leq$. Two points are ordered only if the upward vector aligning them makes an angle between $-45°$ and $45°$ with the big arrow vector on the left. If so, then, of the two points, the one that is lower in the figure is lower in P. Each color represents an image through g. Each part X of this partition $\mathcal{G}$ contains elements of same image through g and can be described as all elements between $\min(X)$ and $\max(X)$. In this example, each color also represents a part. We see that g can be defined in a compact way as a list or tree of parts, each described by its minimal and maximal elements.

2.3.1 Problem statement and intuition

Let us start by translating the problems stated in introduction of this article into formal terms. As showed in Eq. 2.1, g can be computed from $\text{supp}(f)$ only. However, g cannot be completely defined in the general case from $g(\text{supp}(f))$ alone. Furthermore, it is not possible to determine $\text{supp}(f)$ from the definition of g before computing f. These two issues prevent us from limiting computations so that it scales linearly with the quantity of information in these transforms, i.e. $|\text{supp}(f)|$. Nevertheless, it is possible to determine the smallest set containing both $\text{supp}(f)$ and all the defining elements of P for g, either from f or from g.

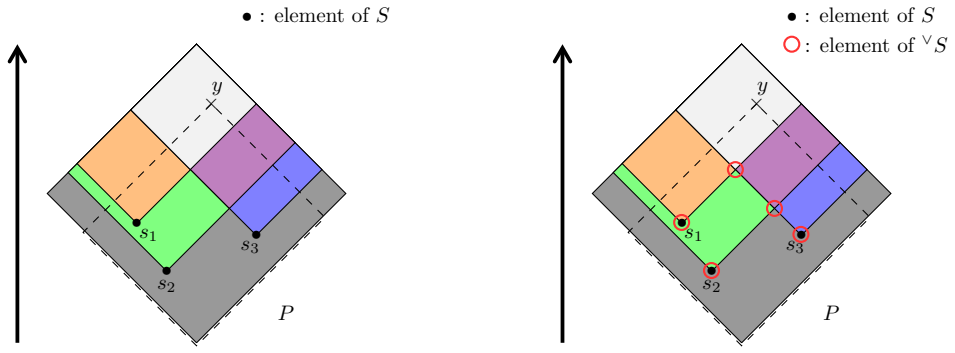
Giving some reduced common superset for these has been done in the past: For conjunctive fusion in DST, one can find in [44] that computations can be limited to the powerset of $\text{supp}(f_1) \cap \text{supp}(f_2)$, where f_1 and f_2 are two mass functions to combine. In [61], a method was proposed to limit computations of the zeta transform to the elements of the powerset that are greater than an element of $\min(\text{supp}(f))$. In contrast, here we propose the smallest common superset for f and g in any semilattice. To the best of our knowledge, our approach is also the first one to propose a reduced common superset based on g alone for the computation of its Möbius transform (i.e. f). The elements of this smallest common superset are what we call *the focal points of f and g*.

The idea is simple: track the influence of f on g by looking at elements from P that are greater than the same elements from $\text{supp}(f)$, since they are necessarily associated with the same image through g, and select one *representative* for each of these configurations. Computing the zeta transform of f on these *representatives* only will yield the complete definition of g. Then, use the Möbius function associated with the partially ordered set made of these *representatives* to compute the Möbius transform of g and get the complete definition of f.

These *representatives* are our focal points. By definition, their image through g contains all possible images through g, excepted 0 under some conditions. The same can be stated with f. However, several questions arise: How to get the images of all other elements of P? Is there a best *representative* and how to select it for each of these subsets of $\text{supp}(f)$? See section 2.3.3. How to find them efficiently and how to find them without knowing $\text{supp}(f)$? See section 2.3.4. Is the Möbius transform for these *representatives* really the same in this reduced set as in P? See section 2.3.2.

2.3.2 Simplifying the Möbius inversion formula

Let us start by showing that these focal points are unique and arise naturally in the expression of the Möbius transform defined in P. In the following, we will mainly focus on the additive Möbius inversion formula since it displays the same properties as the multiplicative form (think of the multiplicative form as the exponential of the additive form). In the end, our study exploits the neutrality of some values in the sum or product. For the addition, the neutral element is 0, hence our use of the support $\text{supp}(f)$. To transpose our findings to the multiplicative form, consider instead $\text{supp}(f - 1)$, as the neutral element for the multiplication is 1.



(a) Level partition

(b) Join-closure of S

Figure 2.2: Same format as in Fig 2.1. (a) Illustration of the concept of level partition. Each color represents a part of the partition $\mathcal{P}_{/(S,\leq)}$, where $S = \{s_1, s_2, s_3\}$. In a level partition, every part X is made of elements of same lower closure in S, i.e. $(\downarrow X) \cap S$. The part delimited by a dashed contour is the lower closure of some element y in P, i.e. $\downarrow y$. We see that all elements of S are in the lower closure of y. So, the lower closure in S of y is S. (b) Projection of all the elements of the join-closure $^\vee S$ onto the level partition of (a), assuming each part X of $\mathcal{P}_{/(S,\leq)}$ where $X \subseteq \uparrow S$ (i.e. all parts except the gray one) has a minimum (no assumption on $\min(P)$ is made).

Definition 2.3.2.1 (*Level partition*)**.** For any subset $S \subseteq P$, let us refer to the elements of S less than some element $x \in P$ as the *lower closure* $(\downarrow x) \cap S$ of x in S, i.e. $(\downarrow x) \cap S = \{s \in S \ / \ s \leq x\}$. In accordance with convention, let us also use the notation $(\downarrow X) \cap S = \{s \in S \ / \ \exists x \in X, \ s \leq x\}$, where $X \subseteq P$.

For conciseness, we will use the aliases $\overset{\downarrow}{}_S x = (\downarrow x) \cap S$ and $\overset{S}{}_\uparrow x = (\uparrow x) \cap S$ in the following, where x can be a subset of P or one of its elements.

We define a *level partition*, noted $\mathcal{P}_{/(S,\leq)}$ and read as P *divided according to* S *and the order* $\leq$, as the partition of P such that for any distinct parts $X, Y \in \mathcal{P}_{/(S,\leq)}$, we have $\overset{\downarrow}{}_S X \neq \overset{\downarrow}{}_S Y$ and such that for any part $X \in \mathcal{P}_{/(S,\leq)}$, any element $x \in X$ satisfies $\overset{\downarrow}{}_S x = \overset{\downarrow}{}_S X$. We say that all elements in X are at the same *level* regarding the elements of S and the order $\leq$. We would have used the *upper closure* for the dual order $\geq$. This concept is illustrated in Fig. 2.2a. To summarize, we have:

$$\mathcal{P}_{/(S,\leq)} = \left\{ \ X \subseteq P \ / \ X \neq \emptyset, \ \forall y \in P, \quad \left(\forall x \in X, \ \overset{\downarrow}{}_S x = \overset{\downarrow}{}_S y\right) \Leftrightarrow y \in X \ \right\}$$

In particular, one may notice that $\mathcal{P}_{/(\mathrm{supp}(f),\leq)}$ is also an image partition of P with respect to g.

Definition 2.3.2.2 (*Möbius function aggregate*)**.** For any element $y \in P$, for any nonempty set of elements $S \subseteq P$ and for any part X of the partition $\mathcal{P}_{/(S,\leq)}$, we define our Möbius function *aggregate* η as follows:

$$\eta_{S,\leq,P}(X, y) = \sum_{\substack{z \in X \\ z \leq y}} \mu_{P,\leq}(z, y)$$

Notice that if $y \in \min(X)$, then $\eta_{S,\leq,P}(X, y) = \mu_{P,\leq}(y, y) = 1$, according to Eq. 1.7. The notation $\eta_{S,\leq,P}$ is similar to $\mathcal{P}_{/(S,\leq)}$ and is read as η *given* S *and the elements of* P *above it*, where the term *above* depends on the order $\leq$.

Thanks to Definition 2.3.2.1 and Definition 2.3.2.2, we can propose a compact reformulation of the Möbius transform of Eq. 1.5 in the form of Lemma 2.3.2.1.

Lemma 2.3.2.1 (*Compact reformulation of the Möbius inversion formula*)**.** *For any set $S \supseteq supp(f)$, we have $\forall y \in P$:*

$$f(y) = \sum_{\substack{X \in \mathcal{P}_{/(S,\leq)} \\ y \in \uparrow X \\ X \subseteq \uparrow S}} g(X) \ . \ \eta_{S,\leq,P}(X, y) \tag{2.2}$$

where $g(X) = g(x)$ for any $x \in X$.

Proof. See Appendix A.1. ■

However, this reformulation is not a simplification and is thus of little use if our Möbius function aggregate η must be computed from the original Möbius function μ as in Definition 2.3.2.2. Fortunately, it is possible to compute η recursively, independently from μ, as shown in the following Lemma 2.3.2.2.

Lemma 2.3.2.2 (*Recursive aggregation of Möbius function images*)**.** *For any nonempty set of elements $S \subseteq P$ and for any part $X \in \mathcal{P}_{/(S,\leq)}$, if every part $Z \in \mathcal{P}_{/(S,\leq)}$ verifying $\overset{\downarrow}{}_S X \subseteq \overset{\downarrow}{}_S Z$ has a minimum, i.e. $|\min(Z)| = 1$, then we have for any $y \in P$ where $\bigwedge X < y$:*

$$\eta_{S,\leq,P}(X, y) = - \sum_{\substack{Z \in \mathcal{P}_{/(S,\leq)} \\ \bigwedge X < \bigwedge Z \leq y}} \eta_{S,\leq,P}(Z, y), \tag{2.3}$$

and $\eta_{S,\leq,P}(X, y) = 1$ if $y = \bigwedge X$.

Proof. See Appendix A.2. ■

In other words, for any part X with a minimum m, if every part containing elements greater than m has a minimum, then η can be written in a recursive form that only depends on itself. In fact, it is easy to see that η is an extension of the Möbius function μ of Eq. 1.7 associated with the partially ordered set made of every minimum of part that is in the upper closure of S (See Definition 2.3.3.2). So, the minimum of each of these parts constitutes one of the *representatives* we were looking for in section 2.3.1, i.e. our focal points. The following section 2.3.3 will focus on them.

Notice that we specifically target the parts that are in the upper closure of S in order to avoid any unnecessary constraint on $\min(P)$. Indeed, the lowest parts contain minimal elements of P and may not have a minimum if P does not have one. However, if they are not in the upper closure of S, then their image through both f and g is 0, which means that they have no influence in Equations 2.1 and 2.2 and so do not contribute to the definition of f nor g. They are not used in Eq. 2.3 either since this recursion goes upward in P. On the other hand, if they are in the upper closure of S, then each of them necessarily has an element s of S as its minimum since there is no other part below it to contain s and since s is the least element of P that is greater than or equal to s. Thus, we simply ignore the parts that are outside the upper closure of S.

2.3.3 Focal points and their implications

The purpose of Definition 2.3.3.1 is to introduce the join-closure operator that will be used to formalize the notion of focal point. Then, Property 2.3.3.1 will give the image through f and g of all non focal points. Finally, Definition 2.3.3.2 will define the Möbius function extension to use in Theorem 2.3.3.1, which contains what we called the *Efficient Möbius inversion formula.*

Definition 2.3.3.1 (*Join-closure*)**.** For any nonempty set of elements $S \subseteq P$, we note ${}^{\vee}S$ the smallest join-closed subset of P containing S, i.e.:

$${}^{\vee}S = \left\{ \bigvee F \ / \ \emptyset \subset F \subseteq S \right\}$$

The operator ${}^{\vee}\cdot : 2^P \to 2^P$ is thus a closure operator, i.e. for any sets $S, S' \subseteq P$, we have: $S \subseteq {}^{\vee}S, \quad S \subseteq S' \Rightarrow {}^{\vee}S \subseteq {}^{\vee}S'$ and ${}^{\vee}({}^{\vee}S) = {}^{\vee}S$. This notion is illustrated in Fig. 2.2b.

By definition of the supremum, i.e. the least upper bound, each element of ${}^{\vee}S$ is the minimum of a part $X \in \mathcal{P}_{/(S,\leq)}$ verifying $\overset{\downarrow}{}_S X \neq \emptyset$. Reciprocally, if a part $X \in \mathcal{P}_{/(S,\leq)}$ verifying $\overset{\downarrow}{}_S X \neq \emptyset$ has a minimum, then it is in ${}^{\vee}S$. Therefore, ${}^{\vee}S$ is the set made of the minimum of each part from $\mathcal{P}_{/(S,\leq)}$ in the upper closure of S. In particular, we have $S \subseteq {}^{\vee}S$. Yet, parts that do not contain any element of S may not have a minimum. Thus, it may be necessary to check the existence of all these minima before anything. It is equivalent to checking that ${}^{\vee}S$ is an upper subsemilattice of P, i.e. a subset of P for which the supremum (as defined in P) of every nonempty subset exists in it. For instance, if P is itself an upper semilattice, then all these minima exist.

The elements of ${}^{\vee}\mathrm{supp}(f)$ are what we call *focal points*[2].

Example 2.3.3.1. Let $(P, \leq) = (2^\Omega, \subseteq)$, where $\Omega = \{a, b, c\}$. In this partially ordered set, the supremum operator $\vee$ is the union operator $\cup$. Let m be a mass function such that $m(\Omega) = 0.1$, $m(\{a, b\}) = 0.1$, $m(\{b, c\}) = 0.2$ and $m(\{a\}) = 0.6$. We have $\mathrm{supp}(m) = \{\Omega, \{a, b\}, \{b, c\}, \{a\}\}$. It is easy to see that the union of any selection of support elements gives another support element. Therefore, we have ${}^{\vee}\mathrm{supp}(m) = \mathrm{supp}(m)$.

It may help, for one that is familiar with DST, to notice that ${}^{\vee}\mathrm{supp}(m) = \mathrm{supp}(m')$, where $m' = \bigcirc\!\!\!\!\bigcirc_{s \in \mathrm{supp}(m)} m$ and $\bigcirc\!\!\!\!\bigcirc$ is the disjunctive fusion operator.

Example 2.3.3.2. Taking back Example 2.3.3.1, but looking at the dual closure operator ${}^{\wedge}\cdot$, we get in particular $\bigcap\{\{a, b\}, \{b, c\}\} = \{b\}$ and $\bigcap\{\{b\}, \{a\}\} = \emptyset$. We have ${}^{\wedge}\mathrm{supp}(m) = \{\Omega, \{a, b\}, \{b, c\}, \{a\}, \{b\}, \emptyset\} = \mathrm{supp}(m) \cup \{\{b\}, \emptyset\}$. This meet-closed subset of P contains the focal points of m in $(2^\Omega, \supseteq)$.

It may help, for one that is familiar with DST, to notice that ${}^{\wedge}\mathrm{supp}(m) = \mathrm{supp}(m')$, where $m' = \bigcirc\!\!\!\!\bigcirc_{s \in \mathrm{supp}(m)} m$ and $\bigcirc\!\!\!\!\bigcirc$ is the conjunctive fusion operator.

By definition of a level partition (Definition 2.3.2.1), we get Property 2.3.3.1, which determines the image through f and g of all elements of P that are not focal points.

Property 2.3.3.1 (*Images of non focal points*)**.** For any upper subsemilattice ${}^{\vee}S$ of P such that ${}^{\vee}S \supseteq \mathrm{supp}(f)$, and for any element $y \notin {}^{\vee}S$, we have $f(y) = 0$. Also, if $y \in \uparrow {}^{\vee}S$, then $g(y) = g(s)$, where y *covers* s in ${}^{\vee}S$, i.e. s is the maximum among the elements of ${}^{\vee}S$ lower than y. Otherwise, $g(y) = 0$.

Example 2.3.3.3. Taking back m from Example 2.3.3.1 and the implicability function b from Example 1.3.1.1, we see that the elements from 2^Ω that are not focal points are $\{a, c\}$, $\{c\}$, $\{b\}$ and $\emptyset$. So, we have $m(\{a, c\}) = m(\{c\}) = m(\{b\}) = m(\emptyset) = 0, \quad b(\{a, c\}) = b(\{a\})$ and $b(\{c\}) = b(\{b\}) = b(\emptyset) = 0$.

Example 2.3.3.4. Taking back m from Example 2.3.3.2 and the commonality function q from Example 1.3.1.2, we must look this time at the minimum among the elements of ${}^{\wedge}\mathrm{supp}(m)$

[2]This name stands for an analogy in the field of optics: a focal point is the point of the spatial domain (P) where an image is formed by the intersection of rays coming from a distant source ($\mathrm{supp}(f)$) passing through a lense (subset of $\mathrm{supp}(f)$).

greater than some non focal point. We see that the elements from 2^Ω that are not focal points are $\{a, c\}$ and $\{c\}$. So, we have $m(\{a, c\}) = m(\{c\}) = 0$, $q(\{a, c\}) = q(\Omega)$ and $q(\{c\}) = q(\{b, c\})$.

Furthermore, thanks to Lemma 2.3.2.2 and Definition 2.3.3.1, we can now define (Definition 2.3.3.2) the extension of the Möbius function to be applied in our compact reformulation of the Möbius transform.

Definition 2.3.3.2 (*Möbius function extension*)**.** For any nonempty set of elements $S \subseteq P$ such that ${}^\vee S$ is an upper subsemilattice of P, we define the extension $\eta_{S,\leq,P} : {}^\vee S \times P \to \mathbb{Z}$ of the Möbius function $\mu_{{}^\vee S,\leq} : {}^\vee S \times {}^\vee S \to \mathbb{Z}$ as follows:

For any part $X \in \mathcal{P}_{/(S,\leq)}$ such that $X \subseteq \uparrow S$ and for any $y \in P$ where $\bigwedge X < y$,

$$\eta_{S,\leq,P}\left(\bigwedge X, y\right) = \eta_{S,\leq,P}(X, y) = -\sum_{\substack{Z \in \mathcal{P}_{/(S,\leq)} \\ \bigwedge X < \bigwedge Z \leq y}} \eta_{S,\leq,P}\left(\bigwedge Z, y\right),$$

which is equivalent to stating that for any $(s, y) \in {}^\vee S \times P$ where $s < y$,

$$\eta_{S,\leq,P}(s, y) = -\sum_{\substack{p \in {}^\vee S \\ s < p \leq y}} \eta_{S,\leq,P}(p, y), \tag{2.4}$$

with $\eta_{S,\leq,P}(s, s) = 1$.

The final part of this section consists in proposing our so-called *Efficient Möbius inversion formula* in the form of Theorem 2.3.3.1, which exploits the compact reformulation of Lemma 2.3.2.1, the focal points of Definition 2.3.3.1 and the extended Möbius function defined in Definition 2.3.3.2.

Theorem 2.3.3.1 (*Efficient Möbius inversion formula*)**.** *For any upper subsemilattice ${}^\vee S$ of P such that ${}^\vee S \supseteq supp(f)$, we have $\forall y \in P$,*

$$f(y) = \sum_{\substack{s \in {}^\vee S \\ s \leq y}} g(s) \,.\, \eta_{S,\leq,P}(s, y) \tag{2.5}$$

Proof. Eq. 2.5 is a simple reformulation of Eq. 2.2 from Lemma 2.3.2.1 with the focal points of Definition 2.3.3.1 and the extended Möbius function η of Definition 2.3.3.2 (given Lemma 2.3.2.2), combined with the fact that ${}^\vee({}^\vee S) = {}^\vee S$, which also means that $\mathcal{P}_{/({}^\vee S,\leq)} = \mathcal{P}_{/(S,\leq)}$. ■

Notice that S in fact need not contain $\text{supp}(f)$, as long as ${}^\vee S$ does. So, for example, ${}^\vee S$ can be a sublattice of P verifying ${}^\vee S \supseteq \text{supp}(f)$, with $S = I \cup \{\bigwedge I\}$, where I is the set containing the join-irreducible elements of ${}^\vee S$, as is the case with the *lattice support* from [53], where S may not contain $\text{supp}(f)$. Also, note that $\mathcal{P}_{/({}^\vee S,\leq)} = \mathcal{P}_{/(S,\leq)}$ since ${}^\vee({}^\vee S) = {}^\vee S$, which means that all computations can be made only based on the upper semillatice ${}^\vee S$, without actually having to determine any set S.

Example 2.3.3.5. Taking back Example 1.3.2.1, we get that for any focal point $s \in {}^\vee\text{supp}(m)$, the Möbius transform m of b in $(2^\Omega, \subseteq)$ is the Möbius transform of b in $({}^\vee\text{supp}(m), \subseteq)$, i.e. noting $S = \text{supp}(m)$, we have

$$\forall y \in {}^\vee S, \quad m(y) = \sum_{\substack{s \in {}^\vee S \\ s \subseteq y}} b(s) \,.\, \mu_{{}^\vee S,\subseteq}(s, y)$$

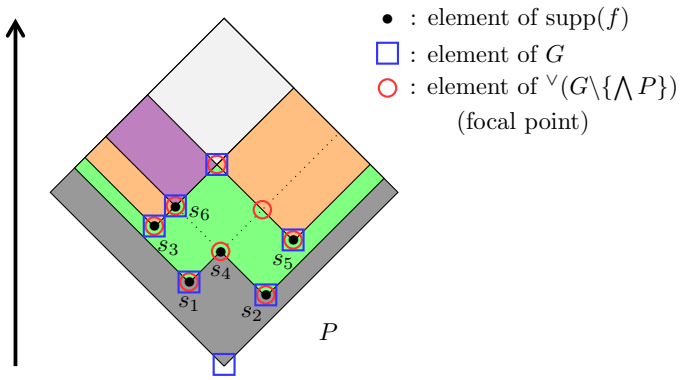


Figure 2.3: Example of image partition $\mathcal{G}$ from Fig. 2.1. Each color represents an image through g. Suppose that g is the zeta transform of f in $(P, \leq)$ and that the image through g of the gray part is 0. Then, this partition is the consequence of the support of f displayed in the figure with the following conditions: $f(s_2) = f(s_1)$, $f(s_4) = -f(s_1)$ and $f(s_3) = f(s_5)$. In this case, the image through g of the elements in the green part is $f(s_1)$, the one in the orange part is $f(s_1) + f(s_3)$, the one in the violet part is $f(s_1) + f(s_3) + f(s_6)$ and the one in the white part is $f(s_1) + 2.f(s_3) + f(s_6)$. Notice that not all elements from $\text{supp}(f)$ are minimal elements of a part. However, they are all contained in the join-closure of $G\backslash\{\bigwedge P\}$, noted ${}^{\vee}(G\backslash\{\bigwedge P\})$, where $G = \bigcup_{X\in\mathcal{G}} \min(X)$. In fact, we even have ${}^{\vee}(G\backslash\{\bigwedge P\}) = {}^{\vee}\text{supp}(f)$.

Example 2.3.3.6. Similarly, taking back Example 1.3.2.2, we get that for any focal point $s \in {}^{\wedge}\text{supp}(m)$, the Möbius transform m of q in $(2^{\Omega}, \supseteq)$ is the Möbius transform of q in $({}^{\wedge}\text{supp}(m), \supseteq)$, i.e. noting $S = \text{supp}(m)$, we have

$$\forall y \in {}^{\wedge}S, \quad m(y) = \sum_{\substack{s\in{}^{\wedge}S \\ s\supseteq y}} q(s) \,.\, \mu_{{}^{\wedge}S,\supseteq}(s, y)$$

2.3.4 Ways to compute focal points

This section focuses on methods allowing one to compute focal points in an efficient way. Property 2.3.4.1 describes a direct scheme for computing the join-closure ${}^{\vee}S$ of any set $S \subseteq P$, while Theorem 2.3.4.1 indicates how to find ${}^{\vee}\text{supp}(f)$ based on $\mathcal{G}$ alone.

Property 2.3.4.1 (*Computing the join-closure of S directly*)**.** Every element in ${}^{\vee}S$ can be described as either y, where $y \in S$, or $s \vee y$, where $s \in {}^{\vee}S$. Doing so, all focal points can be found through a double loop: the outer one iterating through S and the inner one dynamically iterating through already found focal points (starting with S). Therefore, all elements of ${}^{\vee}S$ can be found in $O(|{}^{\vee}S|.|S|)$. It is even possible to further optimize since for any $x, y \in P$, if $x \leq y$ or $x \geq y$, then $x \vee y = y$ or $x \vee y = x$, which means in our case that it is useless to compute the supremum of two elements if there exists an order between them.

This Property 2.3.4.1 can be used to compute focal points directly from $\text{supp}(f)$, but $\text{supp}(f)$ is not always known beforehand. One may want to find all focal points from g alone. The problem is that for any two elements $x, y \in P$, $g(x) \neq g(y) \Rightarrow \overset{\downarrow}{S}x \neq \overset{\downarrow}{S}y$, but the converse is not true. Therefore, some elements from ${}^{\vee}\text{supp}(f)$ may not be directly apparent in the compact definition of g through its image partition $\mathcal{G}$. See Fig. 2.3. The following Theorem 2.3.4.1 explains how focal points can be found from $\mathcal{G}$ alone, in spite of this fact.

Theorem 2.3.4.1 (*Finding ${}^{\vee}supp(f)$ from $\mathcal{G}$*)**.** *Let Y be the set made of every supremum $x \vee a$, where $x \in G\backslash M$ and $a \in M$, where $M = \min(P) \cap \overline{supp(g)}$ and $G = \bigcup_{X\in\mathcal{G}} \min(X)$.*

If $^{\vee}G$ is an upper subsemilattice of P (e.g. if P is itself an upper semilattice), then we have $^{\vee}supp(f) \subseteq {}^{\vee}(Y \cup G \backslash M) \subseteq {}^{\vee}G$.

Proof. See Appendix A.3. ■

Consequently, one can find all focal points of f and g, from either $\text{supp}(f)$ or the minimal elements of $\mathcal{G}$, either using a variation of the procedure described in Property 2.3.4.1 or by building a sublattice L of P as done in Proposition 2 of [53]. This latter method, of complexity $O(n.|L|)$, potentially generates more elements but features a better worst-case complexity $O(n.|P|)$, where n is the number of join-irreducible elements of L. In addition, this lattice L contains both $^{\vee}\text{supp}(f)$ and $^{\wedge}\text{supp}(f)$.

It is also worth noting that we do not even have to compute the join-closure of G if f is nonnegative, since there can be no compensation of images through f. This means that the image through g of a focal point cannot be equal to the one of an element it covers (See Appendix A.3). Hence Property 2.3.4.2.

Property 2.3.4.2 (*Finding $^{\vee}supp(f)$ from $\mathcal{G}$ when f is nonnegative*)**.** If $^{\vee}\text{supp}(f)$ is an upper subsemilattice of P (e.g. if P is itself an upper semilattice) and if f is nonnegative, then we have $^{\vee}\text{supp}(f) \subseteq G \backslash M$, where $M = \min(P) \cap \overline{\text{supp}(g)}$ and $G = \bigcup_{X \in \mathcal{G}} \min(X)$.

Example 2.3.4.1. The mass function m is required to be nonnegative. Therefore, no matter the partition defining the implicability function b and the commonality function q, the partition for b always contains its focal points among the minimal elements of its parts, and the partition for q (the dual of b) always contains its focal points among the maximal elements of its parts (provided that q and b indeed correspond to mass functions). However, the disjunctive weight function v from Example 1.3.3.1 and the conjunctive weight function w from Example 1.3.3.2, which are multiplicative Möbius transforms in DST, are allowed to have values below 1. Thus, they do not satisfy the multiplicative equivalent of Property 2.3.4.2.

2.3.5 Focal points for both additive and multiplicative Möbius transforms

In this section, we will see which place takes the focal points of two functions $f : P \to \mathbb{R}$ and $h : P \to \mathbb{R}^*$ when they are linked by the following equation:

$$\forall y \in P, \quad g(y) = \sum_{x \leq y} f(x) = \prod_{x \leq y} h(x) \tag{2.6}$$

We will not consider the case where h is allowed to have null images since they make impossible the inversion of products. Indeed, any element of P that is greater than an element associated with a null value through h is guaranteed to get a null image through the zeta transform g, no matter what image it has through h. Therefore, retrieving their image through h from g becomes impossible. The following Property 2.3.5.1 reflects this constraint.

Property 2.3.5.1. For any minimal element $y \in \min(P)$, Eq. 2.6 gives us $g(y) = f(y) = h(y)$. In particular, if $y \notin \text{supp}(f)$, then $g(y) = f(y) = h(y) = 0$, which we forbid. Thus, we have $\min(P) \subseteq \text{supp}(f)$.

Example 2.3.5.1. Taking back the implicability function b from Example 1.3.1.1 and the disjunctive weight function v from Example 1.3.3.1, we have:

$$\forall y \in 2^{\Omega}, \quad b(y) = \sum_{x \subseteq y} m(x) = \prod_{x \subseteq y} v(x)^{-1},$$

which implies that $m(\emptyset) = v(\emptyset)^{-1}$ and so $m(\emptyset) \neq 0$, which is in accordance with Property 2.3.5.1.

Example 2.3.5.2. Taking back the commonality function q from Example 1.3.1.2 and the conjunctive weight function w from Example 1.3.3.2, we have:

$$\forall y \in 2^\Omega, \quad q(y) = \sum_{x \supseteq y} m(x) = \prod_{x \supseteq y} w(x)^{-1},$$

which implies that $m(\Omega) = w(\Omega)^{-1}$ and so $m(\Omega) \neq 0$, which is in accordance with Property 2.3.5.1.

From this Property 2.3.5.1 and Theorem 2.3.4.1, we can link the focal points of f and h, and thus their respective *support* elements, in Corollary 2.3.5.1.

Corollary 2.3.5.1 (*Link between* $^\vee supp(f)$ *and* $^\vee supp(h-1)$)**.** *If either* $^\vee(supp(h-1) \cup \min(P))$ *or* $^\vee supp(f)$ *is an upper subsemilattice of* P*, then we have*

$$^\vee supp(f) = {}^\vee(supp(h-1) \cup \min(P))$$

Proof. See Appendix A.4. ■

In particular, $^\vee\text{supp}(f) = \{\bigwedge P\} \cup {}^\vee\text{supp}(h-1)$, if P has a minimum.

Example 2.3.5.3. Taking back the mass function m, the disjunctive weight function v and the conjunctive weight function w from Example 2.3.5.1 and Example 2.3.5.2, we get $^\vee\text{supp}(m) = \{\emptyset\} \cup {}^\vee\text{supp}(v-1)$ and $^\wedge\text{supp}(m) = \{\Omega\} \cup {}^\wedge\text{supp}(w-1)$.

Finally, Theorem 2.3.5.1 proposes formulas to track the information flow from h to g and f, i.e. the effects of changing one element in $\text{supp}(h-1)$ on f and g. This will be used in the application of section 2.4.3. The reversed flow, from f to h, is not displayed here as it does not simplify well.

Theorem 2.3.5.1 (*Information flow from* h *to* g *and* f)**.** *Let* h' *be equal to* h *everywhere, except for the image of some* $x \in P$*. Also, let* f' *and* g' *be the functions corresponding to* h' *so that they satisfy Eq. 2.6 in place of respectively* f *and* g*. Then we have:*

$$\forall y \in P, \quad g'(y) = \begin{cases} g(y) \cdot \frac{h'(x)}{h(x)} & \text{if } x \leq y \\ g(y) & \text{otherwise} \end{cases} \tag{2.7}$$

and for any upper subsemilattice $^\vee S$ *of* P *such that* $^\vee S \supseteq supp(f) \cup \{x\}$*,*

$$\forall y \in P, \quad f'(y) = \begin{cases} 0 & \text{if } y \notin {}^\vee S \\ f(y) + \left[\frac{h'(x)}{h(x)} - 1\right] . f_{\uparrow x}(y) & \text{if } x \leq y \\ f(y) & \text{otherwise} \end{cases}$$

where $f_{\uparrow x} : \uparrow x \to \mathbb{R}$ *is the Möbius transform of* g *in* $(\uparrow x, \leq)$*, i.e.* $\forall y \in \uparrow x$*,*

$$f_{\uparrow x}(y) = \sum_{\substack{s \in \uparrow x \\ s \leq y}} g(s) \cdot \mu_{\uparrow x, \leq}(s, y)$$

Proof. See Appendix A.5. ■

In particular, notice that if $x \in {}^\vee\text{supp}(f)$, then $^\vee\text{supp}(f) \supseteq \text{supp}(f) \cup \{x\}$.

2.3.6 Discussions

Several remarks can be made regarding g as input. Firstly, we only consider a compact definition of g in this article because the point is that it is possible to compute its Möbius transform even in a tremendously vast partially ordered set by avoiding to consider all elements of P. If g is defined by its image on every element of P, then the usual Möbius transform should be employed, as the search for focal points would require to check all elements of P at least once. This is all the more relevant since there are some algorithms like the FMT [43] that operate in $O(n.|P|)$, where P is here a lattice and n is the number of its join-irreducible elements, where we usually have $n \ll |P|$.

Secondly, if P is downward infinite, then there may be parts in $\mathcal{G}$ that do not have any minimal element. Since our method may need these minimal elements in order to find some potentially *hidden* support elements if f can have negative values, one might have to add a surrogate element for each downward infinite horizon in P. For example, if P contains an element that is greater than two infinite chains, then one should add two surrogate elements symbolizing downward infinity, one for each chain. Of course, this can only be possible if there is a finite number of downward infinite horizons. Finally, $|\text{supp}(f)|$ must be finite and $\mathcal{G}$ must have a finite number of parts. Otherwise, focal points cannot be determined.

2.3.7 From theory to practice

In practice, one could wonder how these formulas can be exploited. Actually, even though it is possible to do so, the Möbius function μ (or our extension η) is usually not evaluated when computing the Möbius transform of a function. It mostly serves a theoretical purpose. Instead, it is enough to consider the zeta transform g of a function f and to simply rewrite it in a recursive way to express f in terms of g and f:

$$\forall y \in P, \quad g(y) = \sum_{\substack{x \in S \\ x \leq y}} f(x) \quad \Leftrightarrow \quad f(y) = g(y) - \sum_{\substack{x \in S \\ x < y}} f(x) \tag{2.8}$$

where $S \supseteq \text{supp}(f)$. Then, there are two main ways to use these formulas: (i) naively, summing all terms for each element y, or (ii) efficiently, reusing partial sums common to multiple elements y. With (i), values of g can be computed independently, while (ii) requires the computation of the image through g of all elements of P. But, if one needs the value of g on every element y (i.e. the complete definition of g), whether it is because $|\text{supp}(f)|$ is close to $|P|$ or otherwise, then (ii) is much more efficient. The optimal method achieving (ii) for $P = 2^\Omega$ is the *Fast Möbius Transform* (FMT) [43], which has a time complexity in $O(N.2^N)$, where N is the size of the frame of discernment resulting from the best lossless coarsening[3] of Ω regarding $\text{supp}(f)$.

Our contribution to this is the proof that we do not necessarily need the value of g on all elements of P to define g and to define f from g (without knowledge about $\text{supp}(f)$). Our focal points constitute a subset of P that can be substantially smaller and is necessary and sufficient to define all zeta and Möbius transforms. As seen in Theorem 2.3.3.1, for any element of ${}^\vee S$ the Möbius transform of g in $(P, \leq)$ is the Möbius transform of g in $({}^\vee S, \leq)$. This means that we can work in the domain of our focal points instead of P and obtain the exact same results. This finding is mostly useful for (ii) as it exploits the structure of the domain. For this, we proposed variants of the FMT, called *Efficient Möbius Transformations* (EMT) [53], which work in any distributive lattice L and exploit the structure of its subsemilattices. Taking

[3] A lossless coarsened frame of discernment Ω' is a partition of the original set Ω, subject to this coarsening, such that every support element of the considered mass function defined on 2^Ω can be mapped into $2^{\Omega'}$. The best lossless coarsening results in the smallest Ω' possible (see [44]).

$L = 2^\Omega$, the complexity of the EMT is always lower than $O(N.2^N)$ and can be even lower than $O(N^2)$, e.g. if $\text{supp}(f)$ is a chain.

When using (i), only support elements are necessary in computations, but g is only completely defined when its image on all focal points is given (See Example 2.3.7.1), which is obvious considering its image partition $\mathcal{G}$. In addition, it is important to mention that changing even only one image through g of an element from $\text{supp}(f)$ may add or remove elements from $\text{supp}(f)$, but they will always be in $^\vee\text{supp}(f)$. Thus, manipulating g only on the elements of $\text{supp}(f)$ is highly unreliable. On the contrary, focal points can serve as data structure to completely define g and can be directly found from any compact definition of g, such as one by intervals.

Furthermore, $\text{supp}(f)$ may not be known before actually computing f from g. In this case, it is impossible to compute f with only elements from $\text{supp}(f)$. However, it is always possible to find the focal points of f from g.

See Example 2.3.7.2.

Example 2.3.7.1 (*Computing q from m*)**.** Let us take back Example 2.3.3.2, i.e. $\Omega = \{a, b, c\}$ and m is a mass function such that $m(\Omega) = 0.1$, $m(\{a, b\}) = 0.1$, $m(\{b, c\}) = 0.2$ and $m(\{a\}) = 0.6$. We have $\text{supp}(m) = \{\Omega, \{a, b\}, \{b, c\}, \{a\}\}$ and $^\wedge\text{supp}(m) = \text{supp}(m) \cup \{\{b\}, \emptyset\}$. From m, we get its commonality function q based on Eq. 2.1 with $\text{supp}(m)$ on its focal poins $^\wedge\text{supp}(m)$:

- $q(\Omega) = m(\Omega) = \mathbf{0.1}$
- $q(\{a, b\}) = m(\{a, b\}) + m(\Omega) = \mathbf{0.2}$
- $q(\{b, c\}) = m(\{b, c\}) + m(\Omega) = \mathbf{0.3}$
- $q(\{a\}) = m(\{a\}) + m(\{a, b\}) + m(\Omega) = \mathbf{0.8}$
- $q(\{b\}) = m(\{a, b\}) + m(\{b, c\}) + m(\Omega) = \mathbf{0.4}$
- $q(\emptyset) = m(\{a\}) + m(\{a, b\}) + m(\{b, c\}) + m(\Omega) = \mathbf{1}$

As already shown in Example 2.3.3.4, the rest of 2^Ω, which does not contain any focal point, is defined by Property 2.3.3.1: $q(\{a, c\}) = q(\Omega)$ and $q(\{c\}) = q(\{b, c\})$.

Example 2.3.7.2 (*Computing w from m*)**.** Now, from Example 2.3.7.1, suppose that we want to compute w. From Corollary 2.3.5.1, we know that $\{\Omega\} \cup {}^\wedge\text{supp}(w - 1) = {}^\wedge\text{supp}(m)$. In addition, we have $m(\Omega) = w(\Omega)^{-1} = 0.1 \neq 1$, which means that $\Omega \in \text{supp}(w - 1)$ and so $^\wedge\text{supp}(w - 1) = {}^\wedge\text{supp}(m)$. Then, adapting Eq. 2.8 to the multiplicative form, we get:

$$\forall y \in 2^\Omega, \quad w(y) = \begin{cases} 1 & \text{if } y \notin {}^\wedge\text{supp}(m) \\ q(y)^{-1}. \displaystyle\prod_{\substack{s \in {}^\wedge\text{supp}(m) \\ s \supset y}} w(s)^{-1} & \text{otherwise} \end{cases}$$

This result and its dual for the disjunctive weight function v were already the conclusion of one of our previous papers [42]. So, from this, we get the conjunctive weight function w:

- $w(\Omega) = q(\Omega)^{-1} = \mathbf{10}$
- $w(\{a, b\}) = [q(\{a, b\}) \,.\, w(\Omega)]^{-1} = \mathbf{0.5}$
- $w(\{b, c\}) = [q(\{b, c\}) \,.\, w(\Omega)]^{-1} = \mathbf{\frac{1}{3}}$
- $w(\{a\}) = [q(\{a\}) \,.\, w(\Omega) \,.\, w(\{a, b\})]^{-1} = \mathbf{0.25}$
- $w(\{b\}) = [q(\{b\}) \,.\, w(\Omega) \,.\, w(\{a, b\}) \,.\, w(\{b, c\})]^{-1} = \mathbf{1.5}$
- $w(\emptyset) = [q(\emptyset) \,.\, w(\Omega) \,.\, w(\{a, b\}) \,.\, w(\{b, c\}) \,.\, w(\{a\}) \,.\, w(\{b\})]^{-1} = \mathbf{1.6}$

Since all these images are different from 1, we get here[4] that $^\wedge\text{supp}(m) = \text{supp}(w - 1)$.

[4]From experience, all focal points of m are most of the time also elements of the support of $w - 1$ or $v - 1$, to the point where we in fact never witnessed a case in which this was not true.

2.4 Implications for Dempster-Shafer Theory

In DST, we work with $P = 2^\Omega$, which is a lattice, i.e. a set of which every nonempty subset has both a supremum and an infimum. Doing so, all focal points always exist for both the relations $\subseteq$ and $\supseteq$. We will see in section 2.4.1 how our *Efficient* Möbius inversion formula impacts almost all representations of DST and how to fuse belief sources using focal points. In section 2.4.2, we will propose a generalization of the conjunctive decomposition of evidence to benefit from fusion rules such as the Cautious one [23], even when the considered mass function is dogmatic. Finally, section 2.4.3 will provide formulas to study the impact of each decomposition weight on the corresponding mass function.

2.4.1 Efficient representations in Dempster-Shafer Theory

In DST, the mass function m is central. It is considered as a generalization of the discrete Bayesian probability distribution. It is defined within the bounds of two constraints [39]: one is that m is nonnegative, and the other is

$$\sum_{y \in 2^\Omega} m(y) = 1. \tag{2.9}$$

However, other representations are often used to analyze or fuse mass functions. With the exception of the *pignistic probability* representation, all of them are linked to the zeta and Möbius transforms. We already introduced them in our examples throughout this article:

the implicability[5] function b from Example 1.3.1.1, the commonality function q from Example 1.3.1.2, the disjunctive weight function v from Example 1.3.3.1 and the conjunctive weight function w from Example 1.3.3.2.

In addition, Example 2.4.1.1 displays a classic use case demonstrating how the fusion of two functions of same type in DST can be performed with focal points.

Example 2.4.1.1 (*Efficient combination with Dempster's fusion rule*)**.** Dempster's combination rule $\oplus$ is defined as the normalized conjunctive rule $\textcircled{\cap}$, i.e. $\forall y \in 2^\Omega \ / \ y \neq \emptyset$:

$$\begin{aligned}
(m_1 \oplus m_2)(y) &= \frac{1}{K}(m_1 \textcircled{\cap} m_2)(y) \\
&= \frac{1}{K} \sum_{\substack{s_1 \cap s_2 = y \\ s_1 \in \text{supp}(m_1) \\ s_2 \in \text{supp}(m_2)}} m_1(s_1) \ . \ m_2(s_2) \qquad (2.10) \\
&= \frac{1}{K} \sum_{x \supseteq y} q_1(x).q_2(x).\mu_{2^\Omega,\supseteq}(x,y) \qquad (2.11)
\end{aligned}$$

where $K = 1 - (m_1 \textcircled{\cap} m_2)(\emptyset)$. Eq. 2.10 can be used by an evidence-based algorithm in $O(|\text{supp}(m_1)| \ . \ |\text{supp}(m_2)|)$, observing that each support element in $\text{supp}(m_{12})$ of the combined mass function m_{12} is defined as the intersection $s_1 \cap s_2$ of a pair of support elements, where $s_1 \in \text{supp}(m_1)$ and $s_2 \in \text{supp}(m_2)$. Alternatively, Eq. 2.11 can be used by powerset-based algorithms such as the FMT (this application has been tackled in [43]) in $O(N.2^N)$, where N is here the size of the frame of discernment resulting from the best lossless coarsening of Ω

[5] This representation alone is linked to two other ones: the belief function *Bel*, which is equal to b when setting $m(\emptyset) = 0$, and the plausibility function $Pl(y) = 1 - b(\overline{y})$, $\forall y \in 2^\Omega$.

regarding $\text{supp}(m_1) \cup \text{supp}(m_2)$. It consists in separately computing q_1 and q_2 with the FMT and then multiplying them element-wise to get q_{12}, before using again the FMT to compute m_{12} from q_{12}. This second approach is useful when almost all images of m_{12} are required, e.g. when $|\text{supp}(m_1)| \,.\, |\text{supp}(m_2)|$ is of significantly higher magnitude than $N.2^N$.

We can also reformulate Eq. 2.11 with focal points in the light of Theorem 2.3.3.1 and obtain a hybrid approach:

$$(m_1 \bigoplus m_2)(y) = \begin{cases} \dfrac{1}{K} \displaystyle\sum_{\substack{s \in {}^\wedge S \\ s \supseteq y}} q_1(s).q_2(s).\mu_{{}^\wedge S,\supseteq}(s,y) & \text{if } y \in {}^\wedge S \\ 0 & \text{otherwise} \end{cases} \tag{2.12}$$

where ${}^\wedge S \supseteq \text{supp}(m_{12})$ and $P = 2^\Omega$. Eq. 2.12 can be exploited in less than $O(N.2^N)$ with the EMT using the fact that ${}^\wedge(\text{supp}(m_1) \cup \text{supp}(m_2)) \supseteq \text{supp}(m_{12})$. If support elements are not known, one can use the fact that ${}^\wedge(\text{supp}(m_1) \cup \text{supp}(m_2)) = {}^\wedge({}^\wedge\text{supp}(m_1) \cup {}^\wedge\text{supp}(m_2))$.

2.4.2 Generalized decompositions of evidence

Some useful fusion rules for belief sources apply only to the conjunctive decomposition of evidence [23]. Such is the case for the Cautious conjunctive rule [23], which is used when these sources are not independent. Yet, this decomposition can only be computed for mass functions m such that $\Omega \in \text{supp}(m)$. Here, we propose a generalization of this decomposition that works for any mass function m such that $\bigcup \text{supp}(m) \in \text{supp}(m)$. This generalization is given by Definition 2.4.2.1. A similar definition can be given for its dual, namely the disjunctive decomposition, for any mass function such that $\bigcap \text{supp}(m) \in \text{supp}(m)$.

Generalization

For any mass function m such that $\Omega \in \text{supp}(m)$, the conjunctive decomposition is defined as

$$m = \bigcirc\!\!\!\!\bigcap_{A \subset \Omega} A^w \tag{2.13}$$

where each A^w is a generalized simple mass function[6], defined as

$$\forall A \subset \Omega,\ \forall B \subseteq \Omega, \quad A^w(B) = \begin{cases} 1 - w(A) & \text{if } B = A \\ w(A) & \text{if } B = \Omega \\ 0 & \text{otherwise} \end{cases}$$

and w is the conjunctive weight function, i.e. the inverse of the multiplicative Möbius transform of q in $(2^\Omega, \supseteq)$, i.e. $w(A) = \displaystyle\prod_{\substack{B \subseteq \Omega \\ B \supseteq A}} q(B)^{(-1)^{|B|-|A|+1}}$, where q is the commonality function associated with m, i.e. the zeta transform of m in $(2^\Omega, \supseteq)$.

In parallel, the Möbius inversion theorem linking q and w gives us

$$\forall y \in 2^\Omega, \quad q(y) = \sum_{x \supseteq y} m(y) = \prod_{x \supseteq y} w(y)^{-1}, \tag{2.14}$$

which implies that $w(\Omega) = q(\Omega)^{-1} = m(\Omega)^{-1}$, hence $m(\Omega) \neq 0$. Even if we take the multiplicative Möbius transform of q instead of its inverse, we must have $m(\Omega) \neq 0$ so that the Möbius

[6] We use abusively the term mass function in this article for the sake of simplicity. Actually, A^w is a generalized simple basic belief assignment (GSBBA) (See [41] for a more accurate terminology).

inversion theorem can apply, as explained in section 2.3.5. Thus, in order to apply combination rules that are only defined on the conjunctive decomposition, such as the Cautious conjunctive rule, it was argued [41] that one should only use mass functions m satisfying $m(\Omega) \neq 0$. In practice, this is often done artificially, by discounting, i.e. multiplying all masses by some factor $\alpha \in (0, 1)$ and assigning the complement to 1 to $m(\Omega)$.

Nevertheless, assigning a mass to Ω is not ideal: when fusing by conjunction two mass functions that have different focal elements but Ω among them, it appears that no hypothesis (i.e. no focal element) is discarded in the resulting mass function. It contains the focal elements of the two original mass functions, as well as their pair-wise intersections. This means that hypotheses can only accumulate, making the number of focal elements (and so the number of focal points) explode in vast domains when many fusions of belief sources occur. Therefore, it is preferable to only assign masses to actual tangible hypotheses, instead of always considering that all hypotheses are possible even if unlikely. It is more stable, more accurate and allows for more use cases.

It turns out that it is possible to avoid the constraint $m(\Omega) \neq 0$ simply by using as weight function w the inverse of the multiplicative Möbius transform of q in $(\downarrow \text{supp}(m), \supseteq)$ instead of $(2^\Omega, \supseteq)$ (See section 2.3.5). But, how to reflect this in the conjunctive decomposition? Notice that Eq. 2.13 is in fact equivalent to:

$$\forall y \subseteq C, \quad q(y) = \begin{cases} \prod\limits_{x \neq C} w(x) & \text{if } y = C \\ q(C). \prod\limits_{\substack{x \neq C \\ x \supseteq y}} w(x)^{-1} & \text{otherwise} \end{cases}, \tag{2.15}$$

where $C = \Omega$, which means that all images of q are determined by Eq. 2.14, except for the one on C which exploits the fact that the product of all weights is normalized to 1, due to Eq. 2.14 and Eq. 2.9. Now, C can be something else, as long as the conjunctive combination of simple mass functions as in Eq. 2.13 leads to this form, which implies that C both is less than or equal to the element replacing Ω in each simple mass function, while not being less than A (to satisfy the first line in Eq. 2.15), and is a maximal element of the domain on which w is defined (to satisfy $q(C) = w(C)^{-1}$, imposed by Eq. 2.14). In addition, since the second line of this equation is only valid when y is less than C, we get that C must be above all elements of $\text{supp}(m)$ for the conjunctive decomposition to account for all non-zero q images defining m. Combining this with the fact that w must be computed in $(\downarrow \text{supp}(m), \supseteq)$, we get that $C = \bigcup \text{supp}(m)$ and $C \in \text{supp}(m)$, which is in accordance with the other aforementioned conditions. This leads us to Definition 2.4.2.1.

