Multi-target tracking and novel variational approaches for high-dimensional sequential data : an application to object counting in videos

Mathis Chagneux

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAT036

Thèse de doctorat

Multi-target tracking and novel sequential variational approaches for high-dimensional sequential data: an application to object counting in videos

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Telecom Paris

École doctorale n°574 Ecole Doctorale de Mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques Appliquées

Thèse présentée et soutenue à Palaiseau, le 21 Novembre 2023, par

MATHIS CHAGNEUX

Composition du Jury :

Wojciech Pieczynski Professeur, Telecom SudParis (SAMOVAR)	Président / examinateur
Nathalie Peyrard Directrice de recherche, INRAE (MIAT)	Rapportrice
Adeline Leclercq Samson Professeure, Université Grenoble Alpes (UFR Informatique)	Rapportrice
Anna Korba Maîtresse de conférences, ENSAE (CREST)	Examinatrice
François Portier Professeur, ENSAI (CREST)	Examinateur
Sylvain Le Corff Professeur, Sorbonne Université (LPSM)	Directeur de thèse
Pierre Gloaguen Maître de conférences, Universite Bretagne Sud (UMR CNRS 6205)	Co-directeur de thèse
François Septier Professeur, Universite Bretagne Sud (UMR CNRS 6205)	Invité

Remerciements

Avant tout je remercie les rapportrices, Adeline Leclercq-Samson et Nathalie Peyrard, pour avoir pris le temps de consulter en profondeur mes travaux ainsi que pour leurs rapports détaillés et très constructifs. Je remercie également l'ensemble des examinateurs et invités, Anna Korba, Wojciech Pieczynski, François Portier et François Portier pour avoir accepté de faire partie du jury de soutenance.

Je remercie ensuite mes encadrants, Sylvain, Pierre et Charles, références scientifiques pendant ces trois années mais aussi garantie incontestable d'enthousiasme et d'humour dans le travail: le mythe du thésard solitaire et déprimé est absolument démenti! Merci Sylvain pour le savant dosage de ton encadrement qui aura éveillé mon goût de l'approfondissement en me laissant toujours choisir mon rythme (et je sais que ce n'est pas forcément la norme dans le monde des Machine Learners de l'extrême). Merci Pierre pour ta disponibilité au tableau pendant "les années Claude Bernard" (et après!) et bien-sûr merci pour n'avoir jamais sous-estimé l'importance d'exposer ton premier thésard à une inépuisable quantité de vannes bien placées pour son bon développement. Merci Charles pour m'avoir permis de cultiver les aspects les plus geeks de ma personnalité dans un monde de matheux, et pour avoir été un exemple modèle d'ingénieur "décroissant" avant le monde de l'entreprise qui m'attend. Avoir pu poursuivre mes études trois ans de plus dans ces conditions est une chance, je n'ai aucun doute là-dessus, merci à vous trois!

Je remercie également mes collaborateur.ices tout au long de ces trois années qui m'auront permis de voir la recherche sous différents angles. Merci Antoine pour tes invitations aux évenements Surfrider qui m'ont permis de mettre un pied dans le monde des ONG pendant ma thèse, ce n'est pas courant! Merci Océane pour les mêmes raisons, et pour m'avoir accompagné même quand il s'agissait de localiser des plastiques dans des vidéos avec un outil pas pratique. Merci Elisabeth pour avoir apporté ton expertise mathématique précieuse, and thank you Jimmy for assisting us in porting the ideas of backward particle smoothing to the variational world in the most didactic way possible.

Merci aux doctorant.e.s avec qui j'ai cohabité dans les différents labos, d'abord mes cobureaux à l'Agro Bastien et Armand (désolé pour ma disparition après le déménagement), puis les doctorant.e.s du LPSM à Jussieu - Ludovic, Iqraa, Antonio, Camila, Miguel, Ariane, Alexis, Barbara, Francesco, Alice - rencontres inattendues de troisième année grâce à qui j'ai eu droit à deux thèses en une sur le plan humain! Merci enfin à Yazid et Gabriel, rencontres tardives également mais ayant confirmé que je n'étais pas le seul à naviguer dans le florilège de notations des méthodes séquentielles.

Merci à mes ami.e.s. Mention spéciale d'abord à mes colocataires de la rue Auguste Lançon, Alban et Raphaël, derniers piliers du 13e arrondissement avant l'exode et grâce à qui mon quotidien de doctorant est resté joyeux même en temps de pandémie. Merci Ella pour tout le temps partagé extrêmement qualitatif et pour tes qualités humaines exceptionnelles. Merci Antoine

qui m'a convaincu de prendre la musique au sérieux autant que les descentes de gradient. Merci Marc-Antoine d'avoir toujours eu ta porte ouverte pour mes passages à l'improviste. Merci Céline de m'avoir toujours rappelé que Python c'est pas de la programmation. Merci Luc pour toutes les discussions enrichissantes dont la conclusion c'est globalement qu'il faut rester serein. Merci Jane pour m'avoir infiltré à station F sans que j'aie besoin de créer une startup. Merci Clément, Mathilde, Julien, Virgile, Olivier, Jules, Ingrid, Mathieu...

Merci enfin à mes parents et à toute ma famille pour votre soutien pendant ces années d'étude où je n'ai pas donné assez de nouvelles. Merci à eux et à la chance rare d'avoir pu grandir dans des conditions où je n'ai jamais manqué de rien, ni matériellement ni humainement, j'ai conscience que c'est précieux.

Contents

1 Introduction **9**

1.1 A multifaceted practical application: automated monitoring of macrolitter river pollution from videos . . . 9

1.1.1 Context . . . 10

1.1.2 Project specificities . . . 10

1.2 Challenges in multi-target inference for video object counting . . . 15

1.2.1 From counting in images to multi-object tracking . . . 15

1.2.2 Challenges in high-dimensional sequential variational inference . . . 19

1.3 Presentation of the contributions . . . 23

2 Technical background in sequential Bayesian inference **25**

2.1 State-space models . . . 27

2.1.1 Joint backward smoothing in state-space problems . . . 28

2.1.2 Smoothed expectations of additive state functionals . . . 30

2.1.3 Exact inference and Kalman-based extensions . . . 33

2.2 Sequential Monte Carlo . . . 34

2.2.1 Elements of importance sampling . . . 34

2.2.2 Backward particle smoothing . . . 35

2.2.3 Particle-based additive smoothing and online methods . . . 38

2.2.4 Limitations . . . 39

2.3 Variational inference for sequential data . . . 41

2.3.1 General background on variational inference . . . 41

2.3.2 Sequential variational inference . . . 45

3 Macrolitter video counting on riverbanks using state-space models and moving cameras **51**

3.1 Datasets for training and evaluation . . . 53

3.1.1 Images . . . 53

3.1.2 Video sequences . . . 53

3.2 A state-space model with optical flow . . . 53

3.2.1 Detector . . . 54

3.2.2 Bayesian tracking with optical flow . . . 54

3.2.3 Generating potential object tracks . . . 55

3.2.4 Metrics for MOT-based counting . . . 57

3.3 Experiments . . . 60

3.3.1 Detection . . . 60

3.3.2 Counts . . . 60
3.4 Practical impact and future goals . . . 61

4 A backward sampling approach for online variational additive smoothing 63
4.1 Introduction . . . 63
4.2 Main algorithm . . . 66
4.2.1 Backward kernels with forward potentials . . . 66
4.2.2 Approximate backward conditional expectations . . . 66
4.3 Recursive gradient approximations . . . 69
4.3.1 Score-based gradient recursions . . . 70
4.3.2 Baseline variance reduction . . . 71
4.4 Experiments . . . 72
4.4.1 Linear-Gaussian HMM . . . 72
4.4.2 Chaotic recurrent neural network. . . . 75
4.5 Conclusion . . . 78
4.5.1 Summary . . . 78
4.5.2 Perspectives . . . 79

5 Additive smoothing error in backward variational inference for general state-space models 81
5.1 A control on backward variational additive smoothing . . . 82
5.1.1 Assumption and main result . . . 82
5.1.2 Comments on Proposition 1 and H1 . . . 85
5.2 Numerical experiments . . . 85
5.2.1 Linear Gaussian SSMs . . . 86
5.2.2 Nonlinear SSMs . . . 86
5.3 Discussion . . . 93
5.3.1 Assumptions . . . 93
5.3.2 Additional theoretical guarantees . . . 94
5.3.3 Variational kernels parameterization . . . 95

6 Conclusion and perspectives 97
6.1 Further research in backward SVI . . . 97
6.1.1 Relating the variational factors to the generative model . . . 97
6.1.2 Exploring larger variational objectives and families . . . 98
6.2 A unifying framework for sequence-wise prediction tasks in videos . . . 99

References 105

A Further technical background 121
A.1 Additional elements on deep learning, computer vision and MOT . . . 121
A.1.1 Prediction and feature extraction on images with DNNs . . . 121
A.1.2 Recurrent Neural Networks . . . 124
A.1.3 Additional topics in multi-object tracking . . . 124
A.2 Additional topics on state-space models . . . 126
A.2.1 Alternate smoothing decompositions . . . 126
A.2.2 Detailed Kalman filtering and smoothing computations . . . 129

B Appendix for macrolitter counting **131**
B.1 Categories . . . 131
B.2 Details on the evaluation videos . . . 132
B.2.1 River segments . . . 132
B.2.2 Track annotation protocol . . . 132
B.3 Implementation details for the tracking module . . . 133

C Appendix for paper on errors bounds **139**
C.1 Proofs of the main results . . . 139
C.1.1 Proof of Proposition 1 . . . 139
C.1.2 Proof of Corollary 2 . . . 142
C.2 Technical results . . . 142
C.2.1 Hardware configuration . . . 144
C.2.2 Linear Gaussian models . . . 144
C.2.3 Nonlinear models . . . 147

D Contexte, contributions et perspectives en français **149**
D.1 Comptage automatique de macrodéchets à partir de vidéos . . . 149
D.1.1 Contexte . . . 150
D.1.2 Spécificités du projet . . . 151
D.2 Présentation des contributions . . . 155
D.3 Perspective: un cadre unificateur pour la prédiction séquentielle dans les vidéos 156

This thesis is primarly written in English, but some of the content is also provided in French, namely Section 1.1, Section 1.3 and Section 6.2 which are written in French in Appendix D.

Chapter 1

Introduction

The work presented in this manuscript combines ideas from multiple fields, going from engineering mathematics - specificially literature on multi-object tracking and deep learning-based object recognition - to novel approaches for sequential inference via variational methods. The presented contributions are however motivated by an original application which was the impulse for the PhD project: a collaboration with Surfrider Foundation Europe to study and develop novel solutions for automated macrolitter counting in riverbanks. In the following sections, we start by directly presenting this application (in non technical terms) and the computational challenges it involves.

1.1 A multifaceted practical application: automated monitoring of macrolitter river pollution from videos

Litter pollution concerns every part of the globe. Each year, almost ten thousand million tons of plastic waste is generated, among which 80 ends up in landfills or in nature [GJL17a], notably threatening all of the world's oceans, seas and aquatic environments [Wel20; GS20]. Plastic pollution is known to already impact more than 3763 marine species worldwide (see for example [PR23] for a detailed analysis) with risk of proliferation through the whole food chain. This accumulation of waste is the endpoint of the largely misunderstood path of trash, mainly coming from land-based sources [Roc+16], yet rivers have been identified as a major pathway for the introduction of waste into marine environments [Jam+15]. Therefore, field data on rivers and monitoring are strongly needed to assess the impact of measures that can be taken. The analysis of such field data over time is pivotal to understand the efficiency of the actions implemented such as choosing zero-waste alternatives to plastic, designing new products to be long-lasting or reusable, introducing policies to reduce over-packing.

Different methods have already been tested to monitor waste in rivers: litter collection and sorting on riverbanks [Bru+18], visual counting of drifting litter from bridges [Gon+21], floating booms [Gas+14] and nets [Mor+14]. All are helpful to understand the origin and typology of litter pollution yet hardly compatible with long term monitoring at country scales. Monitoring tools need to be reliable, easy to set up on various types of rivers, and should give an overview of plastic pollution during peak discharge to help locate hotspots and provide trends. Newer studies suggest that plastic debris transport could be better understood by counting litter trapped on river banks, providing a good indication of the local macrolitter

pollution especially after increased river discharge [Emm+19; ES20].

1.1.1 Context

To this aim, Surfrider Foundation Europe created the *Plastic Origins* project, one objective of it being the development of effective automated monitoring solutions for macroplastic counting on riverbanks. The data captured as part of this project (which is presented more extensively in Chapter 3) can be summarized as follows.

1. Several thousands of independent annotated litter images, more precisely pairs of litter items photographed on river banks with their location and area in the image identified with bounding boxes.

2. Dozens of non-annotated high-resolution videos of river banks containing litter, shot from handheld cameras in moving boats, lasting from a few seconds to several minutes.

3. Several data gathering expeditions where volunteers are asked to provide visual estimates of the number of litter items on some of the river sections covered by the video footage desribed above.

In Figure 1.1, we show a few examples of the annotated dataset of static litter images, where bounding boxes are drawn to visualize the annotations. In Figure 1.2, we show two sets of frames from one of the videos (on two different sections of the associated river expedition). Such examples illustrate typical characteristics of the setting imposed by the data.

- In both images and videos, objects that must be detected come in a large variety of shapes and colours. They are captured from various angles and distances. The backgrounds, lightning and visual cluttering of the scenes vary greatly.

- In videos, river banks are filmed from a camera which mostly shoots perpendicularly to the direction of motion. The camera moves globally along the river, but motion can be highly nonlinear, e.g. with variations in speed and non-trivial rotations. During the shooting process, a given object will be visible for a various amount of time mostly depending on occlusions and the speed of the camera. Multiple angles of the same object can be visible as the camera moves, such that its visual aspect may slightly shift over time.

1.1.2 Project specificities

At Surfrider, it was established early on that all data collection campaigns (annotated static images and videos) would be pursued but that the most convenient solution for litter monitoring was to work directly on video material, as filming river expeditions was the easiest option to regularly gather on-site data throughout the year.

Therefore, the focus turned specifically to computational solutions which can automatically predict a total number of items of interest visible in videos. This latter task, which we refer to as *video object counting* in the rest of the document, lies at the intersection of the fields of *computer vision* and *temporal data analysis*. Within the former, it is particular in the following ways.

Figure 1.1: 12 instances of the dataset of labeled images

1. For a given video, each object may be visible in multiple frames but must be counted only once.

2. The location of the individual objects is not necessarily required in the final prediction.

The first aspect makes the task of counting in videos largely different than that of counting

Figure 1.2: Two groups (one for each column) of 4 frames from one instance of the video dataset

in independent static images. Amongst existing literature, identifying objects across multiple frames is already a central topic in video-based *multi-object tracking* research (MOT), which aims at predicting the individual trajectories of objects of interest in video footage, i.e. detecting and localizing these objects in each frame and assigning them a consistent identifier over time.

The second point, however, progressively became a distinctive feature of this project. On the one hand, the requested output is more restricted than for traditional MOT and most video-based applications, since accurate frame-by-frame predictions are not required. On the other hand, the possible research directions are broader, because one may consider solutions that do not rely explictly on framewise object detection as an intermediate quantity to produce global video-wise counts. In the following paragraphs, we describe succintly some other important aspects which are specific to the task at hand.

Annotation formats. The video content of the project does not come with any form of dense annotation, i.e. one does not have access to examples of footage with objects located and identified in each frame. The video data consists of either bare footage with no annotations or video segments where the ground truth is a global count for the segment. In video-based computer vision, this is an example of tasks in the *weakly* annotated setting. On the contrary, the image dataset is densely annotated with precise object locations, but images are independent and lack the temporal dependencies of the videos on which the end task must be performed. Therefore, multiple forms of data are available, and it is not clear from the start how to combine these to build an efficient solution that makes best use of each. [1]

Computational resources. Another component of this project is the computational limitations specified by the Surfrider Foundation. Early on, it was announced that a preferable solution would be one easily portable to embedded setups with limited processing power, ideally one that could directly run on smartphones used to capture the videos to avoid sending the data to a secondary device. It was also suggested along the way that methods which could process data on-the-fly would be preferable, as storing and processing all frames at once can be cumbersome on embedded devices. [2]

Ground-truth variability and reliability. Furthermore, among the global counts provided with the videos, variability was observed when multiple people were asked to identify litter items on the same river sections. Figure 1.3, illustrates this in a boxplot of the reported counts by 20 volunteers on three distinct locations covered by videos of the dataset. This variability in the ground truth estimates suggests that the automated macrolitter counting task may benefit from uncertainty estimates together with the predicted counts. Such uncertainty estimates, additionally, would make the end solution more reliable for pollution monitoring, e.g. by

[1] In Chapter 3, we propose an algorithm which only requires supervision in the form of independent annotated images to train an object detecto. In Chapters 4 and 5, we study generic methods for inference in sequential models that are based on *unsupervised* optimization objectives.

[2] In Chapter 3, an effort was put in choosing efficient solutions from computer vision and developing approximations which are known to scale well for the targeted task. Additionally, most of the computations in the algorithm we propose can be performed online. In Chapter 4 we specifically propose an online algorithm to build generic approximations in the variational context.

allowing to discard bad predictions. [3]

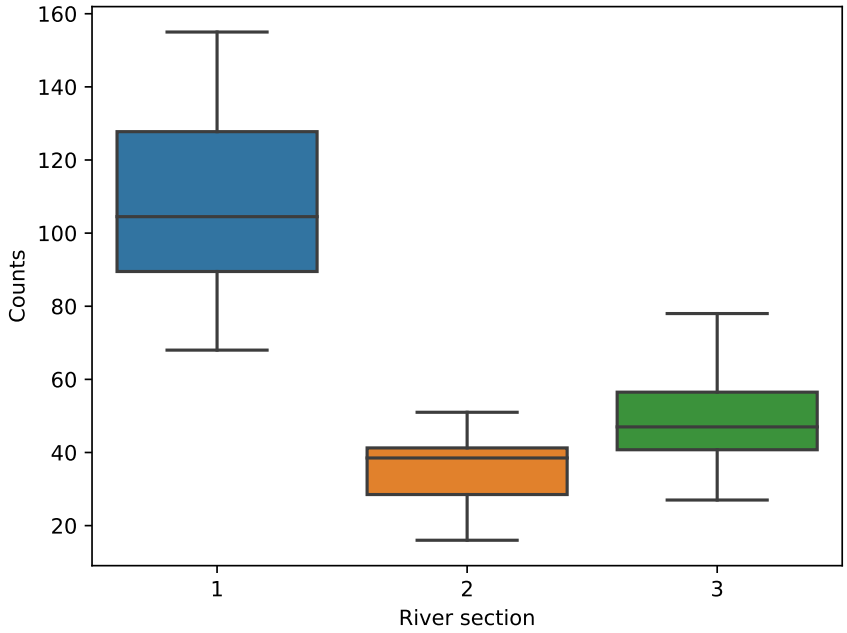


Figure 1.3: Boxplots illustrating the variability of visual counting amongst 20 volunteers for three river sections

High-dimensional observations. From the point of view of inference in sequential data, the observations that make up the sequences of interest (the videos) are colored images in high resolution. As such, the dimensionality of the data at each timestep is in the order of several millions, which makes it impossible to apply classical methods of sequential inference directly in the space of the original videos. [4]

[3] In Chapter 3, the tracking solution we propose naturally accounts for uncertainty in the motion of the video to generate counts, while in Chapter 4 and Chapter 5, the Bayesian formalism of the solutions we study can naturally be used to derive confidence intervals around the statistical estimates.