Definition 2.4.2.1. For any mass function m such that $\bigcup \text{supp}(m) \in \text{supp}(m)$, we define our generalized conjunctive decomposition as

$$m(B) = \begin{cases} \left(\bigcirc\!\!\!\!\bigcirc_{A \subset C} A^w_C \right)(B) & \text{if } B \subseteq C \\ 0 & \text{otherwise} \end{cases}$$

where $C = \bigcup \text{supp}(m)$ and

$$\forall A \subset C,\ \forall B \subseteq C, \quad A^w_C(B) = \begin{cases} 1 - w^C(A) & \text{if } B = A \\ w^C(A) & \text{if } B = C \\ 0 & \text{otherwise} \end{cases}$$

and $w^C(A) = \prod\limits_{\substack{B \subseteq C \\ B \supseteq A}} q(B)^{(-1)^{|B|-|A|+1}}$.

If $\Omega \in \text{supp}(m)$, then Definition 2.4.2.1 yields the classic conjunctive decomposition. This generalized decomposition is unique, as the classic one, due to the fact that w^C is tied to q by the Möbius inversion theorem.

When supp(m) has no maximum

Similar to what was proposed in the past, it is possible to discount m and assign the complement to 1 to any element between $\bigcup\text{supp}(m)$ and Ω, if $\bigcup\text{supp}(m)$ is not already in $\text{supp}(m)$. We argue that $\bigcup\text{supp}(m)$ should be chosen since it is the superset that supports the least hypotheses outside $\text{supp}(m)$ and so the one that biases m the least.

Furthermore, thanks to Theorem 2.3.3.1 and Corollary 2.3.5.1, we can even state the following Corollary 2.4.2.1 about the resulting decomposition weights of such a discounting procedure.

Corollary 2.4.2.1. *For any mass function m such that $\bigcup \text{supp}(m) \notin \text{supp}(m)$, discounting it and assigning the complement to Ω gives the same decomposition weights as discounting it and assigning the complement to $\bigcup \text{supp}(m)$.*

Proof. See Appendix A.6. ■

Fusion of generalized decompositions

Given Corollary 2.4.2.1, for two mass functions m_1 and m_2 such that neither $\text{supp}(m_1)$ nor $\text{supp}(m_2)$ has a maximum and $\bigcup\text{supp}(m_1) = \bigcup\text{supp}(m_2)$, discounting them and assigning the complement to Ω gives the same decomposition weights as discounting them and assigning the complement to $\bigcup\text{supp}(m_1)$. Consequently, any fusion operator defined on decomposition weights produces the same results in our generalized decomposition as in the classic decomposition, except that the mass normally added to Ω is added to $\bigcup\text{supp}(m_1)$ instead, which does not give credit to more hypotheses than needed and is more stable. In addition, this allows for new interesting cases when $\bigcup\text{supp}(m_1) \neq \bigcup\text{supp}(m_2)$.

In this case, before applying any fusion operator, it is necessary to define a common domain for the resulting conjunctive decomposition. For combination rules based on the conjunction of two belief sources (i.e. both sources are considered reliable), such as Dempster's rule, the Cautious conjunctive rule, etc, domains must be intersected and evidence projected on this intersection. This common domain is simply 2^C, where $C = \bigcup\text{supp}(m_1) \cap \bigcup\text{supp}(m_2)$. Then, concerning the projection of evidence on 2^C, this consists in adding masses defined outside 2^C to their intersection with this domain. Fortunately, this is already what the commonality function q does. Indeed, the projection of any mass function m onto 2^C is obtained by transferring, i.e. adding, the mass on all elements $B \subseteq \Omega$ to the mass of $B \cap C$. This projection, noted $m_{\downarrow C}$, is itself a mass function since, by construction, it is as nonnegative as m and the sum of its images remains unchanged. Notice that the zeta transform of $m_{\downarrow C}$ in $(2^C, \supseteq)$ is equal to q on 2^C since $B \supseteq A$ is equivalent to $B \cap A = A$ and since for any element $A \subseteq C$, we have $A \cap (B \cap C) = A \cap B$. Thus, we can use the commonality functions q_1 and q_2, which are respectively the zeta transforms of m_1 in $(2^{T_1}, \supseteq)$ and m_2 in $(2^{T_2}, \supseteq)$, where $T_1 \supseteq \bigcup\text{supp}(m_1)$ and $T_2 \supseteq \bigcup\text{supp}(m_2)$. Then, it only remains to check that $q_1(C) \neq 0$ and $q_2(C) \neq 0$, which is equivalent to verifying that $C \in \downarrow \text{supp}(m_1)$ and $C \in \downarrow \text{supp}(m_2)$. The aforementioned discounting must be employed[7] on any commonality function that does not verify this condition. Now, we can finally fuse m_1 and m_2 by applying any conjunctive fusion operator to the weight functions w_1^C and w_2^C associated with respectively q_1 and q_2 in $(2^C, \supseteq)$. If Theorem 2.3.3.1 is used to compute w_1^C and w_2^C, notice that the focal points of w_1^C are the pair-wise intersections $C \cap s_1$, where $s_1 \in {}^\wedge\text{supp}(m_1)$, and the ones of w_2^C are the pair-wise intersections $C \cap s_2$, where $s_2 \in {}^\wedge\text{supp}(m_2)$.

[7] Small but necessary approximation in order to apply the fusion operator.

The weight functions w_1^C and w_2^C correspond to $m_{1\downarrow C}$ and $m_{2\downarrow C}$, which are mass functions. Therefore, any fusion operator defined in the classic conjunctive decomposition, such as the Cautious conjunctive rule, is valid in this generalization, as if we had $\Omega = C$, and so can be applied to the weights of w_1^C and w_2^C (except the ones on C, in the same way that it is not applied to the weights on Ω in the classic decomposition).

Example 2.4.2.1. Consider $\Omega = \{a, b, c, d\}$ and let m_1 be a mass function such that $m_1(\{a, b\}) = 0.2$, $m_1(\{b, c\}) = 0.2$ and $m_1(\{a\}) = 0.6$. We have $\bigcup \text{supp}(m_1) = \{a, b, c\}$ and $^\wedge\text{supp}(m_1) = \text{supp}(m_1) \cup \{\{b\}, \emptyset\}$. Also, let m_2 be a mass function such that $m_2(\{b, c\}) = 0.3$, $m_2(\{c, d\}) = 0.1$ and $m_2(\{c\}) = 0.6$. We have $\bigcup \text{supp}(m_2) = \{b, c, d\}$ and $^\wedge\text{supp}(m_2) = \text{supp}(m_2)$. Suppose we want to fuse m_1 and m_2 with the Cautious conjunctive rule. We pose $C = \bigcup \text{supp}(m_1) \cap \bigcup \text{supp}(m_2) = \{b, c\}$. We already have $C \in \downarrow \text{supp}(m_1)$ and $C \in \downarrow \text{supp}(m_2)$, so no discounting needed. We even have $C \in \text{supp}(m_1)$ and $C \in \text{supp}(m_2)$, which means that the focal points of w_1^C are $2^C \cap {}^\wedge\text{supp}(m_1) = \{\{b, c\}, \{b\}, \emptyset\}$ and that the ones of w_2^C are $2^C \cap {}^\wedge\text{supp}(m_2) = \{\{b, c\}, \{c\}\}$. Moreover, we have $q_1(\{b, c\}) = 0.2$, $q_1(\{b\}) = 0.4$ and $q_1(\emptyset) = 1$, where q_1 is the commonality function associated with m_1, and $q_2(\{b, c\}) = 0.3$ and $q_2(\{c\}) = 1$, where q_2 is the commonality function associated with m_2. We get:

$w_1^C(\{b, c\}) = q_1(\{b, c\})^{-1} = \mathbf{5}$, $w_1^C(\{b\}) = \left[q_1(\{b\}) \,.\, w_1^C(\{b, c\})\right]^{-1} = \mathbf{0.5}$, $w_1^C(\emptyset) = \left[q_1(\emptyset) \,.\, w_1^C(\{b, c\}) \,.\, w_1^C(\{b\})\right]^{-1} = \mathbf{0.4}$ and $w_2^C(\{b, c\}) = q_2(\{b, c\})^{-1} = \frac{\mathbf{1}}{\mathbf{0.3}}$, $w_2^C(\{c\}) = \left[q_2(\{c\}) \,.\, w_2^C(\{b, c\})\right]^{-1} = \mathbf{0.3}$.

Then, using the minimum operator[8] $\wedge$ of the Cautious conjunctive rule, we obtain:

- $w_{12}^C(\{b\}) = w_1^C(\{b\}) \wedge w_2^C(\{b\}) = 0.5 \wedge 1 = \mathbf{0.5}$
- $w_{12}^C(\{c\}) = w_1^C(\{c\}) \wedge w_2^C(\{c\}) = 1 \wedge 0.3 = \mathbf{0.3}$
- $w_{12}^C(\emptyset) = w_1^C(\emptyset) \wedge w_2^C(\emptyset) = 0.4 \wedge 1 = \mathbf{0.4}$

which means that $w_{12}^C(\{b, c\}) = \left[w_{12}^C(\{b\}).w_{12}^C(\{c\}).w_{12}^C(\emptyset)\right]^{-1} = \frac{1}{0.06}$. Then, let us compute the associated commonality function q_{12}: $q_{12}(\{b, c\}) = w_{12}^C(\{b, c\})^{-1} = \mathbf{0.06}$, $q_{12}(\{b\}) = \left[w_{12}^C(\{b\}).w_{12}^C(\{b, c\})\right]^{-1} = \mathbf{0.12}$, $q_{12}(\{c\}) = \left[w_{12}^C(\{c\}).w_{12}^C(\{b, c\})\right]^{-1} = \mathbf{0.2}$ and $q_{12}(\emptyset) = \left[w_{12}^C(\emptyset).w_{12}^C(\{b\}).w_{12}^C(\{c\}).w_{12}^C(\{b, c\})\right]^{-1} = \mathbf{1}$.

Finally, the associated mass function m_{12} is: $m_{12}(\{b, c\}) = q_{12}(\{b, c\}) = \mathbf{0.06}$, $m_{12}(\{b\}) = q_{12}(\{b\}) - m_{12}(\{b, c\}) = \mathbf{0.06}$, $m_{12}(\{c\}) = q_{12}(\{c\}) - m_{12}(\{b, c\}) = \mathbf{0.14}$ and $m_{12}(\emptyset) = q_{12}(\emptyset) - m_{12}(\{b\}) - m_{12}(\{c\}) - m_{12}(\{b, c\}) = \mathbf{0.74}$, where $m_{12}(x) = 0$ for all $x \notin 2^{\{b,c\}}$. Since these results are here the same as if we had employed the conjunctive fusion rule, it is easy to see that they are correct. Indeed, computing the conjunctive fusion of m_1 and m_2 directly in 2^Ω produces the same resulting mass function m_{12}.

2.4.3 Better understanding the conjunctive and disjunctive decompositions

A better understanding of the conjunctive or disjunctive decomposition of evidence can be exploited to propose e.g. new fusion rules using Definition 2.4.2.1 or new approximation methods based on $\text{supp}(w - 1)$ instead of $\text{supp}(m)$, where m is a mass function and w is the weight function associated with it. For this, it is interesting to study how each piece of evidence in the conjunctive or disjunctive decomposition impacts the mass function. Until now, the only exact way to do this computation in the general case was using the FMT, which is $O(N.2^N)$, and this provided no insight on the internal workings of these modifications. Here, given Theorem 2.3.5.1,

[8] In fact, as all couples of numbers involve 1 and another number less than 1, the conjunctive rule would have given the same result here.

we propose new formulas (Proposition 2.4.3.1) describing the propagation of these updates. We will focus on the conjunctive decomposition, but a similar method exists for its dual, namely the disjunctive decomposition. Once focal points have been computed for the original weight function w, the complexity of our method ranges from $O(1)$ to $O(N\ .\ |{}^\wedge\text{supp}(w-1)|)$ for each image of focal point modified in w, depending on its relation with respect to other focal points.

Proposition 2.4.3.1. Let w' be equal to w everywhere on $2^\Omega\backslash\{\Omega\}$, except for the image of some $x \in {}^\wedge\text{supp}(w-1)\backslash\{\Omega\}$. Also, let m' and q' be the functions corresponding to w' so that they satisfy Eq. 2.6 in place of respectively m and q. Then, we have:

$$\forall y \in 2^\Omega, \quad q'(y) = \begin{cases} q(y) & \text{if } x \supseteq y \\ \frac{w'(x)}{w(x)}.q(y) & \text{otherwise} \end{cases} \tag{2.16}$$

and

$$m'(y) = \begin{cases} 0 & \text{if } y \notin {}^\wedge S \\ \frac{w'(x)}{w(x)}.m(y) + \left[1 - \frac{w'(x)}{w(x)}\right].m_{\downarrow x}(y) & \text{if } x \supseteq y \\ \frac{w'(x)}{w(x)}.m(y) & \text{otherwise} \end{cases} \tag{2.17}$$

where ${}^\wedge S \supseteq \text{supp}(w-1) \cup \{\Omega\}$ and $m_{\downarrow x} : 2^x \to [0,1]$ is the mass function resulting from the projection of m onto $\downarrow x$, i.e. the Möbius transform of q in $(2^x, \supseteq)$.

Proof. See Appendix A.7. ■

Example 2.4.3.1. Let us take back Example 2.3.7.2, i.e. $\Omega = \{a,b,c\}$ and m is a mass function such that $m(\Omega) = 0.1$, $m(\{a,b\}) = 0.1$, $m(\{b,c\}) = 0.2$ and $m(\{a\}) = 0.6$. We have $\text{supp}(m) = \{\Omega, \{a,b\}, \{b,c\}, \{a\}\}$ and ${}^\wedge\text{supp}(m) = \text{supp}(m) \cup \{\{b\}, \emptyset\}$. We also have its commonality function q such that $q(\Omega) = 0.1$, $q(\{a,b\}) = 0.2$, $q(\{b,c\}) = 0.3$, $q(\{a\}) = 0.8$, $q(\{b\}) = 0.4$ and $q(\emptyset) = 1$. Finally, its conjunctive weight function w is defined by $w(\Omega) = 10$, $w(\{a,b\}) = 0.5$, $w(\{b,c\}) = \frac{1}{3}$, $w(\{a\}) = 0.25$, $w(\{b\}) = 1.5$ and $w(\emptyset) = 1.6$.

Now, suppose that we would like to see how changing $w(\{b\})$ from 1.5 to 1, i.e. removing $\{b\}$ from $\text{supp}(w-1)$, affects m. Then, Eq. 2.17 tells us that only $m(\{b\})$ and $m(\emptyset)$ are impacted by more than just a renormalization factor. Let us start by computing $m_{\downarrow\{b\}}$, where $\downarrow\{b\} = 2^{\{b\}}$: $m_{\downarrow\{b\}}(\{b\}) = q(\{b\}) = \mathbf{0.4}$ and $m_{\downarrow\{b\}}(\emptyset) = q(\emptyset) - m_{\downarrow\{b\}}(\{b\}) = \mathbf{0.6}$. Now, we can update m to provide m':

- $m'(\{b\}) = \frac{w'(\{b\})}{w(\{b\})}.m(\{b\}) + \left[1 - \frac{w'(\{b\})}{w(\{b\})}\right].m_{\downarrow\{b\}}(\{b\}) = 0 + \left[1 - \frac{1}{1.5}\right] * 0.4 = \mathbf{\frac{2}{15}}$
- $m'(\emptyset) = \frac{w'(\{b\})}{w(\{b\})}.m(\emptyset) + \left[1 - \frac{w'(\{b\})}{w(\{b\})}\right].m_{\downarrow\{b\}}(\emptyset) = 0 + \left[1 - \frac{1}{1.5}\right] * 0.6 = \mathbf{\frac{3}{15}}$

All other focal points p simply have their image through m' renormalized:

$$m'(p) = \frac{w'(\{b\})}{w(\{b\})}.m(p) = \frac{1}{1.5}.m(p)$$

which gives $m'(\Omega) = m'(\{a,b\}) = \mathbf{\frac{1}{15}}$, $m'(\{b,c\}) = \mathbf{\frac{2}{15}}$ and $m'(\{a\}) = \mathbf{\frac{6}{15}}$. Note that the sum of the images of m' is indeed equal to 1.

Example 2.4.3.2. Still using m, q and w from Example 2.3.7.2, suppose now that we would like to see how changing $w(\{a,b\})$ from 0.5 to 1, i.e. removing $\{a,b\}$ from $\text{supp}(w-1)$, affects m. Then, Eq. 2.17 tells us that only $m(\{a,b\})$, $m(\{a\})$, $m(\{b\})$ and $m(\emptyset)$ are impacted by more than just a renormalization factor. Let us start by computing $m_{\downarrow\{a,b\}}$, where $\downarrow\{a,b\} = 2^{\{a,b\}}$:

- $m_{\downarrow\{a,b\}}(\{a,b\}) = q(\{a,b\}) = \mathbf{0.2}$
- $m_{\downarrow\{a,b\}}(\{a\}) = q(\{a\}) - m_{\downarrow\{a,b\}}(\{a,b\}) = \mathbf{0.6}$

- $m_{\downarrow\{a,b\}}(\{b\}) = q(\{b\}) - m_{\downarrow\{a,b\}}(\{a,b\}) = \mathbf{0.2}$
- $m_{\downarrow\{a,b\}}(\emptyset) = q(\emptyset) - m_{\downarrow\{a,b\}}(\{a,b\}) - m_{\downarrow\{a,b\}}(\{a\}) - m_{\downarrow\{a,b\}}(\{b\}) = \mathbf{0}$

Now, we can update m to provide m':

- $m'(\{a,b\}) = \frac{w'(\{a,b\})}{w(\{a,b\})}.m(\{a,b\}) + \left[1 - \frac{w'(\{a,b\})}{w(\{a,b\})}\right].m_{\downarrow\{a,b\}}(\{a,b\}) = \frac{1}{0.5} * 0.1 + \left[1 - \frac{1}{0.5}\right] * 0.2 = \mathbf{0}$
- $m'(\{a\}) = \frac{w'(\{a,b\})}{w(\{a,b\})}.m(\{a\}) + \left[1 - \frac{w'(\{a,b\})}{w(\{a,b\})}\right].m_{\downarrow\{a,b\}}(\{a\}) = \frac{1}{0.5} * 0.6 + \left[1 - \frac{1}{0.5}\right] * 0.6 = \mathbf{0.6}$
- $m'(\{b\}) = \frac{w'(\{a,b\})}{w(\{a,b\})}.m(\{b\}) + \left[1 - \frac{w'(\{a,b\})}{w(\{a,b\})}\right].m_{\downarrow\{a,b\}}(\{b\}) = 0 + \left[1 - \frac{1}{0.5}\right] * 0.2 = \mathbf{-0.2}$
- $m'(\emptyset) = \frac{w'(\{a,b\})}{w(\{a,b\})}.m(\emptyset) + \left[1 - \frac{w'(\{a,b\})}{w(\{a,b\})}\right].m_{\downarrow\{a,b\}}(\emptyset) = 0 + \left[1 - \frac{1}{0.5}\right] * 0 = \mathbf{0}$

All other focal points p simply have their image through m' renormalized:

$$m'(p) = \frac{w'(\{b\})}{w(\{b\})}.m(p) = \frac{1}{0.5}.m(p)$$

which gives $m'(\Omega) = \mathbf{0.2}$ and $m'(\{b,c\}) = \mathbf{0.4}$. Notice that the sum of the images of m' is indeed equal to 1, but we have a negative mass $m'(\{b\})$. This is expected as it is known (see [41]) that the conjunctive decomposition is composed of what are called Simple Support Functions (SSF) (corresponding to elements of $\text{supp}(w-1)$ with an image in $(0,1)$) and Inverse Simple Support Functions (ISSF) (corresponding to elements of $\text{supp}(w-1)$ with an image in $(1,+\infty)$). Fusing all SSFs and ISSFs of one decomposition with the conjunctive rule gives the original mass function. SSFs correspond to mass functions, while ISSFs correspond to the decombination of mass functions, i.e. they are equivalent to mass functions with some negative value (if mass functions were allowed to have negative values). We had on $\{b\}$ a balance between positive mass values corresponding to $w(\{a,b\})$ and $w(\{b,c\})$ and a negative one corresponding to $w(\{b\})$, which we shifted towards the negatives by removing the positive mass corresponding to $w(\{a,b\})$. So, it just appears that changing SSFs from a decomposition that contains ISSFs is not always permitted in DST.

2.5 Conclusions and Perspectives

In this paper, we proposed an exact simplification of the zeta and Möbius transforms, for any function in any incidence algebra or on any partially ordered set. From this, we introduced the notion of *focal point* and discovered interesting properties when applied to the zeta and Möbius transforms. Then, we applied our theorems to DST in order to allow for both exactitude and computational efficiency in vast domain for most transformations between representations of belief and for their fusion. We also proposed a generalization of the conjunctive decomposition of evidence and provided formulas uncovering the influence of each decomposition weight on the corresponding mass function. These last two applications demonstrate the potential of our approach for the proof of new theoretical results and may themselves be exploited to propose new discounting methods and fusion operators in DST.

To go further in practice, we need to present algorithms, data structures and experimental setups comparing execution times and memory usage between the EMT, the FMT and naive approaches using focal points. Hence, we plan to issue a practical follow-up article in the near future, alongside a complete open-source implementation for DST.

In a more general way, our theoretical results can be both useful as a way to significantly reduce the complexity of any algorithm involving the zeta and Möbius transforms (e.g. with our EMT [53], which is defined in any distributive lattice) and as tools to better understand them.

Chapter 3

The Efficient Möbius Transformations

This chapter contains the adaptation and clarification of our third paper published at the international conference on *Scalable Uncertainty Management* (SUM) [53]. Our second paper, published at the french conference *Logique Floue et ses Applications* (LFA) [64], is not presented here because our third paper is a generalization of it.

Abstract

Dempster-Shafer Theory (DST) generalizes Bayesian probability theory, offering useful additional information, but suffers from a high computational burden. A lot of work has been done to reduce the complexity of computations used in information fusion with Dempster's rule. The main approaches exploit either the structure of Boolean lattices or the information contained in belief sources. Each has its merits depending on the situation. In this chapter, we propose sequences of graphs for the computation of the zeta and Möbius transformations that optimally exploit both the structure of distributive lattices and the information contained in belief sources. We call them the *Efficient Möbius Transformations* (EMT). We show that the complexity of the EMT is always inferior to the complexity of algorithms that consider the whole lattice, such as the *Fast Möbius Transform* (FMT) for all DST transformations. We then explain how to use them to fuse two belief sources. More generally, our EMTs apply to any function in any finite distributive lattice, focusing on a meet-closed or join-closed subset.

3.1 Introduction

Dempster-Shafer Theory (DST) [24] is an elegant formalism that generalizes Bayesian probability theory. It is more expressive by giving the possibility for a source to represent its belief in the state of a variable not only by assigning credit directly to a possible state (strong evidence) but also by assigning credit to any subset (weaker evidence) of the set Ω of all possible states. This assignment of credit is called a *mass function* and provides meta-information to quantify the level of uncertainty about one's believes considering the way one established them, which is critical for decision making.

Nevertheless, this information comes with a cost: considering $2^{|\Omega|}$ potential values instead of only $|\Omega|$ can lead to computationally and spatially expensive algorithms. They can become difficult to use for more than a dozen possible states (e.g. 20 states in Ω generate more than a million subsets), although we may need to consider large frames of discernment (e.g. for classification or identification tasks). Moreover, these algorithms not being tractable anymore beyond a few dozen states means their performances greatly degrade before that, which further limits their application to real-time applications. To tackle this issue, a lot of work has been done to reduce the complexity of transformations used to combine belief sources with Dempster's rule [30]. We distinguish between two approaches that we call *powerset-based* and *evidence-based*.

The *powerset-based* approach concerns all algorithms based on the structure of the powerset 2^{Ω} of the frame of discernment Ω. They have a complexity dependent on $|\Omega|$. Early works [54–57] proposed optimizations by restricting the structure of evidence to only singletons and their negation, which greatly restrains the expressiveness of DST. Later, a family of optimal algorithms working in the general case, i.e. the ones based on the *Fast Möbius Transform* (FMT) [43], was discovered. Their complexity is $O(|\Omega|.2^{|\Omega|})$ in time and $O(2^{|\Omega|})$ in space. It has become the de facto standard for the computation of every transformation in DST. Consequently, efforts were made to reduce the size of Ω to benefit from the optimal algorithms of the FMT. More specifically, [44] refers to the process of conditioning by the *combined core* (intersection of the unions of all *focal sets* of each belief source) and *lossless coarsening* (merging of elements of Ω which always appear together in focal sets). Also, Monte Carlo methods [44] have been proposed but depend on a number of trials that must be large and grows with $|\Omega|$, in addition to not being exact.

The *evidence-based* approach concerns all algorithms that aim to reduce the computations to the only subsets that contain information (*evidence*), called *focal sets*, which are usually far less numerous than $2^{|\Omega|}$. This approach, also referred to as the *obvious* one, implicitly originates from the seminal work of Shafer [24] and is often more efficient than the powerset-based one since it only depends on information contained in sources in a quadratic way. Doing so, it allows for the exploitation of the full potential of DST by enabling us to choose any frame of discernment, without concern about its size. Moreover, the evidence-based approach benefits directly from the use of approximation methods, some of which are very efficient [45]. Therefore, this approach seems superior to the FMT in most use cases, above all when $|\Omega|$ is large, where an algorithm with exponential complexity is just intractable.

It is also possible to easily find evidence-based algorithms computing all DST transformation, except for the conjunctive and disjunctive decompositions for which we recently proposed a method [42].

However, since these algorithms rely only on the information contained in sources, they do not exploit the structure of the powerset to reduce the complexity, leading to situations in which the FMT can be more efficient if almost every subset contains information, i.e. if the number of focal sets tends towards $2^{|\Omega|}$ [44], all the most when no approximation method is employed.

In this paper, we fuse these two approaches into one, proposing new sequences of graphs, in the same fashion as the FMT, that are always more efficient than the FMT and can in addition benefit from evidence-based optimizations. We call them the *Efficient Möbius Transformations* (EMT). More generally, our approach applies to any function defined on a finite distributive lattice.

Outside the scope of DST, [61] is related to our approach in the sense that we both try to remove redundancy in the computation of the zeta and Möbius transforms on the subset lattice 2^Ω. However, they only consider the redundancy of computing the image of a subset that is known to be null beforehand. To do so, they only visit sets that are accessible from the focal sets of lowest rank by successive unions with each element of Ω. Here, we demonstrate that it is possible to avoid far more computations by reducing them to specific sets so that each image is only computed once. These sets are the focal points described in [42]. The study of their properties will be carried out in depth in an upcoming article [52]. Besides, our method is more general since it applies to any finite distributive lattice.

Furthermore, an important result of our work resides in the optimal computation of the zeta and Möbius transforms in any intersection-closed family F of sets from 2^Ω, i.e. with a complexity $O(|\Omega|.|F|)$. Indeed, in the work of [62] on the optimal computation of these transforms in any finite lattice L, they embedded L into the Boolean lattice 2^Ω, obtaining an intersection-closed family F as its equivalent, and found a meta-procedure building a circuit of size $O(|\Omega|.|F|)$ computing the zeta and Möbius transforms. However, they did not managed to build this circuit in less than $O(|\Omega|.2^{|\Omega|})$. Given F, our Theorem 3.3.2.2 in this paper directly computes this circuit with a complexity that can be as low as $O(|\Omega|.|F|)$ in some instances, while being much simpler.

This paper is organized as follows: Section 3.2 will present the elements on which our method is built. Section 3.3 will present our EMT. Section 3.4 will discuss their complexity and their usage both in general and in the case of DST. Finally, we will conclude this article with section 3.5.

3.2 Background of our method

Let $(P, \leq)$ be a finite[1] set partially ordered by $\leq$.

3.2.1 Zeta transform

The zeta transform $g : P \to \mathbb{R}$ of a function $f : P \to \mathbb{R}$ is defined as follows:

$$\forall y \in P, \quad g(y) = \sum_{x \leq y} f(x) \tag{3.1}$$

This can be extended to the multiplication as the *multiplicative zeta transform*:

$$\forall y \in P, \quad g(y) = \prod_{x \leq y} f(x)$$

[1]The following definitions hold for lower semifinite partially ordered sets as well, i.e. partially ordered sets such that the number of elements of P lower in the sense of $\leq$ than another element of P is finite. But for the sake of simplicity, we will only talk of finite partially ordered sets.

Example 3.2.1.1. In DST, the implicability function b is defined as the zeta transform of the mass function m in $(2^\Omega, \subseteq)$, i.e.:

$$\forall B \in 2^\Omega, \quad b(B) = \sum_{A \subseteq B} m(A)$$

Example 3.2.1.2. In DST, the implicability function b is also the inverse of the multiplicative zeta transform of the disjunctive weight function v in $(2^\Omega, \subseteq)$, i.e.:

$$\forall B \in 2^\Omega, \quad b(B) = \prod_{A \subseteq B} v(A)^{-1}$$

Example 3.2.1.3. In DST, the commonality function q is defined as the zeta transform of the mass function m in $(2^\Omega, \supseteq)$, i.e.:

$$\forall B \in 2^\Omega, \quad q(B) = \sum_{A \supseteq B} m(A)$$

Example 3.2.1.4. In DST, the commonality function q is also the inverse of the multiplicative zeta transform of the conjunctive weight function w in $(2^\Omega, \supseteq)$, i.e.:

$$\forall B \in 2^\Omega, \quad q(B) = \prod_{A \supseteq B} w(A)^{-1}$$

3.2.2 Möbius transform

The Möbius transform of g is f. It is defined as follows:

$$\forall y \in P, \quad f(y) = \sum_{x \leq y} g(x).\mu_{P,\leq}(x, y) \tag{3.2}$$

where $\mu_{P,\leq}$ is the Möbius function of $(P, \leq)$ (See [40]). There is also a multiplicative version with the same properties that can be seen as the exponential of the Möbius transform of $\log \circ g$:

$$\forall y \in P, \quad f(y) = \prod_{x \leq y} g(x)^{\mu_{P,\leq}(x,y)}$$

Example 3.2.2.1. In DST, the mass function m is the Möbius transform of the implicability function b in $(2^\Omega, \subseteq)$, i.e.:

$$\forall B \in 2^\Omega, \quad m(B) = \sum_{A \subseteq B} b(A).\mu_{2^\Omega,\subseteq}(A, B)$$

where for any $A, B \in 2^\Omega$, the Möbius function evaluates to $\mu_{2^\Omega,\subseteq}(A, B) = (-1)^{|B|-|A|}$, as recalled in [43].

Example 3.2.2.2. In DST, the disjunctive weight function v is the inverse of the multiplicative Möbius transform of the implicability function b in $(2^\Omega, \subseteq)$, i.e.:

$$\forall B \in 2^\Omega, \quad v(B) = \prod_{A \subseteq B} b(A)^{-\mu_{2^\Omega,\subseteq}(A,B)}$$

Example 3.2.2.3. In DST, the mass function m is the Möbius transform of the commonality function q in $(2^\Omega, \supseteq)$, i.e.:

$$\forall B \in 2^\Omega, \quad m(B) = \sum_{A \supseteq B} q(A).\mu_{2^\Omega,\supseteq}(A, B)$$

where for any $A, B \in 2^\Omega$, the Möbius function also evaluates to $\mu_{2^\Omega,\supseteq}(A, B) = (-1)^{|B|-|A|}$.

Example 3.2.2.4. In DST, the conjunctive weight function w is the inverse of the multiplicative Möbius transform of the commonality function q in $(2^\Omega, \subseteq)$, i.e.:

$$\forall B \in 2^\Omega, \quad w(B) = \prod_{A \supseteq B} q(A)^{-\mu_{2^\Omega,\supseteq}(A,B)}$$

3.2.3 Sequence of graphs and computation of the zeta transform

To yield $g(y)$ for some $y \in P$, we must sum all values $f(x)$ such that $x \leq y$. Our objective is to do it in the minimum number of operations, i.e. to minimize the number of terms to sum. If we only compute $g(y)$ alone, we have to pick every element in $\{x \in P \ / \ x \leq y\}$ and sum their associated values through f. However, if we compute $g(y)$ for all elements $y \in P$ at once, we can organize and mix these computations so that partial sums are reused for more than one value through g. Indeed, for any element $y \in P$, if there is an element $z \in P$ such that $y \leq z$, we have $g(z) = g(y) + \sum_{\substack{x \leq z \\ x \not\leq y}} f(x)$. So, we want to recursively build partial sums so that we can get the full sum on each $g(y)$ by only summing the values on some elements from $\{x \in P \ / \ x \leq y\}$. In other words, we would like to define an ordered sequence of transformations computing g from f.

Let us adopt the formalism of graph theory. Let $G_\leq$ be a directed acyclic graph in which the set of its nodes matches P and each arrow is directed by $\leq$. Let $E_\leq$ be the set of its arrows. We have $E_\leq = \{(x, y) \in P^2 \ / \ x \leq y\}$ and $G_\leq = (P, E_\leq)$. Thus, computing $g(y)$ alone is equivalent to visiting the node y from all nodes x of $G_\leq$ such that there is an arrow $(x, y) \in E_\leq$. Each "visit" to node y from a node x corresponds to the computation of the operation $f(x)+\cdot$, where $\cdot$ represents the current state of the sum associated with y. Thus, $G_\leq$, combined with the binary operator $f(\cdot) + \cdot$, describes the transformation of 0 into g. More concisely, we will equivalently initialize our algorithm with values through f instead of 0 and exploit the combination of $G_<$ and $+$. We will say that the transformation $(G_<, f, +)$ *computes the zeta transform of f in* $(P, \leq)$. In the end, we want to minimize the number of "visits" to be made to all y, i.e. we want to minimize the total number of arrows to follow to compute every $g(y)$. Therefore, the question is: Is there an ordered sequence of graphs that can compute g from f with less arrows in total than $(G_<, f, +)$?

Let I_P be the set containing all identity arrows of $G_\leq$, i.e. $I_P = \{ \ (x, y) \in P^2 \ / \ x = y\}$. Consider that all elements $y \in P$ are initialized with $f(y)$. We are interested in finding a sequence of graphs that is equivalent to the arrows of $G_<$. Let $(H_i)_{i \in [\![1,n]\!]}$ be a sequence of n directed acyclic graphs $H_i = (P, E_i)$. We will note $((H_i)_{i \in [\![1,n]\!]}, f, +)$ the computation that transforms f into h_1 through the arrows of E_1, then transforms h_1 into h_2 through the arrows of E_2, and so on until the transformation of h_{n-1} into h_n through the arrows of E_n. We ignore all identity arrows in these computations. So, this sequence of graphs requires us to consider $|E_1| + |E_2| + \cdots + |E_n|$ arrows, but transforms f into h_n in $|E_1 \backslash I_P| + |E_2 \backslash I_P| + \cdots + |E_n \backslash I_P|$ operations.

Proposition 3.2.3.1. Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$. One particular sequence of interest is $(H_i)_{i \in [\![1,n]\!]}$, where $H_i = (2^\Omega, E_i)$ and:

$$E_i = \{(X, Y) \in 2^\Omega \times 2^\Omega / \ Y = X \cup \{\omega_i\}\}.$$

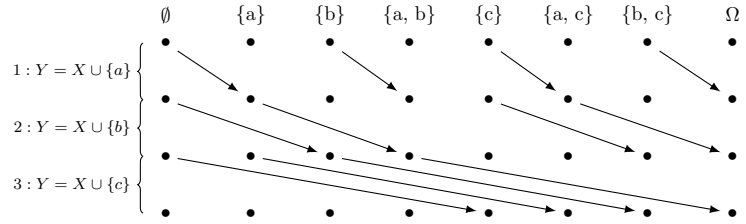


Figure 3.1: Illustration representing the paths generated by the arrows contained in the sequence $(H_i)_{i\in[\![1,3]\!]}$, where $H_i = (2^\Omega, E_i)$ and $E_i = \{(X,Y) \in 2^\Omega \times 2^\Omega / \ Y = X \cup \{\omega_i\}\}$ and $\Omega = \{a,b,c\}$. This sequence computes the same zeta transformations as $G_\subset = (2^\Omega, E_\subset)$, where $E_\subset = \{(X,Y) \in 2^\Omega \times 2^\Omega \ / \ X \subset Y\}$. A dot represents the node of its column. Its row i corresponds to both the tail of a potential arrow in H_i and the head of a potential arrow in H_{i-1}. The last row corresponds to the heads of all potential arrows that could be in H_3. The arrows represents the actual arrows in each graph H_i. Identity arrows are ignored in computations and not displayed here for the sake of clarity. If they were, there would be vertical arrows in every column, linking each dot in row i to the next dot of same node in row $i+1$. This representation is derived from the one used in [43].

This sequence computes the same zeta transformations as $G_\subset = (2^\Omega, E_\subset)$, where $E_\subset = \{(X,Y) \in 2^\Omega \times 2^\Omega \ / \ X \subset Y\}$.

Example 3.2.3.1. Let us say that $\Omega = \{a,b,c\}$. Crossing ignored arrows (i.e. identity arrows), we have:

- $E_1 = \{(X,Y) \in 2^\Omega \times 2^\Omega / \ Y = X \cup \{a\}\} = \{$

  $(\emptyset, \{a\}),\ \cancel{(\{a\},\{a\})},\ (\{b\},\{a,b\}),\ \cancel{(\{a,b\},\{a,b\})},$
  $(\{c\},\{a,c\}),\ \cancel{(\{a,c\},\{a,c\})},\ (\{b,c\},\Omega),\ \cancel{(\Omega,\Omega)}$
  $\}$

- $E_2 = \{(X,Y) \in 2^\Omega \times 2^\Omega / \ Y = X \cup \{b\}\} = \{$

  $(\emptyset, \{b\}),\ (\{a\},\{a,b\}),\ \cancel{(\{b\},\{b\})},\ \cancel{(\{a,b\},\{a,b\})},$
  $(\{c\},\{b,c\}),\ (\{a,c\}),\Omega),\ \cancel{(\{b,c\},\{b,c\})},\ \cancel{(\Omega,\Omega)}$
  $\}$

- $E_3 = \{(X,Y) \in 2^\Omega \times 2^\Omega / \ Y = X \cup \{c\}\} = \{$

  $(\emptyset, \{c\}),\ (\{a\},\{a,c\}),\ (\{b\},\{b,c\}),\ (\{a,b\},\Omega),$
  $\cancel{(\{c\},\{c\})},\ \cancel{(\{a,c\},\{a,c\})},\ \cancel{(\{b,c\},\{b,c\})},\ \cancel{(\Omega,\Omega)}$
  $\}$

Fig. 3.1 illustrates this sequence. Check that, after execution of $((H_i)_{i\in[\![1,n]\!]}, f, +)$, each element y of 2^Ω is associated with the sum $\sum_{x\subseteq y} f(x)$. For instance, let us take a look at Ω. Initially, each element of 2^Ω is associated with its value through f. Then, at step 1, we can see that the value on Ω is summed with $f(\{b,c\})$. At step 2, it is summed with $h_1(\{a,c\})$, which is equal to $f(\{c\})+f(\{a,c\})$, following step 1. Finally, at step 3, it is summed with $h_2(\{a,b\})$, which is equal to $h_1(\{a\})+h_1(\{a,b\})$ following step 2, which is itself equal to $f(\emptyset)+f(\{a\})+f(\{b\})+f(\{a,b\})$, following step 1. Gathering all these terms, we get that $h_3(\Omega) = f(\Omega) + f(\{b,c\}) + f(\{c\}) + f(\{a,c\}) + f(\emptyset) + f(\{a\}) + f(\{b\}) + f(\{a,b\})$.

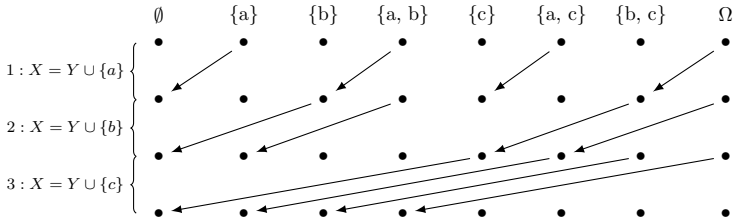


Figure 3.2: Illustration representing the paths generated by the arrows contained in the sequence $(H_i)_{i\in[\![1,3]\!]}$, where $H_i = (2^\Omega, E_i)$ and $E_i = \{(X,Y) \in 2^\Omega \times 2^\Omega / \ X = Y \cup \{\omega_i\}\}$ and $\Omega = \{a,b,c\}$. This sequence computes the same zeta transformations as $G_\supset = (2^\Omega, E_\supset)$, where $E_\supset = \{(X,Y) \in 2^\Omega \times 2^\Omega \ / \ X \supset Y\}$.

Proposition 3.2.3.2. The dual of this particular sequence in $(2^\Omega, \supseteq)$ is $(H_i)_{i\in[\![1,n]\!]}$, where $H_i = (2^\Omega, E_i)$ and:

$$E_i = \{(X,Y) \in 2^\Omega \times 2^\Omega / \ X = Y \cup \{\omega_i\}\}.$$

This sequence computes the same zeta transformations as $G_\supset = (2^\Omega, E_\supset)$, where $E_\supset = \{(X,Y) \in 2^\Omega \times 2^\Omega \ / \ X \supset Y\}$.

Example 3.2.3.2. Fig. 3.2 illustrates the dual sequence for zeta transforms in $(2^\Omega, \supseteq)$ and $\Omega = \{a,b,c\}$.

Remark. These two sequences of graphs are the foundation of the *Fast Möbius Transform* (FMT) algorithms. Their execution is $O(n.2^n)$ in time and $O(2^n)$ in space. As we can see, the FMT presented here proposes two transformations $((H_i)_{i\in[\![1,n]\!]}, f, +)$ that computes the same transformation as respectively $(G_\subset, f, +)$ and $(G_\supset, f, +)$ for any function $f : 2^\Omega \to \mathbb{R}$. The authors proved them to be the optimal transformations for any set Ω, i.e. the one that uses the fewest arrows, independently of the function f to be considered. This means that they do not take into account the neutral values of f for the operator $+$, i.e. where f evaluates to 0, contrary to our approach. This is why our method is able to feature a lower complexity than this *optimal* FMT.

More generally, Theorem 3 of [43] defines a necessary and sufficient condition to verify that a transformation $((H_i)_{i\in[\![1,n]\!]}, f, +)$ computes (ignoring identity arrows) the same transformation as $(G_<, f, +)$. It is stated in our terms as follows:

Theorem 3.2.3.1. *Let $(H_i)_{i\in[\![1,n]\!]}$ be a sequence of directed acyclic graphs $H_i = (P, E_i)$. Let us pose $A_i = E_i \cup I_P$. The transformation $((H_i)_{i\in[\![1,n]\!]}, f, +)$ computes (ignoring identity arrows) the same transformation as $(G_<, f, +)$ if and only if each set of arrows satisfies $E_i \subseteq E_\leq$ and every arrow $e \in E_\leq$ can be decomposed as a unique path $(e_1, e_2, \ldots, e_n) \in A_1 \times A_2 \times \cdots \times A_n$, where the tail of e_1 is the tail of e and the head of e_n is the head of e. Recall that a path is a sequence of arrows in which $\forall i \in [\![1, n-1]\!]$, the head of e_i is the tail of e_{i+1}.*

Example 3.2.3.3. Let us prove Proposition 3.2.3.1. We had $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$. The sequence of graphs $(H_i)_{i\in[\![1,n]\!]}$ computes the same zeta transformations as $G_\subset = (2^\Omega, E_\subset)$, where $E_\subset = \{(X,Y) \in 2^\Omega \times 2^\Omega \ / \ X \subset Y\}$ if:

$$E_i = \{(X,Y) \in 2^\Omega \times 2^\Omega / \ Y = X \cup \{\omega_i\}\},$$

where $i \in \{1, \ldots, n\}$ and $H_i = (2^\Omega, E_i)$.

Proof. Obviously, for any $i \in [\![1,n]\!]$, we have $E_i \subseteq E_\subseteq$. In addition, each set $X \subseteq \Omega$ is composed by definition of at most n elements from Ω. Thus, it is possible to reach in at most n steps from X any set $Y \subseteq \Omega$ such that $X \subseteq Y$ with identity arrows and arrows where the

head is just the tail with exactly one other element from Ω, i.e. with identity arrows or arrows of $(H_i)_{i\in[\![1,n]\!]}$. Moreover, since each step corresponds to a distinct element from Ω, there is exactly one path from X to Y: at each step corresponding to the elements in $Y\backslash X$, follow the arrow that adds an element to the set reached in the previous step. Only identity arrows can be followed in the steps corresponding to the elements in $X \cap Y$. Otherwise, there would be an element missing or in excess relatively to Y at the end of step n, which means that Y could not be reached. Therefore, according to Theorem 3.2.3.1, the transformation $((H_i)_{i\in[\![1,n]\!]}, f, +)$ computes the same transformation as $(G_\subset, f, +)$ for any function $f : 2^\Omega \to \mathbb{R}$. ■

In addition, for a sequence of graphs computing zeta transforms in $(P, \leq)$, reversing its paths yields a sequence of graphs computing zeta transforms in $(P, \geq)$. Hence Proposition 3.2.3.2.

3.2.4 Sequence of graphs and computation of the Möbius transform

Now, consider that we want to find a sequence of graphs that undoes the previous computation. We want to transform g into f, i.e. the Möbius transform of g in $(P, \leq)$. For this, notice that for any step i in the transformation $((H_i)_{i\in[\![1,n]\!]}, f, +)$, we have for each node $y \in P$,

$$h_i(y) = h_{i-1}(y) + \sum_{(x,y)\in E_i\backslash I_P} h_{i-1}(x) \quad \Leftrightarrow \quad h_{i-1}(y) = h_i(y) - \sum_{(x,y)\in E_i\backslash I_P} h_{i-1}(x).$$

So, as long as we know all $h_{i-1}(x)$ for all arrows $(x, y) \in E_i\backslash I_P$ at each step i and for all $y \in P$, we can simply reverse the order of the sequence $(H_i)_{i\in[\![1,n]\!]}$ and use the operator - instead of +. If this is verified, then $((H_{n-i+1})_{i\in[\![1,n]\!]}, g, -)$ computes the Möbius transform of g in $(P, \leq)$. This condition can be translated as follows: for every arrow $(x, y) \in E_i\backslash I_P$, we have $h_i(x) = h_{i-1}(x)$. This condition is equivalent to stating that for every arrow $(x, y) \in E_i\backslash I_P$, there is no arrow (w, x) in $E_i\backslash I_P$.

Theorem 3.2.4.1. *Let $(H_i)_{i\in[\![1,n]\!]}$ be a sequence of directed acyclic graphs $H_i = (P, E_i)$. Let h_n be the function resulting from the transformation $((H_i)_{i\in[\![1,n]\!]}, f, +)$, ignoring identity arrows. If for every arrow $(x, y) \in E_i\backslash I_P$, there is no arrow (w, x) in $E_i\backslash I_P$, then $((H_{n-i+1})_{i\in[\![1,n]\!]}, h_n, -)$ yields the initial function f.*

Thus, if $((H_i)_{i\in[\![1,n]\!]}, f, +)$ computes the zeta transform g of f in $(P, \leq)$ and Theorem 3.2.4.1 is satisfied, then $((H_{n-i+1})_{i\in[\![1,n]\!]}, g, -)$ *computes the Möbius transform f of g in $(P, \leq)$.*

Application to the powerset lattice 2^Ω (FMT) Consider again the sequence $(H_i)_{i\in[\![1,n]\!]}$, from the application of section 3.2.3.1, that computes the zeta transform of f in $(2^\Omega, \subseteq)$. If there is an arrow $(X, Y) \in E_i\backslash I_{2^\Omega}$, then $\omega_i \notin X$. This means that there is no set W in 2^Ω such that $W \cup \{\omega_i\} = X$, and so no arrow (W, X) in $E_i\backslash I_{2^\Omega}$. Thus, according to Theorem 3.2.4.1, $((H_{n-i+1})_{i\in[\![1,n]\!]}, h_n, -)$ computes the Möbius transformation of $((H_i)_{i\in[\![1,n]\!]}, f, +)$, i.e. the function f. Furthermore, given that Ω is a set and that each graph H_i concerns an element ω_i, independently from the others, any indexing (order) in the sequence $(H_i)_{i\in[\![1,n]\!]}$ computes the zeta transformation. This implies that any order in $((H_{n-i+1})_{i\in[\![1,n]\!]}, h_n, -)$ computes the Möbius transformation of $((H_i)_{i\in[\![1,n]\!]}, f, +)$. In particular, $((H_i)_{i\in[\![1,n]\!]}, h_n, -)$ computes the same transformation as $((H_{n-i+1})_{i\in[\![1,n]\!]}, h_n, -)$.

Example 3.2.4.1. Let us say that $\Omega = \{a, b, c\}$. We want to compute the Möbius transform f of g in $(2^\Omega, \subseteq)$. Each arrow set E_i has already been computed in Example 3.2.3.1. Fig. 3.1 illustrates any transformation based on $(H_i)_{i\in[\![1,n]\!]}$, including $((H_i)_{i\in[\![1,n]\!]}, h_n, -)$. Check that, after execution of $((H_i)_{i\in[\![1,n]\!]}, h_n, -)$, each element y of 2^Ω is associated with $f(y)$. For instance, let us take a look again at Ω. Initially, each element of 2^Ω is associated with its value through g. Then, at step 1, we can see that to the value on Ω is subtracted $g(\{b, c\})$. At step 2, to $h_1(\Omega)$

is subtracted $h_1(\{a,c\})$, which is equal to $g(\{a,c\}) - g(\{c\})$, following step 1. Finally, at step 3, to $h_2(\Omega)$ is subtracted $h_2(\{a,b\})$, which is equal to $h_1(\{a,b\}) - h_1(\{a\})$ following step 2, which is itself equal to $g(\{a,b\}) - g(\{b\}) - (g(\{a\}) - g(\emptyset))$, following step 1. Gathering all these terms, we get that

$$\begin{aligned} h_3(\Omega) &= g(\Omega) - g(\{b,c\}) - [\ g(\{a,c\}) - g(\{c\})\ ] - [\ g(\{a,b\}) - g(\{b\}) - [\ g(\{a\}) - g(\emptyset)\ ]\ ] \\ &= g(\Omega) - g(\{b,c\}) - g(\{a,c\}) + g(\{c\}) - g(\{a,b\}) + g(\{b\}) + g(\{a\}) - g(\emptyset) \\ &= \sum_{X \subseteq \Omega} g(x).(-1)^{|\Omega| - |X|}. \end{aligned}$$

As recalled in [43], the function that associates to each couple $(X,Y) \in 2^\Omega \times 2^\Omega$ the value $(-1)^{|Y|-|X|}$ is the Möbius function μ in $(2^\Omega, \subseteq)$. So, according to Eq. 3.2, we have $h_3(\Omega) = f(\Omega)$.

3.2.5 Order theory

Let $(P, \leq)$ be a set partially ordered by the relation $\leq$.

Meet / join Let S be a subset of P. If it is unique, the greatest element of P that is less than all the elements of S is called the meet (or infimum) of S. It is noted $\bigwedge S$. If $S = \{x, y\}$, we may also note it with the binary operator $\wedge$ such that $x \wedge y$. If it is unique, the least element of P that is greater than all the elements of S is called the join (or supremum) of S. It is noted $\bigvee S$. If $S = \{x, y\}$, we may also note it with the binary operator $\vee$ such that $x \vee y$.

Example 3.2.5.1. In $(2^\Omega, \subseteq)$, the meet operator $\wedge$ is the intersection operator $\cap$, while the join operator $\vee$ is the union operator $\cup$.

Lattice / semi-lattice If any non-empty subset of P has a join, we say that P is an upper semilattice. If any non-empty subset of P has a meet, we say that P is a lower semilattice. When P is both, we say that P is a lattice.

Example 3.2.5.2. In $(2^\Omega, \subseteq)$, any non-empty subset S has an intersection and a union in 2^Ω. They respectively represent the common elements of the sets in S and the cumulative elements of all the sets in S. Thus, 2^Ω is a lattice.

Irreducible elements In any partially ordered set, there are bottom elements such that they cannot be described as the join of two lesser elements. Such irreducible elements are called the *join-irreducible elements of* P if they are not equal to the global minimum of P. We will note ${}^\vee\mathcal{I}(P)$ the set of all join-irreducible elements of P. Since none of them is $\bigwedge P$, this means that the join of any two join-irreducible elements yields a non-join-irreducible element of P. In fact, if P is an upper semilattice (or a lattice), it is known that the join of all possible non-empty subset of ${}^\vee\mathcal{I}(P)$ yields all elements of P, except $\bigwedge P$. Formally, we write that all join-irreducible element i verifies $i \neq \bigwedge P$ and for all elements $x, y \in P$, if $x < i$ and $y < i$, then $x \vee y < i$.

Dually, in any partially ordered set, there are top elements such that they cannot be described as the meet of two greater elements. Such irreducible elements are called the *meet-irreducible elements of* P if they are not equal to the global maximum of P. We will note ${}^\wedge\mathcal{I}(P)$ the set of all meet-irreducible elements of P. Since none of them is $\bigvee P$, this means that the meet of any two meet-irreducible elements yields a non-meet-irreducible element of P. In fact, if P is a lower semilattice (or a lattice), it is known that the meet of all possible non-empty subset of ${}^\wedge\mathcal{I}(P)$ yields all elements of P, except $\bigvee P$. Formally, we write that all meet-irreducible element i verifies $i \neq \bigvee P$ and for all elements $x, y \in P$, if $x > i$ and $y > i$, then $x \wedge y > i$.

Example 3.2.5.3. In $(2^\Omega, \subseteq)$, the join-irreducible elements are the singletons $\{\omega\}$, where $\omega \in \Omega$. The meet-irreducible elements are their complement $\overline{\{\omega\}} = \Omega \backslash \{\omega\}$, where $\omega \in \Omega$.

Distributive lattice A distributive lattice L is a lattice that satisfies the distributive law:

$$\forall x, y, z \in L, \quad (x \wedge y) \vee (x \wedge z) = x \wedge (y \vee z) \tag{3.3}$$

Since $(x \wedge z) \vee z = z$, this condition is equivalent to:

$$\forall x, y, z \in L, \quad (z \vee y) \wedge (z \vee x) = z \vee (y \wedge x) \tag{3.4}$$

Example 3.2.5.4. In the powerset lattice 2^Ω, it holds for any sets $A, B, C \in 2^\Omega$ that $(A \cap B) \cup (A \cap C) = A \cap (B \cup C)$. Thus, the lattice 2^Ω is a distributive lattice.

Sublattice A sublattice S is simply a subset of a lattice L that is itself a lattice, with the same meet and join operations as L. This means that for any two elements $x, y \in S$, we have both $x \vee y \in S$ and $x \wedge y \in S$, where $\vee$ and $\wedge$ are the join and meet operators of L.

Upset / down set An upset (or upward closed set) S is a subset of P such that all elements in P greater than at least one element of S is in S. The upper closure of an element $x \in P$ is noted $\uparrow x$ or $x^{\uparrow P}$ (when the encompassing set has to be specified). It is equal to $\{y \in P \ / \ x \leq y \ \}$.

Dually, a down set (or downward closed set) S is a subset of P such that all elements in P less than at least one element of S is in S. The lower closure of an element $x \in P$ is noted $\downarrow x$ or $x^{\downarrow P}$ (when the encompassing set has to be specified). It is equal to $\{y \in P \ / \ x \geq y \ \}$.

3.2.6 Support elements and focal points

Let $f : P \to \mathbb{R}$.

Support of a function in P

The support $\text{supp}(f)$ of a function $f : P \to \mathbb{R}$ is defined as $\text{supp}(f) = \{x \in P \ / \ f(x) \neq 0\}$.

Example 3.2.6.1. In DST, the set containing the focal sets of a mass function m is $\text{supp}(m)$.

Focal points

We note ${}^\vee\text{supp}(f)$ the smallest join-closed subset of P containing $\text{supp}(f)$, i.e.:

$${}^\vee\text{supp}(f) = \left\{ \bigvee S \ / \ S \subseteq \text{supp}(f), \ S \neq \emptyset \right\}$$

We note ${}^\wedge\text{supp}(f)$ the smallest meet-closed subset of P containing $\text{supp}(f)$, i.e.:

$${}^\wedge\text{supp}(f) = \left\{ \bigwedge S \ / \ S \subseteq \text{supp}(f), \ S \neq \emptyset \right\}$$

The set containing the *focal points* $\mathring{\mathcal{F}}$ of a mass function m from [42] for the conjunctive weight function is ${}^\wedge\text{supp}(m)$. For the disjunctive weight function, we use their dual focal points, defined by ${}^\vee\text{supp}(m)$.

It has been proven in [42] that the image of 2^Ω through the conjunctive weight function can be computed without redundancies by only considering the focal points ${}^\wedge\text{supp}(m)$ in the definition of the multiplicative Möbius transform of the commonality function. The image of all set in

$2^\Omega \backslash {}^\wedge\text{supp}(m)$ through the conjunctive weight function is 1. In the same way, the image of any set in $2^\Omega \backslash {}^\wedge\text{supp}(m)$ through the commonality function is only a duplicate of the image of a focal point in ${}^\wedge\text{supp}(m)$. Its image can be recovered by searching for its smallest focal point superset in ${}^\wedge\text{supp}(m)$. The same can be stated for the disjunctive weight function regarding the implicability function and ${}^\vee\text{supp}(m)$.

In fact, as generalized in [52], for any function $f : P \to \mathbb{R}$, its focal points ${}^\wedge\text{supp}(f)$ are sufficient to define its zeta and Möbius transforms in $(P, \geq)$, and ${}^\vee\text{supp}(f)$ is sufficient to define its zeta and Möbius transforms in $(P, \leq)$.

However, considering the case where P is a finite lattice, naive algorithms that only consider ${}^o\text{supp}(f)$, where $o \in \{\vee, \wedge\}$ have upper bound complexities in $O(|{}^o\text{supp}(f)|^2)$, which may be worse than the optimal complexity $O(|{}^\vee\mathcal{I}(P)|.|P|)$ of a procedure that considers the whole lattice P. In this paper, we propose computing schemes for $g(S)$, where ${}^o\text{supp}(f) \subseteq S \subseteq P$ and g is the zeta transform of f in $(P, \leq)$, with complexities always less than $O(|{}^\vee\mathcal{I}(P)|.|P|)$, provided that P is a finite distributive lattice. We also provide schemes for computing the Möbius transform $f(S)$ of g in $(P, \leq)$.

3.3 Our Efficient Möbius Transformations

In this section, we consider a function $f : P \to \mathbb{R}$ where P is a finite distributive lattice (e.g. the powerset lattice 2^Ω). We present here the sequences of graphs that can be exploited to compute our so-called *Efficient Möbius Transformations*. Theorem 3.3.2.1 describes a way of computing the zeta and Möbius transforms of a function based on the smallest sublattice ${}^\mathcal{L}\text{supp}(f)$ of P containing both ${}^\wedge\text{supp}(f)$ and ${}^\vee\text{supp}(f)$, which is defined in Proposition 3.3.1.3. Theorem 3.3.2.2 goes beyond this optimization by computing these transforms based only on ${}^o\text{supp}(f)$, where $o \in \{\vee, \wedge\}$. Nevertheless, this second approach requires the direct computation of ${}^o\text{supp}(f)$, which has an upper bound complexity of $O(|\text{supp}(f)|.|{}^o\text{supp}(f)|)$, which may be more than $O(|{}^\vee\mathcal{I}(P)|.|P|)$ if $|\text{supp}(f)| \gg |{}^\vee\mathcal{I}(P)|$.

3.3.1 Preliminary results

In this subsection, we provide some propositions that are useful for proving our main results (presented in the next subsection).

Lemma 3.3.1.1 (*Safe join*)**.** *Let us consider a finite distributive lattice L. For all join-irreducible element $i \in {}^\vee\mathcal{I}(L)$ and for all elements $x, y \in L$ that are not greater than or equal to i, i.e. $i \not\leq x$ and $i \not\leq y$, we have that their join is not greater or equal to i either, i.e. $i \not\leq x \vee y$.*

Proof. By definition of a join-irreducible element, we know that $\forall i \in {}^\vee\mathcal{I}(L)$ and for all $a, b \in L$, if $a < i$ and $b < i$, then $a \vee b < i$. Moreover, for all $x, y \in L$ such that $i \not\leq x$ and $i \not\leq y$, we have equivalently $i \wedge x < i$ and $i \wedge y < i$. Thus, we get that $(i \wedge x) \vee (i \wedge y) < i$. Since L satisfies the distributive law Eq. (3.3), this implies that $(i \wedge x) \vee (i \wedge y) = i \wedge (x \vee y) < i$, which means that $i \not\leq x \vee y$. ■

Lemma 3.3.1.2 (*Safe meet*)**.** *Let us consider a finite distributive lattice L. For all meet-irreducible element $i \in {}^\wedge\mathcal{I}(L)$ and for all elements $x, y \in L$ that are not less than or equal to i, i.e. $i \not\geq x$ and $i \not\geq y$, we have that their meet is not less or equal to i either, i.e. $i \not\geq x \wedge y$.*

Proof. By definition of a meet-irreducible element, we know that $\forall i \in {}^\wedge\mathcal{I}(L)$ and for all $a, b \in L$, if $a > i$ and $b > i$, then $a \wedge b > i$. Moreover, for all $x, y \in L$ such that $i \not\geq x$ and $i \not\geq y$, we have equivalently $i \vee x > i$ and $i \vee y > i$. Thus, we get that $(i \vee x) \wedge (i \vee y) > i$. Since L satisfies the distributive law Eq. (3.4), this implies that $(i \vee x) \wedge (i \vee y) = i \vee (x \wedge y) < i$, which means that $i \not\geq x \vee y$. ■

Proposition 3.3.1.1 (*Iota elements of a subset of P*)**.** For any subset $S \subseteq P$, we note $\iota(S)$ the set containing the join-irreducible elements of the smallest sublattice L_S of P containing S, i.e. $\iota(S) = {}^{\vee}\mathcal{I}(L_S)$. These so-called *iota elements of S* can be obtained through the following equality:

$$\iota(S) = \left\{ \bigwedge i^{\uparrow S} \ / \ i \in {}^{\vee}\mathcal{I}(P),\ i^{\uparrow S} \neq \emptyset \right\},$$

where $i^{\uparrow S}$ is the upper closure of i in S, i.e. $\{s \in S \ / \ i \leq s\}$.

Proof. See Appendix B.1. ■

Proposition 3.3.1.2 (*Dual iota elements of a subset of P*)**.** Similarly, for any subset $S \subseteq P$, we note $\bar{\iota}(S)$ the set containing the meet-irreducible elements of the smallest sublattice L_S of P containing S, i.e. $\bar{\iota}(S) = {}^{\wedge}\mathcal{I}(L_S)$. These so-called *dual iota elements of S* can be obtained through the following equality:

$$\bar{\iota}(S) = \left\{ \bigvee i^{\downarrow S} \ / \ i \in {}^{\wedge}\mathcal{I}(P),\ i^{\downarrow S} \neq \emptyset \right\},$$

where $i^{\downarrow S}$ is the lower closure of i in S, i.e. $\{s \in S \ / \ i \geq s\}$.

Proof. Analog to the proof of Proposition 3.3.1.1, exploiting the dual definition of the distributive law in a lattice, i.e. Eq. (3.4). ■

Proposition 3.3.1.3 (*Lattice support*)**.** The smallest sublattice of P containing both ${}^{\wedge}\text{supp}(f)$ and ${}^{\vee}\text{supp}(f)$, noted ${}^{\mathcal{L}}\text{supp}(f)$, can be defined as:

$$\begin{aligned} {}^{\mathcal{L}}\text{supp}(f) &= \left\{ \bigvee X \ / \ X \subseteq \iota(\text{supp}(f)),\ X \neq \emptyset \right\} \cup \left\{ \bigwedge \text{supp}(f) \right\} \\ &= \left\{ \bigwedge X \ / \ X \subseteq \bar{\iota}(\text{supp}(f)),\ X \neq \emptyset \right\} \cup \left\{ \bigvee \text{supp}(f) \right\}. \end{aligned}$$

More specifically, ${}^{\vee}\text{supp}(f)$ is contained in the upper closure $\text{supp}(f)^{\uparrow^{\mathcal{L}}\text{supp}(f)}$ of $\text{supp}(f)$ in ${}^{\mathcal{L}}\text{supp}(f)$:

$$\text{supp}(f)^{\uparrow^{\mathcal{L}}\text{supp}(f)} = \{x \in {}^{\mathcal{L}}\text{supp}(f) \ / \ \exists s \in \text{supp}(f),\ s \leq x\},$$

and ${}^{\wedge}\text{supp}(f)$ is contained in the lower closure $\text{supp}(f)^{\downarrow^{\mathcal{L}}\text{supp}(f)}$ of $\text{supp}(f)$ in ${}^{\mathcal{L}}\text{supp}(f)$:

$$\text{supp}(f)^{\downarrow^{\mathcal{L}}\text{supp}(f)} = \{x \in {}^{\mathcal{L}}\text{supp}(f) \ / \ \exists s \in \text{supp}(f),\ s \geq x\}.$$

These sets can be computed in less than respectively $O(|\iota(\text{supp}(f))|.|\text{supp}(f)^{\uparrow^{\mathcal{L}}\text{supp}(f)}|)$ and $O(|\iota(\text{supp}(f))|.|\text{supp}(f)^{\downarrow^{\mathcal{L}}\text{supp}(f)}|)$, which is at most $O(|{}^{\vee}\mathcal{I}(P)|.|P|)$.

Proof. The proof is immediate here, considering Proposition 3.3.1.1 and its proof, as well as Proposition 3.3.1.2. In addition, since ${}^{\wedge}\text{supp}(f)$ only contains the meet of elements of $\text{supp}(f)$, all element of ${}^{\wedge}\text{supp}(f)$ is less than at least one element of $\text{supp}(f)$. Similarly, since ${}^{\vee}\text{supp}(f)$ only contains the join of elements of $\text{supp}(f)$, all element of ${}^{\vee}\text{supp}(f)$ is greater than at least one element of $\text{supp}(f)$. Hence $\text{supp}(f)^{\uparrow^{\mathcal{L}}\text{supp}(f)}$ and $\text{supp}(f)^{\downarrow^{\mathcal{L}}\text{supp}(f)}$. ■

As pointed out in [63], a special ordering of the join-irreducible elements of a lattice when using the Fast Zeta Transform [62] leads to the optimal computation of its zeta and Möbius transforms. Here, we use this ordering to build our EMT for finite distributive lattices in a way similar to [63] but without the need to check the equality of the decompositions into the first j join-irreducible elements at each step.

Corollary 3.3.1.1 (*Join-irreducible ordering*). *Let us consider a finite distributive lattice* $(L, \leq)$ *and let its join-irreducible elements* ${}^{\vee}\mathcal{I}(L)$ *be ordered such that* $\forall i_k, i_l \in {}^{\vee}\mathcal{I}(L)$, $k < l \Rightarrow i_k \not\geq i_l$. *If* L *is a graded lattice (i.e. a lattice equipped with a rank function* $\rho : L \to \mathbb{N}$*), then* $\rho(i_1) \leq \rho(i_2) \leq \cdots \leq \rho(i_{|{}^{\vee}\mathcal{I}(L)|})$ *implies this ordering. We note* ${}^{\vee}\mathcal{I}(L)_k = \{i_1, \ldots, i_{k-1},\ i_k\}$.

For all element $i_k \in {}^{\vee}\mathcal{I}(L)$, *we have* $i_k \not\leq \bigvee {}^{\vee}\mathcal{I}(L)_{k-1}$.

Proof. Since the join-irreducible elements are ordered such that $\forall i_k, i_l \in {}^{\vee}\mathcal{I}(L)$, $k < l \Rightarrow i_k \not\geq i_l$, it is trivial to see that for any $i_l \in {}^{\vee}\mathcal{I}(L)$ and $i_k \in {}^{\vee}\mathcal{I}(L)_{l-1}$, we have $i_k \not\geq i_l$. Then, using Lemma 3.3.1.1 by recurrence, it is easy to get that $i_l \not\leq \bigvee {}^{\vee}\mathcal{I}(L)_{l-1}$. ■

Example 3.3.1.1. For example, in DST we work with $P = 2^{\Omega}$, in which the rank function is the cardinality, i.e. for all $A \in P$, $\rho(A) = |A|$. So, with $(L, \subseteq)$ a subset lattice of $(2^{\Omega}, \subseteq)$, the ordering required in Corollary 3.3.1.1 simply translates to sorting the join-irreducible elements of L from the smallest set to the largest set. Thus, sorting ${}^{\vee}\mathcal{I}(L) = \{i_1, i_2, \cdots, i_n\}$ such that $|i_1| \leq |i_2| \leq \cdots \leq |i_n|$, Corollary 3.3.1.1 tells us that for any $k \in [\![2, n]\!]$, we have $i_k \not\subseteq \bigcup {}^{\vee}\mathcal{I}(L)_{k-1}$, where ${}^{\vee}\mathcal{I}(L)_k = \{i_1, \ldots, i_{k-1},\ i_k\}$.

Corollary 3.3.1.2 (*Meet-irreducible ordering*). *Let us consider a finite distributive lattice* $(L, \leq)$ *and let its meet-irreducible elements* ${}^{\wedge}\mathcal{I}(L)$ *be ordered such that* $\forall i_k, i_l \in {}^{\vee}\mathcal{I}(L)$, $k < l \Rightarrow i_k \not\leq i_l$. *If* L *is a graded lattice (i.e. a lattice equipped with a rank function* $\rho : L \to \mathbb{N}$*), then* $\rho(i_1) \geq \rho(i_2) \geq \cdots \geq \rho(i_{|{}^{\wedge}\mathcal{I}(L)|})$ *implies this ordering. We note* ${}^{\wedge}\mathcal{I}(L)_k = \{i_1, \ldots, i_{k-1},\ i_k\}$.

For all element $i_k \in {}^{\wedge}\mathcal{I}(L)$, *we have* $i_k \not\geq \bigwedge {}^{\wedge}\mathcal{I}(L)_{k-1}$.

Proof. Since the meet-irreducible elements are ordered such that $\forall i_k, i_l \in {}^{\wedge}\mathcal{I}(L)$, $k < l \Rightarrow i_k \not\leq i_l$, it is trivial to see that for any $i_l \in {}^{\wedge}\mathcal{I}(L)$ and $i_k \in {}^{\wedge}\mathcal{I}(L)_{l-1}$, we have $i_k \not\leq i_l$. Then, using Lemma 3.3.1.2 by recurrence, it is easy to get that $i_l \not\geq \bigwedge {}^{\wedge}\mathcal{I}(L)_{l-1}$. ■

Example 3.3.1.2. Taking back Example 3.3.1.1, the ordering required in this Corollary 3.3.1.2 simply translates to sorting the meet-irreducible elements of L from the largest set to the smallest set. Thus, sorting ${}^{\wedge}\mathcal{I}(L) = \{i_1, i_2, \cdots, i_n\}$ such that $|i_1| \geq |i_2| \geq \cdots \geq |i_n|$, Corollary 3.3.1.2 tells us that for any $k \in [\![2, n]\!]$, we have $i_k \not\supseteq \bigcap {}^{\wedge}\mathcal{I}(L)_{k-1}$, where ${}^{\wedge}\mathcal{I}(L)_k = \{i_1, \ldots, i_{k-1},\ i_k\}$.

3.3.2 Main results

In this subsection, we present our two types of transformations computing the zeta and Möbius transforms of a function $f : P \to \mathbb{R}$. The first one corresponds to Theorem 3.3.2.1 and its corollaries and is based on a sublattice $L \subseteq P$, where L contains the lattice support of f, i.e. ${}^{\mathcal{L}}\text{supp}(f) \subseteq L$. The second one corresponds to Theorem 3.3.2.2 and its corollary and is based on any lower subsemilattice of P containing ${}^{\wedge}\text{supp}(f)$. Its dual is presented in Corollary 3.3.2.7 and its corollary and is based on any upper subsemilattice of P containing ${}^{\vee}\text{supp}(f)$.

Theorem 3.3.2.1 (*Efficient Möbius Transformation in a distributive lattice*). *Let us consider a finite distributive lattice* L *(such as* ${}^{\mathcal{L}}supp(f)$*) and let its join-irreducible elements* ${}^{\vee}\mathcal{I}(L) = \{i_1, i_2, \cdots, i_n\}$ *be ordered such that* $\forall i_k, i_l \in {}^{\vee}\mathcal{I}(L)$, $k < l \Rightarrow i_k \not\geq i_l$.

Consider the sequence $(H_k)_{k \in [\![1,n]\!]}$, *where* $H_k = (L, E_k)$ *and:*

$$E_k = \left\{(x, y) \in L^2 \ / \ y = x \vee i_{n+1-k}\right\}.$$

This sequence computes the same zeta transformations as $G_< = (L, E_<)$, *where* $E_< = \{(X, Y) \in L^2 \ / \ X < Y\}$. *This sequence is illustrated in Fig. 3.3. The execution of any transformation based on this sequence is* $O(n.|L|)$ *in time and* $O(|L|)$ *in space.*

Proof. See Appendix B.2. ■

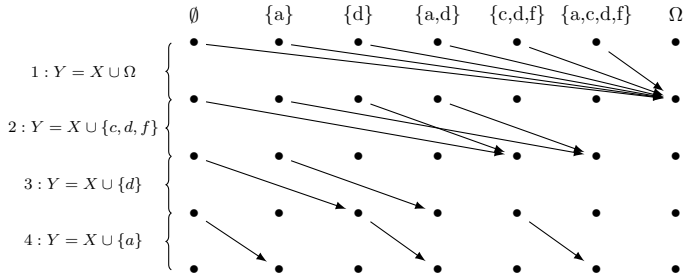


Figure 3.3: Illustration representing the paths generated by the arrows contained in the sequence $(H_k)_{k\in[\![1,4]\!]}$, where $H_k = (L, E_k)$ and $E_k = \{(X, Y) \in L^2 \ / \ Y = X \cup i_{5-k}\}$ and $L = \{\emptyset, \{a\}, \{d\}, \{a, d\}, \{c, d, f\}, \{a, c, d, f\}, \Omega\}$ and $\Omega = \{a, b, c, d, e, f\}$ and $(i_k)_{k\in[\![1,4]\!]} = (\{a\}, \{d\}, \{c, d, f\}, \Omega)$. This sequence computes the same zeta transformations as $G_\subset = (L, E_\subset)$, where $E_\subset = \{(X, Y) \in L^2 \ / \ X \subset Y\}$.

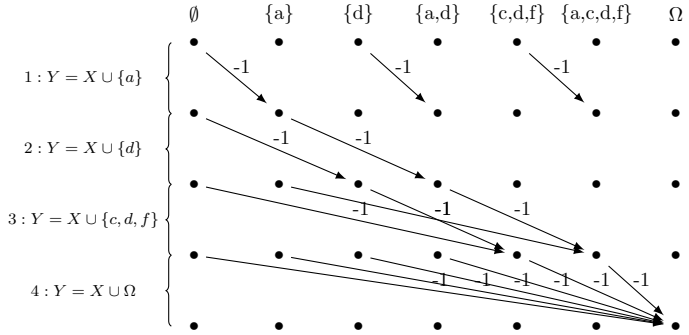


Figure 3.4: Illustration representing the paths generated by the arrows contained in the sequence $(H_k)_{k\in[\![1,4]\!]}$, where $H_k = (L, E_k)$ and $E_k = \{(X, Y) \in L^2 \ / \ Y = X \cup i_k\}$ and $L = \{\emptyset, \{a\}, \{d\}, \{a, d\}, \{c, d, f\}, \{a, c, d, f\}, \Omega\}$ and $\Omega = \{a, b, c, d, e, f\}$ and $(i_k)_{k\in[\![1,4]\!]} = (\{a\}, \{d\}, \{c, d, f\}, \Omega)$. This sequence computes the same Möbius transformations as $G_\subset = (L, E_\subset)$, where $E_\subset = \{(X, Y) \in L^2 \ / \ X \subset Y\}$. The "-1" labels emphasize the intended use of the operator $-$ with this sequence.

Corollary 3.3.2.1. *It can be shown similarly that the sequence $(H_k)_{k\in[\![1,n]\!]}$ computes the same zeta transforms as the sequence of Theorem 3.3.2.1, where $H_k = (L, E_k)$ and:*

$$E_k = \left\{(x, y) \in L^2 \ / \ x = y \wedge \bar{i}_k\right\}.$$

with the meet-irreducible elements $^\wedge\mathcal{I}(L) = \{\bar{i}_1, \bar{i}_2, \cdots, \bar{i}_n\}$ ordered such that $\forall \bar{i}_k, \bar{i}_l \in {}^\wedge\mathcal{I}(L)$, $k < l \Rightarrow \bar{i}_k \not\leq \bar{i}_l$, i.e. in reverse order compared to the join-irreducible elements of Theorem 3.3.2.1.

Corollary 3.3.2.2. *Consider the sequence $(H_k)_{k\in[\![1,n]\!]}$, where $H_k = (L, E_k)$ and:*

$$E_k = \left\{(x, y) \in L^2 \ / \ y = x \vee i_k\right\}.$$

This sequence computes the same Möbius transformations as $G_< = (L, E_<)$, where $E_< = \{(X, Y) \in L^2 \ / \ X < Y\}$. This sequence is illustrated in Fig. 3.4 and leads to the same complexities as the one presented in Theorem 3.3.2.1.

Proof. For every arrow $(x, y) \in E_k$, if $i_k \not\leq x$, then there is an arrow such that $y = x \vee i_k \neq x$. However, there can be no arrow $(w, x) \in E_k$ since x cannot be equal to $w \vee i_k$. Otherwise, if

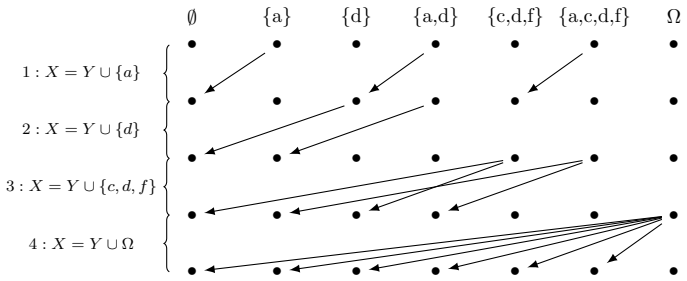


Figure 3.5: Illustration representing the paths generated by the arrows contained in the sequence $(H_k)_{k\in[\![1,4]\!]}$, where $H_k = (L, E_k)$ and $E_k = \{(X,Y) \in L^2 \ / \ X = Y \cup i_k\}$ and $L = \{\emptyset, \{a\}, \{d\}, \{a,d\}, \{c,d,f\}, \{a,c,d,f\}, \Omega\}$ and $\Omega = \{a,b,c,d,e,f\}$ and $(i_k)_{k\in[\![1,4]\!]} = (\{a\}, \{d\}, \{c,d,f\}, \Omega)$. This sequence computes the same zeta transformations as $G_\supset = (L, E_\supset)$, where $E_\supset = \{(X,Y) \in L^2 \ / \ X \supset Y\}$.

$i_k \leq x$, then $y = x \vee i_k = x$, i.e. (x,y) is an identity arrow. Thus, for every arrow $(x,y) \in E_k \backslash I_P$, there is no arrow (w,x) in $E_k \backslash I_P$, meaning that Theorem 3.2.4.1 is satisfied. The sequence $(H_{n-k+1})_{k\in[\![1,n]\!]}$ computes the same Möbius transformations as $G_<$. ■

Corollary 3.3.2.3. *Again, it can be shown similarly that the sequence $(H_k)_{k\in[\![1,n]\!]}$ computes the same Möbius transforms as the sequence of Corollary 3.3.2.2, where $H_k = (L, E_k)$ and:*

$$E_k = \left\{(x,y) \in L^2 \ / \ x = y \wedge \bar{i}_{n+1-k}\right\}.$$

with the meet-irreducible elements ${}^\wedge\mathcal{I}(L) = \{\bar{i}_1, \bar{i}_2, \cdots, \bar{i}_n\}$ ordered such that $\forall \bar{i}_k, \bar{i}_l \in {}^\wedge\mathcal{I}(L)$, $k < l \Rightarrow \bar{i}_k \not\leq \bar{i}_l$, i.e. in reverse order compared to the join-irreducible elements of Corollary 3.3.2.2.

Corollary 3.3.2.4. *Dually, consider the sequence $(H_k)_{k\in[\![1,n]\!]}$, where $H_k = (L, E_k)$ and:*

$$E_k = \left\{(x,y) \in L^2 \ / \ x = y \vee i_k\right\}.$$

This sequence computes the same zeta transformations as $G_> = (L, E_>)$, where $E_> = \{(X,Y) \in L^2 \ / \ X > Y\}$. This sequence is illustrated in Fig. 3.5 and leads to the same complexities as its dual.

Corollary 3.3.2.5. *Consider the sequence $(H_k)_{k\in[\![1,n]\!]}$, where $H_k = (L, E_k)$ and:*

$$E_k = \left\{(x,y) \in L^2 \ / \ x = y \vee i_{n+1-k}\right\}.$$

This sequence computes the same Möbius transformations as $G_> = (L, E_>)$, where $E_> = \{(X,Y) \in L^2 \ / \ X > Y\}$. This sequence is illustrated in Fig. 3.6 and leads to the same complexities as the one presented in Theorem 3.3.2.1.

The procedures exploiting the sequences of graphs described in Theorem 3.3.2.1 and its corollaries to compute the zeta and Möbius transforms of a function f on P is always less than $O(|{}^\vee\mathcal{I}(P)|.|P|)$. Their upper bound complexity for the distributive lattice $L = {}^\mathcal{L}\text{supp}(f)$ is $O(|{}^\vee\mathcal{I}(L)|.|L|)$, which is actually the optimal one for a lattice.

Yet, we can reduce this complexity even further if we have ${}^\wedge\text{supp}(f)$ or ${}^\vee\text{supp}(f)$. This is the motivation behind the sequence proposed in the following Theorem 3.3.2.2. As a matter of fact, [62] proposed a meta-procedure producing an algorithm that computes the zeta and Möbius transforms in an arbitrary intersection-closed family F of sets of 2^Ω with a circuit of size $O(|\Omega|.|F|)$. However, this meta-procedure is $O(|\Omega|.2^{|\Omega|})$. Here, Theorem 3.3.2.2 provides a

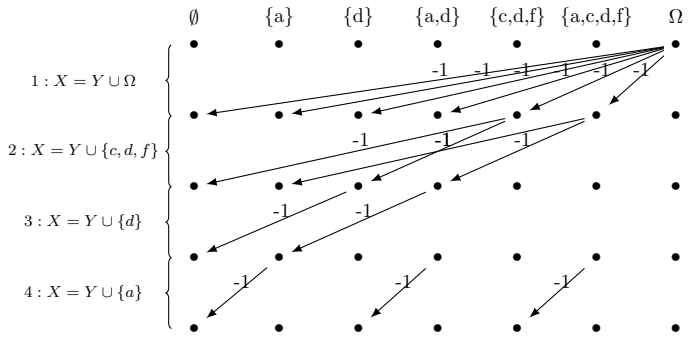


Figure 3.6: Illustration representing the paths generated by the arrows contained in the sequence $(H_k)_{k\in[\![1,4]\!]}$, where $H_k = (L, E_k)$ and $E_k = \{(X, Y) \in L^2 \ / \ X = Y \cup i_{5-k}\}$ and $L = \{\emptyset, \{a\}, \{d\}, \{a, d\}, \{c, d, f\}, \{a, c, d, f\}, \Omega\}$ and $\Omega = \{a, b, c, d, e, f\}$ and $(i_k)_{k\in[\![1,4]\!]} = (\{a\}, \{d\}, \{c, d, f\}, \Omega)$. This sequence computes the same Möbius transformations as $G_\supset = (L, E_\supset)$, where $E_\supset = \{(X, Y) \in L^2 \ / \ X \supset Y\}$. The "-1" labels emphasize the intended use of the operator $-$ with this sequence.

sequence of graphs that leads to procedures that directly compute the zeta and Möbius transforms in $O(|\Omega|.|F|.\epsilon)$, where ϵ can be as low as 1. Besides, our method is far more general since it applies to any meet-closed or join-closed subset of a finite distributive lattice. The intuition behind it is that we can do the same computations as in Theorem 3.3.2.1, even in a subsemilattice, simply by bridging gaps, from the smallest gap to the biggest to make sure that all nodes are visited.

Theorem 3.3.2.2 (*Efficient Möbius Transformation in a join-closed or meet-closed subset of P*)**.** *Let us consider a meet-closed subset M of P (such as $^\wedge supp(f)$). Also, let the iota elements $\iota(M) = \{i_1, i_2, \cdots, i_n\}$ be ordered such that $\forall i_k, i_l \in \iota(M)$, $k < l \Rightarrow i_k \not\geq i_l$.*

Consider the sequence $(H_k)_{k\in[\![1,n]\!]}$, where $H_k = (M, E_k)$ and:

$$E_k = \left\{(x, y) \in M^2 \ / \ x = \bigwedge (y \vee i_k)^{\uparrow M} \text{ and } x \leq y \vee \bigvee \iota(M)_k\right\},$$

where $\iota(M)_k = \{i_1, i_2, \ldots, i_k\}$. This sequence computes the same zeta transformations as $G_> = (M, E_>)$, where $E_> = \{(X, Y) \in M^2 \ / \ X > Y\}$. This sequence is illustrated in Fig. 3.7. The execution of any transformation based on this sequence is at most $O(|\iota(M)|.|M|)$ in space and $O(|\iota(M)|.|M|.\epsilon)$ in time, where ϵ represents the average number of operations required to "bridge a gap", i.e. to find the minimum of $(y \vee i_k)^{\uparrow M}$.

Proof. See Appendix B.3 ■

Remark. This number ϵ is hard to evaluate beforehand since it depends on both the number of "gaps to bridge" and the average number of elements that can be greater than some element. If there is no gap, or if there is only one greater element, this ϵ can be as low as 1. Moreover, the choice of data structures may greatly reduce this ϵ. For instance, if $P = 2^\Omega$, then dynamic binary trees may be employed to avoid the consideration of elements that cannot be less than some already found element in $(y \cup i_k)^{\uparrow M}$, by cutting some branches.

Corollary 3.3.2.6. *Consider the sequence $(H_k)_{k\in[\![1,n]\!]}$, where $H_k = (M, E_k)$ and:*

$$E_k = \left\{(x, y) \in M^2 \ / \ x = \bigwedge (y \vee i_{n+1-k})^{\uparrow M} \text{ and } x \leq y \vee \bigvee \iota(M)_{n+1-k}\right\}.$$

This sequence computes the same Möbius transformations as $G_> = (M, E_>)$, where $E_> =$

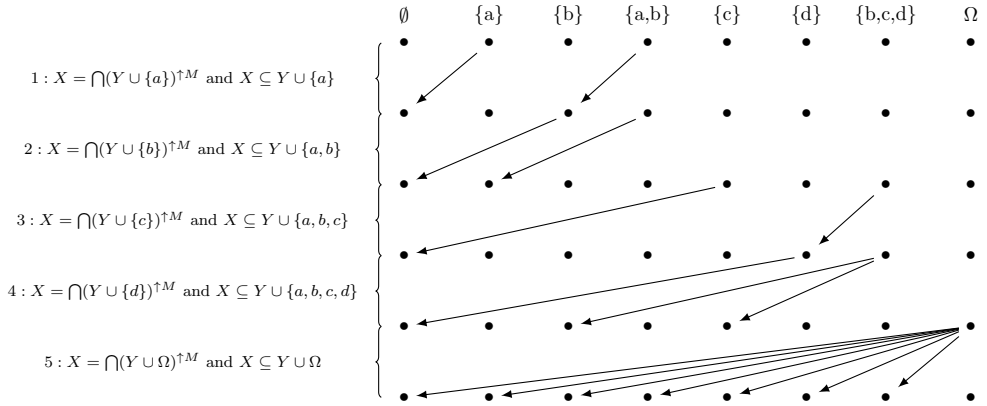


Figure 3.7: Illustration representing the paths generated by the arrows contained in the sequence $(H_k)_{k\in[\![1,5]\!]}$, where $H_k = (M, E_k)$ and $E_k = \left\{(X, Y) \in M^2 \ / \ X = \bigcap(Y \cup i_k)^{\uparrow M} \text{ and } X \subseteq Y \cup \bigcup \iota(M)_k\right\}$ and $\iota(M)_k = \{i_1, i_2, \ldots, i_k\}$ and $M = \{\emptyset, \{a\}, \{b\}, \{a, b\}, \{c\}, \{d\}, \{b, c, d\}, \Omega\}$ and $\Omega = \{a, b, c, d, e, f\}$ and $(i_k)_{k\in[\![1,5]\!]} = (\{a\}, \{b\}, \{c\}, \{d\}, \Omega)$. This sequence computes the same zeta transformations as $G_\supset = (M, E_\supset)$, where $E_\supset = \{(X, Y) \in M^2 \ / \ X \supset Y\}$. Actually, since there is no order between any two iota elements of $\iota(M)\backslash\{\Omega\}$ in this example, the order chosen here is arbitrary. Any order would compute the same zeta transformations as $G_\supset$, as long as Ω is the last iota element to consider.

$\{(X, Y) \in M^2 \ / \ X > Y\}$. *This sequence is illustrated in Fig. 3.8 and leads to the same complexities as the one presented in Theorem 3.3.2.2.*

Proof. Analog to the proof of Corollary 3.3.2.2. ■

Corollary 3.3.2.7. *Dually, let us consider a join-closed subset J of P (such as ${}^\vee supp(f)$). Also, let the dual iota elements $\bar{\iota}(J) = \{\bar{i}_1, \bar{i}_2, \cdots, \bar{i}_n\}$ be ordered such that $\forall \bar{i}_k, \bar{i}_l \in \bar{\iota}(J)$, $k < l \Rightarrow \bar{i}_k \not\leq \bar{i}_l$, i.e. in reverse order compared to the iota elements of Theorem 3.3.2.2.*

In the direct line of Corollary 3.3.2.1, consider the sequence $(H_k)_{k\in[\![1,n]\!]}$, where $H_k = (J, E_k)$ and:

$$E_k = \left\{(x, y) \in J^2 \ / \ x = \bigvee (y \wedge \bar{i}_k)^{\downarrow J} \text{ and } x \geq y \wedge \bigwedge \bar{\iota}(J)_k\right\},$$

where $\bar{\iota}(J)_k = \{\bar{i}_1, \bar{i}_2, \ldots, \bar{i}_k\}$. This sequence computes the same zeta transformations as $G_< = (J, E_<)$, where $E_< = \{(X, Y) \in J^2 \ / \ X < Y\}$. This sequence is illustrated in Fig. 3.9. The execution of any transformation based on this sequence is at most $O(|\bar{\iota}(J)|.|J|)$ in space and $O(|\bar{\iota}(J)|.|J|.\epsilon)$ in time, where ϵ represents the average number of operations required to "bridge a gap", i.e. to find the maximum of $(y \wedge \bar{i}_k)^{\downarrow M}$.

Corollary 3.3.2.8. *Finally, in the direct line of Corollary 3.3.2.3, consider the sequence $(H_k)_{k\in[\![1,n]\!]}$, where $H_k = (J, E_k)$ and:*

$$E_k = \left\{(x, y) \in J^2 \ / \ x = \bigvee (y \wedge \bar{i}_{n+1-k})^{\downarrow J} \text{ and } x \geq y \wedge \bigwedge \bar{\iota}(J)_{n+1-k}\right\}.$$

This sequence computes the same Möbius transformations as $G_< = (J, E_<)$, where $E_< = \{(X, Y) \in J^2 \ / \ X < Y\}$. This sequence is illustrated in Fig. 3.10 and leads to the same complexities as the one presented in Corollary 3.3.2.7.

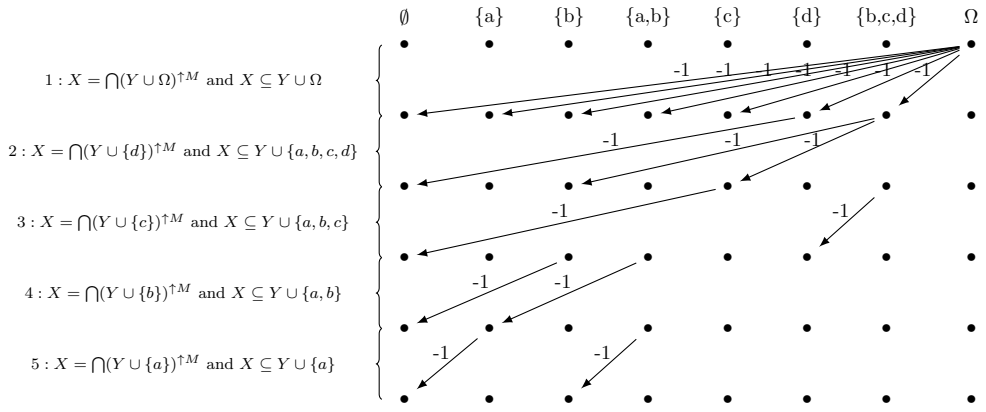


Figure 3.8: Illustration representing the paths generated by the arrows contained in the sequence $(H_k)_{k\in[\![1,5]\!]}$, where $H_k = (M, E_k)$ and $E_k = \left\{(X,Y) \in M^2 \ / \ X = \bigcap(Y \cup i_{6-k})^{\uparrow M} \text{ and } X \subseteq Y \cup \bigcup \iota(M)_{6-k}\right\}$ and $\iota(M)_k = \{i_1, i_2, \ldots, i_k\}$ and $M = \{\emptyset, \{a\}, \{b\}, \{a,b\}, \{c\}, \{d\}, \{b,c,d\}, \Omega\}$ and $\Omega = \{a, b, c, d, e, f\}$ and $(i_k)_{k\in[\![1,5]\!]} = (\{a\}, \{b\}, \{c\}, \{d\}, \Omega)$. This sequence computes the same Möbius transformations as $G_\supset = (M, E_\supset)$, where $E_\supset = \{(X,Y) \in M^2 \ / \ X \supset Y\}$. The "-1" labels emphasize the intended use of the operator $-$ with this sequence. Actually, since there is no order between any two iota elements of $\iota(M)\backslash\{\Omega\}$ in this example, the order chosen in Fig. 3.7 is arbitrary. Thus, any order here would compute the same Möbius transformations as $G_\supset$, as long as Ω is the first iota element to consider.

3.4 Discussions

3.4.1 General usage

If $|\text{supp}(f)|$ is of same order of magnitude as $|{}^\vee\mathcal{I}(P)|$ or lower, then we can directly compute the focal points ${}^\wedge\text{supp}(f)$ or ${}^\vee\text{supp}(f)$. Next, with ${}^\wedge\text{supp}(f)$, we can compute Efficient Möbius Transformations based on Theorem 3.3.2.2 to get the zeta or Möbius transform of any function f in $(P, \geq)$.

Let us take the sequence $(H_k)_{k\in[\![1,n]\!]}$ from Theorem 3.3.2.2, where $H_k = ({}^\wedge\text{supp}(f), E_k)$ and:

$$E_k = \left\{(x,y) \in {}^\wedge\text{supp}(f)^2 \ / \ x = \bigwedge(y \vee i_k)^{\uparrow^\wedge\text{supp}(f)} \text{ and } x \leq y \vee \bigvee \iota({}^\wedge\text{supp}(f))_k\right\},$$

where $\iota({}^\wedge\text{supp}(f))_k = \{i_1, i_2, \ldots, i_k\}$ such that $\forall i_k, i_l \in \iota({}^\wedge\text{supp}(f))$, $k < l \Rightarrow i_k \not\geq i_l$.

Example 3.4.1.1. Consider the mass function m and the commonality function q from Example 3.2.1.3. The transformation $((H_k)_{k\in[\![1,n]\!]}, m, +)$, where ${}^\wedge\text{supp}(f) = {}^\wedge\text{supp}(m)$, computes the commonality function q.

Example 3.4.1.2. Consider the mass function m and the commonality function q from Example 3.2.2.3. The transformation $((H_{n+1-k})_{k\in[\![1,n]\!]}, q, -)$, where ${}^\wedge\text{supp}(f) = {}^\wedge\text{supp}(m)$, computes the mass function m.

Example 3.4.1.3. Consider the conjunctive weight function w and the commonality function q from Example 3.2.1.4. The transformation $((H_k)_{k\in[\![1,n]\!]}, w^{-1}, \times)$, where ${}^\wedge\text{supp}(f) = {}^\wedge\text{supp}(w-1)$, computes the commonality function q.

Example 3.4.1.4. Consider the conjunctive weight function w and the commonality function q from Example 3.2.2.4. The transformation $((H_{n+1-k})_{k\in[\![1,n]\!]}, w^{-1}, /)$, where ${}^\wedge\text{supp}(f) = {}^\wedge\text{supp}(w-1)$, computes the conjunctive weight function w.

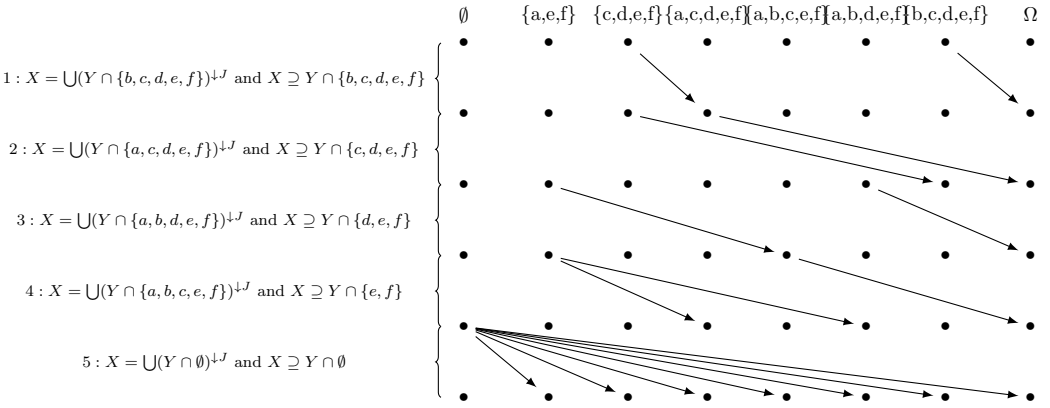


Figure 3.9: Illustration representing the paths generated by the arrows contained in the sequence $(H_k)_{k\in[\![1,5]\!]}$, where $H_k = (J, E_k)$ and $E_k = \left\{(X,Y) \in J^2 \ / \ X = \bigcup(Y \cap \bar{\imath}_k)^{\downarrow J} \text{ and } X \supseteq Y \cap \bigcap \bar{\iota}(J)_k\right\}$ and $\bar{\iota}(J)_k = \{\bar{i}_1, \bar{i}_2, \ldots, \bar{i}_k\}$ and $J = \{\emptyset, \{a,e,f\}, \{c,d,e,f\}, \{a,c,d,e,f\}, \{a,b,c,e,f\}, \{a,b,d,e,f\}, \{b,c,d,e,f\}, \Omega\}$ and $\Omega = \{a,b,c,d,e,f\}$ and $(\bar{i}_k)_{k\in[\![1,5]\!]} = (\{b,c,d,e,f\}, \{a,c,d,e,f\}, \{a,b,d,e,f\}, \{a,b,c,e,f\}, \emptyset)$. This sequence computes the same zeta transformations as $G_\subset = (J, E_\subset)$, where $E_\subset = \{(X,Y) \in J^2 \ / \ X \subset Y\}$. Actually, since there is no order between any two dual iota elements of $\bar{\iota}(J)\backslash\{\emptyset\}$ in this example, the order chosen here is arbitrary. Any order would compute the same zeta transformations as $G_\subset$, as long as $\emptyset$ is the last dual iota element to consider.