[4] In Chapter 5 and 4, we extensively study variational solutions as alternative solution to solve the scaling issues of classical approximations derived from e.g. Monte Carlo methods.

1.2 Challenges in multi-target inference for video object counting

All in all, the previous setting and challenges can be approached from many different angles. When combining the specific goal of video object counting with the additional observations and constraints presented in the previous paragraphs, some central motivations of this thesis may be summarized as follows.

- Given limited annotations, can we build solutions to extract global information (i.e. counts) from high-dimensional sequential data (i.e. videos) ?
- Would they scale well ?
- Are they easily amenable to uncertainty estimation ?

In this work, we attempted to improve our understanding of the theoretical and practical challenges of the research problems involved behind these questions, and to extract elements which would either be best suited for the final task or constituted core topics that would benefit from further research. In this section, we reproduce this analysis roughly in the order it was conducted.

1.2.1 From counting in images to multi-object tracking

Counting in images

While literature specifically targeting video object counting as an end goal is rather sparse, a first way to approach the topic is to consider the simpler task of counting objects in individual still images. In that respect, and largely motivated by applications such as crowd counting [Zha+15; Zha+16; SSB17], research has been very active. Driven by recent advances in computer vision, namely the strong performance of convolutional neural networks (CNNs) [LB+95] and the recent availability of large datasets of annotated images [Den+09; Lin+14], most research has undergone a general shift from handcrafted feature engineering [Low99] and mathematical modeling of images [Bar12] to a wide adoption of learning-based approaches, in particular supervised learning (see Appendix A.1.1 for an introduction). Therefore, for the most part, improvements in image object counting have largely been fostered by very active research topics such as image classification [Che+21], segmentation [Min+21] or object detection [Zha+19], which are illustrated in Figure 1.4. That said, counting methods have in general be divided into two categories.

The first (and most intuitive) approach to estimate the number of objects of interest in an image is to localize each of them individually and enumerate them afterwards, a methodology referred to as *counting-by-detection*. In this setting, improvements in counting performance can be mostly attributed to the development of ever more sophisticated object detectors [Ren+15; Red+16; ZWK19; Car+20]. Nonetheless, as detection-based image counting solutions eventually discard location information in their final output, a specific focus in the related approaches has often been put on object detection solutions that provide minimalistic outputs and/or requiring weak forms of supervision [Bea+16; Lar+18], see e.g. Figure 1.5 for some intuitive illustration. This has been motivated, additionally, by the lack of precisely annotated datasets necessary to bring generic object detectors to desired levels of performance in

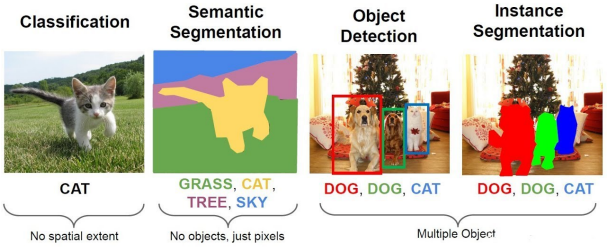


Figure 1.4: Various computer vision predictions from still images

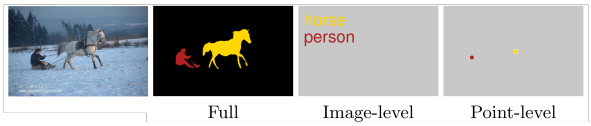


Figure 1.5: Full and weak forms of supervision for the task of semantic segmentation. Source [Bea+16]

very specific settings. In the context of litter counting, for example, while few initiatives have been conducted to assemble substancial amounts of trash images in hetereogeneous contexts [PS20], most works that have tackled the task of counting litter via detection [Wol+20; Lie+20] have required a dedicated data acquisition campaign.

Both to circumvent the previous issues and to improve counting performance in highly occluded settings where conventional object detection ultimately fails, another line of research has focused on building fully differentiable deep learning-based solutions which directly frame counting as a regression problem trained from image-level count supervision (i.e. datasets of images annotated only with the number of objects visible in them). Starting with the seminal work of [LZ10] most of these approaches format the output of a DNN into intermediate so-called *density maps*, which are 2D predictions whose sum over the image space provides count estimations, see e.g. Figure 1.6 for an intuitive illustration. As such, and while few other approaches [Cha+17] depart from these intermediate structures, advances in these so-called *counting-by-regression* approaches have largely been dictated by DNN architectures that produce better density maps [Zha+16], or with elaborate methodologies that additonally leverage weak forms of location supervision and annotation uncertainty [ALZ16] to build more reliable estimates.

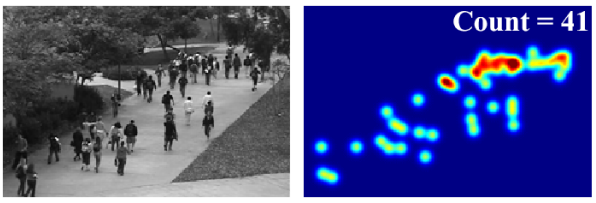


Figure 1.6: An example of density map and associated count from [Pha+15].

Deep learning approaches for video and temporal data

While some works from the previous research community attempt to leverage temporal redundancy in videos to improve *framewise* counting performance [XSY17], extending the above ideas to the more complex task of video-wise object counting is not direct.

In fact, in practice the latter task can hardly be framed as a well-defined subfield of computer vision. As such, one may be tempted first to approach it from other video-related tasks, such as video classification [Kar+14] or video action recognition [FPW17], which both produce video-level predictions from supervision related to entire portions of the footage. Here, most advances have originally been enabled by extending image-related deep learning architectures to the temporal dimension of videos, e.g. spatiotemporal convolutions [Tra+15; Fei20], albeit at the expense of much larger networks. To mitigate the latter issue, many deep learning approaches such as recurrent neural networks (RNNs) [HS97], and more recently transformers [Vas+17], have succesfully been applied to temporally-structured data, such as speech [GMH13] or language [Dev+19], with recent applications to video [Arn+21]. However, directly using them for the task of counting is very challenging. First, in most cases, proper training requires substantial amounts of annotated data. Second, most predictions that have been tackled with these tools alone, such as assigning a label to portions of a video or retrieving the language of an audio extract, can be intuitively linked with a global label summarizing some semantic information spread across the sequential content. Object counts at the video scale, comparatively, constitute a rather weak learning signal. Finally, while they do provide strong computational solutions to assimilate high-dimensional temporal data such as videos, they only provide deterministic predictions which are impossible to analyse or supplement with uncertainty estimates, given their rather opaque internal components, e.g. state of an RNN, intermediate layers of a transformer.

Multi-object tracking

Instead, another pragmatic approach to video object counting is to tackle the related problem of *multi-object tracking* (MOT), which seeks to predict the trajectories (or tracks) that each visible object takes in a video (i.e. its successive positions in each frame, as illustrated in Fig 1.7). Indeed, having achieved MOT, an estimate of object counts is immediatly obtained by enumeration of the number of predicted tracks.

By itself, MOT is a vast domain which can be studied from many angles, involving both concepts from computer vision and specific mathematical models to re-identify objects across frames [Luo+21]. Indeed, the dominant methodology to tackle the tracking problem is to divide it into two stages: an object detector first predicts the positions of objects in each frame, which are then recombined into object tracks by assigning a consistent identifier to detections that correspond to a common object. The first step is directly related to literature on object detection in images as presented above. The second, however, is a central of topic of MOT referred to as *data association*, which is often framed as an assignment problem between pairs of detections associated with a given cost. In practice, solving the matching problem from the costs is not challenging given the dimensions involved, i.e. optimal solutions from linear programming [Kuh55] can be used as-is. However, the definition and computation of the costs themselves is where lies most of MOT research.

In that respect, a decisive factor is the availability or not of *track supervision*, i.e. datasets of videos where ground-truth trajectories have been annotated by hand. Mostly motivated

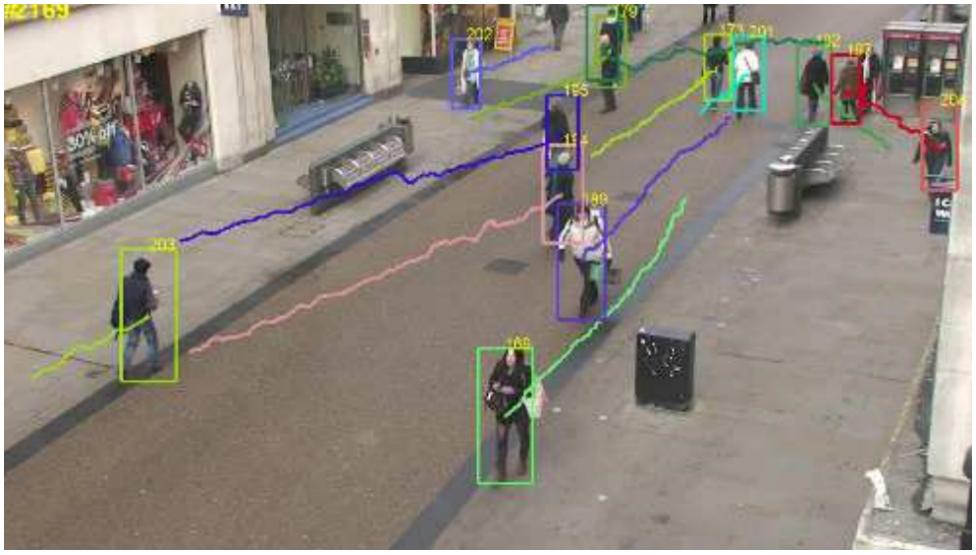


Figure 1.7: An illustration of MOT: people are detecting in each frame and consecutive positions of the same person are associated to given tracks (one colour per track).

by highly demanding industrial applications such as autonomous driving [Cae+20], the recent collection of substantially large such datasets have enabled the development of so-called *end-to-end multi-object tracking* solutions, where both the detection and association stages are combined into a common supervised learning problem given labels that specify both object locations and identities across frames. In this case, associations costs are largely abstracted into advanced deep learning modules such as graph neural networks [Wu+20; LGJ20; BL20], or by supplementing object detectors with additional outputs from which association can be performed greedily [ZKK20; Wan+20; Zha+21]. While such methods generally show best overall tracking performance, in most applications the very time-consuming collection of track annotations is not possible, as was the case in the context of the Plastic Origins project.

In this case, association cues are generally built by supplementing object detectors with either motion or appearance models built separately to recognise objects across frames. In some settings, simple linear assumptions on motion are sufficient to correctly predict positions of already tracked object in subsequent frames [WBP17] and use this information for association. In other works, unsupervised learning techniques [Che+20; He+20] allow to train a separate network that produces embeddings of image content in metric spaces where distances can quantify visual similarity between detections [Bew+16]. From there, varying degrees of sophistication can be added to improve the association performance, from precise estimation of the nonlinear motion of pixels between images, which is known as *optical flow* prediction [HS81; Dos+15], to proper management of uncertainty in the measurements via principled methods from probabilistic inference.

While many works have successfully deployed MOT techniques to count objects in videos, such as for fruit counting [Liu+18; He+22; Liu+19] and animal counting [Xu+20; Li+22; Tia+19; Kim+22a], many challenges remain to improve the stability of the tracking solutions.

In particular, when strong motion, blurry frames or visually cluttered backgrounds are present in the video footage (as was the case in many videos of the Plastic Origins dataset), object detectors may fail to produce consistent streams of detections for the objects, which can result in fragmented MOT predictions (e.g. two trajectories are predicted for the same object)

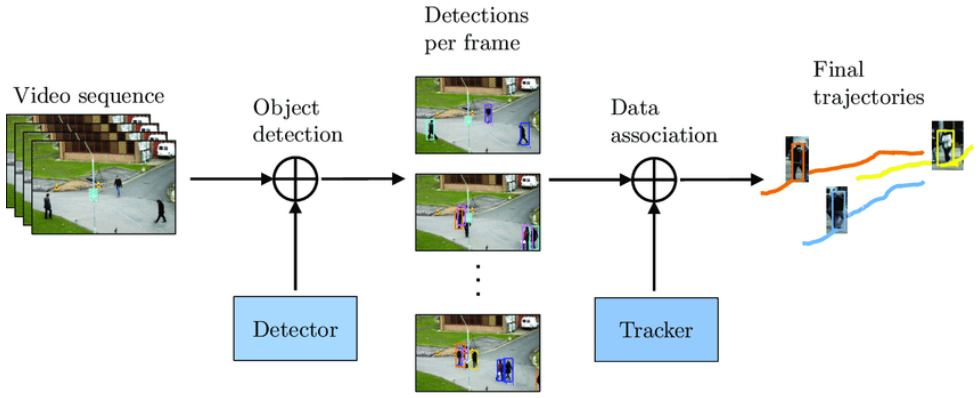


Figure 1.8: The tracking-by-detection paradigm

and lead to overcounting. To solve this, one possibility is to improve the robustness of the data association stage and build more sophisticated models that can recover from missing or false detections. In that regard, a recurring topic in MOT is the search for a unifying formalism that integrates multiple sources of information (e.g. detection confidence, motion, etc) into a global mathematical formalism. While not particularly adressed in the deep learning community, major advances in literature on sensor fusion have been made regarding this problem by relying on the formalism of point processes. In the seminal work [Mah03], the notion of multi-target Bayes filtering introduces well-defined probabilistic models on *sets* of points at each timestep, which avoids tracking objects separately and relying on a separate methodology for their interaction. More recent methods like [Reu+14; VVH17] further formalise the association stage by viewing track labels as random variables, while other works [Mor19] frame MOT as a Bayesian nonparametric estimation problem. Still, given the *tracking-by-detection* scheme, most of the information about the targeted content is essentially contracted at each timestep in the form of point estimates. As a consequence, additional information about the video (e.g. temporal evolution of the image content, detection uncertainty) must be translated into this formalism, which in many cases leads to complex models with hetereogeneous interconnected components and interdependent hyperparameters (e.g. detection and associations thresholds).

1.2.2 Challenges in high-dimensional sequential variational inference

From MOT to high-dimensional latent inference

To circumvent these MOT-specific technicalities, another direction to improve the reliability of the results - notably the coherency of the predictions with respect to the temporal evolution of the observations - is to introduce the important dependencies of the data in the feature extraction stages of the detection networks, where relevant information on the image content is compressed into compact representations prior to the final prediction stage. For example, some works like [Zhu+17] temporally constrain the framewise feature maps of the object detectors with deterministic estimates of pixel motion, see Figure 1.9 for an intuitive illustration. More recently, such methodology has been further motivated by major advances in unsupervised latent representation learning in videos [Loc+20; Gre+19; Kab+21; Els+22; SWA22; Kip+22] which pushes this idea further and focus entirely on producing embeddings of the images that are naturally constrained given known structure of the data and the desired predictions.

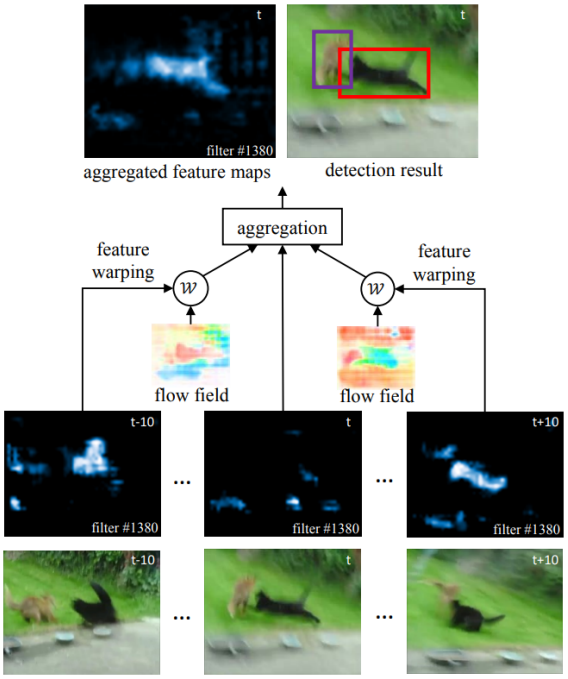


Figure 1.9: In [Zhu+17], a vector field which estimates the motion of pixels accross frames is used to combine object detection features from multiple frames. Here "heatmaps" (maps of the probability in $[0, 1]$ of object presence at each pixel) show improvement in the capability of the detector to extract relevant information in the frame at t by leveraging content in the neighboring frames.

Based on broad ideas of cognitive science [KTG92] and recent trends in representation learning [BCV13], they assume the existance of underlying representations of the images which naturally decompose into individual components, each related to distinct visual elements in the scene, e.g. separate objects. In general, it is expected that these so-called "object-centric representations" can be estimated without additional supervision by leveraging statistical dependencies in the data, which can be evaluated by considering the improvements they bring in MOT predictions [Wei+21].