Let us now take the sequence $(H_k)_{k\in[\![1,n]\!]}$ from Corollary 3.3.2.7, where $H_k = ({}^\vee\text{supp}(f), E_k)$ and:

$$E_k = \left\{(x,y) \in {}^\vee\text{supp}(f)^2 \ / \ x = \bigvee (y \wedge \bar{i}_k)^{\downarrow {}^\vee\text{supp}(f)} \text{ and } x \geq y \wedge \bigwedge \bar{\iota}({}^\vee\text{supp}(f))_k\right\},$$

where $\bar{\iota}({}^\vee\text{supp}(f))_k = \{\bar{i}_1, \bar{i}_2, \ldots, \bar{i}_k\}$ and such that $\forall \bar{i}_k, \bar{i}_l \in \bar{\iota}({}^\vee\text{supp}(f))$, $k < l \Rightarrow \bar{i}_k \not\leq \bar{i}_l$.

Example 3.4.1.5. Consider the mass function m and the implicability function b from Example 3.2.1.1. The transformation $((H_k)_{k\in[\![1,n]\!]}, m, +)$, where ${}^\vee\text{supp}(f) = {}^\vee\text{supp}(m)$, computes the implicability function b.

Example 3.4.1.6. Consider the mass function m and the implicability function b from Example 3.2.2.1. The transformation $((H_{n+1-k})_{k\in[\![1,n]\!]}, b, -)$, where ${}^\vee\text{supp}(f) = {}^\vee\text{supp}(m)$, computes the mass function m.

Example 3.4.1.7. Consider the disjunctive weight function v and the implicability function b from Example 3.2.1.2. The transformation $((H_k)_{k\in[\![1,n]\!]}, v^{-1}, \times)$, where ${}^\vee\text{supp}(f) = {}^\vee\text{supp}(v-1)$, computes the implicability function b.

Example 3.4.1.8. Consider the disjunctive weight function v and the implicability function b from Example 3.2.2.2. The transformation $((H_{n+1-k})_{k\in[\![1,n]\!]}, v^{-1}, /)$, where ${}^\vee\text{supp}(f) = {}^\vee\text{supp}(v-1)$, computes the disjunctive weight function v.

These transformations can be computed in at most $O(|{}^\vee\mathcal{I}(P)|.|\text{supp}(f)| + |I(\text{supp}(f))|.|{}^o\text{supp}(f)|)$ operations, where $I \in \{\iota, \bar{\iota}\}$ and $o \in \{\wedge, \vee\}$, which is at most $O(|{}^\vee\mathcal{I}(P)|.|P|)$.

Otherwise, if $|\text{supp}(f)| \gg |{}^\vee\mathcal{I}(P)|$, then we can compute $\text{supp}(f)^{\uparrow^{\mathcal{L}}\text{supp}(f)}$ or $\text{supp}(f)^{\downarrow^{\mathcal{L}}\text{supp}(f)}$ from the lattice support of Proposition 3.3.1.3, and then compute Efficient Möbius Transformations based on Theorem 3.3.2.1. Doing so, computing the same transforms can be done in

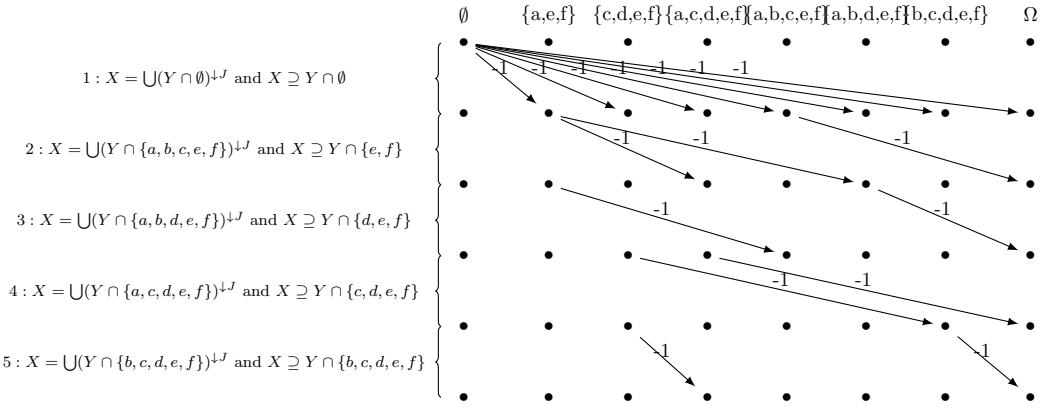


Figure 3.10: Illustration representing the paths generated by the arrows contained in the sequence $(H_k)_{k\in[\![1,5]\!]}$, where $H_k = (J, E_k)$ and $E_k = \left\{(X,Y) \in J^2 \;/\; X = \bigcup(Y \cap \bar{\imath}_{6-k})^{\downarrow J} \text{ and } X \supseteq Y \cap \bigcap \bar{\iota}(J)_{6-k}\right\}$ and $\bar{\iota}(J)_k = \{\bar{\imath}_1, \bar{\imath}_2, \ldots, \bar{\imath}_k\}$ and $J = \{\emptyset, \{a,e,f\}, \{c,d,e,f\}, \{a,c,d,e,f\}, \{a,b,c,e,f\}, \{a,b,d,e,f\}, \{b,c,d,e,f\}, \Omega\}$ and $\Omega = \{a,b,c,d,e,f\}$ and $(\bar{\imath}_k)_{k\in[\![1,5]\!]} = (\{b,c,d,e,f\}, \{a,c,d,e,f\}, \{a,b,d,e,f\}, \{a,b,c,e,f\}, \emptyset)$. This sequence computes the same Möbius transformations as $G_\subset = (J, E_\subset)$, where $E_\subset = \{(X,Y) \in J^2 \;/\; X \subset Y\}$. The "-1" labels emphasize the intended use of the operator $-$ with this sequence. Actually, since there is no order between any two dual iota elements of $\bar{\iota}(J)\backslash\{\emptyset\}$ in this example, the order chosen in Fig. 3.9 is arbitrary. Thus, any order would compute the same Möbius transformations as $G_\subset$, as long as $\emptyset$ is the first dual iota element to consider.

at most $O(|^\vee\mathcal{I}(P)|.|\text{supp}(f)| + |I(\text{supp}(f))|.|\text{supp}(f)^{A^{\mathcal{L}}\text{supp}(f)}|)$ operations, where $I \in \{\iota, \bar{\iota}\}$ and $A \in \{\uparrow, \downarrow\}$, which is at most $O(|^\vee\mathcal{I}(P)|.|P|)$.

Either way, it is always possible to compute zeta and Möbius transforms in a distributive lattice in less than $O(|^\vee\mathcal{I}(P)|.|P|)$ in time and space.

3.4.2 Dempster-Shafer Theory

So, we can always compute most DST transformations (See Examples 3.4.1.1 to 3.4.1.8), wherever the FMT applies, in less than $O(|\Omega|.2^{|\Omega|})$ operations in the general case. The EMT are always more efficient than the FMT.

Moreover, $\text{supp}(f)^{\downarrow^{\mathcal{L}}\text{supp}(f)}$ can be optimized if $\Omega \in \text{supp}(f)$. Indeed, in this case, we have $\text{supp}(f)^{\downarrow^{\mathcal{L}}\text{supp}(f)} = {}^{\mathcal{L}}\text{supp}(f)$, while there may be a lot less elements in $(\text{supp}(f)\backslash\{\Omega\})^{\downarrow^{\mathcal{L}}\text{supp}(f)}$. If so, one can equivalently compute the down set $(\text{supp}(f)\backslash\{\Omega\})^{\downarrow^{\mathcal{L}}\text{supp}(f)}$, execute an EMT with Theorem 3.3.2.1, and then add the value on Ω to the value on all sets of $(\text{supp}(f)\backslash\{\Omega\})^{\downarrow^{\mathcal{L}}\text{supp}(f)}$.

The same can be done with $(\text{supp}(f)\backslash\{\emptyset\})^{\uparrow^{\mathcal{L}}\text{supp}(f)}$. This trick can be particularly useful in the case of the conjunctive and disjunctive weight function, which require that $\text{supp}(f)$ contains respectively Ω and $\emptyset$.

Also, optimizations built for the FMT, such as the reduction of Ω to the core $\mathcal{C}$ or its optimal coarsened version Ω', are already encoded in the use of the function ι (see Example 3.4.2.1). On the other hand, optimizations built for the evidence-based approach, such as approximations by reduction of the number of focal sets, i.e. reducing the size of $\text{supp}(f)$, can still greatly enhance the EMT.

Finally, it was proposed in [43] to fuse two mass functions m_1 and m_2 using Demp-

ster's rule by computing the corresponding commonality functions q_1 and q_2 in $O(|\Omega|.2^{|\Omega|})$, then computing $q_{12} = q_1.q_2$ in $O(2^{|\Omega|})$ and finally computing back the fused mass function m_{12} from q_{12} in $O(|\Omega|.2^{|\Omega|})$. Here, we propose to compute the same detour but only on the elements of $\text{supp}(m_{12}) \subseteq (\text{supp}(m_1) \cup \text{supp}(m_2))^{\downarrow^{\mathcal{L}}\text{supp}(f)}$. Indeed, notice that $\text{supp}(m_{12}) \subseteq {}^{\wedge}(\text{supp}(m_1) \cup \text{supp}(m_2))$, which implies that $\text{supp}(m_{12})^{\downarrow^{\mathcal{L}}\text{supp}(f)} \subseteq (\text{supp}(m_1) \cup \text{supp}(m_2))^{\downarrow^{\mathcal{L}}\text{supp}(f)}$. Thus, noting $L = (\text{supp}(m_1) \cup \text{supp}(m_2))^{\downarrow^{\mathcal{L}}\text{supp}(f)}$, we compute the corresponding commonality functions q_1 and q_2 in $O(|\iota(L)|.|L|)$, then compute $q_{12} = q_1.q_2$ in $O(|L|)$ and finally compute back the fused mass function m_{12} from q_{12} in $O(|\iota(L)|.|L|)$, where $\iota(L) = \iota(\text{supp}(m_1) \cup \text{supp}(m_2))$.

Example 3.4.2.1 (*Coarsening in the consonant case*)**.** Let $\text{supp}(f) = \{F_1, F_2, \ldots, F_K\}$ such that $F_1 \subset F_2 \subset \cdots \subset F_K$. A coarsening Ω' of Ω is a mapping from disjoint groups of elements of Ω to elements of Ω'. The set Ω' can be seen as a partition of Ω. The goal of this coarsening of Ω is to provide a reduced powerset $2^{\Omega'}$. The best coarsening in this example would create as much elements in Ω' as there are elements in $\text{supp}(f)$. Thus, the best coarsening would give us a powerset of size $2^{|\text{supp}(f)|}$.

On the other hand, our iota elements $\iota(\text{supp}(f))$ are the join-irreducible elements of the smallest sublattice of 2^{Ω} containing $\text{supp}(f)$. This lattice is what we called the *lattice support of* f and noted ${}^{\mathcal{L}}\text{supp}(f)$. By definition, we necessarily have $|{}^{\mathcal{L}}\text{supp}(f)| \leq 2^{|\text{supp}(f)|}$. More precisely here, all elements of $\text{supp}(f)$ are both focal points and join-irreducible elements of ${}^{\mathcal{L}}\text{supp}(f)$, i.e. $\iota(\text{supp}(f)) = \text{supp}(f) = {}^{\vee}\text{supp}(f) = {}^{\wedge}\text{supp}(f)$, if $\emptyset \notin \text{supp}(f)$ (Otherwise, we have $\iota(\text{supp}(f)) = \text{supp}(f) \backslash \{\emptyset\}$). In fact, since our iota elements are not mapped elements of a reduced set Ω' but instead raw sets from 2^{Ω}, combinations of joins lead to a vastly different lattice. In this example, we have ${}^{\mathcal{L}}\text{supp}(f) = \text{supp}(f)$, instead of the $2^{\text{supp}(f)}$ given by coarsening.

3.5 Conclusion

In this chapter, we proposed the *Efficient Möbius Transformations* (EMT), which are general procedures to compute the zeta and Möbius transforms of any function defined on any finite distributive lattice with optimal complexity. They are based on our reformulation of the Möbius inversion theorem with focal points only, featured in the previous chapter corresponding to our journal paper [52]. The EMT optimally exploit the information contained in both the support of this function and the structure of distributive lattices. Doing so, the EMT always perform better than the optimal complexity for an algorithm considering the whole lattice, such as the FMT. Following these findings, it remains to propose explicit algorithms and implementation guidelines. We provide this for the powerset lattice, for DST, in Appendix C.

This closes the presentation of our contributions around the efficient computation of DST transformations and their fusion. The next and last chapter of this thesis will propose a solution tackling communication issues such as redundancy in a distributed collaboration and the general lack of conciseness in vehicular networks.

Chapter 4

Learning to value the unknown

4.1 Introduction

Recently, we have been witnesses of accidents involving autonomous vehicles and their lack of sufficient information at the right time. One way to tackle this issue is to benefit from the perception of different viewpoints, namely collaborative perception. While setting a multitude of sensors in the road infrastructure could be imagined, this would require a lot of investments and limit its usage to some areas in the world. Instead, we focus on the exchange of information between vehicles about their common environment, where they are the only sources available.

These communications can simply be centralized by a server that would gather all information from all vehicles to process it and re-distribute it to all, as suggested in [12]. However, this still consists of Vehicle-to-Infrastructure (V2I) communications, which implies (1) an infrastructure cost and the impossibility to share information with other agents when there is no server available nearby. It also features the disadvantage of (2) making the agents broadcast their entire perception, which can be heavy on the means of communication and computation and give rise to delays.

In contrast, the decentralized Vehicle-to-Vehicle (V2V) approach [16–18, 21, 22] does not require any extra infrastructure to work, i.e. does not implies (1). In this setting, agents directly exchange pieces of information between them. It also comes with new problems such as data incest and lower computation capabilities. We will ignore them here as we already tackled the issue of avoiding data incest using Dempster-Shafer Theory (DST) [24] in spite of low computation capabilities with two conference papers [42, 53] and a journal paper [52]. But V2V communications bring a potentially heavier communication burden as well, due to redundancies. In fact, (2) is worse in this setting than in the centralized one if agents are passive, meaning if they simply broadcast their perception for the others to know, without filtering it beforehand. Nevertheless, this decentralized approach offers the possibility to make the agents active in their quest for full perception, i.e. making the agents ask for specific areas in their surroundings on which they would like to know more, instead of always broadcasting everything. This is impossible in the centralized setting, as the server decides and thus needs to gather all perceptions beforehand.

Here, we propose such a system, where each agent builds its own local top-down semantic grid and sends specific requests to others in the form of bounding boxes described in the global reference frame. We choose local grid maps for their ability to map an agent's knowledge and to deduce its uncertainties in space.

4.2 Related Works

Since not all uncertain areas are relevant, Active Exploration [65, 66] is not enough; a truly efficient collaboration policy requires some understanding of the scenery [67], extracted from the spatial arrangement of grid cells and their classes. What could lie in the shadows and how to best discover it? If a pedestrian is heading towards an occluded area, we expect the agent to request for this area, as a tracking system. If the agent has no idea of what could be in the unknown, maybe it could ask for some key points to understand the layout of the environment. If an area on the road is near a crowd of people or in the continuity of a pedestrian crossing, ask for it as some unseen-before pedestrians could be crossing, etc. More generally, we would like the agent to know as much as possible about moving objects in its vicinity, while avoiding to request too much information from others. This represents a complex bounding box selection policy to be learned from pixels.

Given the long-lasting successes of Deep Learning in such ordeals, it seems natural to consider neural networks for our problem. But, while it is theoretically possible (but practically challenging) to learn our policy in an end-to-end fashion with model-free Deep Reinforcement Learning (DRL), we choose to first learn a deep generative model to pre-process our inputs. Indeed, training deep neural networks is easier, faster and more stable when the loss on the output is in the form of a well-justified derivable function, which is hard to achieve with reward signals from a RL environment. Building this generative model also allows for more control and insights on what is learned, and reduces the size of the neural networks that are supposed to be trained through model-free DRL. As demonstrated in World Models [68], learning a policy on top of a model can even be achieved with simple heuristics such as Evolution Strategies (ES), with performances equivalent to RL algorithms.

Our model needs to be generative, for inference in unknown areas. In addition, we want it to be predictive, in order to make it understand latent dynamics, anticipating disappearances or inferring hidden road users from the behavior of visible ones. Doing so, it could even eventually compensate for communication latencies. Such a model would be useful in itself for other tasks as well, e.g. autonomous driving.

Several existing works [69–73] employed generative models with convolutional networks in a U-Net architecture in order to augment instantaneous individual grid maps. Some used deterministic networks such as Generative Adversarial Networks (GAN). Others tried to incorporate stochasticity with Monte Carlo Dropout or simply using a Variational Auto-Encoder (VAE). Most used occupancy grids as input, but some chose semantic grid maps or DOGMa (occupancy grid with velocities). These inputs were either expressed in a static global reference frame or given to a system that had no prediction capability. Doing so, it appears that none of these approaches really modeled the long-term dynamics of the environment that would be necessary to learn our desired policy. On the other hand, a kind of recurrent generative model inspired by the VAE, namely Temporal Difference VAE (TD-VAE) [74], was designed with the specific intent of being taken as base for a reinforcement learning algorithm. It puts an emphasis on the learning of belief states for long-term predictions, which are important for the development of complex strategies. It has been proven in [75] that explicitly predicting future states enhances data-efficiency in a number of RL tasks, though they train their model jointly with the policy and do not use the loss defined in [74]. Appealed by the theoretical justifications of TD-VAE, its decoupling regarding specific RL tasks (which simplifies the search for good RL hyperparameters) and its demonstrated ability to predict plausible sequences of images in a 3D world at different time horizons and from a variable number of observations, we have implemented and adapted this TD-VAE to our problem. However, correcting some of its weaknesses regarding its actual prediction capability, we finally proposed our own model, called Locally Predictable VAE (LP-VAE). To learn our communication policy based on this model, we chose the widely used

Proximal Policy Optimization (PPO) algorithm [29], which is a fairly stable and simple policy gradient-based DRL algorithm with few hyperparameters.

Closely related to our goal, other works try to address the problem of efficiently communicating between autonomous vehicles. In [76], they used a joint Perception and Prediction (P&P) model that transforms sensor data into learned features to broadcast to other vehicles. This model also fuses received features with local ones and tries to predict the trajectory of nearby communicating vehicles. This information compression is also present in our work in the form of a Convolutional VAE preprocessing each observation grid. We go one step further in communication efficiency as our system does not broadcast every piece of information, but chooses instead which one it wishes to receive. Sending learned features also forces them to make another neural network learn to spatially and temporally transform all pieces of information received from the vehicular network. Even the fusion operation is done by making a neural network learn how to fuse two learned features, without any guarantee on the result. Instead, here we rely on top-down semantic grids, which are simple discretizations of the space around the ego-vehicle. Doing so, we can transform the content of our transmissions using linear transformations. Furthermore, our system keeps its integrity by only fusing probability distributions.

In [77], they used Deep Reinforcement Learning to select only a portion of the perceptive field of an autonomous vehicle to send to others. However, this information filtering is done on the sender side, contrary to our approach that filters on the receiver side. Doing so, their approach still consists in broadcasting pieces of information, regardless of the actual needs of others.

The same can be stated for [78], where they describe a V2V cooperative perception system in which vehicles exchange object detections. They try to reduce redundancies by estimating the value of a piece of information for a potential receiver. The value here is the novelty, i.e. the probability that the potential receiver is not aware of some object of interest.

Section 4.3 formally introduces our communication problem, justifying the use of a preprocessing generative model. Section 4.4 provides some background on PPO and explains why we chose this RL algorithm to learn our communication policy. Section 4.5 formalizes the aforementioned generative model, introducing TD-VAE and LP-VAE. Section 4.6 presents our deep networks implementing these models. Then, section 4.7 evaluates and compares the performance of different versions of our models and policy learnings. Finally, we conclude this chapter with section 4.8.

4.3 Problem formulation

We formulate our communication problem as a Markov Decision Process (MDP). Fig. 4.1 gives an overview of it, working with the driving simulator CARLA [6] for our experiments.

4.3.1 State space

We assume the existence of a driving policy from which we only know the actions taken at each time step: ego-vehicle controls (acceleration and steering angle, each ranging in $[-1, 1]$) and global direction (average of the next 10 equally-spaced points the planner set to visit in meters relative to the ego-vehicle's reference). This driving policy influences the road environment in which the ego-vehicle is moving. This is not the case with the communication environment that we consider in this MDP. Each observation is a tuple (G_t, C_t, V_t), where G_t is an ego-centered semantic grid, C_t represents the actions taken by the driving policy at a given instant t (which influence G_{t+1}) and V_t is the motion of the ego-vehicle between $t-1$ and t. Each

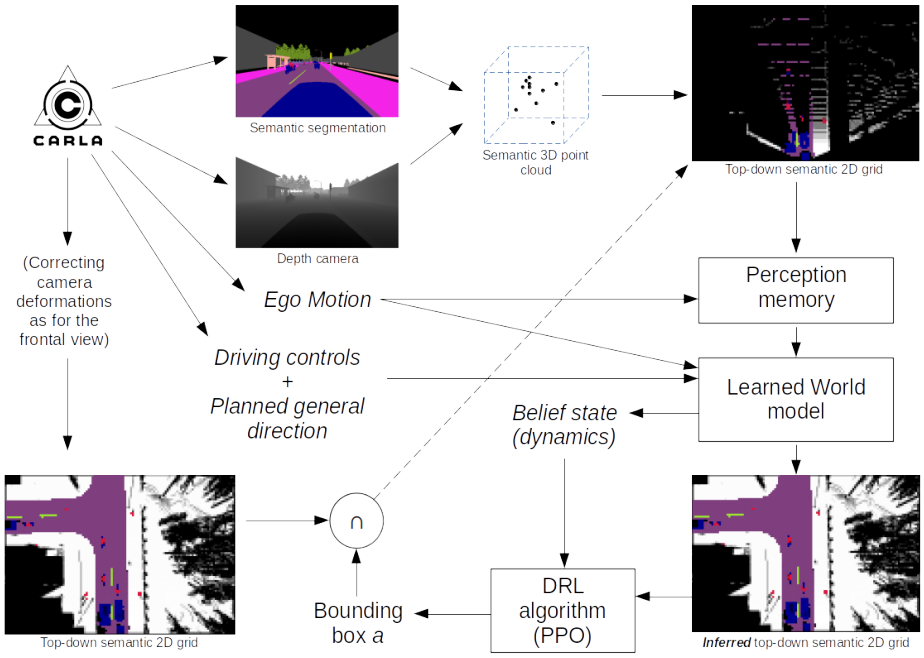


Figure 4.1: Illustration of our application. CARLA provides a semantic segmentation corresponding to a camera attached to the ego-vehicle hood, as well as its corresponding depth (images taken from [6]). This gives us enough information to create a semantic 3D point cloud, i.e. to scatter all pixels in space according to their depth and image coordinates (and the camera deformation). From it, we project these pixels back into a 2D plane (i.e. a grid), but from a top-down point of view (and without camera deformations). In parallel, we get the ego-vehicle motion since the previous time step in order to update a *perception memory* containing 2D points from previous time steps. We add the current semantic grid to this memory and give the resulting augmented grid to our *learned world model* (STD-VAE or LP-VAE), along with the ego-motion and driving policy commands. In turn, this model tries to guess what is hidden in occluded areas and provides a belief state about latent dynamics. These outputs are then given to a *DRL algorithm* that chooses a grid area to request to the world. This area is extracted at the next time step from a grid generated by a camera above the ego-vehicle. Finally, this information is fused at the next time step with the ego-vehicle perception.

semantic grid G_t is a top-down 6-channels pseudo-Bayesian mass grid corresponding to the five classes of the frame of discernment $\Omega = \{pedestrian, car, road\ lines, road, other\}$. The class *car* actually contains any type of vehicle, even bikes. The class *road lines* contains any road marking: road lines, arrows, painted stop signs, etc. The class *other* contains the rest of the static objects perceivable by the agent, such as vegetation, sidewalks, buildings, etc. The last channel represents ignorance, i.e. the mass put on Ω. This means that $G_t \geq 0$ and, for any cell index i of G_t, we have $\sum_{k=1}^{6} G_t[i][k] = 1$. These cells are distributed as a matrix (grid) of 80 rows and 120 columns, i.e. G_t is analog to a $80 \times 120 \times 6$ image of values in $[0, 1]$. See Fig. 4.2 for a visualization of this semantic grid.

These observations constitute a very large and complex space which would be hard to transform into exploitable neural network features without a derivable loss function. Thus, we will first build a generative model of the driving environment (implicitly including the agent's driving policy). Besides, learning this model beforehand will give us more control on the information

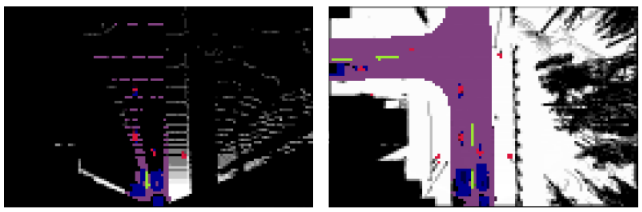


Figure 4.2: **Left:** Illustration of an instance of top-down semantic grid G_t corresponding to a partial observation x_t in our model. Red is for *pedestrians*, blue is for *cars*, yellow is for *road lines*, purple is for *road*, white is for *other* and black is for ignorance. The displayed class is the one with the greatest mass. The intensity of its color depends on its mass: the closer to 0, the darker. Notice all the occlusions due to walls or other road users, in addition to the limited distance of perception of the ego-vehicle. **Right:** Instance of top-down semantic grid corresponding to a complete observation y_t in our model. Actually, this view is obtained with a facing ground camera above the ego-vehicle. Doing so, it contains itself some occlusions due to trees, poles, buildings, etc. Thus, it is rather a hint about the true y_t. This view can also be obtained by the fusion of multiple view points, from autonomous vehicles or infrastructure sensors.

flow that should be considered by the communication policy. Therefore, the state space of our MDP is made of learned features from this generative model. Several versions of this generative model are proposed in Section 4.5.

4.3.2 Action space

Our MDP has 4 continuous actions that each ranges in $[0, 1]$, defining a bounding box in the local grid G_t of the ego-vehicle at time t: width, height, column and row. This bounding box is supposed to represent an area in the ego-vehicle's future surroundings.

4.3.3 Transition function

Transitions from a state-action pair to a new state depend also on the driving environment, i.e. CARLA. First, this environment generates a new partial grid G_{t+1} and other observations already described. The bounding box described by the action given at time t is then translated into an area of G_{t+1} filled with complete information. Fig. 4.3 illustrates this process.

In addition, a visual memory mechanism, specific to our MDP, makes perceptions persist for a few time steps, discounted a little more every time. This implements short-term memory, so that we only consider as unknown what has not been perceived in a long time (or never). This also has the effect of giving consequences to past actions, since bounding boxes in the same area will have close to no potential information gain for a few time steps.

4.3.4 Rewards

Finally, let us define a reward function for our MDP. Let r_t be a reward density, defined for each cell i of G_{t+1} as:

$$r_t(i) = -\eta.r_{\min} \; + \; S[i].\sum_{k=1}^{5} r_{obj}[k].\; \max(0,\; G_{t+1}[i][k] - \widetilde{G}_{t+1}[i][k])^w, \tag{4.1}$$

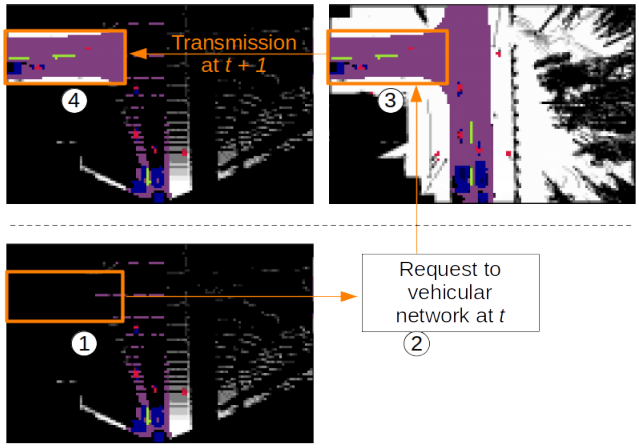


Figure 4.3: Illustration of our decision process: 1) Based on what is known at time t, select a bounding box where there is high uncertainty and high probability to discover road users. 2) Send this request in global coordinates to the vehicular network (which may consists of both infrastructure sensors and other autonomous vehicles). 3) At time $t+1$, we expect some vehicles to transmit their perception of this area. In our implementation, complete perceptions are simply obtained by a camera above the ego-vehicle since we focus on the selection of bounding boxes, i.e. 1). 4) The transmitted partial perception is fused with the one of the ego-vehicle at time $t+1$.

where $w \in \mathbb{R}^{+*}$, $\eta \in [0,1]$ and $\widetilde{G}_{t+1}$ is the grid before fusion with the grid G^M_{t+1} corresponding to M_{t+1}. The quantity r_{obj} is a nonnegative reward per object pixel (only null for the static class *other*, i.e. $r_{obj}[5] = 0$) such that $r_{obj}[k] \geq r_{obj}[k+1]$. Indeed, pedestrian are the smallest identifiable objects among our classes and so must have the highest reward per pixel. The quantity $r_{\min}$ is equal to the least positive reward per pixel, i.e. $r_{\min} = r_{obj}[4]$. It is used to discourage the selection of uninteresting cells. The coefficient η that multiplies it represents the minimum informational gain that is needed to consider this cell worth to be requested. For some value of η, this minimum gain applies to the class with the least reward, while it becomes virtually more and more forgiving as the class has a greater reward per cell. Moreover, notice that $\max(0,\ G_{t+1}[i][k] - \widetilde{G}_{t+1}[i][k]) \in [0,1]$, which implies that $\max(0,\ G_{t+1}[i][k] - \widetilde{G}_{t+1}[i][k])^w \in [0,1]$. This means that w only alters the significance of some gain in mass: for $w \in (0,1)$, $\max(0,\ G_{t+1}[i][k] - \widetilde{G}_{t+1}[i][k])$ will be greater than for $w = 1$, while for $w \in (1,+\infty)$, $\max(0,\ G_{t+1}[i][k] - \widetilde{G}_{t+1}[i][k])$ will be less. In other words, if $w \in (1,+\infty)$, then the gain will have to be more important to have an impact on $r_t(i)$. Finally, S represents a spatial filter to account for the fact that we are not equally interested everywhere in discovering road users. For example, a road user very far ahead is not as valuable an information as a road user just around the corner. We defined a forward filter S_F and a lateral filter S_L, such that $S = S_F.S_L$. We set

$$S_F[i] = 1 - \left[\frac{\beta_F}{1-\alpha}.\max\left(0,\ \frac{F(i)}{\max(F)} - \alpha\right)\right]$$

where $\alpha \in [0,1)$ and $\beta_F \in [0,1]$. The quantity $F(i)$ is the forward distance (number of rows from the row in which the center of the ego-vehicle is) corresponding to cell i. The greater the parameter β_F, the less the farest cells are valued. The greater the parameter α, the farer from the ego-vehicle the decrease in value starts.

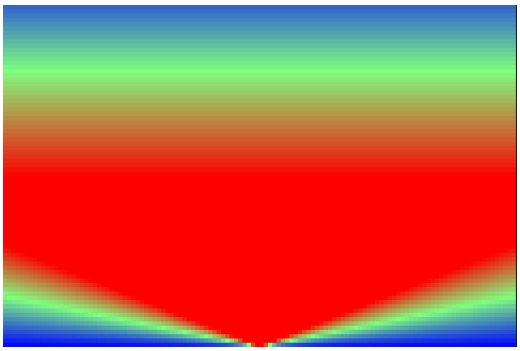


Figure 4.4: Heatmap illustrating our spatial filter S for $\alpha = 0.5$, $\beta_F = 0.8$, $\beta_L = 1$ and $\zeta = 0.01$. Deep blue is 0, while deep red is 1, which means that the reward in a cell located in a blue region will be 0, no matter what is inside. The center of the ego-vehicle is in the middle of the first row starting from bottom.

The second filter is defined as

$$S_L[i] = 1 - \frac{\beta_L}{\zeta}.\max\left(0,\ \zeta - \left|\cos\left(\arctan2\left(L(i), F(i)\right)\right)\right|\ \right)$$

where $\zeta \in (0, 1]$. The quantity $L(i)$ is the lateral distance (number of columns from the column in which the center of the ego-vehicle is) corresponding to cell i. This filter describes a cone in front of the ego-vehicle (and symmetrically at the back of it) in which the cells are the most valued. The greater the parameter ζ, the narrower this cone. The greater the parameter β_L is, the less the cells outside the cone (i.e. on the sides of the ego-vehicle) are valued. Fig. 4.4 provides a visualization of S.

The reward associated with some action a_t is defined as

$$R_t(a_t) = -K.(1-\eta).r_{\min} + \sum_{i \in I(a_t)} r_t(i), \tag{4.2}$$

where K is the minimum number of interesting cells that must be entirely discovered in order to make the request worthwhile, $I(a) = [v(a), v(a) + h(a)] \times [u(a), u(a) + w(a)]$ and $u(a)$, $v(a)$, $w(a)$, $h(a)$ are respectively the column index, row index, width and height indicated by some action a.

Grid fusion

In order to produce G_t from $\widetilde{G}_t$ and the grid G_t^M corresponding to M_t in Eq. (4.1), we need to define a fusion procedure. As each cell i in both $\widetilde{G}_t$ and G_t^M is a mass function, we know that:

$$G_t[i][6] = \widetilde{G}_t[i][6]\ .\ G_t^M[i][6],$$

where 6 is the channel corresponding to the mass on Ω. Furthermore, we can get the contour functions of these pseudo-Bayesian mass functions simply by adding the mass on Ω to the mass on each of our 5 classes. Then, a simple pointwise multiplication of these two contour functions produces the contour function corresponding to G_t. This also implies a mass on $\emptyset$, which is caused by conflicting pieces of evidence between the two mass functions. Since we are not interested in this level of conflict, we choose to renormalize masses as in Dempster's combination rule. Unlike Dempster's rule however, we only distribute this conflict on singletons $G_t[i][1:5]$ and keep the true value $G_t[i][6]$, as the distinction between ignorance and conflict is crucial to our communication policy. Algorithm 1 details this procedure.

Algorithm 1: Fusion procedure for two pseudo-Bayesian mass functions m_1 and m_2.

Input: Two pseudo-Bayesian mass functions m_1, m_2
Output: The fused mass function m_{12}

$N \leftarrow \text{len}(m_1)$;
$m_{12}[N] \leftarrow m_1[N] \ . \ m_2[N]$;
$m_{12}[1:N-1] \leftarrow (m_1[1:N-1] + m_1[N]) \ . \ (m_2[1:N-1] + m_2[N]) - m_{12}[N]$;
$s \leftarrow \text{sum}(m_{12}[1:N-1])$;
if $s > 0$ **then**
  $m_{12}[1:N-1] \leftarrow (1 - m_{12}[N]) \ . \ \frac{m_{12}[1:N-1]}{s}$;
Return m_{12};

4.4 Quick introduction to policy gradient-based reinforcement learning and our choice for PPO

When we want to alter the behavior of an agent without having to discuss, it is possible to associate actions with particular feedbacks. You can think of the training of a dog with food (positive feedback) and gentle slaps (negative feedback), or marketing campaigns/propaganda with more subtle techniques inducing positive or negative feelings when some keywords and ideas are presented. This is the paradigm of Reinforcement Learning (RL) [79]. In RL, an *agent* interacts with its *environment* through *actions*. Based on the *state* of the agent in its environment and the action it has chosen, the environment gives back a particular *reward* to the agent, which can be positive or negative. This environment is typically formulated as a Markov Decision Process (MDP):

- a state space S,
- an action space A,
- a transition function (determining the next state of the agent, based on its current state and the chosen action),
- a reward function r (determining the reward obtained by the agent, based on its current state, the chosen action and its next state).

The function associating a state with an action is called the *policy*, noted π. It represents the agent's behavior. It can be a mapping, if the policy is deterministic, or a probability distribution over the action space A, if the policy is stochastic. A RL algorithm is an iterative algorithm, gathering all rewards r_t obtained by the agent in an episode, under the current policy, before updating the policy and repeating the same process with the new policy. We will mostly ignore the distinction between on-policy and off-policy algorithms in this short introduction. The goal of the algorithm is to find the optimal policy maximizing the expected value of the discounted rewards sum R_t, where $R_t = \sum_{k=0}^{T} \gamma^k . r_{t+k}$ and $\gamma \in [0, 1]$. The time horizon T represents the number of future rewards that are taken into account as being potentially influenced by the choice of action at time t. It can be infinite. The parameter γ brings additional nuances. If $\gamma = 1$, then all future rewards matter as much as the immediate reward r_t. If $\gamma = 0$, then only immediate rewards matter. In other words, γ determines if the policy is more or less short-term or long-term.

4.4.1 Value functions

The value function V^π is the expected value of the discounted rewards sum R_t under the policy π for a given starting state, i.e.:

$$\begin{aligned} V^\pi(s) &= \mathbb{E}_\pi \left[R_t \mid s_t = s \right] \\ &= \mathbb{E}_\pi \left[\sum_{k=0}^{T} \gamma^k . r_{t+k} \;\middle|\; s_t = s \right] \end{aligned} \tag{4.3}$$

It represents the rewards that can be expected from a state s if all future actions, including a_t, are chosen according to the policy π.

The Q-value function Q^π is the expected value of the discounted rewards sum R_t under the policy π for a given starting state and a given first action, i.e.:

$$\begin{aligned} Q^\pi(s,a) &= \mathbb{E}_\pi \left[R_t \mid s_t = s,\ a_t = a \right] \\ &= \mathbb{E}_\pi \left[\sum_{k=0}^{T} \gamma^k . r_{t+k} \;\middle|\; s_t = s,\ a_t = a \right] \end{aligned} \tag{4.4}$$

It represents the rewards that can be expected from a state s if all future actions, excluding a_t which is given as a, are chosen according to the policy π.

Consequently, we have the following equalities:

$$V^\pi(s) = \sum_{a \in A} \pi(a|s) . Q^\pi(s,a)$$

and

$$Q^\pi(s,a) = \sum_{s' \in S} P(s'|s,a) . \left[r(s,a,s') + \gamma . V^\pi(s') \right]$$

where $P(s'|s,a)$ is the probability to be in state s' after choosing action a in state s, and r is here the reward function associating a reward to the tuple consisting of a state, an action and a next state.

The optimal policy π^* can be found through Bellman's optimality equations:

$$V^{\pi^*}(s) = \max_{a \in A} \sum_{s' \in S} P(s'|s,a) . \left[r(s,a,s') + \gamma . V^{\pi^*}(s') \right]$$

$$Q^{\pi^*}(s,a) = \sum_{s' \in S} P(s'|s,a) . \left[r(s,a,s') + \gamma . \max_{a' \in A} Q^{\pi^*}(s',a') \right]$$

Now, consider the update rule of the *Value iteration* algorithm, which is directly derived from Bellman's optimality equations:

$$\forall s \in S, \quad V_{k+1}(s) = \max_{a \in A} \sum_{s' \in S} P(s'|s,a) . \left[r(s,a,s') + \gamma . V_k(s') \right]$$

where V_0 is arbitrary. It is proved that $\lim_{k\to\infty} V_k = V^{\pi^*}$. Consequently, the optimal policy π^* can simply be found after convergence of this value iteration:

$$\forall s \in S, \quad \pi^*(s) = \underset{a\in A}{\operatorname{argmax}} \left[r(s,a) + \gamma . \sum_{s'\in S} P(s'|s,a).V^{\pi^*}(s') \right] \tag{4.5}$$

However, when transition probabilities are unknown or when the state and action spaces are too large, this algorithm is impossible to compute. In this case, Monte Carlo methods must be employed, either to directly estimate V^π or Q^π from respectively Eq. (4.3) and (4.4), or more efficiently to implicitly estimate $P(s'|s,a)$ with the update rule for some fixed policy π:

$$\forall s \in S, \quad V^\pi_{k+1}(s) = (1-\alpha).V^\pi_k(s) + \alpha.\left[r(s,\pi(s),s') + \gamma.V^\pi_k(s')\right]$$

where $\alpha = \frac{1}{n}$ in theory, with n being the number of interactions the agent had with its environment in state s with policy π for this update step $k+1$. In practice however, $n = 1$ and α is fixed to some constant (learning rate), acting as a moving average. It is proved to converge towards V_π, i.e. $\lim_{k\to\infty} V^\pi_k = V^\pi$. This is the basis of *Temporal-Difference (TD) Learning*.

To derive the optimal policy π^* though, we will need to use the equivalent of this update rule for Q instead, since Eq. (4.5) requires to know transition probabilities. There are several ways to proceed, but a notable example is the *off-policy* update rule of the *Q-Learning* algorithm [80]:

$$\forall (s,a) \in S \times A, \quad Q^{\pi^*}_{k+1}(s,a) = (1-\alpha).Q^{\pi^*}_k(s,a) + \alpha.\left[r(s,a,s') + \gamma.\max_{a'\in A} Q^{\pi^*}_k(s',a')\right]$$

where $Q^{\pi^*}_0$ is arbitrary and the rewards $r(s,a,s')$ are obtained beforehand with a policy that is more or less random, to explore the environment. Then, the optimal policy π^* can simply be found with:

$$\forall s \in S, \quad \pi^*(s) = \underset{a\in A}{\operatorname{argmax}}\, Q_{\pi^*}(s,a) \tag{4.6}$$

Instead of Monte Carlo methods, it is possible to learn to approximate the optimal function Q^{π^*} with function approximators such as Neural Networks, hence the emergence of Deep RL (DRL).

4.4.2 The policy gradient approach

Let us consider that we have a set of parameters θ (such as ones of a neural network) determining a probabilistic policy, noted π_θ. Note that here π_θ is a probability distribution over the action space A, where it was a deterministic function until now. In this case, finding the optimal policy π^*_θ consists in finding the set of parameters θ^* such that:

$$\begin{aligned}\theta^* &= \underset{\theta\in\Theta}{\operatorname{argmax}}\, \mathbb{E}_{\pi_\theta}\left[R_t\right] \\ &= \underset{\theta\in\Theta}{\operatorname{argmax}}\, \mathbb{E}_{\pi_\theta}\left[\sum_{k=0}^{T} \gamma^k.r_{t+k}\right]\end{aligned}$$

where Θ is the set of all possible parameter sets. Let us note $\tau \sim \pi_\theta$ some trajectory τ consisting of all state-action pairs (s_t, a_t) sampled from the policy π_θ in an episode. We will note $\tau = (s_0, a_0, s_1, a_1, \ldots, s_T, a_T)$ and $R(\tau) = \sum_{t=0}^{T} \gamma^t.r(s_t,a_t)$. In addition, we note $J(\theta)$ the quantity $\mathbb{E}_{\tau\sim\pi_\theta}\left[R(\tau)\right]$. We have:

$$\begin{aligned}\theta^* &= \underset{\theta\in\Theta}{\operatorname{argmax}}\, \mathbb{E}_{\tau\sim\pi_\theta}\left[R(\tau)\right] \\ &= \underset{\theta\in\Theta}{\operatorname{argmax}}\, J(\theta)\end{aligned}$$

and

$$J(\theta) = \int \pi_\theta(\tau).R(\tau)\ d\tau$$

where $\pi_\theta(\tau)$ is the probability of the trajectory τ under the policy π_θ.

Therefore, the objective of policy gradient-based RL algorithms is to find the parameters θ^* that maximize J through gradient ascent. For this, notice that:

$$\begin{aligned}\frac{\partial J(\theta)}{\partial \theta} &= \int \frac{\partial \pi_\theta(\tau)}{\partial \theta}.R(\tau)\ d\tau \\ &= \int \pi_\theta(\tau).\frac{\partial \pi_\theta(\tau)}{\partial \theta}.\frac{1}{\pi_\theta(\tau)}.R(\tau)\ d\tau \\ &= \mathbb{E}_{\tau\sim\pi_\theta}\left[\frac{\partial \log \pi_\theta(\tau)}{\partial \theta}.R(\tau)\right]\end{aligned}$$

And, since $\pi_\theta(\tau) = \pi_\theta(s_0, a_0, s_1, a_1, \ldots, s_T, a_T) = P(s_0).\prod_{t=0}^{T} \pi_\theta(a_t|s_t).P(s_{t+1}|s_t, a_t)$, we have that

$$\frac{\partial \log \pi_\theta(\tau)}{\partial \theta} = \sum_{t=0}^{T} \frac{\partial \log \pi_\theta(a_t|s_t)}{\partial \theta}.$$

This leads to the following equality:

$$\frac{\partial J(\theta)}{\partial \theta} = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \frac{\partial \log \pi_\theta(a_t|s_t)}{\partial \theta}.\sum_{t=0}^{T} \gamma^t.r(s_t, a_t)\right]$$

So, the optimal policy π_θ^* can be found by iteratively approximating $\frac{\partial J(\theta)}{\partial \theta}$ through Monte Carlo samples of trajectories under the current policy π_θ and updating the parameters θ with $\theta + \alpha.\frac{\partial J(\theta)}{\partial \theta}$, where α is the learning rate. This procedure represents the *REINFORCE* algorithm [81]. A particular advantage of this approach is that finding the optimal policy does not depend on searching for the action giving the maximum over the action space as in Q-Learning, which must then be discrete. Here, the action space can be continuous (or at least very large). This will prove useful in our case where the action space consists of all possible configurations of bounding box in a grid.

Variance reduction

Nevertheless, it is easy to see that this policy gradient-based algorithm can be very unstable, due to the fact that these gradients often have a high variance (which makes Monte Carlo estimates very noisy). There are several techniques that can be employed to reduce this variance. Two common ways are to limit the number of samples to be obtained and to reduce the variance in each state of the sampled trajectories. For the first one, it can be shown that

$$\frac{\partial J(\theta)}{\partial \theta} = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \frac{\partial \log \pi_\theta(a_t|s_t)}{\partial \theta}.\sum_{t'=t}^{T} \gamma^{t'}.r(s_{t'}, a_{t'})\right]$$

For the second one, it can be shown that adding or subtracting terms that only depends on particular states does not introduce any bias since it is independent from the sampled trajectory. It is known that choosing the value function V^{π_θ} for these terms reduces the variance of the aforementioned gradients. Thus, we have:

$$\frac{\partial J(\theta)}{\partial \theta} = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \frac{\partial \log \pi_\theta(a_t|s_t)}{\partial \theta}.\left[\left(\sum_{t'=t}^{T} \gamma^{t'}.r(s_{t'}, a_{t'})\right) - V^{\pi_\theta}(s_t)\right]\right]$$

In fact, it can also be shown that these gradients can be written in terms of Q values:

$$\frac{\partial J(\theta)}{\partial \theta} = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \frac{\partial \log \pi_\theta(a_t|s_t)}{\partial \theta} . \left[Q^{\pi_\theta}(s_t, a_t) - V^{\pi_\theta}(s_t) \right] \right]$$
$$= \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \frac{\partial \log \pi_\theta(a_t|s_t)}{\partial \theta} . A^{\pi_\theta}(s_t, a_t) \right]$$

The function A^{π_θ} is called the *Advantage function*. It represents the advantage of choosing an action over the others, in a certain state, if all future actions are taken in accordance with the policy π_θ.

Actor-Critic architecture

This latter form of gradient is exploited by Actor-Critic RL algorithms. As V^{π_θ} usually cannot be known in practice, Actor-Critic architectures rely on an estimator $\widehat{V}^{\pi_\theta}$. The estimation of the value function $\widehat{V}^{\pi_\theta}$ is called the *Critic*. The *Actor* is the parameterized policy π_θ that is learned.

Actor-Critic DRL architectures

In Actor-Critic DRL algorithms, both the policy π_θ and the value function estimator $\widehat{V}_{\pi_\theta}$ are parameterized by Deep neural networks of parameters θ. Learning the value function, instead of estimating it directly through sampling, is more robust. Indeed, while the target of this value network is the estimation of the value function through sampling, progressively updating through gradient descent smooths out estimation errors. It also avoids to keep in memory potentially very large (even infinite) tables, in accordance with the size of the state space.

Their gradient estimator is of the form:

$$\frac{\partial J(\theta)}{\partial \theta} \approx \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \frac{\partial \log \pi_\theta(a_t|s_t)}{\partial \theta} . \widehat{A}_t \right]$$

To obtain this gradient through automatic differentiation in the back-propagation phase, Actor-Critic DRL algorithms use the objective function:

$$L^{PG}(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \log \pi_\theta(a_t|s_t) . \widehat{A}_t \right]$$

Proximal Policy Optimization (PPO) [29] and Trust Region Policy Optimization (TRPO) [82] (on which PPO is built) are two special Actor-Critic DRL algorithms. The idea behind PPO and TRPO is to further stabilize gradient updates by replacing this L^{PG} by a *surrogate* objective that forces the new policy to be close to the previous one. Let $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ denote the probability ratio of the new policy over the previous one. We have $r_t(\theta_{old}) = 1$. In TRPO, the surrogate objective to maximize is defined as:

$$L^{CPI}(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} r_t(\theta) . \widehat{A}_t \right]$$

The issue with L^{CPI} is that it leads to excessively large policy updates when the advantage is positive and the ratio is more than 1 or when the advantage is negative and the ratio is less

than 1. To remedy this problem, PPO proposes a clipped version, limiting updates that would make the new policy continue to diverge from $\pi_{\theta_{old}}$. Their surrogate objective to maximize is defined as:

$$L^{CLIP}(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T} \min\left(r_t(\theta).\widehat{A}_t,\ \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right).\widehat{A}_t\right)\right]$$

where ϵ is a hyper-parameter in $[0, 1]$. The minimum between the objective of L^{CPI} and its clipped version behaves as follows:

When $\widehat{A}_t < 0$, the action should be discouraged. If the ratio is more than 1, this means that in the previous update, we mistakenly made this action more probable. So, we want the new policy to return towards the old one, as fast as possible. If the ratio is less than 1, this means that in the previous update, we already made this action less probable. So, we want the new policy to *cautiously* continue to move away from the old policy, i.e. not too much as it may be an estimation error and risks to introduce instability.

Conversely, when $\widehat{A}_t > 0$, the action should be encouraged. If the ratio is less than 1, this means that in the previous update, we mistakenly made this action less probable. So, we want the new policy to return towards the old one, as fast as possible. If the ratio is more than 1, this means that in the previous update, we already made this action more probable. So, we want the new policy to *cautiously* continue to move away from the old policy, i.e. not too much as it may be an estimation error and risks to introduce instability.

This results in PPO being one of the most efficient and stable state-of-the-art DRL algorithms, in addition to having few hyper-parameters to adjust. Hence our choice.

In addition, PPO introduces two other hyper-parameters to adjust the importance of two additional losses to minimize simultaneously. One is for minimizing a Mean Squared Error (MSE) between the value $\widehat{V}^{\pi_\theta}$ estimated by the network and the value estimated by sampling. The other is for minimizing an *entropy loss*, which simply tries to minimize all action probabilities to make them tend towards equiprobability. The role of this entropy loss is to act as a regularization slowing the convergence towards local minima by forcing the policy to explore more (hence the emphasis on its randomness).

4.5 Models

In this section, we will present several versions of the generative model mentioned in section 4.3.1, namely STD-VAE and LP-VAE. In the end, this generative model will provide us with learned features describing the state of the environment related to the MDP presented in section 4.3, in order to reduce the size of the network optimized through DRL and to control what is kept in the information flow. We will start by formalizing in section 4.5.1 a draft of this model that ignores the actions the agent takes at each time step. Then, we will briefly introduce in section 4.5.2 the original TD-VAE [74]. Following that, we will propose in section 4.5.3 our sequential variant of TD-VAE, i.e. STD-VAE. Inspired by this model, we will then propose LP-VAE in section 4.5.4. Finally, section 4.5.5 will demonstrate with LP-VAE how to modify this generative model to incorporate the actions chosen by the agent.

4.5.1 Action-independent modeling

As a vehicle clearly cannot access the complete state of its surroundings through its sole perception, we can model our problem as a Partially Observable Discrete-Time Markov Chain

(PO-DTMC), where X_t and Z_t denote random variables representing respectively a partial observation and a latent state at time t. However, we consider that Z_t and X_t are in different spaces, the latent space describing the whole environment and containing information about object dynamics and trajectories allowing for predictions. More precisely, X_t corresponds to the sole perception of the ego-vehicle at time t, without memory of the past. We also introduce a third random variable Y_t which represents the spatially complete observation corresponding to Z_t in the space of X_t. In other words, X_t is a partial observation of Y_t which is itself a partial observation of Z_t.

So, let θ be a set containing the parameters of a generative model that projects a latent state Z_t onto the observation space as $(X_t,\ Y_t)$. We choose to implement this generative model as a deep neural network and we set the following Gaussian distributions as constraints, for numerical stability and simplicity:

- $Z_i \sim \mathcal{N}(0, I_d)$
- $p_{Z_{i+1}|Z_i}(\cdot|z_t;\theta) = \mathcal{N}(\mu_z(z_t;\theta),\ \sigma_z^2(z_t;\theta).I_d)$
- $p_{Y_i|Z_i}(\cdot|z_t;\theta) = \mathcal{N}(\mu_y(z_t;\theta),\ \alpha_y.I_{|X_t|})$
- $p_{X_i|Y_i,Z_i}(\cdot|y_t,z_t;\theta) = \mathcal{N}(\mu_x(y_t,z_t;\theta),\ \alpha_x.I_{|X_t|})$

where μ_z, σ_z, μ_x and μ_y are all deep neural networks taking their parameters in θ, where d is an arbitrary number of dimensions for Z_t, where z_t is a realization of Z_t for some $t \in [1, T]$ and where $\alpha_\cdot \in \left[\frac{1}{2\pi}, +\infty\right)$. This last constraint implies that the generative model recreates independently each dimension of X_t from a latent state z_t with the same fixed precision. Moreover, the PO-DTMC formulation implies that each pair of observations (X_t, Y_t) is only dependent on Z_t, i.e.

$$p_{X,Y|Z}\left(x, y \mid z; \theta\right) = \prod_{t=1}^{T} p_{X_i,Y_i|Z_i}(x_t, y_t \mid z_t; \theta),$$

and that the Markovian property holds in latent space, i.e.

$$p_Z(z;\theta) = p_{Z_i}(z_1).\prod_{t=2}^{T} p_{Z_{i+1}|Z_i}(z_t \mid z_{t-1};\theta).$$

Fig. 4.5 provides the Bayesian network corresponding to our model.

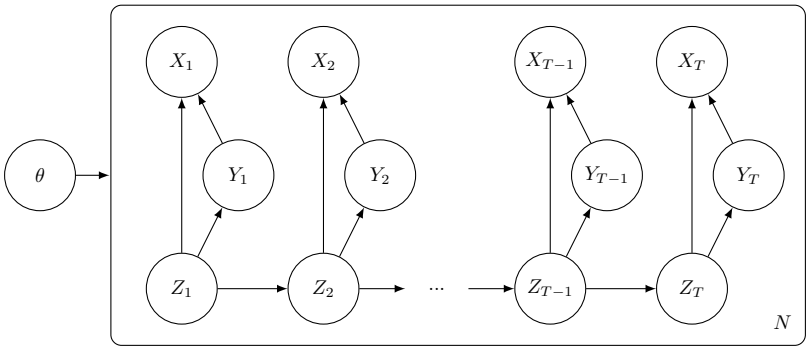


Figure 4.5: Bayesian network of our generative model of parameters in θ. We have N replications of this model, corresponding to the N sequences of length T in our dataset. The parameter set θ influences the inference of all variables in the model for the N sequences we have.

Thus, based on a dataset of N independent sequences of partial and complete observations $D = (x_{1:T},\ y_{1:T})_{1:N}$, we want to optimize the parameters θ so that the probability that the

model generates the sequences of D is maximal under its constraints. In other words, we want to find the parameters θ that maximize $p_{(X,Y)^{(1)},\dots,(X,Y)^{(N)}}(D;\theta)$, which is the same as finding θ maximizing $\log p_{(X,Y)^{(1)},\dots,(X,Y)^{(N)}}(D;\theta)$. We have:

$$\log p_{(X,Y)^{(1)},\dots,(X,Y)^{(N)}}(D;\theta) = \sum_{(x,y)\in D} \log p_{X,Y}(x,y;\theta)$$

where

$$\begin{aligned} p_{X,Y}(x,y;\theta) &= \int p_{X,Y|Z}(x,y \mid z;\theta) \,.\, p_Z(z;\theta)\, dz \\ &= \int \cdots \int p_{Z_i}(z_1).\prod_{t=1}^{T} p_{X_i,Y_i|Z_i}(x_t,y_t \mid z_t;\theta) \,.\, \prod_{t=2}^{T} p_{Z_{i+1}|Z_i}(z_t \mid z_{t-1};\theta) \prod_{t=1}^{T} dz_t \end{aligned}$$

which is intractable, due to the fact that μ_z, σ_z, μ_x and μ_y are multi-layers neural networks with nonlinearities. This intractability is amplified by the fact that we work with sequences of T non-independent continuous latent states, which implies a multiple integral over $\mathbb{R}^{T\times d}$. This means that we cannot evaluate or differentiate the marginal likelihood $p_{X,Y}(x,y;\theta)$. For the same reasons, the posterior distribution

$$p_{Z|X,Y}(\cdot|\ x,y;\theta) = \frac{p_{X,Y|Z}(x,y|\ \cdot\ ;\theta).p_Z(\cdot\ ;\theta)}{p_{X,Y}(x,y;\theta)},$$

is intractable, which implies that methods based on the posterior distribution such as the Expectation-Maximization (EM) algorithm cannot be employed either. So, let us adopt the Variational Bayesian (VB) approach by introducing a variational distribution dependent on a parameter set ϕ to approximate $p_{Z|X,Y}(\cdot|\ x,y;\theta)$. But, more than just a mathematical trick, we want this variational distribution to actually be a recognition model such that it is able to infer latent states only given past partial observations, in order to infer y and to be able to generate plausible next observations.

4.5.2 TD-VAE model

TD-VAE [74] is a variant of the original VAE [28] for temporal sequences which features the particularity to separate *belief states* from latent states. A belief state b_t is a statistics describing $x_{1:t}$ such that $p_{Z_t|X_{1:t}}(\cdot|x_{1:t};\theta) \approx p_{Z_t|B_t}(\cdot|b_t;\theta)$. The end goal motivating this distinction, aside theoretical accuracy, is to learn a model able to deterministically aggregate observations by updating a statistics b_t that contains enough information to infer some latent state z_t, avoiding the accumulation of estimation errors on $z_{1:t-1}$. Since z_t alone allows for predictions of next latent states, b_t constitutes a belief on plausible latent dynamics that is simply updated with each new observation. This feature is important for model-based RL.

In [74], they chose additionally to make their model provide *jumpy* predictions, i.e. directly predicting a latent state $z_{t+\delta}$ from some z_t where δ is not precisely known, in order to abstract latent dynamics for the benefit of computational efficiency. Formally, they seek to optimize θ so that it maximizes the expression

$$\mathop{\mathbb{E}}_{\delta\sim\mathcal{U}_{[\delta_i,\delta_s]}}\left[\mathop{\mathbb{E}}_{t\sim\mathcal{U}_{[1,T-\delta]}}\left[\log p_{X_{t+\delta}|B_t}\left(x_{t+\delta}|b_t;\theta\right)\right]\right], \tag{4.7}$$

where $\mathcal{U}_{[a,b]}$ is the uniform distribution on the interval $[a,b]$ and $B_t = \mathrm{RNN}(X_t, B_{t-1};\phi)$. This cannot be optimized directly, as showed in the previous section. However, we can maximize a lower bound of this expression by introducing a variational distribution.

Let $Q_{t,\delta}(\phi) = q_{Z_t, Z_{t+\delta}|B_t, B_{t+\delta}}(\cdot|b_t, b_{t+\delta}; \phi)$ be this variational distribution, dependent on a parameter set ϕ, such that

$$q_{Z_t, Z_{t+\delta}|B_t, B_{t+\delta}}(\cdot|b_t, b_{t+\delta}; \phi) \approx p_{Z_t, Z_{t+\delta}|B_t, X_{t+\delta}}(\cdot|b_t, x_{t+\delta}; \theta)$$

where it is important to notice that

$$\begin{aligned} p_{Z_t, Z_{t+\delta}|B_t, X_{t+\delta}}(\cdot|b_t, x_{t+\delta}; \theta) &= \frac{p_{X_{t+\delta}, Z_t, Z_{t+\delta}|B_t}(x_{t+\delta}, \cdot|b_t; \theta)}{p_{X_{t+\delta}|B_t}(x_{t+\delta}|b_t; \theta)} \\ &= \frac{P_{t,\delta}(\theta)}{p_{X_{t+\delta}|B_t}(x_{t+\delta}|b_t; \theta)}. \end{aligned}$$

To find the optimal parameters ϕ that minimize its approximation error, we can optimize ϕ so that it minimizes through gradient descent the following average Kullback-Leibler (KL) divergence:

$$\mathop{\mathbb{E}}_{\delta \sim \mathcal{U}_{[\delta_i, \delta_s]}} \left[\mathop{\mathbb{E}}_{t \sim \mathcal{U}_{[1, T-\delta]}} \left[D_{KL} \left(Q_{t,\delta}(\phi) \;\middle||\; \frac{P_{t,\delta}(\theta)}{p_{X_{i+\delta}|B_i}(x_{t+\delta}|b_t; \theta)} \right) \right] \right],$$

This cannot be optimized directly either. Yet, it can be shown that we can equivalently minimize this divergence, while also maximizing a lower bound of (4.7), by minimizing the following loss w.r.t. ϕ and θ:

$$\mathcal{L}_{\text{TD-VAE}}(x; \theta, \phi) = \mathop{\mathbb{E}}_{\delta \sim \mathcal{U}_{[\delta_i, \delta_s]}} \left[\mathop{\mathbb{E}}_{t \sim \mathcal{U}_{[1, T-\delta]}} \left[D_{KL}\left(Q_{t,\delta}(\phi) \;||\; P_{t,\delta}(\theta) \right) \right] \right]$$

where

$$\begin{aligned} &D_{KL}\left(Q_{t,\delta}(\phi) \;||\; P_{t,\delta}(\theta) \right) \\ &= \mathop{\mathbb{E}}_{Z_t, Z_{t+\delta} \sim Q_{t,\delta}(\phi)} \big[\log q_{Z_i|B_i}(z_{t+\delta}|b_{t+\delta}; \phi) \\ &\quad + \log q_{Z_t|B_t, B_{t+\delta}, Z_{t+\delta}}(z_t|b_t, b_{t+\delta}, z_{t+\delta}; \phi) \\ &\quad - \log p_{Z_i|B_i}(Z_t|b_t; \theta) - \log p_{Z_{+\delta}|Z}(Z_{t+\delta}|Z_t; \theta) \\ &\quad - \log p_{X_i|Z_i}(x_{t+\delta}|Z_{t+\delta}; \theta) \big]. \end{aligned}$$

In complement, the authors of [74] had to make the strong assumption that $p_{Z_i|B_i}(\cdot|b_t; \theta) = q_{Z_i|B_i}(\cdot|b_t; \phi)$ for any θ, ϕ. They also set $p_{Z_{+\delta}|Z}(\cdot|z_t; \theta)$ as a multivariate normal distribution with diagonal covariance matrix, corresponding to the distribution of latent states at any instants in $[t+\delta_i, t+\delta_s]$. This is in contradiction with our sequential latent model $p_{Z_{i+1}|Z_i}(\cdot|z_t; \theta)$, which is itself a multivariate normal distribution with diagonal covariance matrix. In this regard, $p_{Z_{+\delta}|Z}(\cdot|z_t; \theta)$ can be seen as a rough approximation.

This abstraction of latent dynamics may be useful in some cases where precision is not needed and the variability of observations $x_{t:t+\delta}$ gathered in a *moment* can be summarized in latent space by smooth transitions between states corresponding to dataset samples. However, we argue that models of complex environments, in which the observation space is combinatorially extremely large and in which multiple agents interact with each other, require precise learning signals to *understand* latent dynamics and so to generalize well outside the training set. More importantly, TD-VAE cannot consider the actions taken by the observing agent between t and $t + \delta$. Yet, learning the link between actions and observations is central in RL.

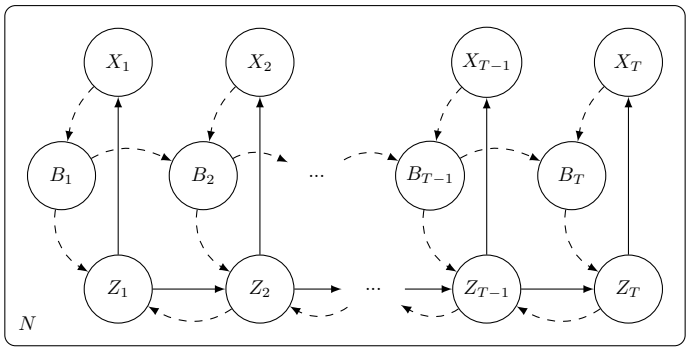


Figure 4.6: Bayesian networks corresponding to STD-VAE. Solid lines represent the Bayesian network of our generative model (without Y_t) of parameters in θ. Dashed lines represent the Bayesian network of the recognition model of parameters in ϕ proposed by TD-VAE. Parameter dependencies are not represented for the sake of clarity. Only B_t is not directly influenced by θ, while only variables at the end of a dashed arrow are influenced by ϕ. We have N replications of this model, corresponding to the N sequences of length T in our dataset.

4.5.3 Our Sequential variant STD-VAE of the TD-VAE model

The authors of [74] also proposed a sequential version of their model. Its corresponding Bayesian network is given in Fig. 4.6. They chose to train its parameters as a particular case of the jumpy one, simply taking $\delta = 1$. Yet, this would only maximize a lower bound of the probability to observe x_{t+1} after b_t, i.e. $\mathop{\mathbb{E}}_{t\sim\mathcal{U}_{[1,T-1]}}\left[\log p_{X_{t+1}|B_t}\left(x_{t+1}|b_t;\theta\right)\right]$, instead of the whole future sequence $x_{t+1:T}$ after b_t, i.e. $\mathop{\mathbb{E}}_{t\sim\mathcal{U}_{[1,T-1]}}\left[\log p_{X_{t+1:T}|B_t}\left(x_{t+1:T}|b_t;\theta\right)\right]$.

From a practical point of view, this would prove to be computationally heavy if done multiple times per sequence and would not learn from the accumulation of prediction errors: particularly in a stochastic network such as TD-VAE and with a time step small enough, the network will tend to optimize weights such that the predicted next state looks almost identical to the initial state. It is only by chaining these predictions that their errors become significant. Thus, we choose a slightly different variational distribution. Let $Q_t(\phi) = q_{Z_{t:T}|B_{t:T}}\left(\cdot|b_{t:T};\phi\right)$ be this variational distribution, dependent on a parameter set ϕ, such that

$$q_{Z_{t:T}|B_{t:T}}\left(\cdot|b_{t:T};\phi\right) \approx p_{Z_{t:T}|B_t,X_{t+1:T}}\left(\cdot|b_t,x_{t+1:T};\theta\right)$$

where it is important to notice that

$$\begin{aligned}
&p_{Z_{t:T}|B_t,X_{t+1:T}}\left(\cdot|b_t,x_{t+1:T};\theta\right)\\
&= \frac{p_{X_{t+1:T},Z_{t:T}|B_t}\left(x_{t+1:T},\cdot|b_t;\theta\right)}{p_{X_{t+1:T}|B_t}\left(x_{t+1:T}|b_t;\theta\right)}\\
&= \frac{P_t(\theta)}{p_{X_{t+1:T}|B_t}\left(x_{t+1:T}|b_t;\theta\right)}.
\end{aligned}$$

To find the optimal parameters ϕ that minimize its approximation error, we can optimize ϕ so that it minimizes through gradient descent the following average Kullback-Leibler (KL) divergence:

$$\mathop{\mathbb{E}}_{t\sim\mathcal{U}_{[1,T-1]}}\left[D_{KL}\left(Q_t(\phi)\;\middle|\middle|\;\frac{P_t(\theta)}{p_{X_{t+1:T}|B_t}\left(x_{t+1:T}|b_t;\theta\right)}\right)\right],$$

It can be shown that we can equivalently minimize this divergence, while also maximizing a lower bound of

$$\mathop{\mathbb{E}}_{t\sim\mathcal{U}_{[1,T-1]}}\left[\log p_{X_{t+1:T}|B_t}\left(x_{t+1:T}|b_t;\theta\right)\right],$$

by minimizing the following loss w.r.t. ϕ and θ:

$$\mathcal{L}_{\text{STD-VAE}}(x;\theta,\phi) = \mathop{\mathbb{E}}_{t\sim\mathcal{U}_{[1,T-1]}}\left[D_{KL}\left(Q_t(\phi)\;||\;P_t(\theta)\right)\right]$$

where

$$\begin{aligned}
&D_{KL}\left(Q_t(\phi)\;||\;P_t(\theta)\right)\\
&=\mathop{\mathbb{E}}_{Z_{t:T}\sim Q_t(\phi)}\Bigg[\log q_{Z_i|B_i}\left(Z_T|b_T;\phi\right)\\
&\quad+\sum_{k=t}^{T-1}\log q_{Z_i|B_i,Z_{i+1}}\left(Z_k|b_k,Z_{k+1};\phi\right)\\
&\quad-\log p_{Z_i|B_i}\left(Z_t|b_t;\theta\right)-\sum_{k=t+1}^{T}\log p_{Z_{i+1}|Z_i}\left(Z_k|Z_{k-1};\theta\right)\\
&\quad-\sum_{k=t}^{T}\log p_{X_i|Z_i}\left(x_k|Z_k;\theta\right)\Bigg]
\end{aligned}\tag{4.8}$$

Fig. 4.7 visually explains the process of evaluating (4.8), which is very similar to the original TD-VAE. The belief network aggregates observations such that each belief b_t is assumed to be a sufficient statistics for $x_{1:t}$. The smoothing network, knowing what the final latent state z_T is, given observations $x_{1:T}$, infers what should have been latent states $z_{t:T-1}$. This gives us two different distributions for the inference of z_t: one given only observations $x_{1:t}$, and the other given all observations $x_{1:T}$. In the learning phase, we measure the divergence between these two distributions as a loss to prompt correct dynamics recognition and consistency in the belief network. Then, the Markovian transition model infers the next state from the current one. We infer the Gaussian parameters of the next state for each latent state inferred by the smoothing network and measure as loss the divergence between the distribution inferred by the smoothing network and the one inferred by the transition model. Finally, for each latent state z_k sampled from the smoothing network, we infer the Gaussian parameters describing the observation x_k with the decoding network and compute the negative log-likelihood of x_k given these parameters as loss.

However, our preliminary experiments on this model with a dataset acquired in CARLA [6] revealed very poor prediction quality when z_t is sampled from $q_{Z_t|B_t}(\cdot|b_t;\phi)$, while providing very good predictions when z_t is sampled from $q_{Z_t|B}(\cdot|b_{t:T};\phi)$, i.e. from the smoothing network. In fact, this seems obvious considering that the prediction part of this model is trained with the latent states sampled from the variational distribution $q_{Z_{t:T}|B_{t:T}}\left(\cdot|b_{t:T};\phi\right)$ and not $q_{Z_{t:T}|B_t}\left(\cdot|b_t;\phi\right)$. This is what motivates the introduction in the next section of a local predictability constraint, allowing us to train our model on samples from $q_{Z_{t:T}|B_t}\left(\cdot|b_t;\phi\right)$. This will also allow us to keep the idea of predicting distant latent states from current observations while avoiding the strong assumption that $p_{Z|B}\left(\cdot|b_t;\theta\right) = q_{Z|B}\left(\cdot|b_t;\phi\right)$.

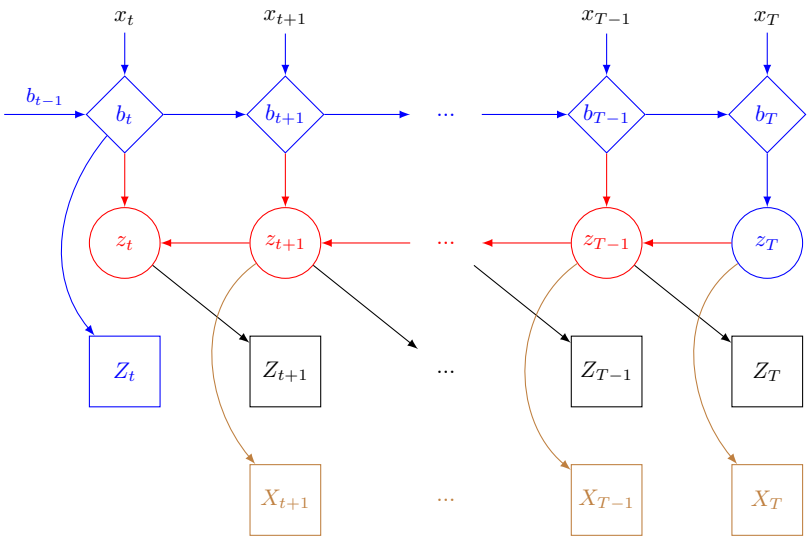


Figure 4.7: Illustration of the forward computations allowing for the evaluation of the STD-VAE loss (4.8). A diamond indicates a deterministically inferred variable. A rectangle indicates the deterministic inference of distribution parameters. A circle indicates the deterministic inference of distribution parameters and a sample from this distribution. The blue network is the belief network. The red network is the smoothing network. The black network is the Markovian transition model. The brown network is the decoding network.

4.5.4 Our Locally Predictable VAE (LP-VAE) model

First, we put a local predictability constraint for the model to be able to predict multiple time steps into the future:

$$p_{Z|X_{1:t}}(\cdot|\ x_{1:t};\theta) \approx p_{Z|X,Y}(\cdot|\ x,y;\theta) \tag{4.9}$$

for any instant $t \geq t_{\min}$. This means that there must be some instant $t_{\min}$ such that the partial observations $x_{1:t_{\min}}$ are sufficient to recognize the latent dynamics of the whole sequence, i.e. such that all observations $y_{1:T}$ and all subsequent partial observations $x_{t_{\min}+1:T}$ bring negligible additional information in the recognition of these latent dynamics. Notice that

$$p_{Z|X,Y}(\cdot|\ x,y;\theta) = \frac{p_{X,Y,Z}(x,y,\cdot\ ;\theta)}{p_{X,Y}(x,y;\theta)} = \frac{P(\theta)}{p_{X,Y}(x,y;\theta)},$$

and let us note $P_t(\theta) = p_{Z|X_{1:t}}(\cdot|\ x_{1:t};\theta)$. To enforce Eq. (4.9), we want to minimize the average KL divergence

$$\begin{aligned}&\mathop{\mathbb{E}}_{t\sim\ \mathcal{U}_{[t_{\min},\ T-1]}}\left[D_{KL}\left(P_t(\theta)\ \middle\|\ \frac{P(\theta)}{p_{X,Y}(x,y;\theta)}\right)\right]\\ &= \log p_{X,Y}(x,y;\theta) + \mathop{\mathbb{E}}_{t\sim\ \mathcal{U}_{[t_{\min},\ T-1]}}\left[D_{KL}\left(P_t(\theta)\ ||\ P(\theta)\right)\right],\end{aligned}$$

which we cannot minimize directly, due to the intractability of $p_{X,Y}(x, y; \theta)$ and $p_{Z|X_{1:t}}(\cdot|\ x_{1:t}; \theta)$. However, we have:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{t\sim\ \mathcal{U}_{[t_{\min},\ T-1]}}[D_{KL}\left(P_t(\theta)\ ||\ P(\theta)\right)] \\
&= -\log p_{X,Y}(x, y; \theta) \\
&\quad + \mathop{\mathbb{E}}_{t\sim\ \mathcal{U}_{[t_{\min},\ T-1]}}\left[D_{KL}\left(P_t(\theta)\ \middle|\middle|\ \frac{P(\theta)}{p_{X,Y}(x, y; \theta)}\right)\right] \\
&\geq -\log p_{X,Y}(x, y; \theta),
\end{aligned} \tag{4.10}$$

since the KL divergence is always nonnegative for two probability distributions. So, by optimizing θ to minimize $\mathop{\mathbb{E}}_{t\sim\ \mathcal{U}_{[t_{\min},\ T-1]}}[D_{KL}\left(P_t(\theta)\ ||\ P(\theta)\right)]$, we maximize a lower bound of $p_{X,Y}(x, y; \theta)$, which is our primary goal. Thus, we can simply introduce a variational distribution to approximate $p_{Z|X_{1:t}}(\cdot|\ x_{1:t}; \theta)$ as long as we simultaneously minimize the aforementioned KL divergence. Such a variational distribution corresponds to a recognition model that tries to predict the next latent states in addition to recognizing the current and past ones, which is more useful than one that would directly approximate $p_{Z|X,Y}(\cdot|\ x, y; \theta)$.

Notice that:

$$\begin{aligned}
&p_{Z|X_{1:t}}(z|\ x_{1:t}; \theta) \\
&= p_{Z|X}(z_t|\ x_{1:t}; \theta)\ .\ p_{Z|Z,X}(z_{1:t-1}|z_t, x_{1:t}; \theta)\ .\ p_{Z|Z,X}(z_{t+1:T}|z_{1:t}, x_{1:t}; \theta) \\
&= p_{Z|X}(z_t|\ x_{1:t}; \theta)\ .\ \prod_{k=1}^{t-1} p_{Z|Z,X}(z_k|z_{k+1}, x_{1:k}; \theta)\ .\ \prod_{k=t+1}^{T} p_{Z|Z}(z_k|z_{k-1}; \theta),
\end{aligned} \tag{4.11}$$

omitting variable indices in distribution indices for the sake of clarity. Based on this decomposition, let us introduce two variational distributions $Q_t^1(\phi) = q_{Z_t|X_{1:t}}(\cdot|x_{1:t}; \phi)$ and $Q_t^2(\phi) = q_{Z_t|X_{1:t},Z_{t+1}}(\cdot|x_{1:t}, z_{t+1}; \phi)$ taking their parameters in the parameter set ϕ such that:

$$\begin{aligned}
q_{Z_t|X_{1:t}}(\cdot|x_{1:t}; \phi) &\approx p_{Z_t|X_{1:t}}(\cdot|x_{1:t}; \theta) \\
q_{Z_t|X_{1:t},Z_{t+1}}(\cdot|x_{1:t}, z_{t+1}; \phi) &\approx p_{Z_t|X_{1:t},Z_{t+1}}(\cdot|x_{1:t}, z_{t+1}; \theta).
\end{aligned}$$

We assume that both $p_{Z_t|X_{1:t}}(\cdot|x_{1:t}; \theta)$ and $p_{Z_t|X_{1:t},Z_{t+1}}(\cdot|x_{1:t}, z_{t+1}; \theta)$ have an approximate Gaussian form with an approximately diagonal covariance matrix, i.e.

$$\begin{aligned}
Q_t^1(\phi) &= \mathcal{N}(\mu_b(x_{1:t}; \phi),\ \sigma_b(x_{1:t}; \phi).I_d) \\
Q_t^2(\phi) &= \mathcal{N}(\mu_s(x_{1:t}, z_{t+1}; \phi),\ \sigma_s(x_{1:t}, z_{t+1}; \phi).I_d),
\end{aligned}$$

where μ_b, σ_b, μ_s and σ_s are deep neural networks taking their parameters in the parameter set ϕ. Taking back Eq. (4.11), we get:

$$\begin{aligned}
&p_{Z|X_{1:t}}(z|\ x_{1:t}; \theta) \\
&\approx q_{Z|X}(z_t|\ x_{1:t}; \phi)\ .\ \prod_{k=1}^{t-1} q_{Z|Z,X}(z_k|z_{k+1}, x_{1:k}; \phi)\ .\ \prod_{k=t+1}^{T} p_{Z|Z}(z_k|z_{k-1}; \theta) \\
&= q_{Z|X}(z_{1:t}|x_{1:t}; \phi)\ .\ p_{Z|Z}(z_{t+1:T}|\ z_t; \theta) \\
&= q_{Z|X_{1:t}}(z|\ x_{1:t}; \theta, \phi) = Q_t(\theta, \phi),
\end{aligned}$$

which means that posing our two variational distributions $Q_t^1(\phi)$ and $Q_t^2(\phi)$ is equivalent to posing the variational distribution $Q_t(\theta, \phi) \approx p_{Z|X_{1:t}}(\cdot|\ x_{1:t}; \theta)$.

Therefore, we want to optimize ϕ and θ to minimize

$$\mathop{\mathbb{E}}_{t\sim\, \mathcal{U}_{[t_{\min},\ T-1]}} \left[D_{KL}\left(Q_t(\theta,\phi) \,\middle\|\, \frac{P(\theta)}{p_{X,Y}(x,y;\theta)} \right) \right]$$

while optimizing ϕ to minimize

$$\mathop{\mathbb{E}}_{t\sim\, \mathcal{U}_{[t_{\min},\ T-1]}} \left[D_{KL}\left(Q_t(\theta,\phi) \parallel P_t(\theta) \right) \right].$$

Actually, to achieve both these objectives, we only need to minimize

$$\mathcal{L}_{\text{LP-VAE}}(x,y;\theta,\phi) = \mathop{\mathbb{E}}_{t\sim\, \mathcal{U}_{[t_{\min},\ T-1]}} \left[D_{KL}\left(Q_t(\theta,\phi) \parallel P(\theta) \right) \right] \tag{4.12}$$

w.r.t. both ϕ and θ. See Appendix D for more details. Developing the KL divergence of Eq. (4.12) to make our recurrent distributions appear, we finally obtain:

$$\begin{aligned}
&D_{KL}\left(Q_t(\theta,\phi) \parallel P(\theta) \right) \\
&= \mathop{\mathbb{E}}_{Z\sim Q_t(\theta,\phi)} \big[\log q_{Z_{1:t}|B_{1:t}}(Z_{1:t}|b_{1:t};\phi) \\
&\qquad\qquad + \log p_{Z_{t+1:T}|Z_t}(Z_{t+1:T}|\ Z_t;\theta) \big] \\
&\quad - \mathop{\mathbb{E}}_{Z\sim Q_t(\theta,\phi)} \big[\log p_{Z_{1:t}}(Z_{1:t}\ ;\theta) \\
&\qquad\qquad + \log p_{Z_{t+1:T}|Z_t}(Z_{t+1:T}|Z_t\ ;\theta) \\
&\qquad\qquad + \log p_{X,Y|Z}(x,y|Z\ ;\theta) \big] \\
&= D_{KL}\left(q_{Z_{1:t}|B_{1:t}}(\cdot|b_{1:t};\phi) \parallel p_{Z_{1:t}}(\cdot\ ;\theta) \right) \\
&\quad - \mathop{\mathbb{E}}_{Z\sim Q_t(\theta,\phi)} \big[\log p_{X,Y|Z}(x,y|Z;\theta) \big]
\end{aligned} \tag{4.13}$$

which leads to

$$\begin{aligned}
&D_{KL}\left(Q_t(\theta,\phi) \parallel P(\theta) \right) \\
&= \mathop{\mathbb{E}}_{Z\sim Q_t(\theta,\phi)} \Bigg[\log q_{Z_i|B_i}\left(Z_t|b_t;\phi\right) \\
&\quad + \sum_{k=1}^{t-1} \log q_{Z_i|B_i,Z_{i+1}}\left(Z_k|b_k,Z_{k+1};\phi\right) \\
&\quad - \log p_{Z_i}\left(Z_1;\theta\right) - \sum_{k=2}^{t} \log p_{Z_{i+1}|Z_i}\left(Z_k|Z_{k-1};\theta\right) \\
&\quad - \sum_{k=1}^{T} \log p_{X_i,Y_i|Z_i}\left(x_k,y_k|Z_k;\theta\right) \Bigg]
\end{aligned} \tag{4.14}$$

Fig. 4.8 illustrates the process of evaluating (4.14). We can easily give an interpretation to this loss: we can identify two global objectives in Eq. (4.13) that are reminiscent of the original VAE [28] in terms of interpretation: the D_{KL} term is an encoder loss for the recognition model of parameters ϕ, while the second term is a decoder loss for the generative model of parameters θ. It can be viewed as a precision loss (second term) optimized against a regularization (first term) to prevent from overfitting.

We can even go deeper in interpretation to highlight what differs from the original VAE. Contrary to the original VAE, our model generates a sequence of observations instead of an

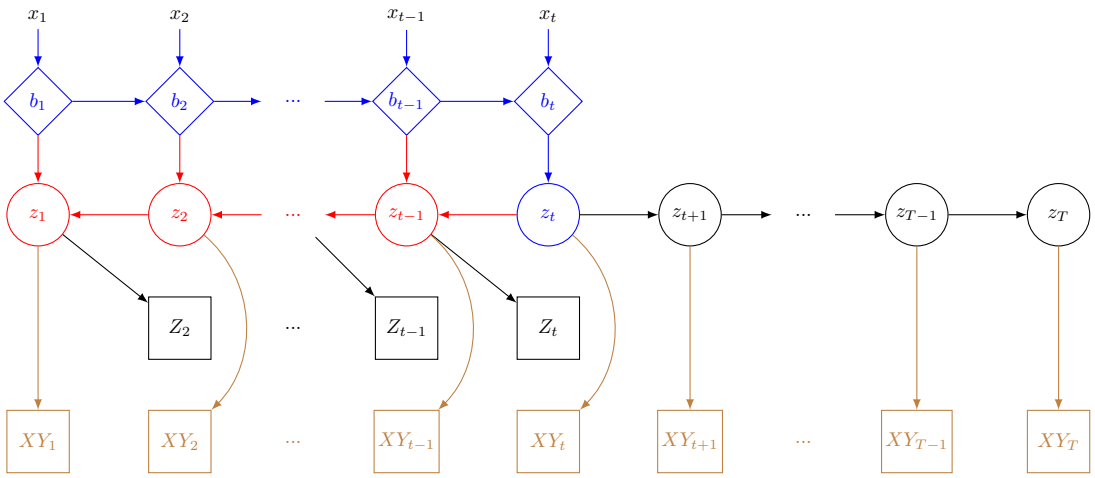


Figure 4.8: Illustration of the forward computations allowing for the evaluation of the LP-VAE loss. A diamond indicates a deterministically inferred variable. A rectangle indicates the deterministic inference of distribution parameters. A circle indicates the deterministic inference of distribution parameters and a sample from this distribution. The blue network is the belief network. The red network is the smoothing network. The black network is the Markovian transition model. The brown network is the decoding network.

isolated one. Doing so, we have a Markovian transition model that predicts a latent state from the previous one with its own set of parameters separated from the decoder ones. Therefore, it seems natural to have a third loss term for prediction. We can make it appear by splitting the second term of Eq. (4.13), i.e.:

$$
\begin{aligned}
&D_{KL}\left(Q_t(\theta,\phi) \mid\mid P(\theta)\right) \\
&= D_{KL}\left(q_{Z_{1:t}|B_{1:t}}(\cdot|b_{1:t};\phi) \mid\mid p_{Z_{1:t}}(\cdot\,;\theta)\right) \\
&\quad - \mathop{\mathbb{E}}_{Z\sim Q_t(\theta,\phi)}\left[\log p_{(X,Y)_{1:t}|Z_{1:t}}((x,y)_{1:t}|Z_{1:t};\theta)\right] \\
&\quad - \mathop{\mathbb{E}}_{Z\sim Q_t(\theta,\phi)}\left[\log p_{(X,Y)_{t+1:T}|Z_{t+1:T}}((x,y)_{t+1:T}|Z_{t+1:T};\theta)\right]
\end{aligned}
$$

The first term is an encoder loss. The second term is a decoder loss. The third term is a prediction loss. This prediction loss can also be viewed as a loss optimized against a regularization since the D_{KL} term affects the inference of Z_t by the recognition model from which the next latent states are predicted.

4.5.5 LP-VAE with actions

The models we described up to this point represents the environment evolving around the observing agent. However, our agent also acts on this environment and influences the observations gathered to train our model. Thus, we need to modify it in order to integrate this subtlety.

Let A_t be the action applied at time t on perceptions. This action describes a mask on the information contained in Y_t. This partial information is then transmitted to the observing agent, influencing X_t. It has no influence on the environment evolving around the agent, only on its perception of it. This means that Y_t and Z_t are not affected by A_t. Moreover, we will now consider that the random variable X_t is the ego-vehicle perception at time t, eventually augmented

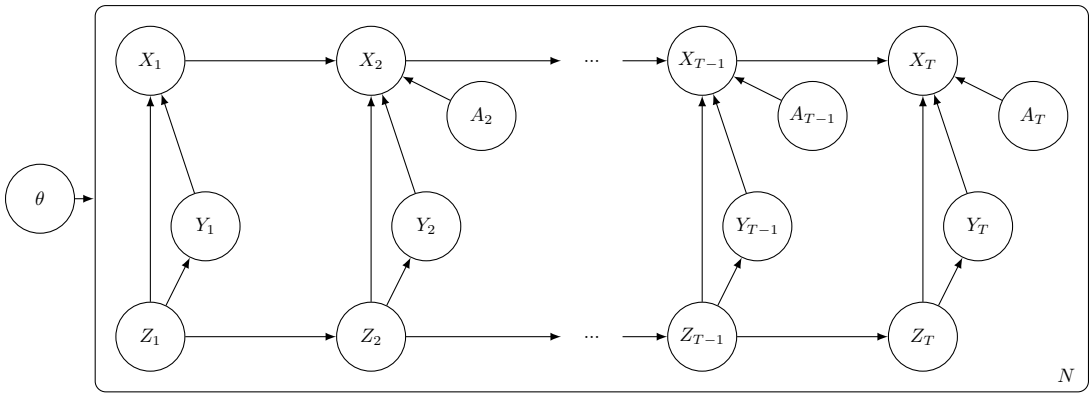


Figure 4.9: Bayesian network of our generative model of parameters in θ. We have N replications of this model, corresponding to the N sequences of length T in our dataset. The parameter set θ influences the inference of all variables in the model for the N sequences we have.

with information from Y_t, in accordance with A_t, and combined with the discounted memory of the previous partial observations $X_{1:t-1}$. Fig. 4.9 provides the corresponding Bayesian network.

We set the following constraints:

- $Z_i \sim \mathcal{N}(0, I_d)$
- $p_{Z_{i+1}|Z_i}(\cdot|z_t;\theta) = \mathcal{N}(\mu_z(z_t;\theta),\ \sigma_z^2(z_t;\theta).I_d)$
- $p_{Y_i|Z_i}(\cdot|z_t;\theta) = \mathcal{N}(\mu_y(z_t;\theta),\ \alpha_y.I_{|X_t|})$
- $p_{X_i|X_{i-1},Y_t,Z_t,A_t}(\cdot|x_{t-1}, y_t, z_t, a_t;\theta) = \mathcal{N}(\mu_x(x_{t-1}, y_t, z_t, a_t;\theta),\ \alpha_x.I_{|X_t|})$

where all parameters $\mu_\cdot$ and $\sigma_\cdot$ are deep neural networks taking their parameters in θ, and $\alpha_\cdot \in \left[\frac{1}{2\pi}, +\infty\right)$.

Our dataset D is composed of N independent sequences of partial and complete observations with a randomly chosen bounding box A_t, i.e. $D = (x_{1:T},\ y_{1:T}, a_{2:T})_{1:N}$. Fortunately, Eq. (4.11) still holds in this new model. Moreover, we know that the environment does not depend on the actions $A_{2:T}$ taken on its perception of it and that the actions only mask regions of Y_t while not altering the remaining. Finally, since X_t contains the information transmitted from Y_t in accordance with A_t, the actions $A_{2:t}$ do not bring any information for the inference of the latent states $Z_{1:t}$. Given the Bayesian network in Fig. 4.9, the actions A without knowing $X_{t+1:T}$ do not bring any information for the inference of the latent states $Z_{t+1:T}$ either. We have:

$$p_{Z|X_{1:t},A}(\cdot|\ x_{1:t}, a;\theta) = p_{Z|X_{1:t}}(\cdot|\ x_{1:t};\theta)$$

Thus, we keep the LP-VAE variational distributions

$$\begin{aligned} Q_t^1(\phi) &= q_{Z_t|X_{1:t}}(\cdot|x_{1:t};\phi), \\ Q_t^2(\phi) &= q_{Z_t|X_{1:t},Z_{t+1}}(\cdot|x_{1:t}, z_{t+1};\phi), \\ Q_t(\theta,\phi) &\approx p_{Z|X_{1:t}}(\cdot|\ x_{1:t};\theta). \end{aligned}$$

Then, for our local predictability constraint (See Eq. (4.9)), we consider $p_{Z|X,Y,A}(\cdot|\ x, y, a;\theta)$

instead of $p_{Z|X,Y}(\cdot|\ x, y; \theta)$. Notice that

$$\begin{aligned} p_{Z|X,Y,A}(\cdot|\ x, y, a; \theta) &= \frac{p_{X,Y,Z|A}(x, y, \cdot\ |a; \theta)}{p_{X,Y|A}(x, y\ |a; \theta)} \\ &= \frac{P(\theta)}{p_{X,Y|A}(x, y\ |a; \theta)} \end{aligned}$$

We take as loss function $\mathcal{L}_{\text{LP-VAE}}(x, y\ |a; \theta, \phi)$ instead of $\mathcal{L}_{\text{LP-VAE}}(x, y; \theta, \phi)$, where

$$\mathcal{L}_{\text{LP-VAE}}(x, y\ |a; \theta, \phi) = \underset{t\sim\ \mathcal{U}_{[t_{\min},\ T-1]}}{\mathbb{E}} [D_{KL}\left(Q_t(\theta, \phi)\ ||\ P(\theta)\right)] \tag{4.15}$$

This loss maximizes a lower bound of

$$p_{X,Y|A}(x, y\ |a; \theta).$$

Developing the KL divergence of Eq. (4.15) in accordance with our new model, we get:

$$\begin{aligned} &D_{KL}\left(Q_t(\theta, \phi)\ ||\ P(\theta)\right) \\ &= \underset{Z\sim Q_t(\theta,\phi)}{\mathbb{E}} \big[\log q_{Z_{1:t}|B_{1:t}}\left(Z_{1:t}|b_{1:t}; \phi\right) \\ &\quad + \log p_{Z_{t+1:T}|Z_t}\left(Z_{t+1:T}|z_t; \theta\right) - \log p_{Z_{1:t}}\left(Z_{1:t}; \theta\right) \\ &\quad - \log p_{Z_{t+1:T}|Z_t}\left(Z_{t+1:T}|z_t; \theta\right) - \log p_{Y|Z}\left(y\ |Z; \theta\right) \\ &\quad - \log p_{X|Y,Z,A}\left(x\ |y, Z, a; \theta\right)\big] \\ &= \underset{Z\sim Q_t(\theta,\phi)}{\mathbb{E}} \big[\log q_{Z_{1:t}|B_{1:t}}\left(Z_{1:t}|b_{1:t}; \phi\right) \\ &\quad - \log p_{Z_{1:t}}\left(Z_{1:t}; \theta\right) - \log p_{Y|Z}\left(y\ |Z; \theta\right) \\ &\quad - \log p_{X|Y,Z,A}\left(x\ |y, Z, a; \theta\right)\big] \end{aligned}$$

which leads to

$$\begin{aligned} &D_{KL}\left(Q_t(\theta, \phi)\ ||\ P(\theta)\right) \\ &= \underset{Z\sim Q_t(\theta,\phi)}{\mathbb{E}} \Bigg[\log q_{Z_i|B_i}\left(Z_t|b_t; \phi\right) - \log p_{Z_i}\left(Z_1; \theta\right) \\ &\quad + \sum_{k=1}^{t-1} \log q_{Z_i|B_i,Z_{i+1}}\left(Z_k|b_k, Z_{k+1}; \phi\right) \\ &\quad - \sum_{k=2}^{t} \log p_{Z_{i+1}|Z_i}\left(Z_k|Z_{k-1}; \theta\right) - \sum_{k=1}^{T} \log p_{Y_i|Z_i}\left(y_k|Z_k; \theta\right) \\ &\quad - \sum_{k=2}^{T} \log p_{X_i|X_{i-1},Y_i,Z_i,A_i}\left(x_k|x_{k-1}, y_k, Z_k, a_k; \theta\right) \\ &\quad - \log p_{X_1|Y_1,Z_1}\left(x_1|y_1, Z_1; \theta\right) \Bigg] \end{aligned} \tag{4.16}$$

In practice however, we will neglect the term $-\log p_{X_1|Y_1,Z_1}\left(x_1|y_1, Z_1; \theta\right)$ for several reasons. First, it avoids to optimize parameters that would only be used in the learning phase, while not corresponding to an important component (the complete observation y_1 being already considered and containing x_1). But maybe more importantly, since X_t keeps a memory of past observations

in this formulation of the LP-VAE, x_1 may also contain information on actions preceding $a_{2:T}$ that should be given as well if x_1 is actually not the start of an episode of interactions in the environment. Not generating x_1 allows us to start the inference of latent states at any point of the episode, independently from the previous actions and observations that produced x_1. This means that we can re-use different subsequences of the same training sequence in the learning phase, without having to make sure that x_1 do not contain information related to past observations and actions.

4.6 Implementation as neural networks

4.6.1 Belief state computation

The grids G_t introduced in section 4.3.1 are not directly taken as input of our LP-VAE. Beforehand, we train a Convolutional VAE (CVAE) to learn a compressed, essentialized representation of these observations in which spatial features have been extracted. This CVAE is itself separated into 4 independent parts in order to preserve the semantics of these features: a CVAE for the pedestrian channel, another for the car channel, another for static elements (road lines, road, other) and a last one for the ignorance. The projection of G_t into the latent space of this Convolutional VAE is the X_t taken by our LP-VAE. Then, we feed X_t, X_{t-1} and the ego-motion V_t to a Multilayer perceptron (MLP) in order to extract features about the motion of road users around the ego-vehicle. The output of this MLP serves as input to a Recurrent Neural Network (RNN) composed of Long Short-Term Memory (LSTM) cells to form and update a belief over the dynamics of other road users. The concatenation of the hidden state of this RNN with X_t and the driving controls C_t represents the belief state B_t at time t. Fig. 4.10 visually sums up this procedure.