To frame these assumptions into generic estimation problems, a common technique is to rely on *latent data models* and define a generative model via the joint distribution of the observations $(Y_0, \dots, Y_t)$ and additional unobserved random variables $(X_0, \dots, X_t)$, called *latent* or *hidden* variables. From there, recovery of the latent representations behind the data is directly cast as a Bayesian inference problem, i.e. statistical estimation under the posterior distribution of $(X_0, \dots, X_t)$ given $(Y_0, \dots, Y_t)$. To this aim, most methods leverage recent advances in approximate Bayesian inference, such as variational autoencoders [KW14] which directly parameterize posterior distributions with neural networks whose parameters can be learnt without supervision via principled approaches, e.g. stochastic gradient descent and proxies of maximum likelihood estimation (MLE) based on divergence minimization. In any case, when the observations are images and the latent variables are features, a challenging aspect of these models is the high-dimensionality of both the data and the hidden variables.

From Monte Carlo methods to variational inference

In more general literature on sequential data, it turns out that the previous estimation problems have been extensively studied from the point of view of hidden Markov models (HMM) [CMR05], where latent recovery is generally referred to as *state inference*, and the distribution of the sequence of latents given the observations is called the *smoothing distribution*. For simple data models where observation processes and dependencies are only specified via linear mappings and Gaussian noise, optimal algorithms based on analytical recursions have been derived very early on [Kal60]. For more general models, however, most computations are intractable, which has led to an abundant literature dedicated to approximations [Sär13; DMS14]. Here, most approaches are either based on coarse simplifications of the data model itself, e.g. via linearization or Gaussian assumptions [WV00], or Monte Carlo estimation, in particular sequential Monte Carlo (SMC) methods known to be consistent [CP+20]. While the former strategy generally fails in complex settings, SMC methods have been successfully applied for practical problems with complex models containing strong nonlinearities and non-Gaussian noise (see e.g. [DFG13, part IV]), and are built on solid theoretical grounds [DG01].

Nonetheless, it is generally accepted that SMC approaches perform poorly in higher dimensions [BBL08], which has limited their applicability for estimation problems involving large latent spaces, such as those presented in the previous sections where hidden variables are abstract representations of the data comprised of many components. To cope with these limitations, many works of the associated literature [HH20; Häl+21a] have turned to variational inference (VI), where approximations of the posteriors are based on optimization of user-chosen families of distributions. While known to scale well comparatively to Monte Carlo solutions, VI methods often rely on simple parametric distributions, and typical choices of approximating families (e.g. products of multivariate Gaussians under *mean-field* assumptions) are largely inadapted in the context of sequential data. Indeed, from a modeling perspective, they cannot capture the actual dependencies of the true smoothing distributions of latent variables $(X_0, \dots, X_t)$ given observations $(Y_0, \dots, Y_t)$, which greatly hinders their use in applications as described in the previous subsection (where correctly leveraging the statistical properties of the posteriors is crucial). For the same reasons, they do not leverage known decompositions of the latter distributions which have proven central in building scalable approximations for HMMs.

To cope with these limitations, a recent line of work, broadly referred to as *sequential variational inference* (SVI), defines parametric families of approximate joint distributions which reintroduce dependencies between the latent states and the observations, either via exponentially conjugated graphical models [Joh+16; LKH18] or by direct specification of Markovian decompositions mimicking those of the true smoothing distributions [Kim+20; Arc+15; KSS15; KSS17; Fra+16]. Here, most works leverage the approximating power of deep neural networks to avoid relying on simplistic assumptions or coarse approximations of the underlying data model. In particular, starting with [Chu+15] a common practice has been to use DNN architures specifically tailored for sequential data (i.e. those mentioned in Section 1.2.1 such as RNNs), whose outputs are used to directly parameterize individual terms in the variational decompositions. From there, inference is transferred into an optimization problem on the parameters of these networks, which is based on minimization of a divergence against the targeted distributions.

Errors bounds for sequential variational inference

By relying on optimization and amortized inference with DNNs, SVI has emerged as a promising alternative methodology in applications where the curse of dimensionality of sample-based approaches is a limiting factor. However, up to now, their wide adoption as generic methods in the SSM community is still limited compared to SMC. Indeed, as the latter are based on principled ideas from Monte Carlo simulation, they are known to converge to true solutions in the limit of infinite samples, and their theoretical analysis has been largely approached via well-established tools from asymptotic and non asymptotic statistics. As such, most SMC algorithms have been systematically accompanied with convergence results [DG01; DL13; LSV20; AD03; DDS10b] which generally provide clear indications of which quantities need to be properly controlled to obtain good overall approximations. In general, such works derive precise bounds that directly link performance with the number of samples. As such, they have been safely adopted as internal components for more complicated tasks, such as parameter estimation in hidden Markov models, where the impact of the approximations they introduce is well understood [Kan+09]. On the other hand, SVI initiatives have been largely developed empirically, sometimes from the point of view of specific applied fields of ML like reinforcement learning [SG20; Web+15] or computational neuroscience [ZP20]. As such, many questions remain to understand the type of solutions that they can provide from a theoretical point of view, as well a general guidelines for their implementation. As refinements of classical VI, SVI methods are expected to only provide biased solutions [BKM17], but precise analysis of this bias requires clear understanding on the behaviour of the variational family through the optimization process, which is intricately linked with the chosen implementations (especially in the presence of DNNs).

Instead, a common theoretical analysis in the sequential context is to derive quantitative bounds that characterize the error induced by computing smoothing expectations of the form $\mathbb{E}\left[h(X_0, \ldots, X_t)|Y_0, \ldots, Y_t\right]$ with an approximation instead of the true posterior. In particular, a central question is the dependency of such error w.r.t the length of the sequences considered, because approximations with supralinear behaviours in that regard cannot be used in realistic scenarios. A particular class of smoothing expectations of interest are those where h decomposes into sums of functions each depending on subsets of the states, referred to as *additive state functionals*. Indeed, those are ubiquitous in HMM literature, because most tasks such as state inference, MLE with the Expectation-Maximization algorithm [DLR77] or recursive MLE [LM97] can be formulated as smoothing expectations of additive state functionals. In the case of SMC, linear bounds have been repeatedly obtained (e.g. as part of the theoretical works mentioned above). While recent results have also been obtained for extensions of classical SMC which introduce biased quantities in the sampling process [GLO22], similar theoretical properties for SVI methods have never been derived. In practice, such results are of great practical importance, as they pave the way to understanding how to control the estimation errors of variational families based on the error made for individual timesteps.

Online approaches and backward variational inference

Another central topic in SVI, which is the practical pendant of the previous considerations, is their scalability when used with sequences $Y_0, \ldots, Y_t$ for t large. Indeed, having introduced temporal decompositions of the variational families, the SVI works mentioned above still rely on the original formulation of the minimization objective of classical VI. While classical meth-

ods allow principled minibatch subsampling for large datasets, i.e. stochastic optimization which is theoretically motivated under independant observations [Hof+13], the dependencies in SVI families constrain optimization to be performed on the joint latent space of $X_0, \dots, X_t$, which can become prohibitive for large sequences and defeat the purpose of resorting to variational inference in the first place. As such, a major challenge is the derivation of *online* SVI methods, i.e. whose parameters can be learnt by processing observations recursively.

To this aim, a few works [MCY18; ZP20; DZP23] leave out some of the dependencies or target simpler distributions, e.g. those of individual states X_t given past observations $Y_0, \dots, Y_t$, known as the *filtering distributions*. However, the resulting online procedures are built via intermediate minimization objectives derived from additional assumptions on the variational approximations at each timestep, and these can hardly be verified in practice. Recently, to circumvent this, [Cam+21] proposed *backward variational inference*, a SVI approach which leverages a known factorization of the smoothing distribution via the so-called *backward* Markov kernels of the reverse process $(X_{t-s})_{s \leq t}$ given $(Y_0, \dots, Y_t)$. Under this decomposition, the joint objective can be expressed via recursions on the observations, and as such allows to derive an online procedure that is still based on the principled minimization objective of classical VI. Conveniently, in this setting, the latter objective is itself an additive state functional, such that the optimization problem may be analyzed via the theoretical bounds described above.

That said, a hidden technicality in the recursions derived from the backward factorization is the presence of nested conditional expectations whose cost of evaluation grows linearly with time. As such, the resulting online algorithms are impractical without further approximations that ensure a constant computational cost at each timestep. In SMC literature, similar decompositions have already been introduced [DDS10a] and theoretically studied [DDS10c] that derive algorithms for recursive smoothing of additive state functionals. Here, the conditional expectations are approximated via evaluation on the discrete support of the empirical distributions obtained as part of the sampling process, which defines statistics that can be updated sequentially in constant time. More recently, [OW+17; AA22] have derived computationally efficient refinements from these ideas by leveraging known properties of the backward kernels. In the context of backward variational inference, such properties have not been leveraged, yet the simplicity of the associated algorithms is appealing to derive efficient online algorithms that do not rely on complex functional approximations at each timestep.

1.3 Presentation of the contributions

Given the previous overview of both practical challenges directly linked with object counting, and more general approaches in sequential variational inference that can be leveraged to derive more generic solutions, the work of thesis is divided into two corresponding sets of contributions.

- A technical contribution which directly tackles video object counting for macrolitter data.

- Two methodological contributions related to backward variational inference which leverage ideas from SMC to derive a theoretical understanding of existing methods and improve their computational properties.

Macrolitter video counting on riverbanks using state space models and moving cameras, *Mathis Chagneux, Sylvain Le Corff, Pierre Gloaguen, Charles Ollion, Océane Lepâtre, and Antoine Bruge. Published (with source code) in Computo, 2023.*

In Chapter 3, we present a new method to count macrolitter items in video of riverbanks filmed from boat-embedded cameras. Here, we rely on multi-object tracking (MOT) but focus on the key pitfalls of false and redundant counts which arise in typical scenarios of poor detection performance. Our system only requires supervision at the image level and performs Bayesian filtering via a state-space model based on optical flow. We present the new open image dataset gathered through a crowdsourced campaign and used to train an object detector that is particularly suited for the task at hand. As part of this work, the realistic video footage assembled by water monitoring experts has been annotated and used for evaluation. Improvements in count quality are demonstrated against systems built from state-of-the-art multi-object trackers sharing the same detection capabilities. A precise error decomposition allows clear analysis and highlights the remaining challenges. This first contribution was conducted in close collaboration with Surfrider Foundation Europe, and provides an initial tool which has since been thoroughly implemented as part of the Plastic Origins project, receives support and is incrementally updated.

A backward sampling approach for online variational additive smoothing, *Mathis Chagneux, Pierre Gloaguen, Sylvain Le Corff, Jimmy Olsson. Submitted for publication in the Transactions on Machine Learning Research (TMLR), 2023.*

In Chapter 4, we leverage ideas from recursive smoothing approaches developed in the SMC community to derive a computationally efficient online algorithm in the context of backward variational inference. Here, we propose a specific decomposition of the variational kernels which resembles that of the true backward kernels and allows to reproduce known approximations schemes of the conditional expectations involved in the recursions. In turn, this removes the need for additional functional approximations previously necessary for recursive computation of smoothing expectations of additive state functionals under variational approximations. Then, we propose a new decomposition of the gradient of the variational optimization objective based on the score-function estimator, which allows recursive learning of the variational parameters. Numerically, the quality of the derived gradients is demonstrated against batch estimates, and the relevance and computational efficiency of the proposed approach is illustrated on long sequences of observations.

Additive smoothing error in backward variational inference for general state-space models, *Mathis Chagneux, Élisabeth Gassiat, Pierre Gloaguen, Sylvain Le Corff. In Major revision for publication in the Journal of Machine Learning Research (JMLR), 2023.*

In Chapter 5, we study the theoretical properties of the backward variational decomposition, where we establish under mixing assumptions that the variational approximation of expectations of additive state functionals induces an error which grows at most linearly in the number of observations. This guarantee is consistent with the known upper bounds for the approximation of smoothing distributions using standard Monte Carlo methods. We illustrate our theoretical result with state-of-the art variational solutions based both on the backward parameterization and on alternatives using forward decompositions. This numerical study proposes guidelines for variational inference based on neural networks in state-space models.

Chapter 2

Technical background in sequential Bayesian inference

We now introduce in a more rigorous setting the important takeaways from the field of sequential Bayesian inference, in particular the notions that intervene in the methodological contributions. We first present known results and decompositions that intervene in classical HMM literature. Then, we recall the main approximations schemes which stem from these fundamental recursions, in particular in the context of sequential Monte carlo methods. Finally, we recall the main ideas behind the variational inference methodology, and detail the main approaches that have been developed to extend it to the sequential setting and the existing links with classical methods.

Notations. In all that follows, we assume an underlying probability space $(\Omega, \mathcal{F}, \mathbb{P})$. We use formulations such as "a random variable X on X" to refer to a measurable function $X : \Omega \to \mathsf{X}$, where X is a set implicitly equipped with a σ-algebra $\mathcal{X}$ on X.

- Whenever the distribution $\mathbb{P} \circ X^{-1}$ of a $\mathbb{R}^d$-valued random variable X admits a density p w.r.t the Lebesgue measure on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, we may use the notation $p(x)$ both to refer to the distribution of X and the latter density, with x both used as an argument for the density and an identifier of which distribution we refer to (via the lower/uppercase relationship). This will be clear from the context. We use this notation without further warnings whenever the previous requirements are satisfied or whenever such considerations can be abstracted to simplify the reasoning (e.g. more rigorous notation is possible to express the same content but is not more insightful).
- We may use the loose notation $(x, y) \mapsto p(x|y)$ to refer to the conditional distribution of X given Y.
- For any measure ν on a measurable space $(\mathsf{X}, \mathcal{X})$ and any measurable function h on X, taking values on some set H, we write $\nu h = \int h(x)\nu(\mathrm{d}x)$.
- For any measurable spaces $(\mathsf{X}, \mathcal{X})$ and $(\mathsf{Y}, \mathcal{Y})$, any measure ν on $(\mathsf{X}, \mathcal{X})$, any kernel $K : (\mathsf{X}, \mathcal{Y}) \to \mathbb{R}_+$ and any measurable function h on $\mathsf{X} \times \mathsf{Y}$ taking values on some set H, we denote $Kh : x \mapsto \int h(x, y)K(x, \mathrm{d}y)$ and $\nu Kh = \int h(x, y)\nu(\mathrm{d}x)K(x, \mathrm{d}y)$.
- If for all $x \in \mathsf{X}$, $K(x, \cdot)$ has a density $k(x, \cdot)$ with respect to a reference measure ν, we write $kh : x \mapsto \int h(x, y)K(x, \mathrm{d}y) = \int h(x, y)k(x, y)\nu(\mathrm{d}y)$.

- We denote with $\mathbb{1}$ the constant function which equals 1 on its input space.
- Indices of sequences are implicitly in $\mathbb{N}$ and $(u_s)_{0\leq s\leq t}$ will be shortened into $(u_s)_{s\leq t}$.
- For any $(s_1, s_2) \in \mathbb{N}^2$, we denote $u_{s_1:s_2}$ the collection $(u_{s_1}, \dots, u_{s_2})$.
- We use the notation $\{y^k\}_{k\leq K}$ to refer to collections of independent observations, as opposed to sequences $y_{0:t}$ of temporally dependent observations. In some cases, each observation will be a sequence, which we denote $y^k = (y_0^k, \dots y_t^k)$ for $k \leq K$.
- In general, we will use the letter "s" to index quantities relatively to a final index "t".

In the next paragraphs, we start by recalling (using loose notations) some basic elements of Bayesian inference in latent data models and the setting we place ourselves in for the rest of the chapter.

Latent models and Bayesian inference. We consider probabilistic models of the data where observations are viewed as random variables $Y \in \mathsf{Y}$ with unknown distribution $p(y)$, Y being the observation space. Taking the point of view of latent data models, we assume the existence of another non-observed random variable $X \in \mathsf{X}$ which is used to specify the data model via $p(y) = \int_{x\in\mathsf{X}} p(x, y)\mathrm{d}x$, where $p(x, y)$ is the joint distribution of (X, Y) in $\mathsf{X} \times \mathsf{Y}$. From $p(x, y) = p(x)p(y|x)$, the model is fully defined via the following quantities.

- The conditional distribution $p(y|x)$, which is usually understood as specifying the measurement process of the observations from underlying causes.
- The distribution $p(x)$, which is usually referred to as the *prior* and can be used to contrain the statistical problem (e.g. with known structure or internal dependencies in the data).

In this context, Bayesian inference frames prediction tasks as posterior estimates, i.e. quantities derived from the posterior distribution

$$p(x|y) = \frac{p(x, y)}{p(y)} .$$

In this work, we focus on estimates formulated as *expectations* under the posterior distribution (as opposed to maximum a posteri estimates).

Parametric models and maximum likelihood estimation. We only consider parametric latent data models, i.e. given a parameter space Θ, data models fully specified given $\theta \in \Theta$ because the prior and observation models are parameterized by θ. We use the notation $p^\theta(x)$ to indicate that a random variable X has a distribution depending on all or a subset of θ. In this context, given a dataset of observations $\{y^k\}_{k\leq K}$, empirical parametric maximum likelihood estimation (MLE) aims at recovering

$$\theta^* = \arg\max_{\theta\in\Theta} \log p^\theta(y^1, \dots, y^K) .$$

In this work, we do not focus specifically on this estimation problem, but links between inference and MLE will be discussed whenever relevant, especially when the computations involve in MLE are related to inference problems.

2.1 State-space models

In latent sequential models, observations are viewed as realisations of a stochastic process $(Y_t)_{t\geq 0}$ in Y whose law is unknown but derived from that of a joint process $(X_t, Y_t)_{t\geq 0}$. For parametric models, the finite-dimensional distributions of the joint sequences $\{(X_{0:t}, Y_{0:t})\}_{t\geq 0}$ are entirely defined given $\theta \in \Theta$ and characterize the generative model of the data. In this setting, a popular class of models are hidden Markov models or *state-space models* (SSM), which satisfy the two following conditions.

- The latent process $(X_t)_{t\geq 0}$ is a Markov chain, i.e. for all $t \geq 0$, the distribution of X_t given $X_{0:t-1}$ is independant of $X_{0:t-2}$. In this context, we denote with χ^θ the distribution of X_0, and for all $t > 0$, M_t^θ is the Markov kernel such that $M_t^\theta(X_{t-1}, \cdot)$ is the conditional distribution of X_t given X_{t-1}. In general, the kernels $(M_t^\theta)_{t\geq 0}$ are called the *transition* kernels. In this work, we also denote with $M_{0:t}^\theta$ the joint distribution of the latent process defined as
$$M_{0:t}^\theta(\mathrm{d}x_{0:t}) = \chi^\theta(\mathrm{d}x_0) \prod_{s=1}^{t} M_t^\theta(x_{s-1}, \mathrm{d}x_s) \ .$$
- Observations are independant conditionally on the latent variables: for all $s \geq 0$, and $s_1 \leq s \leq s_2$, the distribution of Y_s given $X_{s_1:s_2}$ depends only on X_s, and we denote with $G^\theta(X_s, \cdot)$ the conditional distribution of Y_s given X_s, sometimes referred to as *emission* distribution.