4.6.2 Inference of Gaussian parameters

In [74], they proposed to use what they called D maps[1] to infer the Gaussian parameters of any of the distributions over the latent state z_t. It is a part of a LSTM cell (new features multiplied by the input gate), as indicated in Fig. 4.11, where the output is passed to two fully connected (FC) layers in parallel without activation function, one to determine μ_{z_t} and the other to determine $\log(\sigma_{z_t})$. Yet, in our sequential setting, this D map becomes a truly recurrent unit, chaining itself multiple times from t_1 to 1 in the smoothing network and from t_1 to t_2 in the prediction network. As for any recurrent network, this poses the issue of vanishing gradients. Furthermore, it lacks the semantics of a transition model: some components could disappear from the frame (forget gate) and some other could become visible or simply move from their initial state (input gate, followed by an addition to the initial components). These are exactly the transformations applied to the cell state of a LSTM cell. Thus, using the cell state of a LSTM cell as latent state mean μ_{z_t} as in Fig. 4.11, where $h = z_{t+1}$ and input $= b_t$, solves both the vanishing gradient issue and the lack of model semantics. Giving h as both hidden and cell states also has the effect of implementing peephole connections [83], giving the cell state some control over the input, forget and output gates (the three sigmoïd layers), which better captures sporadic events. In addition, uncertainty should be encoded within the latent state to be self-sufficient for a transition model. This encourages the computation of the standard deviation σ_{z_t} from μ_{z_t} with some filtering gate (output gate), which is exactly what a LSTM cell does to output a quantity based on its cell state. Similarly, we use this LSTM cell in the

[1] In [74], they used a 16-layer model where the information transits from layer to layer through the states of a LSTM, possibly in place of this D map, in their DeepMind Lab experiment. Note however that it is recurrent through layers, not time. This is different from what is proposed here.

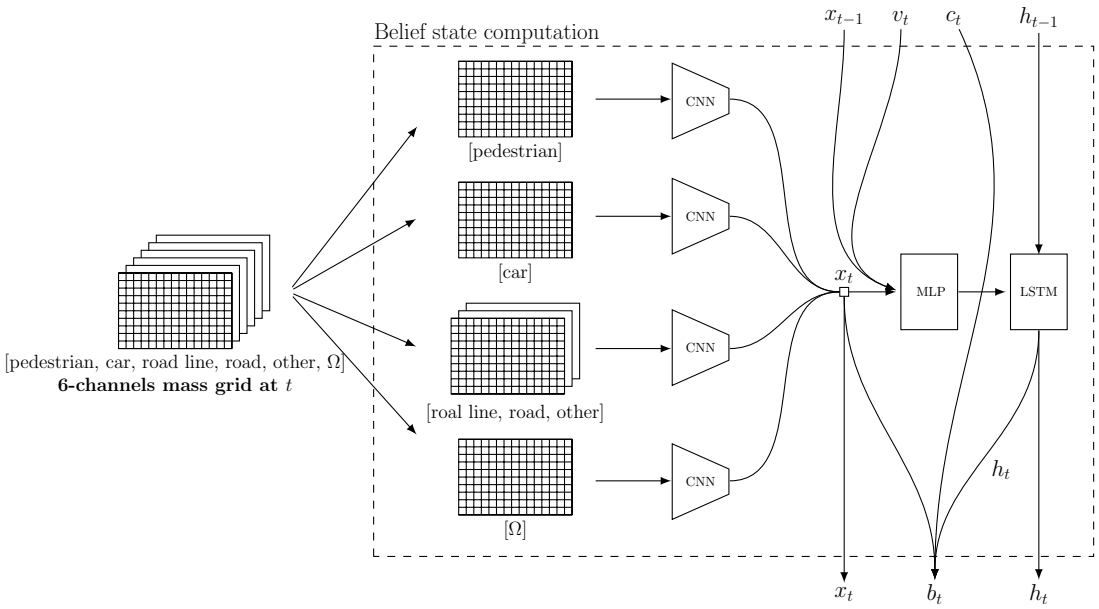


Figure 4.10: Illustration of the process of computing the observation X_t and the belief state B_t from G_t, X_{t-1}, V_t and C_t. Four independent Convolutional VAEs are trained to learn a sufficient representation of pedestrian, car, {*road lines, road, other*} and ignorance. These encodings form X_t. A Multilayer perceptron (MLP) tries to learn features about the motion of road users around the ego-vehicle. The output of this MLP serves as input to a Recurrent Neural Network (RNN) composed of Long Short-Term Memory (LSTM) cells to form and update a belief over the dynamics of other road users. The concatenation of the hidden state of this RNN with X_t and the driving controls C_t represents the belief state B_t at time t.

prediction network for $p_{Z_{i+1}|Z_i}(\cdot|z_t;\theta)$, where $h = z_t$ and input $= \emptyset$. For the belief network, we keep this D map as there is no propagation in time.

4.6.3 Decoding

So far, we determined the networks outputting distribution parameters describing the latent states Z used in the evaluation of $\mathcal{L}_{\text{LP-VAE}}$, both for the generative model and the recognition model. It remains to propose the decoding network that is part of the generative model and produces X and Y. Given the conditional distributions appearing in $\mathcal{L}_{\text{LP-VAE}}$, we need a decoder inferring Y_t from Z_t and another one inferring X_t from X_{t-1}, Y_t, A_t and Z_t.

However, since X_t and Y_t are not given in the original space but in a learned compressed one, extracting features from Y_t according to the bounding box A_t is not directly possible. One has to decode Y_t, extract features according to A_t, decode X_t and then fuse it with the leaked features from Y_t. For the sake of efficiency, we will learn to directly extract these features that we denote by the random variable M_t in the learned compressed space and to fuse them with X_t. Thus, in parallel to $\mathcal{L}_{\text{LP-VAE}}$, we minimize an extra loss term

$$-\sum_{k=2}^{T} \log p_{M_i|A_i,Y_i}\left(m_k|a_k,y_k;\theta\right),$$

where m_t corresponds to y_t masked in accordance with a_t and compressed by the same CVAE as for y_t. Note that our dataset becomes $D = (x_{1:T},\ y_{1:T}, m_{2:T}, a_{2:T})_{1:N}$.

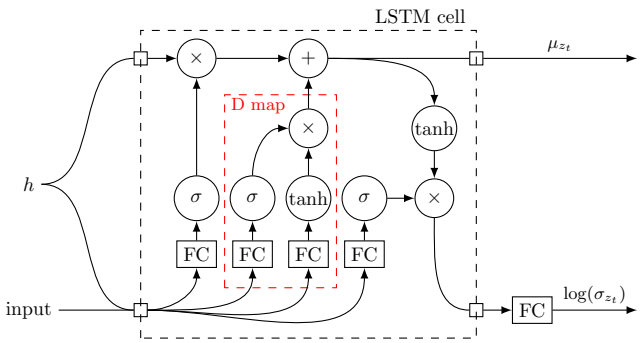


Figure 4.11: Proposed replacement for D maps. The *FC* rectangles indicate a single Fully Connected layer. Circles indicate point-wise operations, where σ is the sigmoïd activation function.

We choose to infer Y_t from Z_t through a D map as introduced in section 4.6.2. All other inferences are done through an updating module that is inspired by the updating of a LSTM cell state. The masking of Y_t is orchestrated by A_t, producing M_t by filtering. Finally, X_{t-1} is updated in two steps. The first update is assumed to change its reference frame and to determine which parts of Y_t are visible to the ego-vehicle. This implicitly produces the X_t corresponding to the null action, i.e. the action that consists in doing nothing. We consider this transformation deterministic, given y_t and z_t. The second update transmits the excerpt M_t from Y_t to this prior perception, producing the actual X_t influenced by A_t. Fig. 4.12 depicts these networks. In addition, Fig. 4.13 illustrates the decoding of X_t by the decoder of the Convolutional VAE.

4.7 Experiments

4.7.1 Data acquisition & RL Environment

To conduct our experiments, we chose to work with the open-source driving simulator CARLA [6]. Our semantic grids G_t are computed online from a frontal 320×480 depth camera with FOV of 135° and its corresponding pixel-wise semantic classification. These simulated sensors are attached to a simulated vehicle autonomously wandering in a city with other vehicles, bikes and pedestrians (see Fig. 4.2). More precisely, G_t is obtained by counting the number of occurrences of each class in each possible configuration of 4×4 consecutive pixels. All classes corresponding to static objects are merged into the class *other*. Then, in each cell of the resulting $80 \times 120 \times 5$ grid, these numbers are divided by 16 and we add a channel representing ignorance (i.e. Ω) to store the quantity needed to make the sum on all channels equal to 1. We also discount the resulting mass functions by a factor of 0.01 to simulate noise, i.e. all masses are multiplied by 0.99 and 0.01 is added to the mass on ignorance. Finally, thanks to the depth and information about the camera, we create a 3D point cloud of this frontal perception. Thus, to get the 2D grid G_t, we ignore points higher than 2.5 meters and we take the highest of the remaining ones (if more than one point at the same ground coordinates). For this reason, it sometimes happens that the ground under a vehicle is perceived, but not its top, leading to *road* cells surrounded by *car* cells, as can be observed in Fig. 4.2 Left. An important road elevation may also conflict with the threshold of 2.5 meters. This view can be obtained by a LIDAR and a 3D semantic classifier [84] as well.

Our top-down semantic grids corresponding to complete observations y_t in our model are obtained with a facing ground camera above the ego-vehicle. Doing so, it contains itself some occlusions due to trees, poles, buildings, etc. Thus, it is rather a hint about the true y_t. This

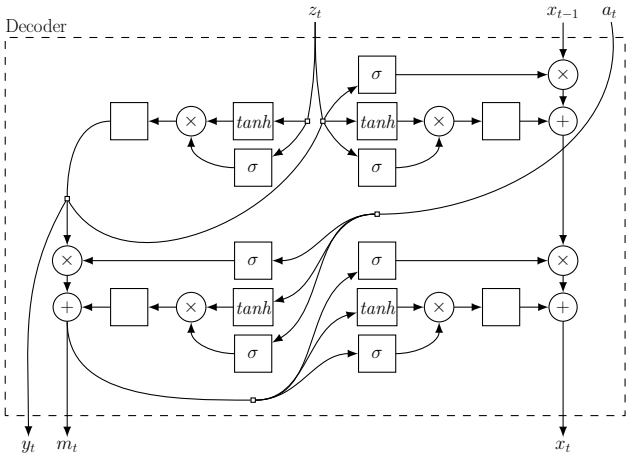


Figure 4.12: Illustration of our decoding architecture. The decoder block infers x_t the partial observation, y_t the spatially complete observation and m_t the masked y_t (as dictated by the bounding box a_t). It takes as inputs a latent state z_t, a previous partial observation x_{t-1} and a bounding box a_t. A rectangle indicates a fully connected layer, while the symbol at its center indicates the activation function applied to its output (σ for sigmoid, *tanh* for hyperbolic tangent and nothing for the identity function). Each updating network is composed of a forget gate (first σ) and a D map, i.e. input features (*tanh*), an input gate (last σ) and a fully connected layer.

grid can also be obtained by the fusion of multiple view points, from a fleet of autonomous vehicles or infrastructure sensors, which can be acquired in the real world. A drone may be able to acquire this information as well. In any case, this ground truth grid is in fact itself uncertain and so is computed as G_t with an ignorance channel.

We created a dataset composed of 1560 sequences of 50 timesteps (5 seconds) each, where each perception is $80 \times 120 \times 6$. There are 30 runs in each of four cities available in CARLA, including small towns, big towns and fast lanes. Each run is 35 seconds long and a sequence is recorded every 2.5 seconds, leading to 13 sequences per run, hence the size of our dataset. This dataset provides the grids corresponding to X_t and Y_t in the action-independent model of section 4.5.1.

To provide the grids corresponding to X_t as defined in the full model of section 4.5.5, we created a second dataset from the first one by choosing random regions of Y_t to be given to X_t. We also added a visual memory that keeps a buffer of grid cells, transforms their coordinates according to the given motion of the ego-vehicle, discounts their mass functions to account for information aging and fuses them with the current perception grid, resulting in this X_t. In fact, the first dataset combined with our visual memory and our fusion procedure of Algorithm 1 for $\widetilde{G}_t$ and G_t^M constitutes the environment in which our agent will learn a communication policy.

4.7.2 Models

During training, we give between 8 and 10 timesteps of observations (i.e. between 0.8 and 1 second) and it is asked to predict between 5 and 10 timesteps ahead, i.e. between 0.5 and 1 second. We use the Mean Squared Error (MSE) loss function to compute the Gaussian negative log likelihoods of observing the grids corresponding to x_t and m_t given latent states. Indeed, this is analog to taking $\alpha = \frac{1}{2}$ and ignoring the constant term $\log\left(\sqrt{2\pi\alpha}\right)$. For the negative log likelihoods on the grid corresponding to y_t, we binarize it by taking the class with maximum mass and use a cross-entropy loss. To account for the fact that the instances of Y_t in our dataset

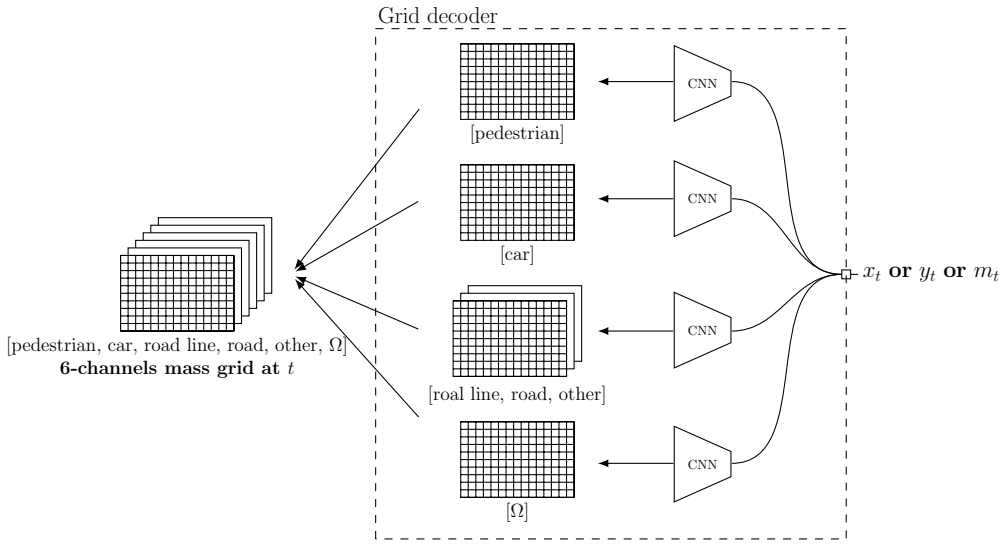


Figure 4.13: Illustration of the decoding of X_t or Y_t or M_t by the decoder of the CVAE that gave X_t to get back into the observation space. The CNN blocks are Transposed CNNs.

are not perfect, we simply do a pointwise multiplication between this loss and the complement to 1 of its ignorance channel (last channel). That way, if y_t does not have any information about a cell, no loss on y_t is actually back-propagated. Furthermore, we weight this cross-entropy loss differently from one channel to another to account for class imbalance. We used the weight vector $[100, 10, 1, 0.2, 0.1, 1]$. Indeed, on average, there are far less cells containing pedestrians than cells containing the road or any other static class. Doing so, without weights, the network would consider pedestrian as noise and neglect them.

In the following, we compare STD-VAE and LP-VAE for complete grid inference and prediction.

Grid completion

In this experiment, we use the decoder network described in Fig. 4.12 on the current latent state Z_t inferred from B_t to retrieve Y_t. Then, we use the network described in Fig. 4.13 to transform Y_t into the complete mass grid G^Y. To compare STD-VAE and LP-VAE, we employed two metrics: binary classification accuracy per class and a *mass score*. Our *Mass score* metric is computed as the mean of $G_t^Y \,.\, \widehat{G^Y}_t$ over all cells in the grid, where G_t^Y is the true binary complete grid classification and $\widehat{G^Y}_t$ is a mass grid inferred by some model. Since G_t^Y is binary, it acts as an indicator function for the correct class and the mass score represents the mean mass given to the right class by the model generating $\widehat{G^Y}_t$. Results are showed in Table 4.1.

Prediction

In this experiment, we compare prediction accuracy between LP-VAE and STD-VAE. For this, we study mass variations on the super-class $\{road, road\ line\}$, i.e. the sum of the *road* and *road line* grid channels. Indeed, this super-class represents the road layout. Its absence in a cell indicates either road users or the *other* class. Thus, its mass variations accounts for the dynamics of the whole scene, independently of classification accuracy.

In practice, for each model, we infer a prediction sequence of 10 complete grids $\hat{y}_{1:10}$ (i.e. 1

	Binary classification per class						Mass
	P	C	RL	R	O	Ω	score
LP-VAE	20.5%	68.5%	28.7%	84.3%	77.8%	**49.5%**	68.3%
STD-VAE	**33.7%**	**72.7%**	**30.7%**	**85.9%**	**80.6%**	46.2%	**68.8%**

Table 4.1: Mass score and binary classification accuracy per class. P indicates the pedestrian channel, C the car channel, RL the road lines channel, R the road channel, O the other channel and Ω the complete out-of-sight channel. It is clear that STD-VAE outperforms LP-VAE for simple grid completion, though the total mass score is not so different.

True y'	No blur		Gaussian blur 5x5		Gaussian blur 11x11	
	+	-	+	-	+	-
LP-VAE $\hat{y}'$	**6.81%**	**6.94%**	**14.41%**	**14.61%**	**23.81%**	**24.36%**
STD-VAE $\hat{y}'$	2.10%	2.37%	4.89%	5.41%	8.66%	9.52%

Table 4.2: Prediction accurracies between STD-VAE and LP-VAE. As expected, LP-VAE significantly outperforms STD-VAE on predictions.

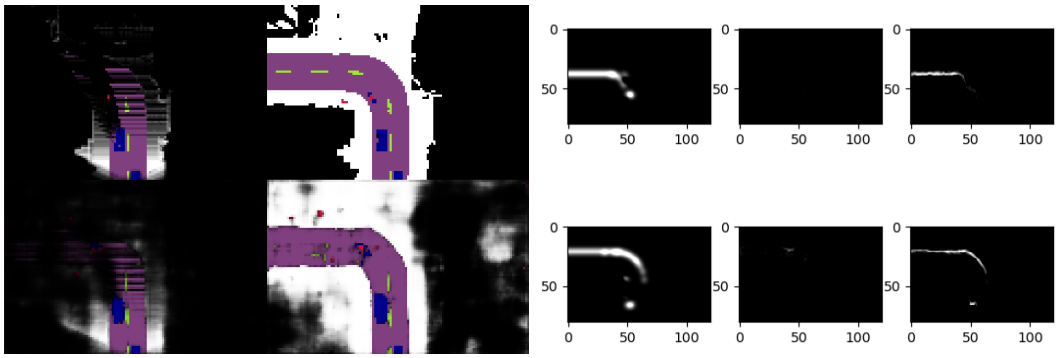
second in the future), based on 10 observations (i.e. the past second). From it, we compute the corresponding sequence of 9 grid variations $\hat{y}'_t = \hat{y}_{t+1} - \hat{y}_t$. We execute the same process with the true complete grids, which produces grids $y'_{1:9}$ of values ranging in $\{-1, 0, 1\}$. We test separately the accuracy on positive and negative changes. For the former, we do a pointwise multiplication between the true complete positive grids $\max(0, y'_{1:9})$ and the inferred positive ones $\max(0, \hat{y}'_{1:9})$. For the latter, we do a pointwise multiplication between the true complete negative grids $\max(0, -y'_{1:9})$ and the inferred negative ones $\max(0, -\hat{y}'_{1:9})$. We then sum all cells of each grid in the sequence, over 4992 sequences, i.e. 49 920 inferred grids and compare it to the separate sums of positive and negative true changes. Results are displayed in the first two columns of Table 4.2.

However, note that this binary mask can be quite hard to match, as both the exact location of these changes and their amplitude must be correct. To alleviate this constraint, we repeat this test with blurring filters applied to each grid of $y'_{1:9}$. The resulting grids, noted $\tilde{y}'_{1:9}$, are then renormalized so that $\sum \max(0, y'_{1:9}).\max(0, \tilde{y}'_{1:9}) = \sum \max(0, y'_{1:9})$ and $\sum \max(0, -y'_{1:9}).\max(0, -\tilde{y}'_{1:9}) = \sum \max(0, -y'_{1:9})$. This allows for slight misplacements of cells in predicted grids. We repeated this test twice with Gaussian filters, with kernels 5x5 and 11x11. These experiments correspond to the last 4 columns of Table 4.2. Our LP-VAE outperforms STD-VAE in every of these tests, no matter how hard the constraint on change location is. This means that the predicted changes of LP-VAE are not just better located, but also *better shaped* than the ones of STD-VAE, as expected by design. Fig. 4.14 illustrates this experiment.

4.7.3 Policy learning

Here, we finally compare different policies learned with PPO, with and without model to test the benefits of using belief states in our case. Each policy is the best found among iterations of training with 3000 transitions amounting to 500 000 time steps in total. We used a batch size of 60, with 10 epochs on each transition dataset, with a learning rate of 0.0003 and an entropy coefficient of 0.01. We also made the time horizon vary, i.e. we made the hyperparameter γ vary from 0 to 0.7, in order to see if a medium/long term strategy performs better.

The network learned with PPO has two parts: one for inferring the Value of a state, representing the mean of all potential future rewards, and one for inferring the best action from this



(a) (b)

Figure 4.14: (a) Left column: partial grid G_t corresponding to X_t. Right column: complete grid G_t^Y corresponding to Y_t. Top row: true classification grids. Bottom row: classification grids predicted by LP-VAE from X alone, 4 time steps in the future. (b) Prediction dynamics. Black represents the absence of variation, white some mass change in the cells of the *road* and *road line* channels of the grid in (a). Left column corresponds to the true variations, blurred by a 11x11 Gaussian filter. The central column corresponds to the prediction dynamics of STD-VAE, multiplied by the ones of the first column. Same for the right column but for LP-VAE. The first row represents positive changes, while the second row represents negative ones.

same state, representing the policy. Each of these networks is composed of two fully connected hidden layers of 128 and 64 neurons.

Different communication behaviors can be obtained by adjusting reward parameters. In particular, increasing K in Eq. 4.2 will make requests bigger, increasing w in Eq. 4.1 will make requests more focused on completely unknown areas, increasing η will make requests more focused on *pedestrians* and *cars*, less rewarding in general and so less frequent. We chose the following values: $\eta = 0.3$, $K = 36$ and $w = 2$. We also added a penalty of -15 for no cooperation at all (i.e. choice of a bounding box with no pixel in it, which means no transmission cost either) to force the agent to play the game. Moreover, approximating the top-down dimensions of cars and pedestrians, we took the following reward densities per squared meter: $r^m_{obj} = [540/(0.7{*}1.6), 540/(3{*}1.8), 20, 20, 0]$. Then, we converted them into rewards per squared cell by multiplying them by our grid resolution. More precisely, we set our cameras in CARLA so that the height corresponds to 40 meters. Thus, our reward densities per squared cell are $r_{obj} = (\frac{40}{80})^2.r^m_{obj}$. Our final rewards are obtained by normalizing r_{obj} to $[0, 1]$ by dividing it by its maximum. For the spatial filter, we used the parameters of Fig. 4.4, i.e. $\alpha = 0.5$, $\beta_F = 0.8$, $\beta_L = 1$ and $\zeta = 0.01$.

In order to evaluate and compare the performance of different policy learning schemes, we take as metrics the mean request size and the mean informational gain over all time steps of a test set with same size and characteristics as the training set described Section 4.7.1. We applied these metrics to 3 class groups: *pedestrians* (P), *cars* (C) and $\{$*road lines*, *road*$\}$ (R). In these conditions, we compared 3 schemes: PPO on top of the LP-VAE belief state B_t, PPO on top of the STD-VAE belief state B_t and PPO on top of X_t alone (i.e. only the features extracted from the current mass grid G_t by a Convolutional VAE). Each of them has been trained with $\gamma = 0$ (i.e. only immediate rewards matter), $\gamma = 0.35$ and $\gamma = 0.7$, to see if we could benefit from medium/long term strategies. We also compare these policies with a simple random policy that has a 50% chance of making a request and chooses uniformly random size and position of bounding box when it does. Table 4.3 presents our results, in percentage relatively to the

		Information gain			Request
		P	C	R	size
	Random	**26.2%**	22%	22.9%	13%
$\gamma = 0$	LP-VAE B_t	**22%**	27.6%	26.5%	6%
	STD-VAE B_t	19.9%	26.5%	24.4%	5%
	X_t alone	21.7%	**29.2%**	**27.6%**	6%
$\gamma = 0.35$	LP-VAE B_t	**20.6%**	**25.7%**	**24.8%**	6%
	STD-VAE B_t	18.2%	22.8%	23.3%	5%
	X_t alone	17.8%	23.4%	22.3%	5%
$\gamma = 0.7$	LP-VAE B_t	**15.7%**	**18.2%**	**19.5%**	5%
	STD-VAE B_t	13.6%	16.3%	17.2%	4%
	X_t alone	14.3%	17.8%	18.6%	4%

Table 4.3: Learned communication policy performances relatively to a broadcasting policy. The information gain is a mean percentage representing the mass actually gained after request, over the total mass that can be gained, at each time step.

maximal information gain and request size possible inherent to a broadcasting policy.

All of our learned policies only ask for about 5% of the space around the ego-vehicle, while receiving about 25% of the relevant information the agent lacks. Requiring about 2.5 times more information from the vehicular network for about the same relevant information gain or lower, the random policy is vastly less efficient. It only outperforms the others for pedestrians, which is consistent with the highly random behavior of pedestrians in CARLA. However, PPO + X_t alone and $\gamma = 0$ (i.e. greedy policy) is the policy that performs best overall. Surprisingly enough, taking into account future rewards actually harms performance in our case. A lower discounting factor in the memory module (i.e. observations that are kept longer in memory) would probably make policies perform best with $\gamma > 0$. Furthermore, note that LP-VAE always performs better than the other learned policies when $\gamma > 0$. This is consistent with the fact that LP-VAE has better prediction capabilities and thus provides useful information in its belief state for predicting future rewards.

4.8 Conclusions

In this chapter, we tried to elaborate an efficient peer-to-peer communication policy for collaborative perception. For this, we made agents learn what could be hidden in their blind spots through a generative sequence model that we proposed, named Locally Predictable VAE (LP-VAE). We compared its performance with another generative sequence model for RL applications called TD-VAE that we slightly adapted to our problem by making it both jumpy and sequential, referring to it as STD-VAE. We demonstrated that LP-VAE produces better predictions than STD-VAE, which translated into better performance for policies learned on top of its belief state. However, we discovered in the end that our best communication policy was a greedy one, i.e. one that does not need prediction capabilities. Combined with the fact that we augmented each observation with the discounted memories of past observations, it followed that only a state-less Convolutional VAE was needed for this greedy policy. Overall, our best learned policies only require about 5% of the space around the ego-vehicle, while gaining about 25% of the relevant information the agent lacks. Thus, we proved that learning to value the unknown is much more efficient than employing a broadcasting policy. It is also more efficient than blindly asking for random areas around the ego-vehicle since it requires about 13% of the total information, while gaining less than 25% of the relevant information the agent lacks. In

addition, we defined interpretable hyperparameters shaping the reward function corresponding to our problem. This makes it possible to obtain various communication policies, with different trade-offs between request size and information gain, as well as different class valuations, spatial priorities and valuation of ignorance (i.e. more or less emphasis on total ignorance). For future works, it would be interesting to compare LP-VAE and STD-VAE in RL tasks where future rewards are more important. Also, we would like to test our communication policies in a truly multi-agent context, where the agent would need to take into account the availability of nearby communicating vehicles, and with real sensor data.

Chapter 5

Conclusions and perspectives

In this thesis, we focused on learning the importance of a missing piece of information, depending on context, using predictive generative models and a policy determining the spatial request to make to the vehicular network based on predictions. Consequently, a vehicle can make a specific, *lightweight* request (or even none) to near vehicles based only on its situation. This allows for a sustainable decentralized collaborative perception system, in which communications are sparser, lighter and more relevant, freeing both communication bandwidth and computing time for the receiver. Furthermore, our new generative model Locally Predictable VAE (LP-VAE) can be exploited in other contexts. It is as theoretically as empirically well-justified and is able to learn sequential models from observations and to provide consistent predictions. Moreover, its learned belief states can be used by Reinforcement Learning algorithms, as Temporal Difference VAE (TD-VAE). And, because it provides better predictions, the belief states of LP-VAE contain more relevant information than the ones of TD-VAE. It would be interesting to evaluate this LP-VAE in various RL tasks in the future, comparing it to TD-VAE and to end-to-end RL baselines.

But, redundancy is not the only challenge posed by decentralization. Indeed, data incest is another one. In addition, directly taking into account the perception of a peer brings the question of trust, i.e. what is the level of self-confidence a peer has and are its observations precise? This means that special care must be taken with the content of these communications and the way two perceptions are fused together.

This led us to Dempster-Shafer Theory (DST) [24], which is a powerful generalization of Bayesian probability theory that explicitly models ignorance and uncertainty in information sources. It also offers easy ways to fuse information sources. In particular, when sources are not independent (data incest), one can use the *Cautious rule* [23]. This rule only applies to the conjunctive decomposition of evidence [41], i.e. to the conjunctive weight function. However, as we want to keep the complexity of the content of our messages reasonable and as we do not want to limit either the number of possible hypotheses, approximation methods [45] to compress perceptual information (i.e. limiting the number of focal sets of the considered mass function) need to be employed, which only apply to the mass function. The state-of-the-art family of algorithms computing the transformation between the mass function and the conjunctive weight function is the *Fast Möbius Transform* (FMT) [43] family. These algorithms are always exponential in the number of possible states of a perceptual component, which put a constrain on its expressiveness.

To avoid limiting the expressiveness of our perceptual components while also avoiding the

exponential burden of the FMT, we created a method to compute the transformation between the mass function and the conjunctive decomposition of DST only based on the actual information contained in sources. This method is exact and relies on a new mathematical notion that we call *focal point*. A focal point is an element of the smallest intersection-closed or union-closed family of sets from 2^Ω containing all focal sets of a mass function. The related paper [42] has been accepted to GRETSI 2019. Far from being ad hoc, our algorithms can be used to compute the conjunctive and disjunctive decompositions of evidence wherever they are useful. For example, these decompositions can be used to define the infinite families of t-norm-based and uninorm-based fusion rules [47], as well as in conflict analysis [48], clustering [49] and reinforcement/discounting [50] of believes. Our reformulation of the conjunctive and disjunctive weight functions in terms of focal points even acts as a generalization as it is still defined when the mass function is dogmatic or normal. Unfortunately, even if their likelihood may be lowered by approximation methods, there exists cases in which our algorithms may feature a worst complexity than the FMT.

Besides, the FMT is far more general and applies to all Möbius transform and its inversion in the Boolean lattice 2^Ω, where Ω is the set containing all possible states of a variable of interest. Doing so, it remained the de facto standard to compute DST transformations in the general case, proven to be optimal in 2^Ω.

This is why we pushed our approach a little further to propose the *Efficient* Möbius inversion theorem [52]. We mainly studied its properties in semilattices with a generalization of the notion of focal point and a generalization of our findings to the Möbius transform itself. Consequently, we were able to propose sequences of graphs in the same fashion as the FMT for the computation of Möbius transforms and their inversion in any distributive lattice (which is even more general than Boolean lattices), in a paper published in the proceedings of SUM 2019. We named these sequences of graphs the *Efficient Möbius Transformations* (EMT) [53]. These sequences are always more efficient than the FMT and can have a complexity as low as linear in the number of informative components. In fact, they draw their computational efficiency from both the structure of subsemilattices of some distributive lattice L such as 2^Ω and the information contained in sources. They are optimal as a combination of the two, relying in $L = 2^\Omega$ on the smallest intersection-closed or union-closed family of sets of 2^Ω containing the informative components of a source, i.e. our focal points.

Doing so, all our algorithms perform better than the FMT for all DST transformations and the combination of belief sources with Dempster's rule [30]. In addition, we demonstrated in [52] the theoretical interest arising from these focal points with the introduction of a new generalization of the conjunctive decomposition of evidence, as well as formulas uncovering how each decomposition weight is tied to the corresponding mass function.

Finally, we developed a complete open-source C++ framework containing these algorithms that can be used in place of FMT implementations. We plan to develop an equivalent Python package in order to further ease their usage.

Appendices

Appendix A

Proofs about focal points and Möbius transforms

A.1 Lemma 2.3.2.1

Proof. For any set $S \supseteq \text{supp}(f)$, given the fact that $\mathcal{P}_{/(S,\leq)}$ partitions P according to S so that all elements of each part have the same lower closure in S, we have that all elements x of a part X in $\mathcal{P}_{/(S,\leq)}$ share the same image $g_S(x) = \sum_{\substack{s \in S \\ s \leq x}} f(s) = \sum_{\substack{s \in \text{supp}(f) \\ s \leq x}} f(s) + \sum_{\substack{s \in S \setminus \text{supp}(f) \\ s \leq x}} f(s) =$ $g(x) + 0 = g(X)$. Therefore, it is possible to group the terms of the Möbius inversion formula (Eq. 1.5) by parts of the level partition of any set $S \supseteq \text{supp}(f)$, i.e. for any $y \in P$:

$$f(y) = \sum_{x \leq y} g(x) \,.\, \mu_{P,\leq}(x,y) = \sum_{\substack{X \in \mathcal{P}_{/(S,\leq)} \\ y \in \uparrow X}} \sum_{\substack{z \in X \\ z \leq y}} g(z) \,.\, \mu_{P,\leq}(z,y)$$

$$= \sum_{\substack{X \in \mathcal{P}_{/(S,\leq)} \\ y \in \uparrow X}} g(X) \,.\, \sum_{\substack{z \in X \\ z \leq y}} \mu_{P,\leq}(z,y) = \sum_{\substack{X \in \mathcal{P}_{/(S,\leq)} \\ y \in \uparrow X}} g(X) \,.\, \eta_{S,\leq,P}(X,y)$$

Moreover, if there is a part $Z \in \mathcal{P}_{/(S,\leq)}$ such that $\overset{\downarrow}{}_S Z = \emptyset$, i.e. such that for all $z \in Z$ there is no element $s \in \text{supp}(f)$ verifying $s \leq z$, then the image $g(Z)$ of this part is necessarily 0. Hence Eq. 2.2. ■

A.2 Lemma 2.3.2.2

Proof. For all $y \in P$, let us consider some $m \in \min(X)$ such that $m < y$. We can rewrite $\eta_{S,\leq,P}(X,y)$ in the following form:

$$\eta_{S,\leq,P}(X,y) = \sum_{\substack{z \in X \\ z \leq y}} \mu_{P,\leq}(z,y) = \sum_{\substack{z \in X \\ z \leq y \\ z \notin \uparrow m}} \mu_{P,\leq}(z,y) + \sum_{\substack{z \in X \\ m \leq z \leq y}} \mu_{P,\leq}(z,y)$$

$$= \sum_{\substack{z \in X \\ z \leq y \\ z \notin \uparrow m}} \mu_{P,\leq}(z,y) + \sum_{m \leq z \leq y} \mu_{P,\leq}(z,y) - \sum_{\substack{z \notin X \\ m \leq z \leq y}} \mu_{P,\leq}(z,y)$$

which reduces to:

$$\eta_{S,\leq,P}(X,y) = \epsilon_m(X,y) - \sum_{\substack{z\notin X \\ m\leq z\leq y}} \mu_{P,\leq}(z,y) \tag{A.1}$$

where $\epsilon_m(X,y) = \sum_{\substack{z\in X \\ z\leq y \\ z\notin\uparrow m}} \mu_{P,\leq}(z,y)$, since Eq. 1.6 gives us $\sum_{m\leq z\leq y} \mu_{P,\leq}(z,y) = 0$. Now, let us determine from which parts the elements of $\{z \notin X \ / \ m \leq z \leq y\}$ are. First, any of these elements belongs to a part Z from $\mathcal{P}_{/(S,\leq)}\backslash\{X\}$. Then, any element z of these Z parts satisfies $m \leq z$. So, we have $\downarrow m \subseteq \downarrow z$, which implies that ${}^{\downarrow}_{S}X \subseteq {}^{\downarrow}_{S}Z$ by definition of a level partition. More precisely, we get ${}^{\downarrow}_{S}X \subset {}^{\downarrow}_{S}Z$, since $Z \neq X$. Thus, the only parts $Z \in \mathcal{P}_{/(S,\leq)}$ that need to be considered in the sum of Eq. A.1 satisfy ${}^{\downarrow}_{S}X \subset {}^{\downarrow}_{S}Z$.

From this, without any hypothesis on the level partition, we can show that η can be written in a recursive form, but not without ϵ, which can only be computed from μ. However, if we consider that every part $Z \in \mathcal{P}_{/(S,\leq)}$ satisfying ${}^{\downarrow}_{S}X \subseteq {}^{\downarrow}_{S}Z$ has a minimum, then we can get rid of ϵ. Indeed, if X has a minimum, then every element of X is greater than m, i.e. $\epsilon_{m,\leq}(X,y) = \sum_{\substack{z\in X \\ z\leq y \\ z\notin\uparrow m}} \mu_{P,\leq}(z,y) = 0$. The same goes for the rest of the recursion, hence the need for the parts ${}^{\downarrow}_{S}X \subset {}^{\downarrow}_{S}Z$ to each have a minimum too.

Moreover, these conditions mean that their respective lower closures in S all have a supremum, since the supremum is by definition the least upper bound. Thus, we get that the only parts $Z \in \mathcal{P}_{/(S,\leq)}$ that need to be considered in the sum of Eq. A.1 satisfy $\bigvee {}^{\downarrow}_{S}X < \bigvee {}^{\downarrow}_{S}Z$, where $\bigvee {}^{\downarrow}_{S}X = m = \bigwedge X$ and $\bigvee {}^{\downarrow}_{S}Z = \bigwedge Z$. So, we have:

$$\begin{aligned}
\eta_{S,\leq,P}(X,y) &= -\sum_{\substack{z\notin X \\ m\leq z\leq y}} \mu_{P,\leq}(z,y) = -\sum_{\substack{Z\in\mathcal{P}_{/(S,\leq)} \\ m<\bigwedge Z\leq y}} \sum_{\substack{z\in Z \\ m\leq z\leq y}} \mu_{P,\leq}(z,y) \\
&= -\sum_{\substack{Z\in\mathcal{P}_{/(S,\leq)} \\ \bigwedge X<\bigwedge Z\leq y}} \sum_{\substack{z\in Z \\ z\leq y}} \mu_{P,\leq}(z,y) = -\sum_{\substack{Z\in\mathcal{P}_{/(S,\leq)} \\ \bigwedge X<\bigwedge Z\leq y}} \eta_{S,\leq,P}(Z,y)
\end{aligned}$$

In addition, we know from Definition 2.3.2.2 that if $y \in \min(X)$, i.e. $y = \bigwedge X$ if X has a minimum, then we have $\eta_{S,\leq,P}(X,y) = 1$. ■

A.3 Theorem 2.3.4.1

Proof. For the sake of clarity, we will use the alias $S = \text{supp}(f)$ in the following. Let G be the set made of the minimal elements of each part of the partition $\mathcal{G}$ defining g, i.e. $G = \bigcup_{X\in\mathcal{G}} \min(X)$. We will assume that ${}^{\vee}G$ is an upper subsemilattice of P. Let us now find the elements of $\text{supp}(f)$ with G.

(i) It is obvious that $\min(P) \subseteq G$. (ii) For any elements $y \in P$ and $s \in \text{supp}(f)$ such that $y < s$ and $|{}^{\downarrow}_{S}s\backslash{}^{\downarrow}_{S}y| = 1$, we have $g(s) = g(x) + f(s) \neq g(x)$, for any element $x \in P$ where $y \leq x < s$. Therefore, there exists a part $X \in \mathcal{G}$ such that $s \in \min(X)$, i.e. $s \in G$. (iii) For any elements $y \in G$ and $s \in \text{supp}(f)$ such that $y < s$ and $|{}^{\downarrow}_{S}s\backslash{}^{\downarrow}_{S}y| = n$, where $n \geq 2$, there exists an element $s' \in \text{supp}(f)$ such that $s' < s$ and $s' \not\leq y$. Thus, there is an element $y' = y \vee s'$ where $y' \leq s$ and $|{}^{\downarrow}_{S}s\backslash{}^{\downarrow}_{S}y'| \leq n-1$. If $y' \neq s$ and $|{}^{\downarrow}_{S}s\backslash{}^{\downarrow}_{S}y'| \geq 2$, then there is an element $s'' \in \text{supp}(f)$ such that $s'' < s$ and $s'' \not\leq y'$, which means that there is an element $y'' = y' \vee s''$ verifying both

$|{\downarrow_S}s \setminus {\downarrow_S}y''| \leq n-2$ and $y'' \leq s$. This upward recursion will ultimately end either because the number of elements in the difference of lower closures in S reaches 1 (which means that $s \in G$, given (ii)) or because it has generated s with the supremum of two lower elements.

Furthermore, from what we demonstrated up to this point, either the aforementionned element $s' \in \text{supp}(f)$ is in G (by (i) or (ii)) or there exists an element $x \in G$ such that $x < s'$ verifying $|{\downarrow_S}s' \setminus {\downarrow_S}x| \geq 2$, which means that there exists an element $\tilde{s} \in \text{supp}(f)$ such that $\tilde{s} < s'$ and $\tilde{s} \not\leq x$. And again, either $\tilde{s} \in G$ or there exists an element $x' \in G$ such that $x' < \tilde{s}$ verifying $|{\downarrow_S}\tilde{s} \setminus {\downarrow_S}x'| \geq 2$. Noticing that $|{\downarrow_S}\tilde{s}| < |{\downarrow_S}s'| < |{\downarrow_S}s|$, we know that this downward recursion will ultimately end with an element $\dot{s} \in \text{supp}(f)$ that is in G since the lower closure in S decreases and will eventually contain only one element of S. Therefore, either s' is in G or it can be found by computing successive suprema as in (iii), starting with the supremum between $\dot{s}$ and another element of G. The same goes for s'' and any other elements eventually encountered in the upward recursion of (iii). Hence, all these elements can be used with y to find s again through successive suprema. So, we get that all elements of $\text{supp}(f)$ are in the join-closure of G, i.e. ${}^{\vee}G$.

We can be even more precise than this by noticing that the supremum of two minimal elements of P associated with 0 through g cannot be an element of $\text{supp}(f)$ that is not already in G. Indeed, for any element $m \in \min(P)$, we have $g(m) = f(m)$. So, if $g(m) = 0$, then $f(m) = 0$, i.e. $m \notin \text{supp}(f)$. Yet, each supremum in (iii) involves at least one element of $\text{supp}(f)$, i.e. one element that is not both a minimal element and an element with null image through g. This means that the supremum of two elements of M, where $M = \min(P) \cap \overline{\text{supp}(g)}$, if not already in G, is either not in $\text{supp}(f)$ or the supremum of another set of elements, where at least one is in $G \setminus M$. Thus, we would like to avoid computing any supremum of the form $\bigvee A$, where $A \subseteq M$, in our join-closure. Let Y be the set made of every supremum $x \vee a$, where $x \in G \setminus M$ and $a \in M$. Notice that the elements of ${}^{\vee}(Y \cup G \setminus M)$ are of the form $\bigvee (X \cup A)$, where $\emptyset \subset X \subseteq G \setminus M$ and $A \subseteq M$. Therefore, ${}^{\vee}(Y \cup G \setminus M)$ contains all the elements of ${}^{\vee}G$, except the ones that are exclusively of the form $\bigvee A$, where $A \subseteq M$. Consequently, we have $\text{supp}(f) \subseteq {}^{\vee}(Y \cup G \setminus M)$, which means, by definition of a closure operator, that ${}^{\vee}\text{supp}(f) \subseteq {}^{\vee}(Y \cup G \setminus M) \subseteq {}^{\vee}G$. ■

A.4 Corollary 2.3.5.1

Proof. Let $\mathcal{G}^*$ be the partition of P w.r.t. the images of g such that $\mathcal{G}^* = \mathcal{P}_{/(\text{supp}(f), \leq)}$, and let $G^* = \bigcup_{X \in \mathcal{G}^*} \min(X)$. We have $G^* = {}^{\vee}\text{supp}(f) \cup \min(P)$, since $\min(P)$ contains the minimal elements of the only part of $\mathcal{P}_{/(\text{supp}(f), \leq)}$ that is not in $\uparrow \text{supp}(f)$, if it even exists. Adapting Theorem 2.3.4.1 to the multiplicative Möbius transform, we also have ${}^{\vee}\text{supp}(h-1) \subseteq {}^{\vee}G^* = {}^{\vee}(\text{supp}(f) \cup \min(P))$, which means that ${}^{\vee}(\text{supp}(h-1) \cup \min(P)) \subseteq {}^{\vee}(\text{supp}(f) \cup \min(P))$. The same reasoning can be applied to the partition $\mathcal{G}_h^* = \mathcal{P}_{/(\text{supp}(h-1), \leq)}$, leading to ${}^{\vee}(\text{supp}(f) \cup \min(P)) \subseteq {}^{\vee}(\text{supp}(h-1) \cup \min(P))$. Therefore, combining all inequalities, we get that ${}^{\vee}(\text{supp}(f) \cup \min(P)) = {}^{\vee}(\text{supp}(h-1) \cup \min(P))$. Finally, by Property 2.3.5.1, we have ${}^{\vee}\text{supp}(f) = {}^{\vee}(\text{supp}(h-1) \cup \min(P))$. ■

A.5 Theorem 2.3.5.1

Proof. In formal terms, the condition on h' translates for any $y \in P$ to:

$$h'(y) = \begin{cases} h'(y) & \text{if } y = x \\ h(y) & \text{otherwise} \end{cases}$$

By Eq. 2.6, we have that for any $y \in P$, $g'(y) = \prod_{z \leq y} h'(z)$. We observe that $h'(x)$ appears in the product of $g'(y)$ only if $x \leq y$. Since all other images of h' are equal to the ones of h, we get Eq. 2.7. Then, Corollary 2.3.5.1 gives us ${}^{\vee}\mathrm{supp}(f) = {}^{\vee}(\mathrm{supp}(h-1) \cup \min(P))$ and ${}^{\vee}\mathrm{supp}(f') = {}^{\vee}(\mathrm{supp}(h'-1) \cup \min(P))$. Moreover, notice that $\mathrm{supp}(h'-1) \subseteq \mathrm{supp}(h-1) \cup \{x\}$. We get ${}^{\vee}(\mathrm{supp}(h'-1) \cup \min(P)) \subseteq {}^{\vee}(\mathrm{supp}(h-1) \cup \min(P) \cup \{x\})$, which implies that ${}^{\vee}\mathrm{supp}(f') \subseteq {}^{\vee}(\mathrm{supp}(f) \cup \{x\})$. Consequently, we know that any upper subsemilattice ${}^{\vee}S$ of P such that ${}^{\vee}S \supseteq \mathrm{supp}(f) \cup \{x\}$ can be used in Eq. 2.5 of Theorem 2.3.3.1 for both f and f'. So, we get for any $y \in P$:

$$\begin{aligned} f'(y) &= \sum_{\substack{s \in {}^{\vee}S \\ s \leq y}} g'(s) \,.\, \eta_{S,\leq,P}(s,y) \\ &= \sum_{\substack{s \in {}^{\vee}S \\ x \leq s \leq y}} g(s) \,.\, \frac{h'(x)}{h(x)} \,.\, \eta_{S,\leq,P}(s,y) + \sum_{\substack{s \in {}^{\vee}S \\ s \leq y \\ s \notin \uparrow x}} g(s) \,.\, \eta_{S,\leq,P}(s,y) \\ &= \frac{h'(x)}{h(x)} \,.\, \sum_{\substack{s \in {}^{\vee}S \\ x \leq s \leq y}} g(s) \,.\, \eta_{S,\leq,P}(s,y) + f(y) - \sum_{\substack{s \in {}^{\vee}S \\ x \leq s \leq y}} g(s) \,.\, \eta_{S,\leq,P}(s,y) \\ &= f(y) + \left[\frac{h'(x)}{h(x)} - 1\right] \,.\, \sum_{\substack{s \in {}^{\vee}S \\ x \leq s \leq y}} g(s) \,.\, \eta_{S,\leq,P}(s,y) \end{aligned}$$

Besides, for any $s, y \in P$ where $x \leq s < y$, we have $\{p \;/\; s < p\} = \{p \;/\; x \leq s < p\}$ and so:

$$\eta_{S,\leq,P}(s,y) = -\sum_{\substack{p \in {}^{\vee}S \\ s < p \leq y}} \eta_{S,\leq,P}(p,y) = -\sum_{\substack{p \in \uparrow x \\ p \in {}^{\vee}S \\ s < p \leq y}} \eta_{S,\leq,P}(p,y) = \eta_{S,\leq,\uparrow x}(s,y),$$

where $\uparrow x$ is the upper closure of x in P. Therefore, the expression $\sum_{\substack{s \in {}^{\vee}S \\ x \leq s \leq y}} g(s) \,.\, \eta_{S,\leq,P}(s,y)$ is in fact the Möbius transform of g in $(\uparrow x, \leq)$. ■

A.6 Corollary 2.4.2.1

Proof. For any mass function m such that $C \in \mathrm{supp}(m)$, where $C = \bigcup \mathrm{supp}(m)$, Corollary 2.3.5.1 implies that ${}^{\wedge}\mathrm{supp}(m) \backslash \{C\} = {}^{\wedge}\mathrm{supp}(w^C - 1) \backslash \{C\}$, where w^C is the inverse of the multiplicative Möbius transform of q in $(\downarrow C, \supseteq)$ and q is the zeta transform of m in $(\downarrow T, \supseteq)$, where $T \supseteq C$. In particular, suppose $C \neq \Omega$ and let w' be the inverse of the multiplicative Möbius transform of q' in $(2^{\Omega}, \supseteq)$, where q' corresponds to m' and m' is equal to m everywhere, except that $m'(C) = 0$ and $m'(\Omega) = m(C)$. We have ${}^{\wedge}\mathrm{supp}(w' - 1) \backslash \{\Omega\} = {}^{\wedge}\mathrm{supp}(m') \backslash \{\Omega\} = {}^{\wedge}\mathrm{supp}(m) \backslash \{C\} = {}^{\wedge}\mathrm{supp}(w^C - 1) \backslash \{C\}$.