A common class of SSMs are those whose transition kernels and emission distributions can be described via equations of the form

$$\begin{aligned} X_t &= \mathsf{F}_t^\theta(X_{t-1}) + \eta_t^\theta \ , \\ Y_t &= \mathsf{G}_t^\theta(X_t) + \epsilon_t^\theta \ , \end{aligned}$$

where $(\eta_t^\theta, \epsilon_t^\theta)_{t\geq 1}$ are typically a sequence of zero-mean random variables understood as noising processes, and $(\mathsf{F}_t^\theta, \mathsf{G}_t^\theta)_{t\geq 1}$ are deterministic functions. However, note that general state-space models are in no way restricted to this class. Other typical situations include $(X_t)_{t\geq 0}$ being governed by a stochastical differential equation, or involving non-addtive observation noise.

In all that follows, we consider the setting of *fully dominated* SSMs, i.e. the initial distribution and all transition kernels admit densities $(m_t^\theta)_{t\geq 0}$ w.r.t to a reference measure on X (where, by convention, $x_0 \mapsto m_0^\theta(x_0)$ is the density of χ^θ), and for all $x \in \mathsf{X}$, $G^\theta(x, \cdot)$ admits a density $g^\theta(x, \cdot)$ w.r.t to a reference measure on Y. Let $y_{0:t}$ be a sequence of $t+1$ observations from the process $(Y_t)_{t\geq 0}$, we adopt a common notational convention of SSM literature and hide the dependency on the observations by introducing measurable functions $g_s^\theta : x_s \mapsto g^\theta(x_s, y_s)$ for all $s \leq t$. In this context, denote $\ell_0^\theta : x_0 \mapsto m_0^\theta(x_0) g_0^\theta(x_0)$ and for all $1 \leq s \leq t$ the functions

$$\ell_s^\theta : (x_{s-1}, x_s) \mapsto m_s^\theta(x_{s-1}, x_s) g_s^\theta(x_s) \ ,$$

which are densities of kernels on X. Finally, define

$$\ell_{0:t}^\theta : x_{0:t} \mapsto \prod_{s=0}^{t} \ell_s^\theta(x_{s-1}, x_s) \ .$$

In SSMs, the joint process $(X_t, Y_t)_{t\geq 0}$ is also a Markov process on $\mathsf{X}\times\mathsf{Y}$. Under the previous assumption, the distribution of $(X_{0:t}, Y_{0:t})$ at all $t \geq 0$ has a density defined as

$$p^\theta_{0:t} : (x_{0:t}, y_{0:t}) \mapsto \ell^\theta_{0:t}(x_{0:t}) \ .$$

In this context, inference tasks are formulated as statistical estimates under the posterior distributions of the states given observations, but various denominations are used depending on the sequence of latents involved and the conditionning on the data. For any $0 \leq s_1 \leq s_2 \leq t$ the *joint smoothing distribution* $\phi^\theta_{s_1:s_2|t}$ is the conditional law of $X_{s_1:s_2}$ given $Y_{0:t}$.

- When not indicated, the smoothing distribution refers to the special case $s_1 = 0, s_2 = t$, i.e. the posterior distribution $\phi^\theta_{0:t|t}$ of the entire latent sequence $X_{0:t}$ given $Y_{0:t}$, which we may abbreviate as $\phi^\theta_{0:t}$ to shorten notations.
- When $s_1 = s_2 = s$, $\phi^\theta_{s|t}$ is the *marginal smoothing* of X_s given $Y_{0:t}$.
- In the latter case with $s = t$, the marginal $\phi^\theta_{t|t}$ is the *filtering distribution* at t, which we may simply abbreviate with ϕ^θ_t.

In SSMs, estimation of such distributions is similarly referred to as smoothing, marginal smoothing and filtering, respectively. Following Bayes formula, the joint smoothing distribution is defined, for any measurable function h on X^{t+1}, as

$$\phi^\theta_{0:t} h = \frac{\ell^\theta_{0:t} h}{\ell^\theta_{0:t} \mathbb{1}_{\mathsf{X}^{t+1}}} \propto \ell^\theta_{0:t} h$$

which reveals a recursion

$$\phi^\theta_{0:t} h \propto \phi^\theta_{0:t-1} \ell^\theta_t h \ , \tag{2.1}$$

between the smoothing distributions at succesive timesteps. This property, which results from the particular dependencies of SSMs, is at the core of most mathematical decompositions that allow to perform inference in a sequential manner.

Additionally, the *likelihood* is the function denoted as L^θ_t whose evaluation on $y_{0:t}$ is precisely the normalizing constant of the smoothing distribution at t, i.e. $\mathsf{L}^\theta_t(y_{0:t}) = \ell^\theta_{0:t} \mathbb{1}_{\mathsf{X}^{t+1}}$. As in general Bayesian inference, the log of this function evaluated at $y_{0:t}$, denoted

$$l^\theta_t = \log \mathsf{L}^\theta_t(y_{0:t}) \ ,$$

is a central quantity in parameter estimation under the methodology of maximum likelihood estimation.

2.1.1 Joint backward smoothing in state-space problems

In literature on SSMs, a common practice is to leverage decompositions of the targeted distributions to derive efficient inference algorithms. In this work, we are mainly interested with the joint smoothing distributions $\phi^\theta_{0:t}$, and as such we focus on known decompositions of latter. While intermediate quantities, such as marginal smoothing or filtering distributions, are relevant on their own and have been the central focus of many estimation problems, we only discuss these when relevant with respect to the joint smoothing problem.

The backward factorization

Given the recursive construction of the smoothing distributions via the unnormalized kernels $(\ell_s^\theta)_{s\leq t}$ as described above, a first approach which comes to mind to derive a recursive algorithm is to renormalize (2.1) at each timestep, i.e. for any functional h

$$\phi_{0:t}^\theta h = \frac{\phi_{0:t-1}^\theta \ell_t^\theta h}{\phi_{0:t-1}^\theta \ell_t^\theta \mathbb{1}_{\mathsf{X}^{t+1}}} \ . \tag{2.2}$$

However, computing this recursion or building approximations from it is seldom done, because the normalization steps involve integrals on increasingly large supports over time. Instead, another approach is to analyse the dependencies induced by SSMs and leverage other decompositions of the smoothing distributions. In that regard, a known property of SSMs is that the reverse process $(X_{t-s})_{s\leq t}$ is also a Markov chain conditionally on $Y_{0:t}$, such that the joint smoothing distribution $\phi_{0:t}^\theta$ can be decomposed via the following so-called *backward* factorization

$$\phi_{0:t}^\theta(\mathrm{d}x_{0:t}) = \phi_t^\theta(\mathrm{d}x_t) \prod_{s=1}^{t} B_{s-1|s}^\theta(x_s, \mathrm{d}x_{s-1}) \ , \tag{2.3}$$

where $(B_{s-1|s}^\theta)_{1\leq s\leq t}$ is a sequence of *inhomogeneous* Markov kernels known as the *backward kernels*, such that, at s, $B_{s-1|s}^\theta(X_s, \cdot)$ is the conditional distribution of X_{s-1} given X_s and $Y_{0:t}$.

While approaching the smoothing problem from the point of view of the reverse chain may seem conceptually unnatural, a known result (which can be derived from the conditional independence properties of SSMs) is that, for all $1 \leq s \leq t$, the previous conditional distributions are independent of observations $(Y_s)_{s\geq t}$, and as such the backward kernel $B_{s-1|s}$ only depends on observations up to $s-1$. Consequently, the factorization (2.3) decomposes the smoothing distribution in terms of normalized quantities which only depend on the current set of observations, and can form the basis for recursive algorithms. In particular, the backward kernels are directly related to the filtering distributions at intermediate timesteps. In the particular case of fully-dominated SSMs, the smoothing distribution and all its marginals admit densities and we will keep the previous notations to refer to both the densities and the distributions. In this context, for all $1 \leq s \leq t$, the backward kernels have density defined as

$$b_{s-1|s}^\theta : (x_s, x_{s-1}) \mapsto \frac{\phi_{s-1}^\theta(x_{s-1}) m_s^\theta(x_{s-1}, x_s)}{\int_{\mathsf{X}} \phi_{s-1}^\theta(x_{s-1}) m_s^\theta(x_{s-1}, x_s) \mathrm{d}x_{s-1}} \ . \tag{2.4}$$

Conveniently, the sequence of backward kernels is therefore obtained as a byproduct of the filtering distributions $(\phi_s^\theta)_{s\leq t}$, whose recursive estimation becomes a central component to build the joint smoothing distribution $\phi_{0:t}^\theta$.

The filtering recursions

Marginalizing (2.1) over $X_{0:t-2}$ yields a simple relationship between the filtering distributions: for any measurable function $h : \mathsf{X} \to \mathsf{H}$,

$$\phi_t^\theta h = \frac{\phi_{t-1}^\theta \ell_t^\theta h}{\phi_{t-1}^\theta \ell_t^\theta \mathbb{1}_{\mathsf{X}}} \ . \tag{2.5}$$

While (2.5) can form the basis to approximate the filtering distributions recursively, in general the computations are rather understood as divided into two distinct steps. At t, first compute

the *predictive* distribution of $\phi^\theta_{t|t-1}$ of X_t given $Y_{0:t-1}$, denoted as $\bar{\phi}^\theta_t$, and obtained from the previous filtering distribution, for any measurable function $h : \mathsf{X} \to \mathsf{H}$, via

$$\bar{\phi}^\theta_t h = \phi^\theta_{t-1} m^\theta_t h \ , \tag{2.6}$$

then compute the new filtering distribution given by

$$\phi^\theta_t h = \frac{\bar{\phi}^\theta_t g^\theta_t h}{\bar{\phi}^\theta_t g^\theta_t \mathbb{1}_{\mathsf{X}}} \ . \tag{2.7}$$

In general, (2.6) is called the *predict* step while (2.7) is the *update* step. The main idea behind this two-step procedure is to first propagate the previous distribution at the current timestep given the dynamics of the model, then introduce the new observation. Moreover, as its name suggests, the predictive distribution is the main mathematical object to reason about the state at t given information up to $t-1$, and is therefore crucial on its own. In general, the process of repeatedly computing the previous equations over time is usually referred to as *forward filtering*.

2.1.2 Smoothed expectations of additive state functionals

As in general Bayesian inference, probabilistic estimates from SSMs are expressed as statistics under the posterior distributions. In the context of joint smoothing, this means that *smoothing expectations* of the form

$$\phi^\theta_{0:t} h = \mathbb{E}\left[h(X_{0:t}) | Y_{0:t}\right] = \int_{\mathsf{X}^{t+1}} h(x_{0:t}) \phi^\theta_{0:t}(\mathrm{d}x_{0:t})$$

are the quantities of interest, where the measurable function h on X^{t+1} depends on the targeted quantity. In the context of state-space models, a particular class of functions are *additive state functionals* $h_{0:t} : \mathsf{X}^{t+1} \to \mathsf{H}$ which decompose into

$$h_{0:t} : x_{0:t} \mapsto \sum_{s=1}^{t} \tilde{h}_s(x_{s-1}, x_s) \ , \tag{2.8}$$

where $\tilde{h}_s : \mathsf{X} \times \mathsf{X} \to \mathsf{H}$. Here, the term "functional" is usually used in place of function because $h_{0:t}$ is sometimes viewed as a higher-order function on the space of probability measures $h_{0:t} : \mathcal{P}(\mathsf{X}^{t+1}) \to \mathsf{H}$ whose evaluation on $\nu \in \mathcal{P}(\mathsf{X}^{t+1})$ is a statistic. In the rest of this manuscript, we rather view them as measurable functions but keep the frequently used terminology of referring to them as state functionals.

In SSMs, many inference tasks can be expressed as expectations of additive state functionals under the joint smoothing distributions. For instance, a bayesian estimator of the smoothing marginal $\phi^\theta_{s'|t}$ state for some timestep $s' \leq t$ is given via

$$\hat{X}_{s'} = \mathbb{E}\left[X_{s'} | Y_{0:t}\right] \ ,$$

which corresponds to $\phi^\theta_{0:t} h^{(s')}_{0:t}$ where $h^{(s')}_{0:t} = \sum_{s=1}^{t} \tilde{h}^{(s')}_s$ with components

$$\tilde{h}^{(s')}_s : (x_{s-1}, x_s) = x_s \mathbb{1}_{s=s'} \ ,$$

for all $s \leq t$. Furthermore, when θ is unknown, MLE can be obtained via maximization of functions which also correspond to additive state functionals. In the Expectation-Maximization algorithm, the central quantity at each timestep is the function $\theta \mapsto Q(\theta, \theta')$ defined, for all $(\theta, \theta') \in \Theta^2$ as

$$Q(\theta, \theta') = \mathbb{E}^{\theta'}\left[\sum_{s=1}^{t} \log \ell_s^{\theta}(X_{s-1}, X_s) \middle| Y_{0:t}\right] = \phi_{0:t}^{\theta'} h_{0:t}^{\mathsf{EM}} ,$$

where $h_{0:t}^{\mathsf{EM}}$ is the additive state functional with components $\tilde{h}_s^{\mathsf{EM}}(x_{s-1}, x_s) = \log \ell_s^{\theta}(x_{s-1}, x_s)$. Similarly, gradient-based MLE involves computation of

$$\nabla_\theta \log \mathsf{L}_t^{\theta} = \mathbb{E}^{\theta}\left[\sum_{s=1}^{t} \nabla_\theta \log \ell_s^{\theta}(X_{s-1}, X_s) \middle| Y_{0:t}\right] = \phi_{0:t}^{\theta} h_{0:t}^{\mathsf{MLE}} . \tag{2.9}$$

where $h_{0:t}^{\mathsf{MLE}}$ is the additive state functional with components $\tilde{h}_s(x_{s-1}, x_s) = \nabla_\theta \log \ell_s^{\theta}(x_{s-1}, x_s)$.

All in all, most predictions in SSMs can be formulated as smoothing expectations of additive state functionals, and the computation of such quantities is often referred to as *additive smoothing*.

Recursive additive smoothing

Under the backward factorization (2.3), a very convenient aspect of additive smoothing is that it can be computed recursively. This can be seen when applying the tower property of expectations to the joint smoothing distribution: for all $s \leq t$, one has

$$\phi_{0:t}^{\theta} h_{0:t} = \mathbb{E}_{\phi_t^{\theta}}\left[H_t^{\theta}(X_t)\right] , \tag{2.10}$$

where, for all $s \leq t$,

$$H_s^{\theta}(X_s) = \mathbb{E}_{\phi_{0:s}^{\theta}}\left[h_{0:s}(X_{0:s}) | X_s\right] .$$

In the context of additive state functionals, the conditional expectations of the previous form can be linked via functional recursions using the sequence of backward kernels $(B_{s-1|s}^{\theta})_{1 \leq s \leq t}$. By applying the tower property again and using the conditional independance properties of the reverse chain, for all $s \leq t$,

$$H_s^{\theta}(x_s) = \mathbb{E}_{B_{s-1|s}^{\theta}(x_s, \cdot)}\left[H_{s-1}^{\theta}(X_{s-1}) + \tilde{h}_s(X_{s-1}, x_s)\right] . \tag{2.11}$$

While the use of such recursions in SSMs literature is rather recent, those have been ubiquitous in other fields such as stochastic control [BS96] where they correspond to special cases of dynamic programming in discrete Markov processes. Conveniently for the task of smoothing, since, at s, the backward kernel $B_{s-1|s}^{\theta}$ only depends on observations up to $s-1$, the computation of H_s^{θ} does not involve observations for timesteps $s' > s$. In turn, the sequence of smoothing expectations $(\phi_{0:s}^{\theta} h_{0:s})_{s \leq t}$ can be obtained at any timestep by sequentially applying (2.11) and (2.10), a procedure which is usually referred to as *recursive additive smoothing* or "forward-only smoothing". Additionally, since the backward kernels can be defined in terms of the filtering distributions $(\phi_s^{\theta})_{s \leq t}$, all of the previous operations can be performed concurrently with the previously mentioned filtering recursions.

Notes on recursive maximum likelihood estimation

As already mentioned, additive smoothing is a central component in MLE methods, which relies on quantities that can be expressed as additive state functionals. We now briefly present how elements of the previous subsection can be used to enable *online* learning of the model parameter θ in the context of gradient-based MLE methods, a topic known as *recursive maximum likelihood estimation* (RMLE) which involves derivations relevant to understand the work of this thesis.

In batch (offline) gradient-based MLE, a sequence of fixed length $Y_{0:t}$ is provided, and updates on the parameter θ are of the form

$$\theta_{k+1} = \theta_k + \gamma_{t+1} \nabla_\theta \log \mathsf{L}_t^\theta(Y_{0:t})\big|_{\theta_k} \ ,$$

which involves recomputing the gradient of $l_t^\theta = \log \mathsf{L}_t^\theta(Y_{0:t})$ at every update. For very large observation sequences $Y_{0:t}$, or in streaming scenarios where observations $\{Y_t\}_{t\geq 0}$ are only available progressively, RMLE methods consider instead the *incremental* log-likelihood $r_t^\theta = \log P^\theta_{Y_t|Y_{0:t-1}}(Y_t)$, where $P^\theta_{Y_t|Y_{0:t-1}}$ is the density of Y_t given $Y_{0:t-1}$. Indeed, the relationship

$$l_t^\theta = \sum_{s=1}^{t} r_s^\theta \ ,$$

which is simply obtained via conditionning, suggests that the sequence $(r_t^\theta)_{t\geq 0}$ may be used to extended the standard methodology of stochastic gradient method to the sequential setting via updates of the form

$$\theta_{t+1} = \theta_t + \gamma_{t+1} \nabla_\theta r_t^\theta\big|_{\theta_t} \ ,$$

which are performed at every timestep $t \geq 0$. From there, estimation of $(\nabla_\theta r_t^\theta)_{t\geq 0}$ becomes the central challenge in RMLE methods, which can be tackled via two different approaches.

1. A first possibility is simply to write that, at $t \geq 0$

$$\nabla_\theta r_t^\theta\big|_{\theta_t} = \nabla_\theta \left(l_t^\theta - l_{t-1}^\theta\right)\big|_{\theta_t} \approx \nabla_\theta l_t^\theta\big|_{\theta_t} - \nabla_\theta l_{t-1}^\theta\big|_{\theta_{t-1}} \ .$$

   Then, computation of $\nabla_\theta l_t^\theta\big|_{\theta_t}$ can be performed via the additive decomposition (2.9) and the recursive additive smoothing techniques described above, further assuming that $\phi_s^\theta \approx \phi_s^{\theta_s}$ and $B_{s-1|s}^\theta \approx B_{s-1|s}^{\theta_{s-1}}$ for all $s < t$, such that updates of the form (2.11) can be carried without recomputation of the previous quantities.

2. Another approach is to remark that $r_t^\theta = \log \bar{\phi}_t^\theta g_t^\theta$ and derive, via simple differentiation rules, another form of the gradient of the incremental log-likelihood

$$\nabla_\theta r_t^\theta = \frac{\bar{\phi}_t^\theta \left(\left(\nabla_\theta \bar{\phi}_t^\theta\right) g_t^\theta - \nabla_\theta g_t^\theta\right)}{\bar{\phi}_t^\theta g_t^\theta}$$

   which involves expectations $\bar{\phi}_t^\theta h = \mathbb{E}\left[h(X_t)|Y_{0:t-1} = y_{0:t-1}\right]$ under the *predictive* distribution, as well as the quantity $\eta_t^\theta = \nabla_\theta \bar{\phi}_t^\theta$ which is a signed measure sometimes referred to as the *tangent filter*. In practice, the latter can also be computed recursively via the methodology of additive smoothing.

2.1.3 Exact inference and Kalman-based extensions

For most SSMs, almost none of the computations developed in the previous section can be computed exactly (because the integrals involved are intractable) and it is therefore necessary to resort to approximations. However, a well-known result for inference in SSMs is that whenever χ^θ is a Gaussian distribution and for all $s \geq 1$, both m_s^θ and g_s^θ are linear and Gaussian kernels, then for any $0 \leq s_1 \leq s_2 \leq t$, $\phi^\theta_{s_1:s_2|t}$ is Gaussian, and the parameters can be analytically computed given θ and $y_{0:t}$. This observation, which is based on fundamental properties of Gaussian vectors (i.e. stability under conditionning and marginalisation) leads to the well-known *Kalman filtering and smoothing* algorithms which are fundamental in SSM literature (see e.g. [CMR05, section 4.2]). In brief, in such linear and Gaussian setting, all of the computations described previously translate into analytical recursions on Gaussian parameters.

- The filtering distributions are Gaussian: for all $s \leq t$,

$$\phi_s^\theta = \mathcal{N}(\mu_s, \Sigma_s)$$

  and the parameters $(\mu_s, \Sigma_s)_{s \leq t}$ can be obtained recursively on the observations $\{y_s\}_{s \leq t}$ via analytical predict and update steps from Section 2.1.1.

- The backward kernels are linear and Gaussian kernels: for all $1 \leq s \leq t$ and $x_s \in \mathsf{X}$,

$$B^\theta_{s-1|s}(x_s, \cdot) \sim \mathcal{N}(\overleftarrow{A}_{s-1|s} x_s + \overleftarrow{a}_{s-1|s}, \overleftarrow{\Sigma}_{s-1|s})$$

  and the parameters can be computed analyically from $(\mu_{s-1}, \Sigma_{s-1})$ and those of m_s^θ because (2.4) admits a closed-form expression.

- For $s' \leq t$, any smoothing marginal is a Gaussian a $\phi^\theta_{s'|t} \sim \mathcal{N}(\mu_{s'|t}, \Sigma_{s'|t})$ and its parameters can by analytically derived from (μ_t, Σ_t) and $(\overleftarrow{A}_s, \overleftarrow{a}_s, \overleftarrow{\Sigma}_s)_{s' \leq s \leq t}$ because marginalization of (2.3) admits a closed-form solution.

While linear and Gaussian models can be sufficient in simple settings, many real world observations can only be realistically described with nonlinear mappings or via more complicated measurement noise, e.g. via equations derived from domain-specific knowledge on the data. Nonetheless, when the departure from nonlinearity and non-gaussianity is mild, a common technique is to retain the practicality of the analytical Kalman recursions via additional simplifications.

Extended Kalman filter. Whenever the transition and emission models still involve Gaussian noises $(\eta^\theta, \epsilon^\theta)$ but contain nonlinear mappinps, e.g. for all $t \geq 0$, $X_t = \mathsf{F}^\theta(X_{t-1}) + \eta^\theta$ and $Y_t = \mathsf{G}^\theta(X_t) + \epsilon^\theta$ where $(\mathsf{F}^\theta, \mathsf{G}^\theta)$ are nonlinear, one technique, commonly referred to as *extended* Kalman filtering, is to propagate parameters using the Kalman filtering recursions on a Taylor expansion of model around the current set of parameters, e.g. for the transition model, one may assume

$$X_t \approx \mathsf{F}^\theta(\mu_{t-1}) + (X_{t-1} - \mu_{t-1}) \left(\partial_X \mathsf{F}^\theta\right)(\mu_{t-1}) + \eta^\theta$$

and similarly for the observation model around the parameters of the predictive distribution $\phi^\theta_{t|t-1}$. Other extensions of the Kalman filter have been proposed, for instance to avoid using linearization techniques, see for instance the unscented Kalman filter [WV00].

While, under a linear and Gaussian SSM, the Kalman filtering and smoothing recursions yield optimal estimates of the smoothing marginal distributions, the above deterministic approximations are biased. While they can be sufficient in simple settings, such bias can become prohibitive whenever the underlying model comprises highly nonlinear dynamics or measurement models.

2.2 Sequential Monte Carlo

In complex settings where the previous approximations fail, another approach consists in approximating all quantities of the previous section using Monte Carlo methods and replace intractable distributions π with weighted measures on discrete supports, of the form

$$\hat{\pi}^N = \sum_{i=1}^{N} \bar{\omega}^i \delta_{\xi^i} \ , \tag{2.12}$$

where $\sum_{i=1}^{N} \bar{\omega}^i = 1$ and $\{\xi^i\}_{i \leq N}$ are random samples. In such methods, the computation of the weights and particles $\{\bar{\omega}^i, \xi^i\}_{i \leq N}$ is the main challenge, as all estimates are readily derived from them (e.g. expectations).

2.2.1 Elements of importance sampling

Given a probability distribution $\pi \in \mathcal{P}(\mathsf{X})$ and a measurable function $h : \mathsf{X} \to \mathsf{H}$, the original idea behind Monte Carlo methods is that πh can be estimated by $N^{-1} \sum_{i=1}^{N} f(\xi^i)$ given i.i.d samples $(\xi^i)_{i \leq N}$ from π. From there, $\hat{\pi}^N = N^{-1} \sum_{i=1}^{N} \delta_{\xi^i}$ may be used as an approximation to the original distribution π. This estimator is unbiased and consistant, however the error decreases in $N^{-1/2}$, and in practice, the variance can be large.

In many situations, it is either impossible to sample directly from π, or the slow decrease in the approximation error is prohibitive. Whenever one has access to another distribution $q \in \mathcal{P}(\mathsf{X})$ which is absolutely continuous w.r.t π, then the Radon-Nikodym theorem allows to consider

$$\hat{\pi}_q^{N,IS} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{\mathrm{d}\pi}{\mathrm{d}q} \right) \left(\xi^i \right) \delta_{\xi^i}$$

as another possible approximation of π given i.i.d samples $(\xi^i)_{i \leq N}$ from q, and idea formally referred to as *importance sampling* (IS). In practice, $\hat{\pi}_q^{N,IS}$ converges in distribution to π as $N \to \infty$ [RCC99]. In literature on importance sampling, the distribution q is called the *importance distribution* or the *proposal*, and choosing this distribution is the main challenge. When both π and q admit densities f_π and f_q w.r.t to some dominiting measure π, then $\mathrm{d}\pi / \mathrm{d}q = f_\pi / f_q$ and the values $\omega^i = f_\pi(\xi^i) / f_q(\xi^i)$ for all $i \leq N$ are called the *importance weights*.

When considering only expectations w.r.t to a particular measurable function h_0 and when π has density f_π w.r.t to the Lebesgue measure, it can be shown that the optimal proposal q^* has density $f_{q*} \propto |h_0| f_\pi$, [Owe13] such that most methods will focus on building proposals that concentrate mass on regions of X where both h_0 takes larges values and where π has large mass.

Nonetheless, in most situations of interest, the target distribution cannot be evaluated because it is only known up to normalization, i.e. for all $h : \mathsf{X} \to \mathsf{H}$,

$$\pi h = \frac{1}{Z_\pi} \int_{\mathsf{X}} h(x) \tilde{\pi}(\mathrm{d}x) \ ,$$

where $\tilde{\pi}$ is some unnormalized measure on X and Z_π is unknown. Given a proposal q, denoting $f_{\tilde{\pi}}$ and f_q the densities of $\tilde{\pi}$ and q, respectively, *self-normalized importance sampling* (SNIS) considers

$$\hat{\pi}_q^{N,SNIS} = \sum_{i=1}^{N} \frac{\omega^i}{\sum_{j=1}^{N} \omega^j} \delta_{\xi^i} \ , \tag{2.13}$$

where for all $i \leq N$, $\omega^i = f_{\tilde{\pi}}(\xi^i)/f_q(\xi^i)$ and the quantities $\bar{\omega}^i = \omega^i / \sum_{j=1}^{N} \omega^j$ are such that $\sum_{i=1}^{N} \bar{\omega}^i = 1$, hence called the normalized weights.

While originally used to reduce the number of samples in Monte Carlo estimation [KM53], both IS and SNIS have recently arised as key components of many estimation problems, e.g [BGS15; Bak+19], and most importantly they play an essential role in sampled-based approximate inference for SSMs, as shown in next section. Contrary to the unnormalized version, SNIS is biased [Aga+17], however the associated empirical distribution $\hat{\pi}_q^{N,SNIS}$ still converges to π in distribution.

The performances of IS and SNIS are strongly tied given the bias-variance tradeoff (and as such SNIS may sometimes be used even when π can be evaluated to trade variance for bias). In any case, the performance of importance sampling is generally measured by quantifying the variance of the importance weights. When several functions need to be measured against π, it is often understood that a good proposal q is one close to π in the statistical sense, while staying absolutely continuous w.r.t the latter. To this aim, many newer solutions in importance sampling consider *adaptive* proposals [EM21], i.e. ones explicitly tuned to target regions of high probability mass under π via with optimization.

While it is generally understood that the dimension of the sampling space impacts the performance of both IS and SNIS, scalability with dimension strongly varies with the choice of proposal q. Recently [Aga+17] showed that the second moment of the weights $\mathbb{E}_q \left[\mathrm{d}\pi / \mathrm{d}q(X^2) \right]$ is a key quantity in importance sampling which can be considered as an "intrinsic" dimension for the related estimators, i.e. the number of samples necessary to obtain good approximations $\hat{\pi}_q^{N,\mathsf{IS}}$ and $\hat{\pi}_q^{N,\mathsf{SNIS}}$ is essentially governed by this quantity rather than directly by the state and data dimensions.

2.2.2 Backward particle smoothing

In the specific case of SSMs, we consider as before the joint smoothing distribution $\phi_{0:t}^\theta$. Recalling that $\phi_{0:t}^\theta h \propto \ell_{0:t}^\theta h$, one possiblity to obtain a sampled-based approximation would be to consider some importance distribution $q_{0:t}$ on X^{t+1} and a SNIS estimate with weights $\bar{\omega}_{0:t}^i \propto \left(\ell_{0:t}^\theta / q_{0:t} \right) (\xi_{0:t}^i)$. However, this conceptually simple approach fails in practice, notably because the effective sampling space is X^{t+1} and therefore very high-dimensional for long sequences and/or p large. As such, building a good proposal $q_{0:t}$ is near impossible without further leveraging structural properties of the data.

When the targeted density admits a temporal factorization, *sequential importance sampling*

(SIS) introduces proposals of the form

$$q_{0:t} = q_0 \prod_{s=1}^{t} q_{s|s-1} \ ,$$

with $q_0 \in \mathcal{P}(\mathsf{X})$ and $(q_{s|s-1})_{1\leq s\leq t}$ Markov kernels in $\mathsf{X} \times \mathsf{X}$. In this case, both the samples $\xi_{0:t}^i \sim q_{0:t}$ and the importance weights can be obtained sequentially. While not restricted to SSMs, these methods are particularly suited in this case because they leverage the temporal structure of the generative model. For the joint smoothing distribution, with samples $\xi_0^i \sim q_0$ and for all $1 \leq s \leq t, \xi_t^i \sim q_{s|s-1}(\xi_{s-1}^i, \cdot)$, the importance weights indeed become

$$\bar{\omega}_{0:t}^i \propto \left(\frac{\ell_0^\theta \prod_{s=1}^t \ell_s^\theta}{q_0 \prod_{s=1}^t q_{s|s-1}} \right) \left(\xi_{0:t}^i\right) \ ,$$

which can be rewritten into $\bar{\omega}_{0:t}^i \propto \omega_0^i \prod_{s=1}^t \omega_{s|s-1}^i$ with $\omega_0^i = \ell_0^\theta(\xi_0^i)/q_0(\xi_0^i)$ and

$$\omega_{s|s-1}^i = \frac{\ell_s^\theta(\xi_{s-1}^i, \xi_s^i)}{q_{s|s-1}(\xi_{s-1}^i, \xi_s^i)} \ .$$

In the context of SSMs, SIS therefore fits well with the recursive nature of the smoothing distribution given by (2.1), and allows previous samples to be re-used across timesteps. Given a set $\{\bar{\omega}_{0:t}^i, \xi_{0:t}^i\}_{i\leq N}$ approximating $\phi_{0:t}^\theta$, the joint distribution at the next timestep $\phi_{0:t+1|t+1}^\theta$ can be readily approximated by propagating particles $\{\xi_t^i\}_{i\leq N}$, computing the new terms $\{\omega_{t+1|t}^i\}_{i\leq N}$ and updating the normalizing constant, i.e. new samples are $\xi_{t+1}^i \sim q_{t+1|t}(\xi_t^i, \cdot)$ and the new set of normalized weights is

$$\bar{\omega}_{0:t+1}^i = \frac{\omega_{t+1|t}^i \omega_{0:t}^i}{\sum_{j=1}^N \omega_{t+1|t}^j \omega_{0:t}^j} \ .$$

Unfortunately, SIS alone does not solve the curse of dimensionality and the intractability of importance sampling in X^{t+1}, which manifests itself in the increasing degeneracy of the weights $\{\bar{\omega}_{0:t}^i\}_{i\leq N}$ over time (intuitively, a single trajectory will concentrate the majority of the mass in the joint space and this worsens as t grows). As such, the previous methodology cannot be used in itself to solve the joint smoothing problem. The dominant approach, instead, is to take the same path than developed in the previous section by considering, first, approximations of the filtering distributions $(\phi_s^\theta)_{s\leq t}$, from which the backward kernels are also approximated.

Particle filtering

As marginal distributions, the filtering distributions are given for free in the previous scheme, i.e. by discarding at every timestep s the particles $\{\xi_{0:s-1}^i\}_{i\leq N}$ and considering the estimation $\sum_{i=1}^N \bar{\omega}_{0:s}^i \delta_{\xi_s^i}$ of ϕ_s^θ, yet in this case the resulting approximation inherits the degeneracy of the weights. Nonetheless, a major advantage of the filtering problem is that it is possible to adopt the previous methodology while breaking the degeneracy by *resampling* the particles at every timestep. Formally, at s, the weighted approximation $\sum_{i=1}^N \bar{\omega}_{0:s}^i \delta_{\xi_s^i}$ can be replaced by $N^{-1} \sum_{i\in I(\bar{\omega}_{0:s})} \delta_{\xi_s^i}$ where $I(\bar{\omega}_s) = \{i_k\}_{k\leq N}$ with $i_k \sim \mathsf{Cat}(\{\bar{\omega}_{0:s}^i\}_{i\leq N})$ for all $1 \leq k \leq N$, i.e. the probability to be resampled is given by the importance weights. The process of

adding this resampling step to the previously described methodology is known as *sequential importance resampling* and is a fundamental building block of *particle filtering*, which consists in sequentially

1. propagating the particles,
2. computing the importance weights,
3. resampling particles according to the importance weights.

A key consequence of the resampling process is that particle paths - which were previously generated independently under $q_{0:t}$ - now interact, in the sense that, at s, a particle ξ_{s-1}^j may be considered the ancestor of several resampled particles in $(\xi_s^i)_{i \in I(\bar{\omega}_{0:s})}$. For this reason, theoretical aspects behind SMC methods are often approached from the point of view of interacting particle systems [Del04], which is generally more involved than classical analysis of Monte Carlo approximations. Since, through the resampling process, the weights which are sequentially obtained are only related to the filtering distributions, we denote them by $\{\bar{\omega}_t^i\}_{i \leq N}$.

Backward particle approximations

To approximate the backward kernels, one possibility is to leverage their direct relationship with the filtering laws (2.4), and use the sequence of normalized weights and samples $\{\bar{\omega}_s^i, \xi_s^i\}_{s \leq t}$ approximating $(\phi_s^\theta)_{s \leq t}$ to similarly build empirical versions of the backward kernels. At s, the backward kernel $B_{s-1|s}^\theta$ may be approached by considering

$$\hat{B}_{s-1|s}^\theta(x_s, \mathrm{d}x_{s-1}) = \sum_{j=1}^N \bar{w}_{t-1|t}^{\theta,j}(x_s) \delta_{\xi_{t-1}^j}(\mathrm{d}x_{s-1}) \ , \tag{2.14}$$

where for all $x \in \mathsf{X}$ and $j \leq N$,

$$\bar{w}_{t-1|t}^{\theta,j}(x) = \frac{\bar{\omega}_{s-1}^j m_s^\theta(\xi_{s-1}^j, x)}{\sum_{k=1}^N \bar{\omega}_{s-1}^k m_s^\theta(\xi_{s-1}^j, x_s)} \ , \tag{2.15}$$

are called the *backward weights*.