Furthermore, $q'(\Omega) = q(C)$ and for any element $y \subseteq C$, we have $q'(y) = q(y)$, which implies that for any element $s \in {}^{\wedge}\mathrm{supp}(w^C - 1) \backslash \{C\}$, we have $q'(y) = q(y)$. This means that $w^C(C) = q(C)^{-1} = w'(\Omega)$ and that by Theorem 2.3.3.1, for any element $y \subset C$, we have $w^C(y) = \prod_{\substack{s \in {}^{\wedge}S \\ s \supseteq y}} q(s)^{-\eta_{S,\supseteq,\downarrow a}(s,y)} = \prod_{\substack{s \in {}^{\wedge}S' \\ s \supseteq y}} q'(s)^{-\eta_{S',\supseteq,P}(s,y)} = w'(y)$, where $S = \mathrm{supp}(w^C - 1)$ and $S' =$

$S\backslash\{C\} \cup \{\Omega\}$ and $\eta_{S,\supseteq,\downarrow C}(C, y) = \eta_{S',\supseteq,P}(\Omega, y)$, since C and Ω have the same relations with respect to the elements in $^\wedge\text{supp}(w^C - 1)\backslash\{C\}$ and so the same role in the recursion giving η in Eq. 2.4. ■

A.7 Proposition 2.4.3.1

Proof. Let w' be equal to w everywhere on 2^Ω, except for the image of some $x \in {}^\wedge\text{supp}(w - 1)\backslash\{\Omega\}$. Also, let m' and q' be the functions corresponding to w' so that they satisfy Eq. 2.6 in place of respectively m and q. For any element $y \in 2^\Omega$, Theorem 2.3.5.1 gives us:

$$q'(y) = \begin{cases} \frac{w(x)}{w'(x)}.q(y) & \text{if } x \supseteq y \\ q(y) & \text{otherwise} \end{cases}$$

and

$$m'(y) = \begin{cases} 0 & \text{if } y \notin {}^\wedge S \\ m(y) + \left[\frac{w(x)}{w'(x)} - 1\right].m_{\downarrow x}(y) & \text{if } x \supseteq y \\ m(y) & \text{otherwise} \end{cases}, \tag{A.2}$$

where $^\wedge S \supseteq \text{supp}(m)$ and $m_{\downarrow x} : \downarrow x \to \mathbb{R}$ is the Möbius transform of q in $(\downarrow x, \supseteq)$. However, in DST, we have to respect a normalization constraint on m' and w' that is imposed by Eq. 2.9 and Eq. 2.14. Obviously, looking at the normalized product, we see that the only way to normalize w' without changing any other image of $2^\Omega\backslash\{\Omega\}$ through w is by changing the one on Ω. More precisely, the normalized equivalent $\overline{w'}$ of w' must satisfy $\overline{w'}(\Omega) = \frac{w(x)}{w'(x)}.w(\Omega)$. So, taking back Eq. A.2, but this time updating m' to get its normalized equivalent $\overline{m'}$ with $x = \Omega$, which implies that $\downarrow x = \downarrow \Omega = 2^\Omega$, we have for any $y \in 2^\Omega$:

$$\overline{m'}(y) = m'(y) + \left[\frac{w'(\Omega)}{\overline{w'}(\Omega)} - 1\right].m'(y) = \frac{w'(\Omega)}{\overline{w'}(\Omega)}.m'(y) = \frac{w'(x)}{w(x)} \ . \ m'(y)$$

Discarding m' as it is not a mass function (does not satisfy Eq. 2.9), we consider directly $\overline{m'}$ as the result of the modification of one image in the conjunctive decomposition. Hence Eq. 2.16 and Eq. 2.17 with $^\wedge S \supseteq \text{supp}(m)$. In addition, since 2^Ω has a maximum Ω, we have $^\wedge\text{supp}(m) = {}^\wedge\text{supp}(w - 1) \cup \{\Omega\}$ by Corollary 2.3.5.1.

Finally, remark that $m_{\downarrow x}$ is a mass function, since it is the Möbius transform, in a reduced subset, of q, which corresponds to the mass function m. It is the same projection process as the one producing $m_{\downarrow C}$ in section 2.4.2. ■

Appendix B

Proofs about the Efficient Möbius Transformations

B.1 Proposition 3.3.1.1

Proof. Let us define L_S as the set containing the join of all combinations of iota elements of S, in addition to $\bigwedge$. Formally, we have:

$$L_S = \left\{ \bigvee X \ / \ X \subseteq \iota(S),\ X \neq \emptyset \right\} \cup \left\{ \bigwedge S \right\}.$$

By construction, we already know that the join of any number (except 0) of elements from L_S is also in L_S. So, L_S is an upper-subsemilattice of P. In the following, we will prove that it is also a lower-subsemilattice of P.

But, before anything, notice that $\bigwedge .^{\uparrow S} : P \to P$ is a closure operator, i.e. for any elements $x, y \in P$, we have:

$$x \leq \bigwedge x^{\uparrow S} \tag{B.1}$$

$$x \leq y \ \Rightarrow \ \bigwedge x^{\uparrow S} \leq \bigwedge y^{\uparrow S} \tag{B.2}$$

$$\bigwedge \left(\bigwedge x^{\uparrow S} \right)^{\uparrow S} = \bigwedge x^{\uparrow S} \tag{B.3}$$

Now, consider the meet of two iota elements $\iota_1 \wedge \iota_2$, where $\iota_1, \iota_2 \in \iota(S)$. By definition, there are two join-irreducible elements $x, y \in {}^{\vee}\mathcal{I}(P)$ such that $\iota_1 \wedge \iota_2 = \bigwedge x^{\uparrow S} \wedge \bigwedge y^{\uparrow S} = \bigwedge (x^{\uparrow S} \cup y^{\uparrow S})$. Let us note $\delta = \bigwedge (x^{\uparrow S} \cup y^{\uparrow S})$. Since $S \supseteq x^{\uparrow S} \cup y^{\uparrow S}$, we know that $\bigwedge S \leq \delta$. If $\delta = \bigwedge S$, then $\delta \in L_S$. Otherwise, given that P is a lattice, we know that δ is equal to the join of all the join-irreducible elements of P that are less than δ, i.e. $\bigvee \delta^{\downarrow^{\vee} \mathcal{I}(P)} = \delta$.

For all join-irreducible elements $i \in \delta^{\downarrow^{\vee} \mathcal{I}(P)}$, we have $i \leq \delta$. By Eq. (B.2), we also have $\bigwedge i^{\uparrow S} \leq \bigwedge \delta^{\uparrow S}$. Moreover, we know that $\delta^{\uparrow S} \supseteq x^{\uparrow S} \cup y^{\uparrow S}$ because $\delta = \bigwedge (x^{\uparrow S} \cup y^{\uparrow S})$. This implies that we have $\bigwedge \delta^{\uparrow S} \leq \delta$, and so $\bigwedge i^{\uparrow S} \leq \delta$.

In addition, by Eq. (B.1), we get that $i \leq \bigwedge i^{\uparrow S}$, which means that, for all join-irreducible elements $i \in \delta^{\downarrow^{\vee} \mathcal{I}(P)}$, we have $i \leq \bigwedge i^{\uparrow S} \leq \delta$. Therefore, combined with the fact that $\bigvee \delta^{\downarrow^{\vee} \mathcal{I}(P)} = \delta$, we finally obtain that $\bigvee \delta^{\downarrow \iota(S)} = \delta$. In plain English, this means that δ is equal to the join

of all the iota elements of S that are less than δ. So, by definition of L_S, we get that $\delta \in L_S$. Thus, the meet of two iota elements $\iota_1 \wedge \iota_2$, where $\iota_1, \iota_2 \in \iota(S)$, is in L_S.

It only remains to consider the meet of two arbitrary elements of L_S, i.e. $x \wedge y$, where $x, y \in L_S$. Notice that if $x = \bigwedge S$ or $y = \bigwedge S$, then $x \wedge y = \bigwedge S \in L_S$. Otherwise, it can be decomposed as follows:

$$x \wedge y = \left(\bigvee x^{\downarrow \iota(S)}\right) \wedge \left(\bigvee y^{\downarrow \iota(S)}\right)$$

Since P follows the distributive law Eq. (3.3), we can rewrite this equation as:

$$x \wedge y = \bigvee_{\iota_1 \in x^{\downarrow \iota(S)}} \bigvee_{\iota_2 \in y^{\downarrow \iota(S)}} (\iota_1 \wedge \iota_2)$$

The meet of two iota elements of S being in L_S, we get that $x \wedge y$ is equal to the join of elements that are all in L_S. As we already established that L_S is an upper-subsemilattice of P, we get that the meet of two arbitrary elements of L_S is also in L_S. Thus, L_S is a lower-subsemilattice of P as well and therefore a sublattice of P.

In addition, notice that for all element $s \in S$ and for all $i \in s^{\downarrow {}^{\vee}\mathcal{I}(P)}$, we have by construction $i \leq \bigwedge i^{\uparrow S} \leq s$. Therefore, P being a lattice, we have $s = \bigvee s^{\downarrow {}^{\vee}\mathcal{I}(P)} = \bigvee s^{\downarrow \iota(S)}$, i.e. $s \in L_S$. Besides, if $\bigwedge P \in S$, then it is equal to $\bigwedge S$, which is also in L_S by construction. So, $S \subseteq L_S$. It follows that the meet or join of every nonempty subset of S is in L_S, i.e. $M_S \subseteq L_S$ and $J_S \subseteq L_S$, where M_S is the smallest meet-closed subset of P containing S and J_S is the smallest join-closed subset of P containing S. Furthermore, iota elements are defined as the meet of a set of elements of S, which implies that they are necessarily all contained in M_S, i.e. $\iota(S) \subseteq M_S$. This means that we cannot build a smaller sublattice of P containing S. Therefore, L_S is the smallest sublattice of P containing S.

Finally, let us verify that all iota elements are join-irreducible elements of L_S. For any join-irreducible element $i \in {}^{\vee}\mathcal{I}(P)$, assume there are two distinct elements $x, y \in L_S$ such that $x < \bigwedge i^{\uparrow S}$ and $y < \bigwedge i^{\uparrow S}$. This implies by Eq. (B.2) and (B.3) that $\bigwedge x^{\uparrow S} \leq \bigwedge i^{\uparrow S}$ and $\bigwedge y^{\uparrow S} \leq \bigwedge i^{\uparrow S}$. Moreover, if $i \leq x$ and $i \leq y$, then by Eq. (B.2), we get that $\bigwedge i^{\uparrow S} \leq \bigwedge x^{\uparrow S}$ and $\bigwedge i^{\uparrow S} \leq \bigwedge y^{\uparrow S}$, which means that $x = y = \bigwedge i^{\uparrow S}$. This contradicts the fact that $x \neq y$. Thus, we get that $i \not\leq x$ and $i \not\leq y$. By Lemma 3.3.1.1, this implies that $i \not\leq x \vee y$. Since $i \leq \bigwedge i^{\uparrow S}$, we have necessarily $x \vee y \neq \bigwedge i^{\uparrow S}$ and so $x \vee y < \bigwedge i^{\uparrow S}$. Therefore, $\bigwedge i^{\uparrow S}$ is a join-irreducible element of L_S, i.e. $\iota(S)$ is the set containing only the join-irreducible elements of L_S. ■

B.2 Theorem 3.3.2.1

Proof. Consider the sequence $(H_k)_{k \in [\![1,n]\!]}$, where $H_k = (L, E_k)$ and:

$$E_k = \left\{(x, y) \in L^2 \ / \ y = x \vee i_{n+1-k}\right\}.$$

By definition, for all $k \in [\![1, n]\!]$ and $\forall (x, y) \in E_k$, we have $x, y \in L$ and $x \leq y$, i.e. $(x, y) \in E_{\leq}$. Reciprocally, every arrow $e \in E_{\leq}$ can be decomposed as a unique path $(e_1, e_2, \ldots, e_n) \in A_1 \times A_2 \times \cdots \times A_n$, where $A_k = E_k \cup I_P$:

Similarly to the FMT, this sequence builds unique paths simply by generating the whole lattice L step by step with each join-irreducible element of L. However, unlike the FMT, the join-irreducible elements of L are not necessarily atoms. Doing so, pairs of join-irreducible elements may be ordered, causing the sequence to skip or double some elements. And even if all the join-irreducible elements of L are atoms, since L is not necessarily a Boolean lattice, the join of two atoms may be greater than a third atom (e.g. if L is the diamond lattice), leading

to the same issue. Indeed, it is easy to build a path between two elements x, y of L such that $x \leq y$: At step 1, we take the arrow $(x, x \vee i_n)$ if $i_n \leq y$ (we take the identity arrow (x, x) otherwise). At step 2, we take the arrow $(p, p \vee i_{n-1})$ if $i_{n-1} \leq y$, where $p = x \vee i_n$ or $p = x$ depending on whether or not $i_n \leq y$, and so on until we get to y. Obviously, we cannot get to y if we take an arrow $(p, p \vee i)$ where $i \not\leq y$. So, any path from x to y only consists of arrows obtained with join-irreducible elements that are less than y. But, are they all necessary to reach y from x? Let i_k be a join-irreducible element such that $i_k \leq y$. This join-irreducible element is only considered in E_{n+1-k}. Suppose we do not take the arrow at step $n+1-k$, i.e. we replace it by an identity arrow (p, p), where p was reached through a path from x with arrows at steps 1 to $n-k$. Since L is a distributive lattice, and since its join-irreducible elements are ordered such that $\forall i_j, i_l \in {}^{\vee}\mathcal{I}(L)$, $j < l \Rightarrow i_j \not\geq i_l$, we have by Corollary 3.3.1.1 that for any $k \in [\![1, n]\!]$, $i_k \not\leq \bigvee {}^{\vee}\mathcal{I}(L)_{k-1}$. So, if $i_k \not\leq p$, then by Lemma 3.3.1.1, we also have $i_k \not\leq p \vee \bigvee {}^{\vee}\mathcal{I}(L)_{k-1}$. Since ${}^{\vee}\mathcal{I}(L)_{k-1}$ contains all join-irreducible elements considered after step $n+1-k$, this implies that there is no path from p to an element greater than i_k. Yet, $i_k \leq y$. Thus, if $i_k \not\leq p$, then y can only be reached from p through a path containing the arrow $(p, p \vee i_k)$. Otherwise, if $i_k \leq p$, then $p = p \vee i_k$, which means that only an identity arrow can be taken at step $n+1-k$ anyway. Either way, there is only one arrow that can be taken at step $n+1-k$ to build a path between p and y. All join-irreducible elements less than y must be used to build a path from x to y. Thereby, this path is unique, meaning that Theorem 3.2.3.1 is satisfied. The sequence $(H_k)_{k \in [\![1,n]\!]}$ computes the same zeta transformations as $G_<$. ■

B.3 Theorem 3.3.2.2

Proof. Consider the sequence $(H_k)_{k \in [\![1,n]\!]}$, where $H_k = (M, E_k)$ and:

$$E_k = \left\{ (x, y) \in M^2 \ / \ x = \bigwedge (y \vee i_k)^{\uparrow M} \text{ and } x \leq y \vee \bigvee \iota(M)_k \right\},$$

where $\iota(M)_k = \{i_1, i_2, \ldots, i_k\}$.

By definition, for all $k \in [\![1, n]\!]$ and $\forall (x, y) \in E_k$, we have $x, y \in M$ and $x \geq y$, i.e. $(x, y) \in E_{\geq}$. Reciprocally, every arrow $e \in E_{\geq}$ can be decomposed as a unique path $(e_1, e_2, \ldots, e_n) \in A_1 \times A_2 \times \cdots \times A_n$, where $A_k = E_k \cup I_P$:

Recall the procedure, described in Theorem 3.3.2.1, that builds unique paths simply by generating all elements of a finite distributive lattice $L \supseteq M$, based on the join of its join-irreducible elements, step by step. Here, the idea is to do the same, except that we remove all elements that are not in M. Doing so, the only difference is that the join $y \vee i_k$ of an element $y \in M$ with a join-irreducible element $i_k \in \iota(M)$ of this hypothetical lattice L may not be in M. However, thanks to the meet-closure of M, we can "jump the gap" between two elements y and p of M, should they be separated by elements of $L \backslash M$. Indeed, for all join-irreducible element $i_k \in \iota(M)$, if $x \geq y \vee i_k$, then since M is meet-closed and $x \in M$, there is a unique element $p \in M$ that we call *proxy* such that $p = \bigwedge (y \vee i_k)^{\uparrow M}$. In complement, the synchronizing condition $p \leq y \vee \bigvee \iota(M)_k$ ensures the unicity of this jump, and so the unicity of the path between any two elements x and y of M.

Finding a path from x to y is easy: take all iota elements less than x, and simply compute successive joins with them, starting from y. At step n, we take the arrow $(\bigwedge (y \vee i_n)^{\uparrow M}, y)$ if $i_n \leq x$, and (y, y) otherwise. At step $n-1$, we take the arrow $(\bigwedge (p \vee i_{n-1})^{\uparrow M}, p)$ if $i_{n-1} \leq x$, where $p = \bigwedge (y \vee i_n)^{\uparrow M}$ or $p = y$, depending on whether or not $i_n \leq x$. Proceeding as such until x is reached guarantees the existence of a path, since the synchronizing condition is always satisfied. Then, the question is: Are there other paths?

Obviously, for some $p \in M$ such that $x \geq p$, no path from x to y can contain arrows $(\bigwedge(p \vee i)^{\uparrow M}, p)$ if $i \not\leq x$. Thus, it contains only arrows corresponding to joins with iota elements that are less than x. Next, let us consider some element $p \in M$. If $i_{k-1} \leq p$, then $p = \bigwedge(p \vee i_{k-1})^{\uparrow M}$, which means that only an identity arrow (p, p) can be taken at step $k-1$. Otherwise, if $i_{k-1} \not\leq p$, then the arrow $(\bigwedge(p \vee i_{k-1})^{\uparrow M}, p)$ can only exist in the sequence if $\bigwedge(p \vee i_{k-1})^{\uparrow M} \leq p \vee \bigvee \iota(M)_{k-1}$. We know by Corollary 3.3.1.1 that $i_k \not\leq \bigvee \iota(M)_{k-1}$. By Lemma 3.3.1.1, this implies that if $i_k \not\leq p$, then $i_k \not\leq p \vee \bigvee \iota(M)_{k-1}$, which means that $i_k \not\leq \bigwedge(p \vee i_{k-1})^{\uparrow M}$. Through the same reasoning, this means that $i_k \not\leq \bigwedge \left(\bigwedge(p \vee i_{k-1})^{\uparrow M} \vee i_{k-2}\right)^{\uparrow M}$. In fact, by recurrence, we have $i_k \not\leq \bigwedge \left(\cdots \bigwedge \left(\bigwedge(p \vee i_{k-1})^{\uparrow M} \vee i_{k-2}\right)^{\uparrow M} \cdots \vee i_1 \right)^{\uparrow M}$. Thus, if $i_k \not\leq p$, then no element greater than i_k can be at the tail of a path leading to p through arrows of the sequence. This means that on a path from x to y, if $i_k \leq x$, then at step k, either $i_k \leq p$ (i.e. we can only take the identity arrow (p, p)) or $i_k \not\leq p$, which implies that we must take the arrow $(\bigwedge(p \vee i_k)^{\uparrow M}, p)$. This implies that all iota elements less than x must be used in the joins corresponding to the arrows of the path from x to y. Therefore, the path described above is unique, which satisfies Theorem 3.2.3.1. The sequence $(H_k)_{k \in [\![1,n]\!]}$ computes the same zeta transformations as $G_>$. ■

Appendix C

Implementation of the Efficient Möbius Transformations (EMT)

We present in this section evidence-based algorithms for the computation of DST transformations such as the commonality function q and the implicability function b, as well as their inversions, namely the mass function m and the conjunctive and disjunctive weight functions w and v. All these algorithms have better complexities than the FMT, i.e. less than $O(|\Omega|.2^{|\Omega|})$.

We implemented these algorithms as a general-purpose C++ *evidence-based* framework along with combination rules from DST. We plan to transpose this implementation as a Python package in the near future to ease its usage. Code and implementation details can be found at [85].

C.1 Data structure

C.1.1 Overview

The core of our implementation uses the class *bitset* (a contiguous sequence of bits) from the standard library, along with a special dynamic binary tree of our design presented in section C.1.3. This tree allows us to search for supersets or subsets without having to consider all sets. For simple look-ups, i.e. just to get the value associated with a set, we use Hashmaps, since they feature constant time complexities for look-up and insertion.

Each representation of evidence (mass function, commonality function, implicability function, conjunctive weight function, disjunctive weight functions, etc) has its own class. A mass function is an object containing a tree of values different from 0, i.e. corresponding to focal sets. It inherits the abstract class *mobius_transform* which simply defines the behavior of storing values such as focal sets. An object inheriting this abstract class can be created either directly, by providing a key-value object such as an Hashmap or a tree, or indirectly, through inversion of a given zeta transform and a given order relation ($\subseteq$ or $\supseteq$). It also features methods to remove negligeable values and renormalize. The conjunctive and disjunctive weight functions also inherit this abstract class. For these last classes, all set that is not present in their tree is associated with 1.

Other representations inherit the abstract class *zeta_transform*. This abstract class defines the behavior of computing focal points from focal sets, given some order relation, and storing

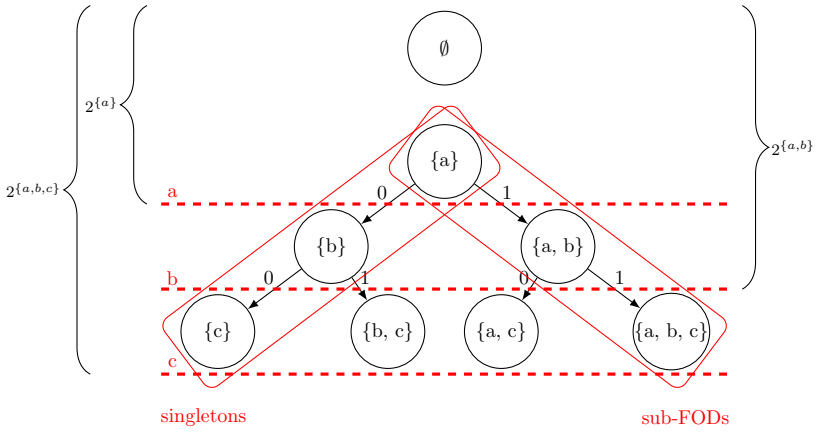


Figure C.1: Illustration depicting our data structure for powerset functions on the frame of discernment $\Omega = \{a, b, c\}$. It is a binary tree with values on nodes and leaves. For the sake of clarity, the structure is shown as if it was static, where all elements from 2^Ω are considered special elements. We see clearly that it reproduces the natural generation of the powerset lattice, encapsulating the powerset of every $\{\omega_1, \ldots, \omega_n\}$ in the powerset of $\{\omega_1, \ldots, \omega_{n+1}\}$. As a consequence, the search for all singletons and all these sub-FODs can be restrained respectiveley to the left and right chain of nodes.

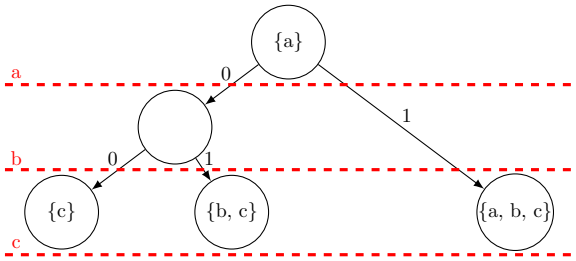


Figure C.2: Illustration depicting our data structure for powerset functions on the frame of discernment $\Omega = \{a, b, c\}$. Here, special sets are $\{a\}$, $\{c\}$, $\{b, c\}$ and $\{a, b, c\}$. The blank node is a disjunction node.

their values. It also keeps its mobius_transform object in memory for eventual additional projections, i.e. to get the value associated with other sets than focal points. An object inheriting this abstract class can be created either directly, by providing another zeta_transform object, or indirectly, by providing an object inheriting the mobius_transform abstract class, an order relation ($\subseteq$ or $\supseteq$) and an operation ($+$ or $\times$). For the latter, you may also provide a specific computation scheme (without structure, as a semilattice or as a lattice). This class also features methods to find the value associated with a non-focal point. The commonality, implicability and plausibility functions inherit this abstract class.

C.1.2 Frame of discernment

A frame of discernment (FOD), noted Ω, is represented as an object containing an array of labels and a Hashmap that enables one to find the index associated with a particular label.

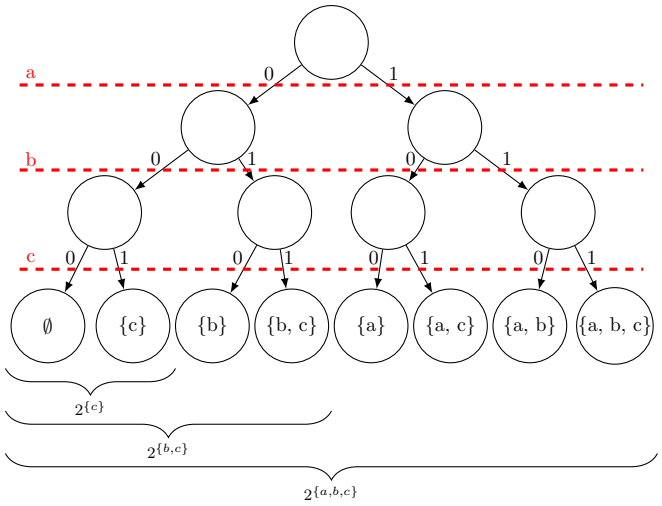


Figure C.3: Same example as in Fig. C.1 with an analogous data structure proposed by Wilson [44]. It is a binary tree in which values are only assigned to terminal leaves, not intermediate nodes. As in Fig. C.1, for the sake of clarity, the structure is shown as if it was static, where all elements from 2^{Ω} are considered special elements.

C.1.3 Powerset function

Our data structure for *powerset functions* (i.e. functions that assign values to elements of 2^{Ω}) is based on the representation of sets as binary strings, as in [44] and [86], and on the binary tree depicted in Fig. C.1. It is a dynamic powerset binary tree, only storing nodes corresponding to *special sets*. All other set not present in the tree is assumed to be associated with a fixed value (i.e. 0 for a mass function, 1 for a weight function). Thus, special sets include focal sets but may not all be focal sets, as is the case with the lattice support. Each node in the tree contains a boolean value to indicate whether it has been set to a value or not, an eventual *value*, pointers to parent and children, its depth index and a bitset representing an element from 2^{Ω}.

The creation of this tree is incremental, starting with the singleton $\{\omega_1\}$, where $\Omega = [\omega_1, \ldots, \omega_n]$, as root, whether it is a special set or not. The next special set is inserted to its right or left given that it contains ω_1 or not. In fact, for a pair (A, *value*), where A is a special set, this procedure will assign *value* to a node of depth equal to the greatest index in $[\![1, n]\!]$ corresponding to an element of A. This node is found following the binary code that represents A to navigate the tree until the last element index is encountered in A, i.e. until we reach the last bit set to 1 in A. So, for some other special set B, if B contains all elements of A, in addition to other ones including one of greater maximum index, then B is inserted at the right of A. If B has all elements of A, in addition to other ones not including one greater than the maximum index of A, then B is not on the same branch as A and will be assigned to a node of same depth as A. All supersets of A have a depth equal to or greater than its. Of course, this also means that all subsets of A have a depth equal to or less than its. Moreover, sets that are at the left of A are not contained in A and do not contain A, since they have all elements of A but the one of greatest index, in addition to others of greater index than the maximum index of A. Furthermore, if B diverges with A before its depth, it may be necessary to create a *disjunction*. A disjunction node (i.e. a node that does not hold any value) is inserted to split the branch in two at the depth equal to the first element index not common to both A and B. The one that does not contain it will be placed at the left of this disjunction node, and the one that does will be placed at its right. This behavior is illustrated in Fig. C.2.

Algorithm 2: Computation of the focal points associated with $(S, \leq)$, where $\leq \in \{\subseteq, \supseteq\}$.

Input: S, $\leq$
Output: F, F_{Map}
$F \leftarrow S$;
$F_{\text{Map}} \leftarrow$ Hashmap<bitset, float>;
for $i = 1$ **to** $|S|$ **do**
  $F_{\text{Map}}[S[i]] \leftarrow 0$;
if $\leq = \subseteq$ **then**
  `// | is the bitwise OR operator`
  $\cdot \leftarrow |$;
else
  `// & is the bitwise AND operator`
  $\cdot \leftarrow \&$;
for $i = 1$ **to** $|S|$ **do**
  for $ii = i + 1$ **to** $|F|$ **do**
    $A \leftarrow F[ii] \cdot S[i]$;
    if $A \notin F_{Map}.keys()$ **then**
      append A to F;
      $F_{\text{Map}}[A] \leftarrow 0$;

A similar binary tree for powerset functions has been proposed in [44]. It is illustrated in Fig. C.3. While both our binary tree and theirs are dynamic, theirs does not exploit depths and requires to store $2F - 1$ elements, where F is the number of special elements to store. Ours needs to store at most $F + \frac{F}{2}$ elements, and only F if every disjunction node between two special sets is also a special set (or if there is simply no disjunction between special sets, e.g. in the consonant case). Furthermore, it features interesting properties like the fact that the search for all singletons and some sub-FODs can be restrained respectiveley to the left and right chains of nodes. As mentioned above, having depths allows us to exploit the fact that subsets can only be of lower depth, while supersets can only be of greater depth. This form of powerset binary tree also speeds up the search for any value since we have to check at most n booleans to find a value, where $n \in [1, |\Omega|]$, while the version of [44] always have to check n booleans.

Recently, another similar structure to ours has been proposed in [87, 88]. However, their data structure is not dynamic, i.e. it stores all subsets instead of only special sets. Doing so, they do not have to store a binary string in each node, but they always have an overall exponential spatial complexity (and of course, a time complexity at least exponential accordingly).

In the case of an infinite FOD, the idea is to represent it as an ever changing FOD containing a special element ω_∞ as first element that symbolizes the *rest* of the FOD. As we know that ω_∞ will always be an element of this FOD, having $\{\omega_\infty\}$ as the root node of its powerset binary tree reduces the number of operations of reorganization after addition or removal of any FOD element.

C.2 Procedures

C.2.1 Computation of focal points

In this section, we present algorithms computing all focal points, given focal sets.

General procedure

Here, we apply Property 2.3.4.1 to the case where $P = 2^\Omega$, leading to Algorithm 2. For a set $S \subseteq 2^\Omega$, e.g. $\text{supp}(f)$, this algorithm computes its focal points ${}^oS$, where $o \in \{\wedge, \vee\}$ with a complexity less than $O(|S| \,.\, |{}^oS|)$ in time and $O(|{}^oS|)$ in space.

Linear analysis

There are some cases in which a linear pre-analysis of complexity $O(|S|)$ both in time and space is sufficient to find all focal points. This analysis focuses on the progressive union of all focal sets in a linear run. Formally, let $S\backslash\{\Omega\} = \{A_1, A_2, \ldots, A_K\}$ and $S\backslash\{\Omega\}_k = \{A_1, A_2, \ldots, A_k\}$. This analysis focuses on $I_i = U_{i-1} \cap A_i$ such that $i \in [\![2, K]\!]$, and $U_{i-1} = \bigcup S\backslash\{\Omega\}_{i-1}$ and $I_1 = A_1$. More precisely, we have

$$\begin{aligned} I_i &= U_{i-1} \cap A_i \\ &= (\textstyle\bigcup S\backslash\{\Omega\}_{i-1}) \cap A_i \\ &= \textstyle\bigcup_{F \in \{A_1, \ldots, A_{i-1}\}} (F \cap A_i)\,. \end{aligned}$$

In other words, these intersections I_i contain all focal points based on pairs of focal sets.

In most cases, without testing each pair of focal sets, we cannot know which focal points are generated based solely on I_i since several combinations of sets in 2^{I_i} can lead to I_i as union. For example, if $\Omega = \{a, b, c, d\}$ and $S\backslash\{\Omega\} = \{\{a, b\}, \{a, c\}, \{b, c, d\}\}$, then $U_2 = \{a, b, c\}$ and $I_3 = \{a, b, c\} \cap \{b, c, d\} = \{b, c\}$. Yet, $\{b, c\}$ is not a focal point. It is the result of the union of the focal points $\{a, b\} \cap \{b, c, d\} = \{b\}$ and $\{a, c\} \cap \{b, c, d\} = \{c\}$.

However, there are two special cases in which we know that I_i is a focal point: (a) if $|I_i| = 0$, then $2^{I_i} = \{I_i\}$, and (b) if $|I_i| = 1$, then $2^{I_i} = \{I_i, \emptyset\}$.

So, if $\forall i \in [\![2, K]\!]$, $|I_i| = 0$, then $\forall F_1, F_2 \in S\backslash\{\Omega\}$, $F_1 \cap F_2 = \emptyset$. This is the *quasi-Bayesian* case that has already been treated in Proposition 1 of [23]. By definition of this case, we have $\forall F \in S\backslash\{\emptyset, \Omega\}$, $F^{\uparrow^\wedge S} = \{\Omega\}$ and the only possible focal point other than the focal sets is $\emptyset$, i.e. ${}^\wedge S\backslash\{\emptyset\} = S\backslash\{\emptyset\}$. This means that for any powerset function f such that $\text{supp}(f) \subseteq S$, the computation of its zeta and Möbius transforms in $(2^\Omega, \supseteq)$ is always $O(|\text{supp}(f)|)$, where $|\text{supp}(f)| \leq |\Omega| + 2$.

In addition, this analysis points to a slightly more general case that contains the quasi-Bayesian one in which $\forall i \in [\![2, K]\!]$, $|I_i| \leq 1$. Indeed, when $|I_i| = 1$, we have $2^{I_i} = \{I_i, \emptyset\}$, which means that all focal points composing I_i are in $\{I_i, \emptyset\}$ and at least one of them is the singleton I_i, since $\emptyset$ is the neutral element for the union of sets. Also, the only new focal point that could be generated based on the intersection of the singleton I_i with another focal point is $\emptyset$.

Algorithm 3 sums up this procedure for ${}^\wedge S$. This linear analysis can be performed for the dual order $\subseteq$ as well, focusing on $S\backslash\{\emptyset\} = \{A_1, A_2, \ldots, A_K\}$, with $\overline{I_i} = \overline{U_{i-1}} \cup A_i$ such that $i \in [\![2, K]\!]$, and $\overline{U_{i-1}} = \bigcap S\backslash\{\emptyset\}_{i-1}$ and $\overline{I_1} = A_1$. This dual procedure is provided by Algorithm 4.

Algorithm 3: Linear computation of $^{\wedge}S$ based on S.

Input: $S\backslash\{\Omega\} = \{A_1, A_2, \ldots, A_K\}$
Output: $^{\wedge}S$, *is_almost_bayesian*
$^{\wedge}S \leftarrow S$;
$U \leftarrow A_1$;
is_almost_bayesian $\leftarrow$ True;
for $i = 2$ **to** K **do**
    $I \leftarrow U \cap A_i$;
    if $|I| > 1$ **then**
        is_almost_bayesian $\leftarrow$ False;
        break;
    else if $|I| = 1$ **then**
        if $I \notin {}^{\wedge}S$ **then**
            append I to $^{\wedge}S$;
    if $\emptyset \notin {}^{\wedge}S$ **then**
        append $\emptyset$ to $^{\wedge}S$;
    $U \leftarrow U \cup A_i$;

Algorithm 4: Linear computation of $^{\vee}S$ based on S.

Input: $S\backslash\{\emptyset\} = \{A_1, A_2, \ldots, A_K\}$
Output: $^{\vee}S$, *is_almost_bayesian*
$^{\vee}S \leftarrow S$;
$U \leftarrow A_1$;
is_almost_bayesian $\leftarrow$ True;
for $i = 2$ **to** K **do**
    $I \leftarrow U \cup A_i$;
    if $|I| < |\Omega| - 1$ **then**
        is_almost_bayesian $\leftarrow$ False;
        break;
    else if $|I| = |\Omega| - 1$ **then**
        if $I \notin {}^{\vee}S$ **then**
            append I to $^{\vee}S$;
    if $\Omega \notin {}^{\vee}S$ **then**
        append Ω to $^{\vee}S$;
    $U \leftarrow U \cap A_i$;

C.2.2 Computation of iota elements

Computation of the iota elements (See Proposition 3.3.1.1) and dual iota elements (See Proposition 3.3.1.2) of S are presented respectively in Algorithms 5 and 6.

Algorithm 5: Computation of $\iota(S)$.

Input: S
Output: I
foreach $\omega \in \Omega$ **do**
  $i \leftarrow \Omega$;
  foreach $F \in S$, *such that* $F \supseteq \{\omega\}$ **do**
    included $\leftarrow$ True;
    $i \leftarrow i \cap F$;
    if $i = \{\omega\}$ **then**
      break;
  if *included* **then**
    append i to I;

Algorithm 6: Computation of $\bar{\iota}(S)$.

Input: S
Output: I
foreach $\omega \in \Omega$ **do**
  $i \leftarrow \emptyset$;
  foreach $F \in S$, *such that* $F \subseteq \Omega \backslash \{\omega\}$ **do**
    included $\leftarrow$ True;
    $i \leftarrow i \cup F$;
    if $i = \Omega \backslash \{\omega\}$ **then**
      break;
  if *included* **then**
    append i to I;

C.2.3 Computation of the lattice support

Computation of the upper and lower closure in the lattice support of S (See Proposition 3.3.1.3) are presented respectively in Algorithms 7 and 8.

C.2.4 Computation of DST transformations in the consonant case

The consonant case has already been treated in Proposition 2 of [23]. A consonant structure of evidence is a nested structure where each focal set is contained in every focal set of greater or equal cardinality. Formally, $\forall F_i, F_j \in \mathcal{F}$, if $|F_i| \leq |F_j|$ then $F_i \subseteq F_j$. By definition, there can be only one focal set of each cardinality in $[\![0, |\Omega|]\!]$, each of them being a subset of every focal set of greater cardinality. This means that there can be no focal point other than focal sets and that each focal point has a proxy focal point. Let us note $|F_1| < |F_2| < \cdots < |F_K|$ the focal elements in S. To check if this structure is consonant, we simply have to check that $\forall i \in [\![1, K-1]\!]$, $F_i \subset F_{i+1}$. This analysis is performed by Algorithm 9.

Algorithm 7: Computation of $S^{\uparrow^{\mathcal{L}}S}$ based on $\iota(S)$ and S.

Input: $\iota(S)$, S
Output: L
$L \leftarrow S$;
foreach $i \in \iota(S)$ **do**
  foreach $A \in L$ **do**
    $B \leftarrow A \cup i$;
    if $B \notin L$ **then**
      append B to L;

Algorithm 8: Computation of $S^{\downarrow^{\mathcal{L}}S}$ based on $\overline{\iota}(S)$ and S.

Input: $\overline{\iota}(S)$, S
Output: L
$L \leftarrow S$;
foreach $i \in \overline{\iota}(S)$ **do**
  foreach $A \in L$ **do**
    $B \leftarrow A \cap i$;
    if $B \notin L$ **then**
      append B to L;

Algorithms 10 to 17 present procedures in the consonant case to compute the following transformations:

- m to b,
- b to m,
- m to q,
- q to m,
- b to v,
- v to b,
- q to w,
- w to q.

Their complexity in time is $\min\left[O(|\Omega|), O\left(|S|.\log\left(|S|\right)\right)\right]$ (which corresponds to the complexity of the sorting algorithm used on S). Their space complexity is $O(|S|)$.

Algorithm 9: Consonance check.

Input: S
Output: *is_consonant*
sort S such that $S = \{F_1, F_2, \ldots, F_K\}$, where $|F_1| \leq |F_2| \leq \cdots \leq |F_K|$;
is_consonant $\leftarrow$ True;
for $i \in [\![1, K-1]\!]$ **do**
  if $F_i \not\subseteq F_{i+1}$ **then**
    is_consonant $\leftarrow$ False;
    break;

C.2.5 Computation of DST transformations in a semilattice

Algorithms 18 to 25 present procedures using ${}^{\wedge}S$ or ${}^{\vee}S$ to compute the following transformations:

- m to b,
- b to m,
- m to q,
- q to m,
- b to v,
- v to b,
- q to w,
- w to q.

They all use the sequences of graphs of Theorem 3.3.2.2 and Corollary 3.3.2.7. Their complexity in time is $O(I(S).|^oS|.\epsilon)$, where $I \in \{\iota, \bar{\iota}\}$ and $o \in \{\wedge, \vee\}$ and where ϵ represents the average number of operations required to "bridge a gap" (See Theorem 3.3.2.2). This is always less than $O(|\Omega|.2^{\Omega})$. Their space complexity is $O(|^oS|)$.

C.2.6 Computation of DST transformations in a lattice

The general procedure to compute ^oS is less than $O(|S|.|^oS|)$. But, if one wishes to be certain that our algorithms are at least as efficient as the FMT, i.e. at most $O(|\Omega|.2^{|\Omega|})$, then one has to look at $|S|$. Indeed, if $O(|S|) \leq O(|\Omega|)$ (e.g. $|S| < 10.|\Omega|$), then $O(|S|.|^oS|) \leq O(|\Omega|.2^{|\Omega|})$. Otherwise, one can rely on the lattice support $^{\mathcal{L}}S$, since $\iota(S)$ can be computed in less than $O(|\Omega|.|S|)$ with Algorithm 5 and is then used to compute ^oS in $O(\iota(S).|^{\mathcal{L}}S|)$ in Algorithms 7 and 8, where $|\iota(S)| \leq |\Omega|$.

Algorithms 26 to 33 present procedures using $^{\mathcal{L}}S$ (more precisely, the upper and lower closures $S^{\uparrow^{\mathcal{L}}S}$ and $S^{\downarrow^{\mathcal{L}}S}$) to compute the following transformations:

- m to b,
- b to m,
- m to q,
- q to m,
- b to v,
- v to b,
- q to w,
- w to q.

They all use the sequences of graphs of Theorem 3.3.2.1 and its corollaries. Their complexity in time is less than $O(I(S).|^{\mathcal{L}}S|)$, where $I \in \{\iota, \bar{\iota}\}$ and $o \in \{\wedge, \vee\}$, which is always less than $O(|\Omega|.2^{|\Omega|})$. Their space complexity is less than $O(|^{\mathcal{L}}S|)$.

C.2.7 Computation of DST transformations independently from Ω

If $\text{supp}(f)$ is almost Bayesian or if $|\Omega|$ happens to be considerable to the point that one would like to compute the previous DST transformations independently from $|\Omega|$, Algorithms 34 to 41 present procedures of time complexities in $[O(|^oS|), O(|^oS|^2)]$. Their complexity in space is $O(|^oS|)$. They present procedures for the following transformations:

- m to b,
- b to m,
- m to q,
- q to m,
- b to v,
- v to b,
- q to w,
- w to q.

Fig. C.4 offers a decision tree to help the reader in choosing the right algorithm for their case. Of course, this decision tree can be implemented as an algorithm in order to create a unique general procedure automatically choosing the type of algorithm to use, no matter the DST transformation.

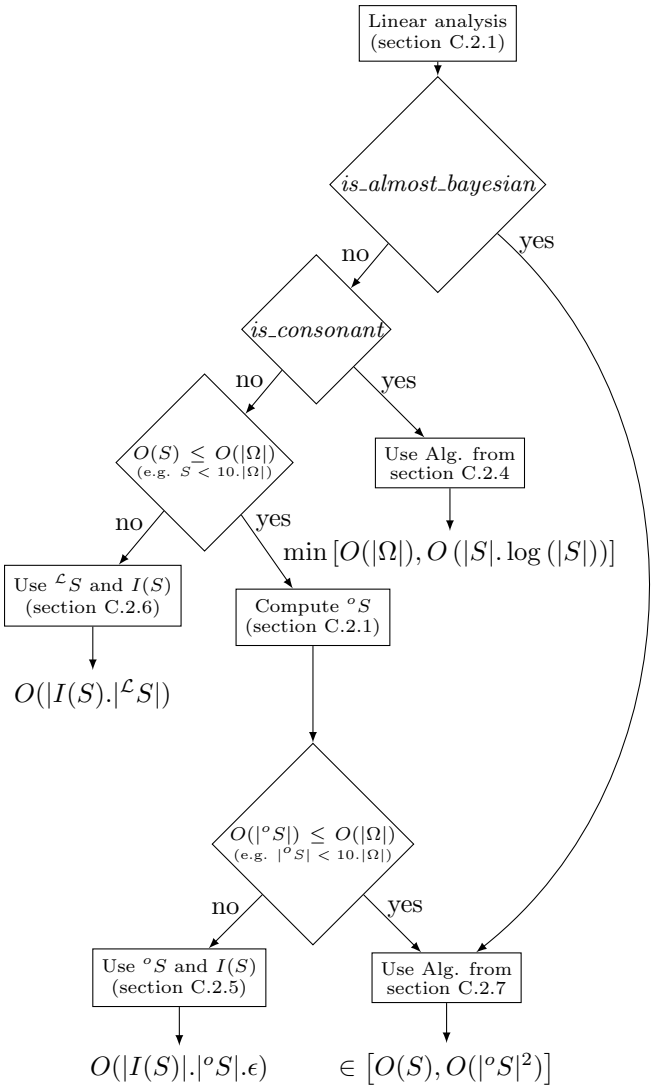


Figure C.4: Decision tree for the choice of which algorithms of section C to use. Of course, this decision tree can be implemented as an algorithm in order to create a unique general procedure automatically choosing the type of algorithm to use, no matter the DST transformation. Diamond nodes represent Boolean tests. Rectangle nodes represent the chosen action. Finally, terminal nodes (leaves) indicate the final time complexity of the whole procedure, including the computation of any DST transformation, where $o \in \{\wedge, \vee\}$ and $I \in \{\iota, \bar{\iota}\}$. In particular, when *is_almost_Bayesian* is true, the complexity of the whole procedure is $O(S)$. Notice that all these complexities are less than $O(|\Omega|.2^{|\Omega|})$.

Algorithm 10: Computation of $\{b(B) \ / \ B \in S \ \}$ based on $\{m(B) \ / \ B \in S \ \}$ in the consonant case.

Input: $\{m(B) \ / \ B \in S \ \}$, *is_consonant*, $S = \{F_1, F_2, \ldots, F_K\}$, where $|F_1| \leq |F_2| \leq \cdots \leq |F_K|$

Output: $\{b(B) \ / \ B \in S \ \}$

if *is_consonant* **then**
- $b(F_1) \leftarrow m(F_1)$;
- **for** $i = 2$ **to** K **do**
  - $b(F_i) \leftarrow m(F_i) + b(F_{i-1})$;

Algorithm 11: Computation of $\{m(B) \ / \ B \in S \ \}$ based on $\{b(B) \ / \ B \in S \ \}$ in the consonant case.

Input: $\{b(B) \ / \ B \in S \ \}$, *is_consonant*, $S = \{F_1, F_2, \ldots, F_K\}$, where $|F_1| \leq |F_2| \leq \cdots \leq |F_K|$

Output: $\{m(B) \ / \ B \in S \ \}$

if *is_consonant* **then**
- $m(F_1) \leftarrow b(F_1)$;
- **for** $i = 2$ **to** K **do**
  - $m(F_i) \leftarrow b(F_i) - b(F_{i-1})$;

Algorithm 12: Computation of $\{q(B) \ / \ B \in S \ \}$ based on $\{m(B) \ / \ B \in S \ \}$ in the consonant case.

Input: $\{m(B) \ / \ B \in S \ \}$, *is_consonant*, $S = \{F_1, F_2, \ldots, F_K\}$, where $|F_1| \leq |F_2| \leq \cdots \leq |F_K|$

Output: $\{q(B) \ / \ B \in S \ \}$

if *is_consonant* **then**
- $q(F_K) \leftarrow m(F_K)$;
- **for** $i = K - 1$ **to** 1 **do**
  - $q(F_i) \leftarrow m(F_i) + q(F_{i+1})$;

Algorithm 13: Computation of $\{m(B) \ / \ B \in S \ \}$ based on $\{q(B) \ / \ B \in S \ \}$ in the consonant case.

Input: $\{q(B) \ / \ B \in S \ \}$, *is_consonant*, $S = \{F_1, F_2, \ldots, F_K\}$, where $|F_1| \leq |F_2| \leq \cdots \leq |F_K|$

Output: $\{m(B) \ / \ B \in S \ \}$

if *is_consonant* **then**
- $m(F_K) \leftarrow q(F_K)$;
- **for** $i = K - 1$ **to** 1 **do**
  - $m(F_i) \leftarrow q(F_i) - q(F_{i+1})$;

Algorithm 14: Computation of $\{v(B) \ / \ B \in S \ \}$ based on $\{b(B) \ / \ B \in S \ \}$ in the consonant case.