Forward filtering / backward simulation

Given the previous elements, the joint smoothing distribution $\phi_{0:t}^\theta$ may be readily approximted as

$$\hat{\phi}_{0:t}^\theta(\mathrm{d}x_{0:t}) = \hat{\phi}_t^\theta(\mathrm{d}x_t) \left\{ \prod_{s=1}^N \hat{B}_{s-1|s}^\theta(x_s, \mathrm{d}x_{s-1}) \right\} \ , \tag{2.16}$$

which has support $\{\xi_{0:t}^i\}_{i \leq N}$. In itself, this already allows to approximate smoothing expectations of measurable functions $h : \mathsf{X}^{t+1} \to \mathsf{H}$. However, the cardinality of the support of $\phi_{0:t}^\theta$ grows with t, such the number of operations necessary to compute $\hat{\phi}_{0:t}^\theta h$ quickly becomes intractable for realistically long sequences.

Backward sampling. To cope with this limitation, an idea is to consider instead the *forward filtering / backward simulation* approach (FFBSi) [Dou+11], which consists in approximating expectations under $\phi^\theta_{0:t}$ by sampling *backward trajectories* $\{\overleftarrow{\xi}^i_{0:t}\}_{i\leq N}$ using the approximate terminal distribution $\hat{\phi}^\theta_t$ and the approximate Markov kernels $(\hat{B}^\theta_{s-1|s})_{1\leq s\leq t}$. For $i \leq N$, this sampling process can be broadly described as follows.

1. At t, $\overleftarrow{\xi}^i_t = \xi^{i_t}_t$ where $i_t \sim \mathsf{Cat}\left(\left\{\bar{\omega}^j_t\right\}_{j\leq N}\right)$.
2. For all $s \leq t$, $\overleftarrow{\xi}^i_{s-1} = \xi^{i_{s-1}}_{s-1}$ with $i_{s-1} \sim \mathsf{Cat}\left(\left\{\bar{w}^{\theta,j}_{s-1|s}(\overleftarrow{\xi}^i_s)\right\}_{j\leq N}\right)$.

Given a set of such trajectories $\{\overleftarrow{\xi}^i_{0:t}\}_{i\leq N}$, smoothing expectatons can be estimated with plain i.i.d Monte Carlo estimation, i.e. for any $h : \mathsf{X}^{t+1} \to \mathsf{H}$,

$$\phi^\theta_{0:t} h \approx \frac{1}{N}\sum_{i=1}^N h(\overleftarrow{\xi}^i_{0:t}) \tag{2.17}$$

and a good approximation of $\phi^\theta_{0:t}$ is given by the uniformly weighted empirical measure built from the backward trajectories $\{\overleftarrow{\xi}^i_{0:t}\}_{i\leq N}$. For a detailed introduction to this approach, see for example [DMS14].

While the previous scheme is a convenient approach to reduce the computational complexity with respect to the length $t+1$ of the observation sequences, the normalization of the backward weights induces an $O(N^2)$ complexity at each timestep which can become prohibitive when the number of samples is large. To reduce the computational burden, an idea introduced in [OW+17] is to avoid the computation of the normalizing constant (2.15) by noting that, for all $x_s \in \mathsf{X}$,

$$B^\theta_{s-1|s}(x_s, \mathrm{d}x_{s-1}) \propto \phi^\theta_{s-1}(\mathrm{d}x_{s-1}) m^\theta_s(x_{s-1}, x_s) \ , \tag{2.18}$$

and therefore a more efficient version to the FFBSi approach can be obtained by sampling backward indices according to unnormalizing distributions on the index space. At s, given $\xi^i_t \in \mathsf{X}$, the probability $\overleftarrow{p}_s(i,j)$ of sampling $j \in \{1,\dots,N\}$ is such that

$$\overleftarrow{p}_s(i,j) \propto \bar{\omega}^j_{s-1} m^\theta_s(\xi^j_{s-1}, \xi^i_s) \ .$$

In practice, this allows to carry the backward sampling process without computing the backward weights, e.g. by leveraging accept-reject methods on the index space. Recently, a detailed theoretical analysis and extensions with MCMC sampling of these methods was proposed in [DC23].

2.2.3 Particle-based additive smoothing and online methods

When the targeted measurable functions are additive state functionals, plugging the previous particle approximations of the filtering distributions and backward kernels into the recursions of Equation (2.11) directly enables *online* computation of smoothing expectations. Indeed, at s, an approximation of $\hat{H}^\theta_s$ of H^θ_s is readily available as

$$\hat{H}_s^\theta(x_s) = \sum_{j=1}^N \bar{w}_{s-1|s}^{\theta,j}(x_s) \left(H_{s-1}^\theta(\xi_{s-1}^j) + \tilde{h}_s(\xi_{s-1}^j, x_s) \right) .$$

Given this, recent recursive particle smoothing methods such as [DDS10a; OW+17; AA22] propagate approximations of the functions $(H_s^\theta)_{s\leq t}$ on the support of the empirical filtering distributions. For all $i \leq N$, $s \leq t$, $\hat{H}_s^i \approx H_s^\theta(\xi_s^i)$ is defined recursively as

$$\hat{H}_s^i = \sum_{j=1}^N \bar{w}_{s-1|s}^{\theta,j} \left(\hat{H}_{s-1}^j + \tilde{h}_s(\xi_{s-1}^j, \xi_s^i) \right) , \tag{2.19}$$

which allows to compute $\phi_{0:s}h_{0:s}$ at each timestep by considering the Monte Carlo estimate $N^{-1}\sum_{i=1}^N \hat{H}_s^i$. However, computation of equations of the form (2.19) may become prohibitive whenever N is large. Since the set of approximations $\{\hat{H}_t^i\}_{i\leq N}$ are only defined on the support of the filtering distributions $\{\xi_s^i\}_{i\leq N}$, another approach is consider, for all $1 \leq s \leq t$ and $i \leq N$, approximations of $H_t^\theta(\xi_t^i)$ obtained by resampling in $\{\xi_{s-1}^j\}_{j\leq N}$ using backward sampling procedures similar to the previous subsection, i.e. to consider

$$\widehat{H}_t^\theta(\xi_s^i) = \frac{1}{M} \sum_{j\in\mathcal{J}_s^i} \hat{H}_t^j + \tilde{h}_s(\xi_{s-1}^j, \xi_s^i) , \tag{2.20}$$

where $\mathcal{J}_s^i$ is a set of M indices drawn from $\{1, \ldots, N\}$ with probabilities $\{\bar{w}_{s-1|s}^{\theta,j}(\xi_s^i)\}_{j\leq N}$. As previously, it is possible to reduce the computational burden by considering resampling schemes which leverage the construction $\bar{w}_{s-1|s}^{\theta,j}(\xi_s^i) \propto \bar{\omega}_{s-1}^j m_s^\theta(\xi_{s-1}^j, \xi_s^i)$ of the backward weights.

2.2.4 Limitations

A known caveat of SMC methods is that the number of particles necessary to obtain good approximations typically scales exponentially with the model dimensions. In practice, the weights of the particle filter can be experimentally observed to "collapse" (a single sample is given nearly all the mass) in higher dimensions [BBL08; DJ09]. As particle smoothers rely on filtering approximations, the smoothing performance also degrades significantly.

Theoretically, the scaling issues of SMC are not completely understood, notably because model dimensions typically do not appear explicitly in the general error bounds for particle filters [Rv15]. In particular, it is not entirely clear *which* dimensions of the models really affect the filtering performance. While it is generally understood that the dimension of the state space plays a key role (as the sampling space), some works [Sny+08] have been conducted to gain further insights into this problem. Ultimately, as particle methods internally rely on importance sampling, the choice of proposal $q_{0:t}$ plays a key role as for any IS and SNIS estimation, and the scalability issues of the latter are inherited in SMC. In SSMs, a popular and readily available proposal is $q_{0:t} = m_{0:t}^\theta$, i.e. samples are drawn from the prior dynamics, irrespective of the observations. In this case, the incremental importance weights correspond, at each timestep, to the observation likelihood $\omega_{s|s-1}^i = g_s^\theta(\xi_s^i)$, a setting known as *bootstrap*. For the latter, [Sny+08] show that the number of particles must grow exponentially with the variance of the log-likelihood of the observations to obtain good filtering approximations. To improve on this behaviour, a large body of research focuses on deriving better proposals for the filtering

problem, notably by relying on the observations in the sampling mechanism [PS99; CMO08]. In fact, the optimal proposal for filtering is known and obtained when $q_{s|s-1}(X_{s-1}, \cdot)$ is the distribution of X_s given X_{s-1} and Y_s. While this distribution is generally intractable in the first place, some works [Sny11; SBM15; GJL17b] suggest that even in the case of this optimal proposal, the weight degeneracy is inevitable when the model dimensions are very large.

Another issue which prevents the wide adoptation of SMC methods in modern marchine learning settings is that the resampling steps are non-differentiable. As such, newers algorithms which relying on stochastic optimization (e.g. for MLE) cannot directly use particle methods as an internal component. While new mechanisms have been developed [JRB18; Cor+21] to retrieve differentiability in the resampling steps, they are often obtained at the cost of biased gradients, additional hyperparameters or intricate implementations [Sin+23].

2.3 Variational inference for sequential data

In parallel to the previous methods which directly build approximations of the fundamental recursions of SSMs, a new category of approximations has started to arise and show promising results for inference in sequential latent data models with high-dimensional state spaces. Rooted in optimization, these methods adapt the methodology of variational to the sequential setting.

2.3.1 General background on variational inference

Consider a space X and an intractable distribution $\pi \in \mathcal{P}(\mathsf{X})$. Given a family of probability distributions $\mathcal{Q} \subset \mathcal{P}(\mathsf{X})$, VI aims at finding an approximation of π in $\mathcal{Q}$ by explicitly minimizing a statistical divergence $d : \mathcal{P}(\mathsf{X}) \times \mathcal{P}(\mathsf{X}) \to \mathbb{R}_+$, i.e. to find

$$q^* = \arg\min_{q \in \mathcal{Q}} d(\pi, q) \ . \tag{2.21}$$

While any probabilistic approximation can technically be cast into this setting, the term VI is commonly reserved to methods which explicitly frame approximation as an optimization problem on spaces of probability distributions. As such, most of the important questions in the field are related to (i) the choices for d (ii) the choice of $\mathcal{Q}$ and (iii) efficient minimization of $d(\pi, \cdot)$ over $\mathcal{Q}$.

In the Bayesian setting presented in the preamble of this chapter where $p(x, y)$ is the joint distribution of some latent data model, VI typically targets the posterior distributions $\pi = p(\cdot|y)$, such that the solutions also depend on y, which we denote q_y^*. A common practice in VI literature, however, is to remove this dependency in the notation, notably to avoid viewing variational solutions as the posterior distributions "$q(x|y)$" of some underlying model distinct from $p(x, y)$. In practice, the dependency of the final solution on the observations may result only from the optimization process, or there may be a direct mapping between the observations and e.g. the parameters of q, as explained before.

Reverse KL divergence and the Evidence Lower Bound

In probability theory, quantifying the discrepancy between distributions may be approached in many ways, each leading to multiple definitions of statistical distances [LV06]. In variational inference, a popular choice is the *Kullback-Leibler divergence* defined, for all $(\pi, q) \in \mathcal{P}(\mathsf{X}) \times \mathcal{P}(\mathsf{X})$ with q absolutely continuous w.r.t π, as

$$\mathbb{D}_{\mathsf{KL}}(q, \pi) = q\left(-\log \frac{\mathrm{d}\pi}{\mathrm{d}q}\right) = -\int_{\mathsf{X}} \log \frac{f_\pi(x)}{f_q(x)} f_q(x) \mathrm{d}x \ , \tag{2.22}$$

where the last equality is valid whenever both π and q admit densities f_π and f_q w.r.t the Lebesgue measure. In the following, we consider that this is always the case and mix notations on density and distributions. $\mathbb{D}_{\mathsf{KL}}$ being non-symetric, considering $\mathbb{D}_{\mathsf{KL}}(q, \pi)$ or $\mathbb{D}_{\mathsf{KL}}(\pi, q)$ leads to different computation problems. The main approach in VI literature is to fix the target distribution π and to consider only the *reverse* KL divergence $\overleftarrow{\mathbb{D}}_{\mathsf{KL}}^{\pi} : q \mapsto \mathbb{D}_{\mathsf{KL}}(q, \pi)$. One major practical advantage of the reverse KL is that it is expressed as an expected value under its input, and the approximating family $\mathcal{Q}$ may be chosen to allow easy computation of such

expectations (as opposed to expectations w.r.t π, the latter being specified by the data model and intractable in the first place).

However, $\overleftarrow{\mathbb{D}}^{\pi}_{\mathsf{KL}}$ still requires evaluation of the density of π, which in most cases is not available. In the Bayesian setting where the target distribution is $p(\cdot|y)$, a common approach is to consider to following derivations

$$\overleftarrow{\mathbb{D}}^{p(\cdot|y)}_{\mathsf{KL}}(q) = -\int_{\mathsf{X}} \log\left(\frac{p(x,y)}{p(y)q(x)}\right) q(x)\mathrm{d}x = -\int_{\mathsf{X}} \log\left(\frac{p(x,y)}{q(x)}\right) q(x)\mathrm{d}x + \log p(y)\ ,$$

to conclude that $\int_{\mathsf{X}} \log\left(\frac{p(x,y)}{q(x)}\right) q(x)\mathrm{d}x \leq \log p(y)$ since $\overleftarrow{\mathbb{D}}^{p(\cdot|y)}_{\mathsf{KL}}(q) \geq 0$. As a lower bound of $\log p(y)$, the quantity

$$\mathcal{L}_q(y) = \int_{\mathsf{X}} \log\left(\frac{p(x,y)}{q(x)}\right) q(x)\mathrm{d}x \tag{2.23}$$

is referred to as the Evidence Lower Bound Objective (ELBO), and we denote

$$q^*_y = \arg\max_{q\in\mathcal{Q}} \mathcal{L}_q(y).$$

In VI, the ELBO has been massively used as the main optimization objective in all works that rely on the reverse KL for scalability concerns. Additionally, newer works [CA18; Che19] suggest that the ELBO may be used for model selection, i.e. instead of $\log p^\theta(y)$ which may be intractable, given a variational family $\mathcal{Q}$ one may use $\mathcal{L}_{q^*_y}(y)$ to select the best parameter θ given observations y (in practice the gap is exactly $\overleftarrow{\mathbb{D}}^{p(\cdot|y)}_{\mathsf{KL}}(q^*_y)$). Similarly, a common practice in parametric latent data models is to optimize both the models parameters $\theta \in \Theta$ and the variational distribution $q \in \mathcal{Q}$ with $\mathcal{L}_q$ as a common objective. In fact, a more formal view on this alternating scheme consists in recognising the ELBO as the free-energy functional [Csi84] associated with $\mathbb{D}_{\mathsf{KL}}$, which is a key quantity already minimized in popular inference methods and MLE algorithms [NH98].

Parametric variational inference and mean-field assumptions

Given this optimization problem, a popular approach is to resort to restricted probability families for $\mathcal{Q}$ in order to allow both (i) efficient optimization of the ELBO and (ii) scalable computation of posterior estimates using q^*_y instead of $p(\cdot|y)$. In that respect, a common choice is to choose a parametric family of distributions $\mathcal{Q}_\Lambda$ where Λ is a parameter space, such that maximization in $\mathcal{Q}_\lambda$ is performed directly in Λ (where a notion of gradient is readily available), i.e. one seeks

$$\lambda^*_y = \arg\max_{\lambda\in\Lambda} \mathcal{L}^\lambda(y), \tag{2.24}$$

where $\mathcal{L}^\lambda = \mathcal{L}_{q^\lambda}$ with q^λ the distribution in $\mathcal{Q}_\Lambda$ with parameter λ. In general, the idea is to consider families $\mathcal{Q}_\Lambda$ whose elements can be easily sampled from and evaluated (e.g. Gaussian), such that unbiased Monte Carlo estimates of $\mathcal{L}^\lambda(y)$ are obtained given i.i.d from samples $(\xi^i)_{i\leq N}$ from q^λ as

$$\widehat{\mathcal{L}}^\lambda(y) = \frac{1}{N}\sum_{i=1}^{N} \log\frac{p(\xi^i, y)}{q^\lambda(\xi^i)}\ .$$

Then, another cornerstone of VI is the so-called mean field assumption: when the target π is a joint distribution $X = (X_1, \dots, X_n)$, the parametric variational approximations $q^\lambda(\mathrm{d}x_{0:n})$ is factorized into

$$q^\lambda(\mathrm{d}x_{0:n}) = \prod_{k=1}^{n} q_k^\lambda(\mathrm{d}x_k) \ , \tag{2.25}$$

i.e. the internal dependencies between the components of π are not captured by the variational distribution, with covariates assumed independant under the variational model. In this methodology, individual distributions $\{q_k\}_{k\leq n}$ will often be referred to as the variational *factors*.

Amortization

In the previous formulation, a new optimization problem needs to be solved for every observation $y \in \mathsf{Y}$ and each observation is associated with an optimal parameter λ_y^*. When inference needs to be performed for a large number of observations $(y^k)_{k\leq K}$, running such process for all $1 \leq k \leq K$ can be prohibitive, preventing the use of VI for very large datasets [BKM17].

A popular alternative is to instead learn directly an approximation of the function $f : y \to \lambda_y^*$ via a parametric space of functions $\{f_\gamma : \mathsf{Y} \to \Lambda\}_{\gamma\in\Gamma}$ - typically DNNs with Γ the space of network parameters - where it is assumed that there exists $\gamma^* \in \gamma$ such that $f_{\gamma^*}(y) \approx \lambda_y^*$ for all $y \in \mathsf{Y}$. In general, given a dataset $\{y^i\}_{i\leq N}$ of observations, γ^* will be obtained as

$$\gamma^* = \arg\max_{\gamma\in\gamma} \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}^{f_\gamma(y^k)}(y^k) \ .$$

In variational inference literature, this methodology is usually referred to as *amortized inference*. While "amortization" is a loosely defined notion, a justification of such denomination is that the parameter vector γ and the associated mappings f_γ are effectively shared (or re-used) to perform inference on multiple observations. Additionally, once trained with the above objective, a common practice is perform inference on new observations using the mapping f_{γ^*} as-is, therefore replacing the need for further optimization altogether.

In new VI methods, such techniques are ubiquitous and are the core component behind the recent success of so-called variational autoencoders [KW14], where the function f_γ will typically be referred to as the *encoder*, the "amortized networks" or the "recognition network" [Zha+18] because upon optimization it provides a deterministic mapping to encoder observations into quantities related to the posterior. In practice, the quality of the variational solutions leveraging this amortized setting strongly depends on the choice of this mapping. In general, amortized variational solutions are expected to perform worse than the traditional setting [CLD18; KLH18] albeit at a much reduced computational cost. Observing that amortized inference consists essentially on "learning to predict the solution of an optimization problem", some works [MYM18] have explored amortized VI from the angle of meta-learning [And+16] to build principled approaches that perform iterative refinements of the predictions given by amortizing networks.

In this thesis, most of the implementated methods rely on amortized inference, but most often we abstract it from the notations by considering distributions and functions that depend on some parameter $\lambda \in \Lambda$ - which may either be the parameter of an encoder as presented above or directly the parameter of the distributions. Additionally, we remove the dependency of the

observations in the notation of the ELBO, as the exact optimization problems involved may either correspond to the amortized objective above involving multiple observations $\{y^k\}_{k\leq K}$ or to the traditional one with a single observation y. For simplicity, the previous sections are written in the latter setting.

Computing the gradient of the ELBO

To solve the maximization problem (2.24), a common approach is to resort to gradient-based optimization and consider gradient-ascent in λ of the ELBO, i.e. updates of the form

$$\lambda_{k+1} = \lambda_k + \gamma_{k+1} \nabla_\lambda \mathcal{L}^\lambda\big|_{\lambda_k} \ ,$$

which requires computing gradients of the ELBO $\mathcal{L}^\lambda$.

Reparameterization. A popular setting is to choose $\mathcal{Q}_\Lambda$ such that its elements can be expressed as image measures $q^\lambda = q_0 \circ (x^\lambda)^{-1}$ where x^λ is continuous in λ and q_0 is a p.d.f. not depending on λ, but known and easy to sample from. Indeed, in this case, for any $q \in \mathcal{Q}_\Lambda$ and any function $h : \mathsf{X} \to \mathsf{H}$, $\nabla_\lambda \left\{q^\lambda h\right\} = \nabla_\lambda \left\{q_0(h \circ x^\lambda)\right\}$, such that the gradient of the ELBO may be expressed and approximated as

$$\nabla_\lambda \mathcal{L}^\lambda = \mathbb{E}_{q_0}\left[\nabla_\lambda \left\{\log \frac{p(x^\lambda(X_0), y)}{q_0(x^\lambda(X_0))}\right\}\right] \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_\lambda \left\{\log \frac{p(x^\lambda(\xi^i_{q_0}), y)}{q_0(x^\lambda(\xi^i_{q_0}))}\right\} \ ,$$

where $\{\xi^i_{q_0}\}_{i\leq N}$ are i.i.d samples from q_0. The gradients in this case are sometimes referred to as *reparameterized* or *pathwise* gradients.

Score-function. When the previous setting cannot be satisfied but it is still possible to sample and evaluate elements of $\mathcal{Q}_\Lambda$, another approach is to consider the score-function estimator of the gradient which states that

$$\nabla_\lambda \mathbb{E}_{q^\lambda}\left[h^\lambda\right] = \mathbb{E}_{q^\lambda}\left[\nabla_\lambda \log q^\lambda \times h^\lambda + \nabla_\lambda h^\lambda\right] \ ,$$

and apply it to the ELBO with $h = \log p(\cdot, y)/q^\lambda$. Noticing that

$$\mathbb{E}_{q^\lambda}\left[\nabla_\lambda h(X)\right] = -\mathbb{E}_{q^\lambda}\left[\nabla_\lambda \log q^\lambda(X)\right] = 0 \ ,$$

another Monte Carlo estimate of the gradient can then be built via

$$\nabla_\lambda \mathcal{L}^\lambda \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_\lambda \left\{\log q^\lambda(\xi^i)\right\} \times \log \frac{p(\xi^i, y)}{q^\lambda(\xi^i)} \ ,$$

with $(\xi^i)_{i\leq N}$ i.i.d samples from q^λ. One may refer to this specific setting as "black-box variational inference" as named in [RGB14] and the associated gradients as *score gradients*.

While both methods yield unbiased gradients, score gradients typically provide much higher variance estimates than reparameeerized gradients and often require variance reduction techniques to allow efficient optimization (see e.g. [Moh+20] for a detailed analysis of the performance of various Monte Carlo gradient estimates).

2.3.2 Sequential variational inference

While elements of the previous section are relevant for any target distribution π, we now focus on works which rely on specialized variational families leveraging known structure in π to improve inference. While such methods - sometimes grouped under the umbrella term of *structured VI* - have attracted much attention for a wide variety of generative models, (e.g. probabilistic graphical models [HB15; RTB16]), we specifically consider the sequential setting of section 2.1, i.e. the case where $\pi = \phi^\theta_{0:t}$ is the joint smoothing distribution of some parametric SSM given an observed sequence $y_{0:t}$. In this case, the variational approximations are elements of $\mathcal{P}(\mathsf{X}^{t+1})$ denoted $q^\lambda_{0:t}$ in the parametric case. Consequently, the ELBO becomes an expectation under $q^\lambda_{0:t}$, denoted with $\mathcal{L}^{\lambda,\theta}_t$ and defined as

$$\mathcal{L}^{\lambda,\theta}_t = \mathbb{E}_{q^\lambda_{0:t}}\left[\log \frac{\ell^\theta_{0:t}}{q^\lambda_{0:t}}(X_{0:t})\right] , \tag{2.26}$$

which we may refer in the following as the *joint ELBO* to distinguish it from the traditional setting. In the rest of this section, for any $0 \leq s_1 \leq s_2 \leq t$, we denote with $q^\lambda_{s_1:s_2}$ the marginal distribution of $X_{s_1:s_2}$ under the variational law $q^\lambda_{0:t}$.

In this setting, whenever $\mathcal{L}^{\lambda,\theta}_t$ can be optimized to provide a solution λ^*, inference can then be performed assuming that $q^{\lambda^*}_{0:t}$ is a sufficiently good approximation of $\phi^\theta_{0:t}$, i.e. by using $q^{\lambda^*}_{0:t}$ instead of the true smoothing distribution to derive statistical estimations. In particular, smoothing functionals of the form $\phi^\theta_{0:t} h_{0:t}$ may be replaced with expectations $q^\lambda_{0:t} h_{0:t}$ under the variational approximations, where one can ensure that plain Monte Carlo estimates are available via specific choices of variational families.

Popular decompositions of the variational distributions

In sequential variational inference, a central idea is to decompose the variational joint distribution $q^\lambda_{0:t}$ into individual factors in order to obtain expressive variational solutions which take into account the temporal dependencies of the data, while breaking inference and optimization of the ELBO into individual steps. With respect to the latter goal, a first option is to consider a simple mean-field assumption (as in Equation (2.25)) across timesteps, e.g. $q^\lambda_{0:t}(\mathrm{d}x_{0:t}) = \prod_{s=1}^t q^\lambda_s(\mathrm{d}x_s)$. In this case

$$\mathcal{L}^{\lambda,\theta}_t = \sum_{s=1}^t \mathbb{E}_{q^\lambda_{s-1:s}}\left[\log \frac{\ell^\theta_s(X_{s-1}, X_s)}{q^\lambda_s(X_s)}\right] + \mathbb{E}_{q^\lambda_0}\left[\log \frac{\chi^\theta(X_0) g^\theta_0(X_0)}{q^\lambda_0(X_0)}\right] ,$$

where the bivariate marginal distributions $q^\lambda_{s-1:s} = q^\lambda_{s-1} \times q^\lambda_s$ are readily available at every timestep when the distributions $(q^\lambda_s)_{s\leq t}$ are all known and easy to sample from. However, in the context of smoothing, this "fully-factorized" decomposition (i.e. each term only depends on a single latent state) effectively doesn't model the latent dependencies in $\phi^\theta_{0:t}$. In particular (i) the Markov property of the conditional distribution of $X_{0:t}$ given $Y_{0:t}$ is ignored and (ii) correlations between the latent states cannot be captured because the marginal distributions are assumed to be independent. Recently, [Bay+21] suggest that such misspecification of the dependencies has detrimental effects on inference. Additionally, a mean-field assumption for temporal data *a priori* forces each factor to depend on the entire sequence of observations $y_{0:t}$, and as such the variational laws do not inherit the convenient recursive properties of the true smoothing distributions described in the previous sections.

To improve on this, many works reintroduce some of the properties of $\phi^\theta_{0:t}$ directly into the variational model. To this aim, a common departure from the mean-field assumption consists in defining variational factors which depend on several latent states and possibly on only a subset of observations, either via unnormalized functions, proper Markov kernels, or a combination of both, as described next. In general, most of the resulting models can be understood as specifying some form of probabilistic graphical model (PGM) in which inference is easier than under the true model (e.g. easy computation of marginal distributions or normalizing constants and exact sampling), but whose parameters are learnt without any approximation of the original model via optimization of the joint ELBO. Here, a recurring objective is to leverage the approximating power of deep neural networks. Given the temporal setting, a common feature of SVI methods is to extend the idea of amortized inference 2.3.1 by relying on mappings which are shared *across timesteps* and used to produce relevant parameters for each of them. As such, many implementation choices are transferred into the parameterization of these networks and the correct ways to include their outputs into a variational model. In that regard, most of the existing methods can generally be understood as falling into several broad categories, which we describe next.

Variational decompositions with conjugate factors. In one line of research [Joh+16; LKH18; Häl+21a], the joint smoothing distribution is not directly specified with a Markovian factorization but instead provided implicitly via a factor graph on which computations can be performed analytically with techniques from inference in probabilistic graphical models [WBJ05; BN06]. In the latter approaches, a common underlying idea is to assimilate observations $\{y_s\}_{s\leq t}$ one-by-one with a single DNN-based mapping whose output can be analytically conjugated with other terms that specify underlying dynamics between the latent states. As a result, the joint approximation $q^\lambda_{0:t}$ can capture complex dependencies on the observations and correlations between the latent states, but the modeling is broken down into simple and easy-to-implement individual components, which are combined with efficient approaches from belief propagation [Min01] or so-called "message-passing" techniques.

While such methods are rather rooted in literature on PGMs, most often the graphs implemented in the latter are motivated by some existing decomposition of the joint smoothing distribution in terms of unnormalized components. For instance, in [Joh+16; LKH18], the density of the joint variational distribution is decomposed as

$$q^\lambda_{0:t}(x_{0:t}) \propto f^\lambda_0(x_0) \prod_{s=1}^{t} f^\lambda_s(x_{s-1}, x_s) \prod_{s=0}^{t} \psi^\lambda(x_s, y_s) \ , \tag{2.27}$$

where the functions $(f^\lambda_s)_{s\leq t}$ are independent of the observations. For the true joint smoothing decomposition, one may observe, using loose notations, that

$$\phi_{0:t}(x_{0:t}) \propto m^\theta_0(x_0) \prod_{s=1}^{t} m^\theta_s(x_{s-1}, x_s) \prod_{s=0}^{t} \frac{p^\theta_{X_s|y_s}(x_s)}{p^\theta_{X_s}(x_s)} \ ,$$

where $p^\theta_{X_s|y_s}$ denotes the density of X_s given $Y_s = y_s$ and $p^\theta_{X_s}$ that of X_s. In practice, such decomposition is not traditionally used in SSMs literature, notably because the previous densities are not always defined and hard to estimate in practice, however it motivates (2.27) as a decomposition of the variational joint distribution. Additionally, in these variational methods,

the functions $(f_s^\lambda)_{s\leq t}$ are such that $f_{0:t}^\lambda = f_0^\lambda \prod_{s=1}^t f_s^\lambda$ is a density on X^{t+1}, and the associated joint distribution is sometimes seen a variational "prior" playing a role similar to $m_{0:t}^\theta$ from the true model. The shared mapping ψ^λ, on the other hand, is generally implemented similarly as traditional encoders from amortized variational inference, but its output is formatted to allow conjugation with the rest of the graph.

All in all, this category of approaches provides a practical way to integrate existing mappings from VAE literature into a sequential setting without resorting to more complex DNN architectures. As conjugacy plays a key role in the tractability of these methods, some works [KL17] have since focused on generalizing the latter to more general factor graphs which are reduced to conjugate ones with additional mechanisms.

Forward variational factorization. Distinct to the previous methods, another line of research attempts to directly reproduce the Markovian factorizations of $\phi_{0:t}$ inside the variational family by explicitly defining Markov kernels whose dependencies on the observations are provided from outputs of DNNs encoded in their parameters. Up to now, most works [Kim+20; Arc+15; KSS15; KSS17; Fra+16] have considered a forward-type of factorization, which is known in SSM literature but relatively unused in practical settings (see Appendix A.2.1 for other factorizations of the smoothing distributions). They factorize the variational joint density as

$$q_{0:t}^\lambda(x_{0:t}) = q_0^\lambda(x_0) \prod_{s=1}^t q_{s|s-1}^\lambda(x_{s-1}, x_s) \ ,$$

where q_0^λ is a distribution, $(q_{s|s-1}^\lambda)_{s\leq t}$ is a sequence of Markov kernels, and all of the latter are made dependent on the *entire* sequence of observations $y_{0:t}$. In these works, optimization is performed via Monte Carlo approximations of the ELBO gradients which are readily available when the terms in the previous factorization belong to parametric families that can be evaluated and sampled from (i.e. samples from $q_{0:t}^\lambda$ are obtained sequentially using q_0^λ and the forward kernels $q_{s|s-1}^\lambda$). Here, a central idea for implementation is to rely on recurrent neural networks which assimilate the observations sequentially and whose outputs parameterize the variational Markov kernels. In [KSS17], for instance, sequences $y_{0:t}$ are encoded using a bidirectinal RNN which produces vectors $(\vec{a}_s)_{s\leq t}$ and $(\overleftarrow{a}_s)_{s\leq t}$ of fixed length. For all $s \leq t$,

$$\vec{a}_s = \vec{A}^\lambda(y_s, \vec{a}_{s-1}) \text{ and } \overleftarrow{a}_s = \overleftarrow{A}^\lambda(y_s, \overleftarrow{a}_{s+1}) \ ,$$

where $\vec{A}^\lambda$ and $\overleftarrow{A}^\lambda$ are distinct DNNs. Then, for all $1 \leq s \leq t$, the parameters of the variational kernels are such that for all $x_{s-1} \in \mathsf{X}$,

$$q_{s|s-1}^\lambda(x_{s-1}, \cdot) \sim \mathcal{N}(\mu_{s|s-1}^\lambda, \Sigma_{s|s-1}^\lambda) \ ,$$

distinct where

$$\mu_{s|s-1}^\lambda = f^\lambda(x_{s-1}, \vec{a}_s, \overleftarrow{a}_s) \text{ and } \Sigma_{s|s-1}^\lambda = g^\lambda(x_{s-1}, \vec{a}_s, \overleftarrow{a}_s) \ ,$$

with f^λ and g^λ nonlinear mappings. Under such Gaussian assumptions, the joint distribution $q_{0:t}^\lambda$ can be easily sampled from and evaluated, and complex dependencies between the latent states and the observations are captured in the mappings $(\overleftarrow{A}^\lambda, \vec{A}^\lambda, f^\lambda, g^\lambda)$ when optimizing the joint ELBO. In SVI literature, such approaches have attracted a lot of attention, notably because

they provide solid ground to combine the flexibility of deep learning-based architectures for temporal data with known probabilistic inference. Nonetheless, by relying on the forward factorization of $\phi^\theta_{0:t}$, they effectively inherit from the limitations of the latter. In particular, the dependency of each variational term on the entire sequence of observations implies that they cannot be used in online settings where observations arrive sequentially, which essentially prevents their use for long sequences and therefore limits their scalability.

Variational filtering. In view of the previous remarks, a distinct line of research [MCY18; ZP20; DZP23] chooses to trade smoothing for filtering by targeting the marginal distributions $(\phi^\theta_s)_{s\leq t}$ at each timestep with variational distributions $q^\lambda_s(\mathrm{d}x_s)$ that depend only on the observations up to s. In general, these solutions depart from the original objective defined by Equation (2.26) and formulate intermediate optimization problems at each timestep by deriving a "single step ELBO" from $\overleftarrow{\mathbb{D}}^{\phi_s}_{\mathsf{KL}}(q^\lambda_s)$. To derive such ELBOs, a defining trait of these works is the additional assumption that, at s, q^λ_{s-1} is a good approximation of ϕ^θ_{s-1}, which in practice can hardly be verified especially under Gaussian variational families. For example, for all $s \leq t$ the quantity

$$\mathbb{E}_{q^\lambda_s}\left[\log \frac{q^\lambda_s}{g^\theta_s} - \mathbb{E}_{\phi^\theta_{s-1}}\left[m^\theta_s\right]\right]$$

is a lower bound of incremental log-likelihood r^θ_s derived from $\overleftarrow{\mathbb{D}}^{\phi^\theta_s}_{\mathsf{KL}}(q^\lambda_s)$, and in [DZP23] authors replace the intractable ϕ^θ_{s-1} with an approximation obtained at the previous timestep.

Additionally, while they do provide solutions to process the observations sequentially, such methods a priori only provide mean-field approximations of the joint smoothing distributions by considering $q^\lambda_{0:t} = \prod_{s=0}^t q^\lambda_s$ (where each q^λ_s only depends on observations up to s). Therefore, even in the event that $q^\lambda_s \approx \phi^\theta_s$ for all $s \leq t$, principled approaches to approximate $\phi^\theta_{0:t}$ given these variational filtering approximations are not available, and as such the associated works only produce joint distributions with misspecified dependencies as in traditional mean-field VI.

Backward variational smoothing

To solve the computational limitations of the previous methods (i.e. dependency of each variational factors on all the observations), another option, first introduced in [Cam+21], is to reproduce the *backward* factorization of (2.3) in the variational model by introducing the following decomposition

$$q^\lambda_{0:t}(x_{0:t}) = q^\lambda_t(x_t) \prod_{s=1}^t q^\lambda_{s-1|s}(x_s, x_{s-1}) \;, \tag{2.28}$$

where the terminal quantity q^λ_t is a density in X and, for all $1 \leq s \leq t$, $q^\lambda_{s-1|s}$ is are densities of Markov kernels in $\mathsf{X} \times \mathsf{X}$.