Input: $\{b(B) \ / \ B \in S \ \}$, *is_consonant*, $S = \{F_1, F_2, \ldots, F_K\}$, where $|F_1| \leq |F_2| \leq \cdots \leq |F_K|$

Output: $\{v(B) \ / \ B \in S \ \}$

if *is_consonant* **then**

  $v(F_1) \leftarrow b(F_1)^{-1}$;

  for $i = 2$ **to** K **do**

    $v(F_i) \leftarrow b(F_i)^{-1}.\ b(F_{i-1})$;

Algorithm 15: Computation of $\{b(B) \ / \ B \in S \ \}$ based on $\{v(B) \ / \ B \in S \ \}$ in the consonant case.

Input: $\{v(B) \ / \ B \in S \ \}$, *is_consonant*, $S = \{F_1, F_2, \ldots, F_K\}$, where $|F_1| \leq |F_2| \leq \cdots \leq |F_K|$

Output: $\{b(B) \ / \ B \in S \ \}$

if *is_consonant* **then**

  $b(F_1) \leftarrow v(F_1)^{-1}$;

  for $i = 2$ **to** K **do**

    $b(F_i) \leftarrow v(F_i)^{-1}.\ b(F_{i-1})$;

Algorithm 16: Computation of $\{w(B) \ / \ B \in S \ \}$ based on $\{q(B) \ / \ B \in S \ \}$ in the consonant case.

Input: $\{q(B) \ / \ B \in S \ \}$, *is_consonant*, $S = \{F_1, F_2, \ldots, F_K\}$, where $|F_1| \leq |F_2| \leq \cdots \leq |F_K|$

Output: $\{w(B) \ / \ B \in S \ \}$

if *is_consonant* **then**

  $w(F_K) \leftarrow q(F_K)^{-1}$;

  for $i = K - 1$ **to** 1 **do**

    $w(F_i) \leftarrow q(F_i)^{-1}.\ q(F_{i+1})$;

Algorithm 17: Computation of $\{q(B) \ / \ B \in S \ \}$ based on $\{w(B) \ / \ B \in S \ \}$ in the consonant case.

Input: $\{w(B) \ / \ B \in S \ \}$, *is_consonant*, $S = \{F_1, F_2, \ldots, F_K\}$, where $|F_1| \leq |F_2| \leq \cdots \leq |F_K|$

Output: $\{q(B) \ / \ B \in S \ \}$

if *is_consonant* **then**

  $q(F_K) \leftarrow w(F_K)^{-1}$;

  for $i = K - 1$ **to** 1 **do**

    $q(F_i) \leftarrow w(F_i)^{-1}.\ q(F_{i+1})$;

Algorithm 18: Computation of $\{b(B) \ / \ B \in {}^{\vee}S \ \}$ based on $\{m(B) \ / \ B \in {}^{\vee}S \ \}$.

Input: $\{m(B) \ / \ B \in {}^{\vee}S \ \}$, ${}^{\vee}S$, $\bar{\imath}(S)$
Output: $\{b(B) \ / \ B \in {}^{\vee}S \ \}$
sort $\bar{\imath}(S)$ such that $\bar{\imath}(S) = \{i_1, i_2, \ldots, i_K\}$, where $|i_1| \leq |i_2| \leq \cdots \leq |i_K|$;
foreach $A \in {}^{\vee}S$ **do**
    $b(A) \leftarrow m(A)$;
$\Omega_{cum} \leftarrow \Omega$;
for $k = K$ **to** 1 **do**
    $\Omega_{cum} \leftarrow \Omega_{cum} \cap i_k$;
    foreach $A \in {}^{\vee}S$ **do**
        $B \leftarrow A \cap i_k$;
        if $B \neq A$ **then**
            $X \leftarrow \underset{C \in B^{\downarrow {}^{\vee}S}}{\operatorname{argmax}} (|C|)$;
            if $X \neq$ *NULL* ***and*** $X \supseteq A \cap \Omega_{cum}$ **then**
                $b(A) \leftarrow b(A) + b(X)$;

Algorithm 19: Computation of $\{m(B) \ / \ B \in {}^{\vee}S \ \}$ based on $\{b(B) \ / \ B \in {}^{\vee}S \ \}$.

Input: $\{b(B) \ / \ B \in {}^{\vee}S \ \}$, ${}^{\vee}S$, $\bar{\imath}(S)$
Output: $\{m(B) \ / \ B \in {}^{\vee}S \ \}$
sort $\bar{\imath}(S)$ such that $\bar{\imath}(S) = \{i_1, i_2, \ldots, i_K\}$, where $|i_1| \leq |i_2| \leq \cdots \leq |i_K|$;
foreach $A \in {}^{\vee}S$ **do**
    $m(A) \leftarrow b(A)$;
$\Omega_{cum} \leftarrow \Omega$;
for $k = 1$ **to** K **do**
    $\Omega_{cum} \leftarrow \Omega_{cum} \cap i_k$;
    foreach $A \in {}^{\vee}S$ **do**
        $B \leftarrow A \cap i_k$;
        if $B \neq A$ **then**
            $X \leftarrow \underset{C \in B^{\downarrow {}^{\vee}S}}{\operatorname{argmax}} (|C|)$;
            if $X \neq$ *NULL* ***and*** $X \supseteq A \cap \Omega_{cum}$ **then**
                $m(A) \leftarrow m(A) - m(X)$;

Algorithm 20: Computation of $\{q(B) \ / \ B \in {}^{\wedge}S\ \}$ based on $\{m(B) \ / \ B \in {}^{\wedge}S\ \}$.

Input: $\{m(B) \ / \ B \in {}^{\wedge}S\ \}$, ${}^{\wedge}S$, $\iota(S)$
Output: $\{q(B) \ / \ B \in {}^{\wedge}S\ \}$
sort $\iota(S)$ such that $\iota(S) = \{i_1, i_2, \ldots, i_K\}$, where $|i_1| \leq |i_2| \leq \cdots \leq |i_K|$;
foreach $A \in {}^{\wedge}S$ **do**
  $q(A) \leftarrow m(A)$;
$\Omega_{cum} \leftarrow \emptyset$;
for $k = 1$ **to** K **do**
  $\Omega_{cum} \leftarrow \Omega_{cum} \cup i_k$;
  foreach $A \in {}^{\wedge}S$ **do**
    $B \leftarrow A \cup i_k$;
    if $B \neq A$ **then**
      $X \leftarrow \underset{C \in B^{\uparrow \wedge S}}{\text{argmin}}\ (|C|)$;
      if $X \neq NULL$ ***and*** $X \subseteq A \cup \Omega_{cum}$ **then**
        $q(A) \leftarrow q(A) + q(X)$;

Algorithm 21: Computation of $\{m(B) \ / \ B \in {}^{\wedge}S\ \}$ based on $\{q(B) \ / \ B \in {}^{\wedge}S\ \}$.

Input: $\{q(B) \ / \ B \in {}^{\wedge}S\ \}$, ${}^{\wedge}S$, $\iota(S)$
Output: $\{m(B) \ / \ B \in {}^{\wedge}S\ \}$
sort $\iota(S)$ such that $\iota(S) = \{i_1, i_2, \ldots, i_K\}$, where $|i_1| \leq |i_2| \leq \cdots \leq |i_K|$;
foreach $A \in {}^{\wedge}S$ **do**
  $m(A) \leftarrow q(A)$;
$\Omega_{cum} \leftarrow \emptyset$;
for $k = K$ **to** 1 **do**
  $\Omega_{cum} \leftarrow \Omega_{cum} \cup i_k$;
  foreach $A \in {}^{\wedge}S$ **do**
    $B \leftarrow A \cup i_k$;
    if $B \neq A$ **then**
      $X \leftarrow \underset{C \in B^{\uparrow \wedge S}}{\text{argmin}}\ (|C|)$;
      if $X \neq NULL$ ***and*** $X \subseteq A \cup \Omega_{cum}$ **then**
        $m(A) \leftarrow m(A) - m(X)$;

Algorithm 22: Computation of $\{v(B) \ / \ B \in {}^{\vee}S \ \}$ based on $\{b(B) \ / \ B \in {}^{\vee}S \ \}$.

Input: $\{b(B) \ / \ B \in {}^{\vee}S \ \}$, ${}^{\vee}S$, $\bar{\iota}(S)$
Output: $\{v(B) \ / \ B \in {}^{\vee}S \ \}$
sort $\bar{\iota}(S)$ such that $\bar{\iota}(S) = \{i_1, i_2, \ldots, i_K\}$, where $|i_1| \leq |i_2| \leq \cdots \leq |i_K|$;
foreach $A \in {}^{\vee}S$ **do**
  $v(A) \leftarrow b(A)^{-1}$;
$\Omega_{cum} \leftarrow \Omega$;
for $k = 1$ **to** K **do**
  $\Omega_{cum} \leftarrow \Omega_{cum} \cap i_k$;
  foreach $A \in {}^{\vee}S$ **do**
    $B \leftarrow A \cap i_k$;
    if $B \neq A$ **then**
      $X \leftarrow \underset{C \in B^{\downarrow {}^{\vee}S}}{\operatorname{argmax}} (|C|)$;
      if $X \neq$ *NULL* ***and*** $X \supseteq A \cap \Omega_{cum}$ **then**
        $v(A) \leftarrow v(A)/v(X)$;

Algorithm 23: Computation of $\{b(B) \ / \ B \in {}^{\vee}S \ \}$ based on $\{v(B) \ / \ B \in {}^{\vee}S \ \}$.

Input: $\{v(B) \ / \ B \in {}^{\vee}S \ \}$, ${}^{\vee}S$, $\bar{\iota}(S)$
Output: $\{b(B) \ / \ B \in {}^{\vee}S \ \}$
sort $\bar{\iota}(S)$ such that $\bar{\iota}(S) = \{i_1, i_2, \ldots, i_K\}$, where $|i_1| \leq |i_2| \leq \cdots \leq |i_K|$;
foreach $A \in {}^{\vee}S$ **do**
  $b(A) \leftarrow v(A)^{-1}$;
$\Omega_{cum} \leftarrow \Omega$;
for $k = K$ **to** 1 **do**
  $\Omega_{cum} \leftarrow \Omega_{cum} \cap i_k$;
  foreach $A \in {}^{\vee}S$ **do**
    $B \leftarrow A \cap i_k$;
    if $B \neq A$ **then**
      $X \leftarrow \underset{C \in B^{\downarrow {}^{\vee}S}}{\operatorname{argmax}} (|C|)$;
      if $X \neq$ *NULL* ***and*** $X \supseteq A \cap \Omega_{cum}$ **then**
        $b(A) \leftarrow b(A).b(X)$;

Algorithm 24: Computation of $\{w(B) \ / \ B \in {}^{\wedge}S \ \}$ based on $\{q(B) \ / \ B \in {}^{\wedge}S \ \}$.

Input: $\{q(B) \ / \ B \in {}^{\wedge}S \ \}$, ${}^{\wedge}S$, $\iota(S)$
Output: $\{w(B) \ / \ B \in {}^{\wedge}S \ \}$
sort $\iota(S)$ such that $\iota(S) = \{i_1, i_2, \ldots, i_K\}$, where $|i_1| \leq |i_2| \leq \cdots \leq |i_K|$;
foreach $A \in {}^{\wedge}S$ **do**
    $w(A) \leftarrow q(A)^{-1}$;
$\Omega_{cum} \leftarrow \emptyset$;
for $k = K$ **to** 1 **do**
    $\Omega_{cum} \leftarrow \Omega_{cum} \cup i_k$;
    foreach $A \in {}^{\wedge}S$ **do**
        $B \leftarrow A \cup i_k$;
        if $B \neq A$ **then**
            $X \leftarrow \underset{C \in B^{\uparrow \wedge S}}{\text{argmin}} (|C|)$;
            if $X \neq NULL$ ***and*** $X \subseteq A \cup \Omega_{cum}$ **then**
                $w(A) \leftarrow w(A)/w(X)$;

Algorithm 25: Computation of $\{q(B) \ / \ B \in {}^{\wedge}S \ \}$ based on $\{w(B) \ / \ B \in {}^{\wedge}S \ \}$.

Input: $\{w(B) \ / \ B \in {}^{\wedge}S \ \}$, ${}^{\wedge}S$, $\iota(S)$
Output: $\{q(B) \ / \ B \in {}^{\wedge}S \ \}$
sort $\iota(S)$ such that $\iota(S) = \{i_1, i_2, \ldots, i_K\}$, where $|i_1| \leq |i_2| \leq \cdots \leq |i_K|$;
foreach $A \in {}^{\wedge}S$ **do**
    $q(A) \leftarrow w(A)^{-1}$;
$\Omega_{cum} \leftarrow \emptyset$;
for $k = 1$ **to** K **do**
    $\Omega_{cum} \leftarrow \Omega_{cum} \cup i_k$;
    foreach $A \in {}^{\wedge}S$ **do**
        $B \leftarrow A \cup i_k$;
        if $B \neq A$ **then**
            $X \leftarrow \underset{C \in B^{\uparrow \wedge S}}{\text{argmin}} (|C|)$;
            if $X \neq NULL$ ***and*** $X \subseteq A \cup \Omega_{cum}$ **then**
                $q(A) \leftarrow q(A).q(X)$;

Algorithm 26: Computation of $\{b(B) \ / \ B \in S^{\downarrow^{\mathcal{L}}S}\}$ based on $\{m(B) \ / \ B \in S^{\downarrow^{\mathcal{L}}S}\}$.

Input: $\{m(B) \ / \ B \in S^{\downarrow^{\mathcal{L}}S}\}$, $S^{\downarrow^{\mathcal{L}}S}$, $\bar{\iota}(S)$
Output: $\{b(B) \ / \ B \in S^{\downarrow^{\mathcal{L}}S}\}$
sort $\bar{\iota}(S)$ such that $\bar{\iota}(S) = \{i_1, i_2, \ldots, i_K\}$, where $|i_1| \leq |i_2| \leq \cdots \leq |i_K|$;
foreach $A \in S^{\downarrow^{\mathcal{L}}S}$ **do**
  $b(A) \leftarrow m(A)$;
for $k = K$ **to** 1 **do**
  foreach $A \in S^{\downarrow^{\mathcal{L}}S}$ **do**
    $B \leftarrow A \cap i_k$;
    if $B \neq A$ **then**
      $b(A) \leftarrow b(A) + b(B)$;

Algorithm 27: Computation of $\{m(B) \ / \ B \in S^{\downarrow^{\mathcal{L}}S}\}$ based on $\{b(B) \ / \ B \in S^{\downarrow^{\mathcal{L}}S}\}$.

Input: $\{b(B) \ / \ B \in S^{\downarrow^{\mathcal{L}}S}\}$, $S^{\downarrow^{\mathcal{L}}S}$, $\bar{\iota}(S)$
Output: $\{m(B) \ / \ B \in S^{\downarrow^{\mathcal{L}}S}\}$
sort $\bar{\iota}(S)$ such that $\bar{\iota}(S) = \{i_1, i_2, \ldots, i_K\}$, where $|i_1| \leq |i_2| \leq \cdots \leq |i_K|$;
foreach $A \in S^{\downarrow^{\mathcal{L}}S}$ **do**
  $m(A) \leftarrow b(A)$;
for $k = 1$ **to** K **do**
  foreach $A \in S^{\downarrow^{\mathcal{L}}S}$ **do**
    $B \leftarrow A \cap i_k$;
    if $B \neq A$ **then**
      $m(A) \leftarrow m(A) - m(B)$;

Algorithm 28: Computation of $\{q(B) \ / \ B \in S^{\uparrow^{\mathcal{L}}S}\}$ based on $\{m(B) \ / \ B \in S^{\uparrow^{\mathcal{L}}S}\}$.

Input: $\{m(B) \ / \ B \in S^{\uparrow^{\mathcal{L}}S}\}$, $S^{\uparrow^{\mathcal{L}}S}$, $\iota(S)$
Output: $\{q(B) \ / \ B \in S^{\uparrow^{\mathcal{L}}S}\}$
sort $\iota(S)$ such that $\iota(S) = \{i_1, i_2, \ldots, i_K\}$, where $|i_1| \leq |i_2| \leq \cdots \leq |i_K|$;
foreach $A \in S^{\uparrow^{\mathcal{L}}S}$ **do**
  $q(A) \leftarrow m(A)$;
for $k = 1$ **to** K **do**
  foreach $A \in S^{\uparrow^{\mathcal{L}}S}$ **do**
    $B \leftarrow A \cup i_k$;
    if $B \neq A$ **then**
      $q(A) \leftarrow q(A) + q(B)$;

Algorithm 29: Computation of $\{m(B) \ / \ B \in S^{\uparrow^{\mathcal{L}}S} \}$ based on $\{q(B) \ / \ B \in S^{\uparrow^{\mathcal{L}}S} \}$.

Input: $\{q(B) \ / \ B \in S^{\uparrow^{\mathcal{L}}S} \}$, $S^{\uparrow^{\mathcal{L}}S}$, $\iota(S)$
Output: $\{m(B) \ / \ B \in S^{\uparrow^{\mathcal{L}}S} \}$
sort $\iota(S)$ such that $\iota(S) = \{i_1, i_2, \ldots, i_K\}$, where $|i_1| \leq |i_2| \leq \cdots \leq |i_K|$;
foreach $A \in S^{\uparrow^{\mathcal{L}}S}$ **do**
  $m(A) \leftarrow q(A)$;
for $k = K$ **to** 1 **do**
  foreach $A \in S^{\uparrow^{\mathcal{L}}S}$ **do**
    $B \leftarrow A \cup i_k$;
    if $B \neq A$ **then**
      $m(A) \leftarrow m(A) - m(B)$;

Algorithm 30: Computation of $\{v(B) \ / \ B \in S^{\downarrow^{\mathcal{L}}S} \}$ based on $\{b(B) \ / \ B \in S^{\downarrow^{\mathcal{L}}S} \}$.

Input: $\{b(B) \ / \ B \in S^{\downarrow^{\mathcal{L}}S} \}$, $S^{\downarrow^{\mathcal{L}}S}$, $\bar{\iota}(S)$
Output: $\{v(B) \ / \ B \in S^{\downarrow^{\mathcal{L}}S} \}$
sort $\bar{\iota}(S)$ such that $\bar{\iota}(S) = \{i_1, i_2, \ldots, i_K\}$, where $|i_1| \leq |i_2| \leq \cdots \leq |i_K|$;
foreach $A \in S^{\downarrow^{\mathcal{L}}S}$ **do**
  $v(A) \leftarrow b(A)^{-1}$;
for $k = 1$ **to** K **do**
  foreach $A \in S^{\downarrow^{\mathcal{L}}S}$ **do**
    $B \leftarrow A \cap i_k$;
    if $B \neq A$ **then**
      $v(A) \leftarrow v(A)/v(B)$;

Algorithm 31: Computation of $\{b(B) \ / \ B \in S^{\downarrow^{\mathcal{L}}S} \}$ based on $\{v(B) \ / \ B \in S^{\downarrow^{\mathcal{L}}S} \}$.

Input: $\{v(B) \ / \ B \in S^{\downarrow^{\mathcal{L}}S} \}$, $S^{\downarrow^{\mathcal{L}}S}$, $\bar{\iota}(S)$
Output: $\{b(B) \ / \ B \in S^{\downarrow^{\mathcal{L}}S} \}$
sort $\bar{\iota}(S)$ such that $\bar{\iota}(S) = \{i_1, i_2, \ldots, i_K\}$, where $|i_1| \leq |i_2| \leq \cdots \leq |i_K|$;
foreach $A \in S^{\downarrow^{\mathcal{L}}S}$ **do**
  $b(A) \leftarrow v(A)^{-1}$;
for $k = K$ **to** 1 **do**
  foreach $A \in S^{\downarrow^{\mathcal{L}}S}$ **do**
    $B \leftarrow A \cap i_k$;
    if $B \neq A$ **then**
      $b(A) \leftarrow b(A).b(B)$;

Algorithm 32: Computation of $\{w(B) \ / \ B \in S^{\uparrow^{\mathcal{L}}S} \}$ based on $\{q(B) \ / \ B \in S^{\uparrow^{\mathcal{L}}S} \}$.

Input: $\{q(B) \ / \ B \in S^{\uparrow^{\mathcal{L}}S} \}$, $S^{\uparrow^{\mathcal{L}}S}$, $\iota(S)$
Output: $\{w(B) \ / \ B \in S^{\uparrow^{\mathcal{L}}S} \}$
sort $\iota(S)$ such that $\iota(S) = \{i_1, i_2, \ldots, i_K\}$, where $|i_1| \leq |i_2| \leq \cdots \leq |i_K|$;
foreach $A \in S^{\uparrow^{\mathcal{L}}S}$ **do**
  $w(A) \leftarrow q(A)^{-1}$;
for $k = K$ **to** 1 **do**
  foreach $A \in S^{\uparrow^{\mathcal{L}}S}$ **do**
    $B \leftarrow A \cup i_k$;
    if $B \neq A$ **then**
      $w(A) \leftarrow w(A)/w(B)$;

Algorithm 33: Computation of $\{q(B) \ / \ B \in S^{\uparrow^{\mathcal{L}}S} \}$ based on $\{w(B) \ / \ B \in S^{\uparrow^{\mathcal{L}}S} \}$.

Input: $\{w(B) \ / \ B \in S^{\uparrow^{\mathcal{L}}S} \}$, $S^{\uparrow^{\mathcal{L}}S}$, $\iota(S)$
Output: $\{q(B) \ / \ B \in S^{\uparrow^{\mathcal{L}}S} \}$
sort $\iota(S)$ such that $\iota(S) = \{i_1, i_2, \ldots, i_K\}$, where $|i_1| \leq |i_2| \leq \cdots \leq |i_K|$;
foreach $A \in S^{\uparrow^{\mathcal{L}}S}$ **do**
  $q(A) \leftarrow w(A)^{-1}$;
for $k = 1$ **to** K **do**
  foreach $A \in S^{\uparrow^{\mathcal{L}}S}$ **do**
    $B \leftarrow A \cup i_k$;
    if $B \neq A$ **then**
      $q(A) \leftarrow q(A).q(B)$;

Algorithm 34: Computation of $\{b(B) \ / \ B \in {}^{\vee}S \}$ based on $\{m(B) \ / \ B \in S \}$ independently from $|\Omega|$.

Input: $\{m(F) \ / \ F \in S \}$
Output: $\{b(B) \ / \ B \in {}^{\vee}S \}$
foreach $B \in {}^{\vee}S$ **do**
  $b(B) \leftarrow m(B)$;
  foreach $F \in S \ / \ F \subset B$ **do**
    $b(B) \leftarrow b(B) + m(F)$;

Algorithm 35: Computation of $\{m(B) \ / \ B \in {}^{\vee}S\}$ based on $\{b(B) \ / \ B \in {}^{\vee}S \ \}$ independently from $|\Omega|$.

Input: $\{b(B) \ / \ B \in {}^{\vee}S \ \}$
Output: $\{m(B) \ / \ B \in {}^{\vee}S \ \}$
sort ${}^{\vee}S$ such that ${}^{\vee}S = \{A_1, A_2, \ldots, A_K\}$, where $|A_1| \leq |A_2| \leq \cdots \leq |A_K|$;
for $i = 1$ **to** K **do**
    $m(A_i) \leftarrow b(A_i)$;
    foreach $B \in {}^{\vee}S \ / \ B \subset A_i$ **do**
        $m(A_i) \leftarrow m(A_i) - m(B)$;

Algorithm 36: Computation of $\{q(B) \ / \ B \in {}^{\wedge}S \ \}$ based on $\{m(B) \ / \ B \in S \ \}$ independently from $|\Omega|$.

Input: $\{m(F) \ / \ F \in S \ \}$
Output: $\{q(B) \ / \ B \in {}^{\wedge}S \ \}$
foreach $B \in {}^{\wedge}S$ **do**
    $q(B) \leftarrow m(B)$;
    foreach $F \in S \ / \ F \supset B$ **do**
        $q(B) \leftarrow q(B) + m(F)$;

Algorithm 37: Computation of $\{m(B) \ / \ B \in {}^{\wedge}S\}$ based on $\{q(B) \ / \ B \in {}^{\wedge}S \ \}$ independently from $|\Omega|$.

Input: $\{q(B) \ / \ B \in {}^{\wedge}S \ \}$
Output: $\{m(B) \ / \ B \in {}^{\wedge}S \ \}$
sort ${}^{\wedge}S$ such that ${}^{\wedge}S = \{A_1, A_2, \ldots, A_K\}$, where $|A_1| \leq |A_2| \leq \cdots \leq |A_K|$;
for $i = K$ **to** 1 **do**
    $m(A_i) \leftarrow q(A_i)$;
    foreach $B \in {}^{\wedge}S \ / \ B \supset A_i$ **do**
        $m(A_i) \leftarrow m(A_i) - m(B)$;

Algorithm 38: Computation of $\{v(B) \ / \ B \in {}^{\vee}S \ \}$ based on $\{b(B) \ / \ B \in {}^{\vee}S \ \}$ independently from $|\Omega|$.

Input: $\{b(B) \ / \ B \in {}^{\vee}S \ \}$, ${}^{\vee}S$
Output: $\{w(B) \ / \ B \in {}^{\vee}S \ \}$
sort ${}^{\vee}S$ such that ${}^{\vee}S = \{A_1, A_2, \ldots, A_K\}$, where $|A_1| \leq |A_2| \leq \cdots \leq |A_K|$;
for $i = 1$ **to** K **do**
    $v(A_i) \leftarrow b(A_i)^{-1}$;
    foreach $B \in {}^{\vee}S \ / \ B \subset A_i$ **do**
        $v(A_i) \leftarrow v(A_i)/v(B)$;

Algorithm 39: Computation of $\{b(B) \ / \ B \in {}^{\vee}S \ \}$ based on $\{v(B) \ / \ B \in {}^{\vee}S \ \}$ independently from $|\Omega|$.

Input: $\{v(B) \ / \ B \in {}^{\vee}S \ \}$, ${}^{\vee}S$
Output: $\{b(B) \ / \ B \in {}^{\vee}S \ \}$
foreach $B \in {}^{\vee}S$ **do**
    $b(B) \leftarrow v(B)^{-1}$;
    foreach $A \in {}^{\vee}S \ / \ A \subset B$ **do**
        $b(B) \leftarrow b(B)/v(A)$;

Algorithm 40: Computation of $\{w(B) \ / \ B \in {}^{\wedge}S \ \}$ based on $\{q(B) \ / \ B \in {}^{\wedge}S \ \}$ independently from $|\Omega|$.

Input: $\{q(B) \ / \ B \in {}^{\wedge}S \ \}$, ${}^{\wedge}S$
Output: $\{w(B) \ / \ B \in {}^{\wedge}S \ \}$
sort ${}^{\wedge}S$ such that ${}^{\wedge}S = \{A_1, A_2, \ldots, A_K\}$, where $|A_1| \leq |A_2| \leq \cdots \leq |A_K|$;
for $i = K$ **to** 1 **do**
    $w(A_i) \leftarrow q(A_i)^{-1}$;
    foreach $B \in {}^{\wedge}S \ / \ B \supset A_i$ **do**
        $w(A_i) \leftarrow w(A_i)/w(B)$;

Algorithm 41: Computation of $\{q(B) \ / \ B \in {}^{\wedge}S \ \}$ based on $\{w(B) \ / \ B \in {}^{\wedge}S \ \}$ independently from $|\Omega|$.

Input: $\{w(B) \ / \ B \in {}^{\wedge}S \ \}$, ${}^{\wedge}S$
Output: $\{q(B) \ / \ B \in {}^{\wedge}S \ \}$
foreach $B \in {}^{\wedge}S$ **do**
    $q(B) \leftarrow w(B)^{-1}$;
    foreach $A \in {}^{\wedge}S \ / \ A \supset B$ **do**
        $q(B) \leftarrow q(B)/w(A)$;

Appendix D

LP-VAE loss

D.1 Minimization of $D_{KL}\left(Q_t(\theta,\phi) \;||\; P_t(\theta)\right)$

Proof. Indeed, we have, for some instant t:

$$
\begin{aligned}
&D_{KL}\left(Q_t(\theta,\phi) \;||\; P(\theta)\right)\\
&= \mathop{\mathbb{E}}_{Z\sim Q_t(\theta,\phi)}\big[\log q_{Z_{1:t}|X_{1:t}}(\cdot|x_{1:t};\phi) + \log p_{Z_{t+1:T}|Z_t}(\cdot|\cdot\,;\theta)\\
&\quad - \log p_{Z_{1:t}}(Z_{1:t};\theta) - \log p_{Z_{t+1:T}|Z_t}(Z_{t+1:T}|Z_t;\theta)\\
&\quad - \log p_{X,Y|Z}(x,y|Z;\theta)\big]\\
&= \mathop{\mathbb{E}}_{Z\sim Q_t(\theta,\phi)}\big[\log q_{Z_{1:t}|X_{1:t}}(\cdot|x_{1:t};\phi)\\
&\quad - \log p_{Z_{1:t}}(Z_{1:t};\theta)\\
&\quad - \log p_{X|Z}(x|Z;\theta) - \log p_{Y|X,Z}(y|x,Z;\theta)\big]\\
&= \mathop{\mathbb{E}}_{Z\sim Q_t(\theta,\phi)}\big[\log q_{Z_{1:t}|X_{1:t}}(\cdot|x_{1:t};\phi)\\
&\quad - \log p_{Z_{1:t}}(Z_{1:t};\theta) - \log p_{X_{1:t}|Z_{1:t}}(x_{1:t}|Z_{1:t};\theta)\\
&\quad - \log p_{X_{t+1:T}|Z_{t+1:T}}(x_{t+1:T}|Z_{t+1:T};\theta) - \log p_{Y|X,Z}(y|x,Z;\theta)\big]\\
&= \mathop{\mathbb{E}}_{Z\sim Q_t(\theta,\phi)}\big[\log q_{Z_{1:t}|X_{1:t}}(\cdot|x_{1:t};\phi)\\
&\quad - \log p_{X_{1:t},Z_{1:t}}(X_{1:t},Z_{1:t};\theta)\\
&\quad - \log p_{X_{t+1:T}|Z_{t+1:T}}(x_{t+1:T}|Z_{t+1:T};\theta) - \log p_{Y|X,Z}(y|x,Z;\theta)\big]
\end{aligned}
$$

Suppose that both $p_{X_{t+1:T}|Z_{t+1:T}}(x_{t+1:T}|Z_{t+1:T};\theta)$ and
$p_{Y|X,Z}(y|x,Z;\theta)$ range in $[0,1]$. This can be easily verified if they can be written as a factorization of probability density functions that each ranges in $[0,1]$, e.g. Gaussian distributions with diagonal covariance matrices where each term of the diagonal is in $\left[\frac{1}{2\pi},+\infty\right)$. Then, both $-\log p_{X_{t+1:T}|Z_{t+1:T}}(x_{t+1:T}|Z_{t+1:T};\theta)$ and
$-\log p_{Y|X,Z}(y|x,Z;\theta)$ are nonnegative, i.e.

$$
\begin{aligned}
&D_{KL}\left(Q_t(\theta,\phi) \;||\; P(\theta)\right)\\
&\quad \geq D_{KL}\left(q_{Z_{1:t}|X_{1:t}}(\cdot|x_{1:t};\phi) \;||\; p_{X_{1:t},Z_{1:t}}(x_{1:t},\cdot\,;\theta)\right).
\end{aligned}
$$

Thus, by minimizing $D_{KL}\left(Q_t(\theta,\phi) \;||\; P(\theta)\right)$, we minimize an upper bound of $D_{KL}\left(q_{Z_{1:t}|X_{1:t}}(\cdot|x_{1:t};\phi) \;||\; p_{X_{1:t},Z_{1:t}}(x_{1:t},\cdot\,;\theta)\right)$.

Furthermore, since we have

$$
\begin{aligned}
&D_{KL}\left(q_{Z_{1:t}|X_{1:t}}(\cdot|x_{1:t};\phi) \mid\mid p_{X_{1:t},Z_{1:t}}(x_{1:t},\cdot\,;\theta)\right) \\
&= D_{KL}\left(q_{Z_{1:t}|X_{1:t}}(\cdot|\ x_{1:t};\phi) \mid\mid p_{Z_{1:t}|X_{1:t}}(\cdot|\ x_{1:t};\theta)\right) - \log p_{X_{1:t}}(x_{1:t};\theta) \\
&= D_{KL}\left(Q_t(\theta,\phi) \mid\mid P_t(\theta)\right) - \log p_{X_{1:t}}(x_{1:t};\theta),
\end{aligned}
$$

we know that by optimizing ϕ to minimize $D_{KL}\left(q_{Z_{1:t}|X_{1:t}}(\cdot|x_{1:t};\phi) \mid\mid p_{X_{1:t},Z_{1:t}}(x_{1:t},\cdot\,;\theta)\right)$, we minimize $D_{KL}\left(Q_t(\theta,\phi) \mid\mid P_t(\theta)\right)$. To sum up, minimizing $D_{KL}\left(Q_t(\theta,\phi) \mid\mid P(\theta)\right)$ w.r.t. ϕ minimizes an upper bound of $D_{KL}\left(Q_t(\theta,\phi) \mid\mid P_t(\theta)\right)$. ■

D.2 Maximization of $p_{X,Y}(x,y;\theta)$

Proof. Finally, replacing $P_t(\theta)$ by $Q_t(\theta,\phi)$ in Eq. (4.10), we get:

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{t\sim\ \mathcal{U}_{[t_{\min},\ T-1]}}\left[D_{KL}\left(Q_t(\theta,\phi) \mid\mid P(\theta)\right)\right] \\
&= -\log p_{X,Y}(x,y;\theta) + \mathop{\mathbb{E}}_{t\sim\ \mathcal{U}_{[t_{\min},\ T-1]}}\left[D_{KL}\left(Q_t(\theta,\phi)\ \middle|\middle|\ \frac{P(\theta)}{p_{X,Y}(x,y;\theta)}\right)\right] \\
&\geq -\log p_{X,Y}(x,y;\theta)
\end{aligned}
$$

Therefore, by optimizing ϕ to minimize $\mathop{\mathbb{E}}_{t\sim\ \mathcal{U}_{[t_{\min},\ T-1]}}\left[D_{KL}\left(Q_t(\theta,\phi) \mid\mid P(\theta)\right)\right]$, we minimize $\mathop{\mathbb{E}}_{t\sim\ \mathcal{U}_{[t_{\min},\ T-1]}}\left[D_{KL}\left(Q_t(\theta,\phi)\ \middle|\middle|\ \frac{P(\theta)}{p_{X,Y}(x,y;\theta)}\right)\right]$, and by optimizing θ to minimize $\mathop{\mathbb{E}}_{t\sim\ \mathcal{U}_{[t_{\min},\ T-1]}}\left[D_{KL}\left(Q_t(\theta,\phi) \mid\mid P(\theta)\right)\right]$, we maximize a lower bound of $p_{X,Y}(x,y;\theta)$. ■

Bibliography

[1] F. Kröger, "Automated driving in its social, historical and cultural contexts," in *Autonomous Driving*. Springer, 2016, pp. 41–68.

[2] E. Dickmanns, "Computer vision in road vehicles–chances and problems," in *ICTS Symposium on Human Factors Technology for Next-Generation Transportation Vehicles, Amalfi, Italy*, 1986.

[3] "SAE J3016. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," April 2021. [Online]. Available: https://www.sae.org/standards/content/j3016_202104/

[4] F. M. Favarò, N. Nader, S. O. Eurich, M. Tripp, and N. Varadaraju, "Examining accident reports involving autonomous vehicles in California," *PLOS ONE*, vol. 12, no. 9, pp. 1–20, 09 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0184952

[5] M. Shan, K. Narula, Y. F. Wong, S. Worrall, M. Khan, P. Alexander, and E. Nebot, "Demonstrations of cooperative perception: safety and robustness in connected and automated vehicle operations," *Sensors*, vol. 21, no. 1, 2021.

[6] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Conference on robot learning (CoRL)*. PMLR, 2017, pp. 1–16.

[7] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin, and F. R. Yu, "A survey of mobile information-centric networking: Research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2353–2371, 2018.

[8] S. Kumar, L. Shi, N. Ahmed, S. Gil, D. Katabi, and D. Rus, "Carspeak: a content-centric network for autonomous driving," in *Proc. of ACM SIGCOMM*. ACM, 2012, pp. 259–270.

[9] J. Ding, H. Xu, J. Hu, and Y. Zhang, "Centralized cooperative intersection control under automated vehicle environment," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 972–977.

[10] M. Algomaiah and Z. Li, "Next-generation interchange control based on centralized management of connected and autonomous vehicles," *IEEE Access*, vol. 7, pp. 82 939–82 955, 2019.

[11] D. Kim and O. Jeong, "Cooperative traffic signal control with traffic flow prediction in multi-intersection," *Sensors*, vol. 20, no. 1, p. 137, 2020.

[12] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds," in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, 2019, pp. 88–100.

[13] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds," in *IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 2019, pp. 514–524.

[14] Y. Mo, P. Zhang, Z. Chen, and B. Ran, "A method of vehicle-infrastructure cooperative perception based vehicle state information fusion using improved kalman filter," *Multimedia Tools and Applications*, pp. 1–18, 2021.

[15] A. Rauch, F. Klanner, R. Rasshofer, and K. Dietmayer, "Car2X-based perception in a high-level fusion architecture for cooperative perception systems," in *2012 IEEE Intelligent Vehicles Symposium*, June 2012, pp. 270–275.

[16] F. Seeliger, G. Weidl, D. Petrich, F. Naujoks, G. Breuel, A. Neukum, and K. Dietmayer, "Advisory warnings based on cooperative perception," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, June 2014, pp. 246–252.

[17] H. Li, M. Tsukada, F. Nashashibi, and M. Parent, "Multivehicle Cooperative Local Mapping: A Methodology Based on Occupancy Grid Map Merging," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2089–2100, Oct 2014.

[18] S. W. Kim, B. Qin, Z. J. Chong, X. Shen, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "Multivehicle Cooperative Driving Using Cooperative Perception: Design and Experimental Validation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 663–680, April 2015.

[19] S.-W. Kim, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "The impact of cooperative perception on decision making and planning of autonomous vehicles," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 3, pp. 39–50, 2015.

[20] S. W. Kim and W. Liu, "Cooperative Autonomous Driving: A Mirror Neuron Inspired Intention Awareness and Cooperative Perception Approach," *IEEE Intelligent Transportation Systems Magazine*, vol. 8, no. 3, pp. 23–32, Fall 2016.

[21] M. Vasic, D. Mansolino, and A. Martinoli, "A system implementation and evaluation of a cooperative fusion and tracking algorithm based on a Gaussian mixture PHD filter," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4172–4179.

[22] N. El Zoghby, V. Cherfaoui, and T. Denoeux, "Evidential distributed dynamic map for cooperative perception in vanets," in *IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 1421–1426.

[23] T. Denoeux, "Conjunctive and disjunctive combination of belief functions induced by nondistinct bodies of evidence," *Artificial Intelligence*, vol. 172, no. 2, pp. 234 – 264, 2008.

[24] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, 1976.

[25] B. Hurl, R. Cohen, K. Czarnecki, and S. Waslander, "TruPercept: Trust modelling for autonomous vehicle cooperative perception from synthetic data," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 341–347.

[26] N. El Zoghby, V. Cherfaoui, B. Ducourthial, and T. Denoeux, "Distributed Data fusion for detecting Sybil attacks in VANETs," in *Belief Functions: Theory and Applications*. Springer, 2012, pp. 351–358.

[27] P. Smets, "Belief functions: The disjunctive rule of combination and the generalized Bayesian theorem," *IJAR*, vol. 9, no. 1, 1993.

[28] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv preprint arXiv:1312.6114*, 2014.

[29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017.

[30] A. Dempster, "A Generalization of Bayesian Inference," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 30, 1968.

[31] J. Čurn, D. Marinescu, N. O'Hara, and V. Cahill, "Data incest in cooperative localisation with the Common Past-Invariant Ensemble Kalman filter," in *Proceedings of the 16th International Conference on Information Fusion*, 2013, pp. 68–76.

[32] S. McLaughlin, V. Krishnamurthy, and S. Challa, "Managing data incest in a distributed sensor network," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, vol. 5, 2003, pp. V–269.

[33] T. Brehard and V. Krishnamurthy, "Optimal Data Incest Removal in Bayesian Decentralized Estimation Over a Sensor Network," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 3, 2007, pp. III–173–III–176.

[34] H. Li and F. Nashashibi, "Cooperative Multi-Vehicle Localization Using Split Covariance Intersection Filter," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 2, pp. 33–44, 2013.

[35] S. Julier and J. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*, vol. 4, 1997, pp. 2369–2373 vol.4.

[36] B. Ducourthial, V. Cherfaoui, and T. Denoeux, "Self-stabilizing distributed data fusion," in *Symposium on Self-Stabilizing Systems*. Springer, 2012, pp. 148–162.

[37] R. Guyard and V. Cherfaoui, "Study of distributed data fusion using dempster's rule and cautious operator," in *Belief Functions: Theory and Applications*, S. Destercke, T. Denoeux, F. Cuzzolin, and A. Martin, Eds. Cham: Springer International Publishing, 2018, pp. 95–102.

[38] P. Smets, "The Combination of Evidence in the Transferable Belief Model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 447–458, 1990.

[39] ——, "The Transferable Belief Model," *Artificial Intelligence*, vol. 66, no. 2, pp. 191–234, 1994.

[40] G.-C. Rota, "On the foundations of combinatorial theory I. Theory of Möbius functions," *Probability theory and related fields*, vol. 2, no. 4, pp. 340–368, 1964.

[41] P. Smets, "The Canonical Decomposition of a Weighted Belief," *Proc. of IJCAI*, pp. 1896–1901, 1995.

[42] M. Chaveroche, F. Davoine, and V. Cherfaoui, "Calcul exact de faible complexité des décompositions conjonctive et disjonctive pour la fusion d'information," in *Proceedings of XXVIIth Francophone Symposium on signal and image processing (GRETSI)*, 2019.

[43] R. Kennes, "Computational aspects of the Mobius transformation of graphs," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 2, pp. 201–223, 1992.

[44] N. Wilson, "Algorithms for Dempster-Shafer Theory," in *Handbook of Defeasible Reasoning and Uncertainty Management Systems: Algorithms for Uncertainty and Defeasible Reasoning*. Springer Netherlands, 2000, pp. 421–475.

[45] A. Sarabi-Jamab and B. N. Araabi, "Information-Based Evaluation of Approximation Methods in Dempster-Shafer Theory," *IJUFKS*, vol. 24, no. 04, pp. 503–535, 2016.

[46] A. Kallel and S. Le Hégarat-Mascle, "Combination of partially non-distinct beliefs: The cautious-adaptive rule," *IJAR*, vol. 50, no. 7, pp. 1000–1021, 2009.

[47] F. Pichon and T. Denoeux, "T-norm and uninorm-based combination of belief functions," *Proc. of NAFIPS*, 2008.

[48] A. Roquel, S. L. Hégarat-Mascle, I. Bloch, and B. Vincke, "Decomposition of conflict as a distribution on hypotheses in the framework on belief functions," *IJAR*, vol. 55, no. 5, pp. 1129 – 1146, 2014.

[49] J. Schubert, "Clustering decomposed belief functions using generalized weights of conflict," *IJAR*, vol. 48, no. 2, pp. 466–480, 2008.

[50] D. Mercier, F. Pichon, and É. Lefèvre, "Corrigendum to "Belief functions contextual discounting and canonical decomposition"[IJAR 53 (2012) 146–158]," *IJAR*, vol. 70, pp. 137–139, 2016.

[51] P. Smets, "The application of the matrix calculus to belief functions," *IJAR*, vol. 31, no. 1, 2002.

[52] M. Chaveroche, F. Davoine, and V. Cherfaoui, "Focal points and their implications for möbius transforms and dempster-shafer theory," *Information Sciences*, vol. 555, pp. 215 – 235, 2021.

[53] ——, "Efficient Möbius transformations and their applications to DS theory," in *International Conference on Scalable Uncertainty Management*. Springer, 2019, pp. 390–403.

[54] J. A. Barnett, "Computational Methods for a Mathematical Theory of Evidence," *Proc. of IJCAI*, vol. 81, pp. 868–875, 1981.

[55] J. Gordon and E. H. Shortliffe, "A method for managing evidential reasoning in a hierarchical hypothesis space," *Artificial intelligence*, vol. 26, no. 3, pp. 323–357, 1985.

[56] P. P. Shenoy and G. Shafer, "Propagating Belief Functions with Local Computations," *IEEE Expert*, vol. 1, no. 3, pp. 43–52, 1986.

[57] G. Shafer and R. Logan, "Implementing Dempster's rule for hierarchical evidence," *Artificial Intelligence*, vol. 33, no. 3, pp. 271–298, 1987.

[58] T. Denœux and A. B. Yaghlane, "Approximating the combination of belief functions using the fast moebius transform in a coarsened frame," *International Journal of Approximate Reasoning (IJAR)*, vol. 31, no. 1-2, pp. 77–101, 2002.

[59] M. Grabisch, "Belief functions on lattices," *International Journal of Intelligent Systems*, vol. 24, no. 1, pp. 76–95, 2009.

[60] T. Denœux and M.-H. Masson, "Dempster-Shafer reasoning in large partially ordered sets: Applications in Machine Learning," in *Integrated Uncertainty Management and Applications*. Springer, 2010, pp. 39–54.

[61] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto, "Trimmed Moebius inversion and graphs of bounded degree," *Theory of Computing Systems*, vol. 47, no. 3, pp. 637–654, 2010.

[62] A. Björklund, T. Husfeldt, P. Kaski, M. Koivisto, J. Nederlof, and P. Parviainen, "Fast zeta transforms for lattices with few irreducibles," *ACM TALG*, vol. 12, no. 1, p. 4, 2016.

[63] P. Kaski, J. Kohonen, and T. Westerbäck, "Fast Möbius inversion in semimodular lattices and U-labelable posets," *arXiv preprint arXiv:1603.03889*, 2016.

[64] M. Chaveroche, F. Davoine, and V. Cherfaoui, "Efficient Möbius transformations and their applications to Dempster-Shafer theory," in *LFA 2019 - Rencontres francophones sur la Logique Floue et ses Applications*, 2019.

[65] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using rao-blackwellized particle filters." in *Robotics: Science and Systems*, vol. 2, 2005, pp. 65–72.

[66] J. Clemens, T. Reineking, and T. Kluth, "An evidential approach to SLAM, path planning, and active exploration," *International Journal of Approximate Reasoning*, vol. 73, pp. 1–26, 2016.

[67] C. Wang, J. Cheng, W. Chi, T. Yan, and M. Q.-H. Meng, "Semantic-Aware Informative Path Planning for Efficient Object Search Using Mobile Robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.

[68] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," in *Advances in Neural Information Processing Systems*, 2018, pp. 2450–2462.

[69] S. Wirges, C. Stiller, and F. Hartenbach, "Evidential occupancy grid map augmentation using deep learning," in *IEEE intelligent vehicles symposium (IV)*, 2018, pp. 668–673.

[70] T. Sugiura and T. Watanabe, "Probable Multi-hypothesis Blind Spot Estimation for Driving Risk Prediction," in *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 4295–4302.

[71] S. Hoermann, M. Bach, and K. Dietmayer, "Dynamic occupancy grid prediction for urban autonomous driving: A deep learning approach with fully automatic labeling," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2056–2063.

[72] M. Everett, J. Miller, and J. P. How, "Planning Beyond The Sensing Horizon Using a Learned Context," *arXiv preprint arXiv:1908.09171*, 2019.

[73] R. Shrestha, F.-P. Tian, W. Feng, P. Tan, and R. Vaughan, "Learned map prediction for enhanced mobile robot exploration," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1197–1204.

[74] K. Gregor, G. Papamakarios, F. Besse, L. Buesing, and T. Weber, "Temporal difference variational auto-encoder," *arXiv preprint arXiv:1806.03107*, 2018.

[75] K. Gregor, D. Jimenez Rezende, F. Besse, Y. Wu, H. Merzic, and A. van den Oord, "Shaping Belief States with Generative Environment Models for RL," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[76] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Springer International Publishing, 2020, pp. 605–621.

[77] S. Aoki, T. Higuchi, and O. Altintas, "Cooperative perception with deep reinforcement learning for connected vehicles," in *IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 328–334.

[78] T. Higuchi, M. Giordani, A. Zanella, M. Zorzi, and O. Altintas, "Value-anticipating V2V communications for cooperative perception," in *IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 1947–1952.

[79] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[80] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[81] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3, pp. 229–256, 1992.

[82] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[83] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 3. IEEE, 2000, pp. 189–194.

[84] Y. Li, L. Ma, Z. Zhong, F. Liu, M. A. Chapman, D. Cao, and J. Li, "Deep learning for LiDAR point clouds in autonomous driving: a review," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[85] M. Chaveroche, "evidence-based-DST," https://github.com/mchaveroche/evidence-based-DST, 2019.

[86] R. Haenni and N. Lehmann, "Implementing belief function computations," *International Journal of Intelligent Systems*, vol. 18, no. 1, pp. 31–49, 2003.

[87] L. G. Polpitiya, K. Premaratne, M. N. Murthi, and D. Sarkar, "A Framework for efficient computation of belief theoretic operations," *Proc. of FUSION*, pp. 1570–1577, 2016.

[88] ——, "Efficient Computation of Belief Theoretic Conditionals," *Proc. of ISIPTA*, pp. 265–276, 2017.