The backward variational ELBO. In this context, the joint ELBO becomes

$$\mathcal{L}^{\lambda,\theta}_t = \mathbb{E}_{q^\lambda_{0:t}}\left[\log \frac{\prod_{s=1}^t \ell^\theta_s(X_{s-1}, X_s)}{q^\lambda_t(X_t)\prod_{s=1}^t q^\lambda_{s-1|s}(X_s, X_{s-1})}\right] \;,$$

which may identified as

$$\mathcal{L}_t^{\lambda,\theta} = \mathbb{E}_{q_{0:t}^\lambda}\left[h_{0:t}^{\lambda,\theta}(X_{0:t}) - \log q_t^\lambda\right] ,$$

with $h_{0:t}^{\lambda,\theta}$ the additive state functional having components $(\tilde{h}_s^{\lambda,\theta})_{s\leq t}$, defined as,

$$\tilde{h}_s^{\lambda,\theta} : (x_{s-1}, x_s) \mapsto \log \frac{\ell_s^\theta(x_{s-1}, x_s)}{q_{s-1|s}^\lambda(x_s, x_{s-1})} , \tag{2.29}$$

for all $s < t$.

Backward variational additive smoothing recursions. Given this decomposition, it becomes possible to apply similar derivations than for additive smoothing with $\phi_{0:t}^\theta$ (as described in 2.1.2) by writing the ELBO as

$$\mathcal{L}_t^{\lambda,\theta} = \mathbb{E}_{q_t^\lambda}\left[H_t^\lambda(X_t)\right] - \mathbb{E}_{q_t^\lambda}\left[\log q_t^\lambda\right] ,$$

where $H_t^\lambda : x_t \mapsto \mathbb{E}_{q_{0:t}^\lambda}\left[h_{0:t}(X_{0:t}) \mid X_t = x_t\right]$ is part of a sequence of functions $(H_s^\lambda)_{s\leq t}$ which can be defined recursively from expectations under the variational kernels $(q_{s-1|s}^\lambda)_{s\leq t}$, i.e. for all $s \leq t$,

$$H_s^\lambda : x_s \mapsto \mathbb{E}_{q_{s-1|s}^\lambda(x_s,\cdot)}\left[H_{s-1}^\lambda(X_{s-1}) + \tilde{h}_s(X_{s-1}, x_s)\right] .$$

Implementations of the backward factorization amenable to online learning. To implement in practice the decomposition (2.28), i.e. setting parametric forms to each of its terms, multiple options can be considered (e.g. recurrent neural networks which encode the parameters of the kernels sequentially as in the previous subsection). However, while it is possible to use the backward decomposition in the offline setting (e.g. given a sequence $y_{0:t}$ of fixed-length $t+1$), the original motivation behind its introduction is that it allows to obtain a variational joint approximation $q_{0:t}^\lambda$ for *all* timesteps $t \geq 0$ without having to recompute all factors when updating them. To this aim, one may simply define:

- A flow of distributions $(q_t^\lambda)_{t\geq 0}$ available at all $t \geq 0$.
- A flow of kernels $(q_{t-1|t}^\lambda)_{t\geq 1}$ available at all $t \geq 1$ which respect the same dependencies on the observations than for the true backward kernels $B_{t-1|t}^\theta$, i.e. for all $t \geq 1$, the parameters of $q_{t-1|t}^\lambda$ only depend on $y_{0:t-1}$.

In this case, successive joint variational distributions can be constructed recursively by considering

$$q_{0:t+1}^\lambda = \frac{q_t^\lambda q_{t-1|t}^\lambda}{q_{t-1}^\lambda} q_{0:t}^\lambda ,$$

where $q_{0:t}^\lambda$ respects the backward decomposition (2.28) at all timesteps $t \geq 0$. [1]

Given all this, the variational backward factorization is a strong candidate because it paves the paw to recursive additive smoothing, given that H_t^λ is defined for all $t \geq 0$ and only depends on observations $y_{0:t-1}$.

[1] In practice, this relationship also allows to define the ELBO in compact form as the smoothed expectation of the additive state functional with components $\tilde{h}_t^{\lambda,\theta} = \frac{\ell_t^\theta q_{t-1}^\lambda}{q_t^\lambda q_{t-1|t}^\lambda}$, but in this work we rather rely on the other definition of the components.

Chapter 3

Macrolitter video counting on riverbanks using state-space models and moving cameras

This chapter is based on the article "Macrolitter video counting on riverbanks using state-space models and moving cameras" published in 2023 in Computo, the journal of the French statistical society, [Cha+23]. All the codes to generate the images and run our algorithm is available online.

This contribution can be summarized as follows.

1. We provide a novel open-source image dataset of macro litter, which includes various objects seen from different rivers and different contexts. This dataset was produced with a new open-sourced platform for data gathering and annotation developed in conjunction with Surfrider Foundation Europe, continuously growing with more data.

2. We propose a new algorithm specifically tailored to count in videos with fast camera movements. In a nutshell, DNN-based object detection is paired with a robust state space movement model which uses optical flow to perform Bayesian filtering, while confidence regions built on posterior predictive distributions are used for data association. This framework does not require video annotations at training time: the multi-object tracking module does not require supervision, only the DNN-based object detection does require annotated images. It also fully leverages optical flow estimates and the uncertainty provided by Bayesian predictions to recover object identities even when detection recall is low. Contrary to existing MOT solutions, this method ensures that tracks are stable enough to avoid repeated counting of the same object.

3. We provide a set of video sequences where litter counts are known and depicted in real conditions. For these videos only, litter positions are manually annotated at every frame in order to carefully analyze performance. This allows us to build new informative count metrics. We compare the count performance of our method against other MOT-based alternatives.

A first visual illustration of the second claim is presented in Figure 3.1: on three selected frames, we present a typical scenario where our strategy can avoid overcounting the same object (we depict internal workings of our solution against the end result of the competitors).

Figure 3.1: Our method: one object (red dot) is correctly detected at every frame and given a consistent identity throughout the sequence with low location uncertainty (red ellipse). Next to it, a false positive detection is generated at the first frame (brown dot) but immediatly lost in the following frames: the associated uncertainty grows fast (brown ellipse). In our solution, this type of track will not be counted. A third correctly detected object (pink) appears in the third frame and begins a new track.

Figure 3.2: SORT: the resulting count is also 2, but both counts arise from tracks generated by the same object, the latter not re-associated at all in the second frame. Additionally, the third object is discarded (in post-processing) by their strategy.

3.1 Datasets for training and evaluation

Our main dataset of annotated images is used to train the object detector. Then, only for evaluation purposes, we provide videos with annotated object positions and known global counts. Our motivation is to avoid relying on training data on videos, that requires this resource-consuming process.

3.1.1 Images

Data collection. With help from volunteers, we compile photographs of litter stranded on river banks after increased river discharge, shot directly from kayaks navigating at varying distances from the shore. Images span multiple rivers with various levels of water current, on different seasons, mostly in southwestern France. The resulting pictures depict trash items under the same conditions as the video footage we wish to count on, while spanning a wide variety of backgrounds, light conditions, viewing angles and picture quality.

Bounding box annotation. For object detection applications, the images are annotated using a custom online platform where each object is located using a bounding box. In this work, we focus only on litter counting without classification, however the annotated objects are already classified into specific categories which could be used in future works.

3.1.2 Video sequences

Data collection. For evaluation, an on-field study was conducted with 20 volunteers to manually count litter along three different riverbank sections in April 2021, on the Gave d'Oloron near Auterrive (Pyrénées-Atlantiques, France), using kayaks. The river sections, each 500 meters long, were precisely defined for their differences in background, vegetation, river current, light conditions and accessibility (see B.2 for aerial views of the shooting site and details on the river sections). In total, the three videos amount to 20 minutes of footage at 24 frames per second (fps) and a resolution of 1920x1080 pixels.

Track annotation. On video footage, we manually recovered all visible object trajectories on each river section using an online video annotation tool (more details in B.2 for the precise methodology). From that, we obtained a collection of distinct object tracks spanning the entire footage.

3.2 Optical flow-based counting via Bayesian filtering and confidence regions

Our counting method is divided into several interacting blocks. First, a detector outputs a set of predicted positions for objects in the current frame. The second block is a tracking module designing consistent trajectories of potential objects within the video. At each frame, a third block links the successive detections together using confidence regions provided by the tracking module, proposing distinct tracks for each object. A final postprocessing step only keeps the best tracks which are enumerated to yield the final count.

3.2.1 Detector

Center-based anchor-free detection. In most benchmarks, the prediction quality of object attributes like bounding boxes is often used to improve tracking. For counting, however, point detection is theoretically enough and advantageous in many ways. First, to build large datasets, a method which only requires the lightest annotation format may benefit from more data due to annotation ease. Second, contrary to previous popular methods [Ren+15] involving intricate mechanisms for bounding box prediction, center-based and anchor-free detectors [ZWK19] only use additional regression heads which can simply be removed for point detection. Adding to all this, [Zha+21] highlight conceptual and experimental reasons to favor anchor-free detection in tracking-related tasks.

For these reasons, we use a stripped version of CenterNet [ZWK19] where offset and bounding box regression heads are discarded to output bare estimates of center positions on a coarse grid. An encoder-decoder network takes an input image $I \in [0,1]^{w \times h \times 3}$ (an RGB image of width w and height h), and produces a heatmap $\hat{H} \in [0,1]^{\lfloor w/p \rfloor \times \lfloor h/p \rfloor}$ such that $\hat{H}_{ij}$ is the probability that (i,j) is the center of an object (p being a stride coefficient). At inference, peak detection and thresholding are applied to $\hat{H}$, yielding the set of detections. The bulk of this detector relies on the DLA34 architecture [Yu+18]. In a video, for each frame $I_t \in [0,1]^{w \times h \times 3}$ (where t indexes the frame number), the detector outputs a set $\mathcal{D}_t = \{u_t^k\}_{1 \leq k \leq D_t}$ where each $u_t^k = (x_t^k, y_t^k)$ specifies the coordinates of one of the D_t detected objects.

Training. For every image, the corresponding set $\mathcal{B} = \{(c_k^w, c_k^h, w_k, h_k)\}_{1 \leq k \leq B}$ of B annotated bounding boxes – i.e. a center (c_k^w, c_k^h), a width w_k and a height h_k– is rendered into a ground truth heatmap $H \in [0,1]^{\lfloor w/p \rfloor \times \lfloor h/p \rfloor}$ by applying kernels at the bounding box centers and taking element-wise maximum. For all $1 \leq i \leq w/p$, $1 \leq j \leq h/p$, the ground truth at (i,j) is

$$H_{ij} = \max_{1 \leq k \leq B} \left(\exp \left\{ -\frac{(i - c_k^w)^2 + (j - c_k^h)^2}{2\sigma_k^2} \right\} \right) ,$$

where σ_k is a parameter depending on the size of the object. Training the detector is done by minimizing a penalty-reduced weighted focal loss

$$\mathcal{L}(\hat{H}, H) = -\sum_{i,j} \gamma_{ij}^{\beta} \left(1 - \hat{p}_{ij}\right)^{\alpha} \log\left(\hat{p}_{ij}\right) ,$$

where α, β are hyperparameters and

$$(\hat{p}_{ij}, \gamma_{ij}) = \begin{cases} (\hat{H}_{ij}, 1) & \text{if } H_{ij} = 1, \\ (1 - \hat{H}_{ij}, 1 - H_{ij}) & \text{otherwise.} \end{cases}$$

3.2.2 Bayesian tracking with optical flow

Optical flow. Between two timesteps $t-1$ and t, the optical flow Δ_t is a mapping satisfying the following consistency constraint:

$$\widetilde{I}_t[u] = \widetilde{I}_{t-1}[u + \Delta_t(u)] ,$$

where, in our case, $\widetilde{I}_t$ denotes the frame t downsampled to dimensions $\lfloor w/p \rfloor \times \lfloor h/p \rfloor$ and $u = (i, j)$ is a coordinate on that grid. To estimate Δ_t, we choose a simple unsupervised Gunner-Farneback algorithm which does not require further annotations, see [Far03] for details.

State space model. Using optical flow as a building block, we posit a state space model where estimates of Δ_t are used as a time and state-dependent offset for the state transition.

Let $(X_t)_{t\geq 1}$ and $(Y_t)_{t\geq 1}$ be the true (but hidden) and observed (detected) positions of a target object in $\mathbb{R}^2$, respectively.

Considering the optical flow value associated with X_{t-1} on the discrete grid of dimensions $\lfloor w/p \rfloor \times \lfloor h/p \rfloor$, write

$$X_t = X_{t-1} + \Delta_t(\lfloor X_{t-1} \rfloor) + \eta_t \tag{3.1}$$

and

$$Y_t = X_t + \varepsilon_t \ ,$$

where $(\eta_t)_{t\geq 1}$ are i.i.d. centered Gaussian random variables with covariance matrix Q independent of $(\varepsilon_t)_{t\geq 1}$ i.i.d. centered Gaussian random variables with covariance matrix R. In the following, Q and R are assumed to be diagonal, and are hyperparameters set to values given in Appendix B.3.

Approximations of the filtering distributions. In our setting, we find that a linearisation of the model (3.1) yields an approximation which is computationally cheap and as robust on our data:

$$X_t = X_{t-1} + \Delta_t(\lfloor \mu_{t-1} \rfloor) + \partial_X \Delta_t(\lfloor \mu_{t-1} \rfloor)(X_{t-1} - \mu_{t-1}) + \eta_t \ .$$

where ∂_X is the derivative operator with respect to the 2-dimensional spatial input X.

This allows the implementation of Kalman updates on the linearised model, a technique named extended Kalman filtering (EKF). On the currently available data, we find that the optical flow estimates are very informative and accurate, making this approximation sufficient. For completeness, we present in Appendix B.3 a SMC-based solution and discuss the empirical differences and use-cases where the latter might be a more relevant choice.

In any case, the state space model naturally accounts for missing observations, as the contribution of Δ_t in every transition ensures that each filter can cope with arbitrary inter-frame motion to keep track of its target.

3.2.3 Generating potential object tracks

The full MOT algorithm consists of a set of single-object trackers followingstate space modelstate space model the previous model, but each provided with distinct observations at every frame. These separate filters provide track proposals for every object detected in the video.

Data association using confidence regions

Throughout the video, depending on various conditions on the incoming detections, existing trackers must be updated (with or without a new observation) and others might need to be

created. This setup requires a third party data association block to link the incoming detections with the correct filters.

At the frame t, a set of L_t Bayesian filters track previously seen objects and a new set of detections $\mathcal{D}_t$ is provided by the detector. Denote by $1 \leq \ell \leq L_t$ the index of each filter at time t, and by convention write $Y^{\ell}_{1:t-1}$ the previous observed positions associated with index ℓ (even if no observation is available at some past times for that object). Let $\rho \in (0, 1)$ be a confidence level.

1. For every detected object $u^k_t \in \mathcal{D}_t$ and every filter ℓ, compute $P(k, \ell) = \mathbb{P}(Y^{\ell}_t \in V_\delta(u^k_t) \mid Y^{\ell}_{1:t-1})$ where $V_\delta(u)$ is the neighborhood of u defined as the squared area of width 2δ centered on u (see Appendix B.3 for exact computations).

2. Using the Hungarian algorithm ([Kuh55]), compute the assignment between detections and filters with P as cost function, but discarding associations (k, ℓ) having $P(k, \ell) < \rho$. Formally, ρ represents the level of a confidence region centered on detections and we use $\rho = 0.5$. Denote a_ρ the resulting assignment map defined as $a_\rho(k) = \ell$ if u^k_t was associated with the ℓ-th filter, and $a_\rho(k) = 0$ if u^k_t was not associated with any filter.

3. For $1 \leq k \leq D_t$, if $a_\rho(k) = \ell$, use u^k_t as a new observation to update the ℓ-th filter. If $a_\rho(k) = 0$, create a new filter initialized from the prior distribution, i.e. sample the true location as a Gaussian random variable with mean u^k_t and variance R.

4. For all filters ℓ' which were not provided a new observation, update only the predictive law of $X^{\ell'}_t$ given $Y^{\ell'}_{1:t-1}$.

In other words, we seek to associate filters and detections by maximising a global cost built from the predictive distributions of the available filters, but an association is only valid if its corresponding predictive probability is high enough. Though the Hungarian algorithm is a very popular algorithm in MOT, it is often used with the Euclidean distance or an Intersection-over-Union (IoU) criterion. Using confidence regions for the distributions of Y_t given $Y_{1:(t-1)}$ instead allows to naturally include uncertainty in the decision process. Note that we deactivate filters whose posterior mean estimates lie outside the image subspace in $\mathbb{R}^2$.

Note that this way of combining a set of Bayesian filters with a data association step that resorts on the most likely hypothesis is a form of Global Nearest Neighbor (GNN) tracking. Another possibility is to perform multi-target filtering by including the data association step directly into the probabilistic model, as in [Mah03]. A generalisation of single-target recursive Bayesian filtering, this class of methods is grounded in the point process literature and well motivated theoretically. In case of strong false positive detection rates, close and/or reappearing objects, practical benefits may be obtained from these solutions. Finally, note that another well-motivated choice for $P(k, \ell)$ could be to use the marginal likelihood $\mathbb{P}(Y^{\ell}_t \in V_\delta(u^k_t))$, which is standard in modern MOT.

Counting

At the end of the video, the previous process returns a set of candidate tracks. For counting purposes, we find that simple heuristics can be further applied to filter out tracks that do not follow actual objects. More precisely, we observe that tracks of real objects usually contain more (i) observations and (ii) streams of uninterrupted observations. Denote by

$T_\ell = \left\{t \in \mathbb{N} \mid \exists u \in \mathcal{D}_t, Y_t^\ell = u\right\}$ all timesteps where the ℓ-th object is observed. To discard false counts according to (i) and (ii), we compute the moving average M_ℓ^κ of 1_{T_ℓ} using windows of size κ, i.e. the sequence defined by $M_\ell^\kappa[t] = \frac{1}{\kappa}\sum_{s\in[\![t-\kappa,t+\kappa]\!]} 1_{T_\ell}[s]$. We then build $T_\ell^\kappa = \{t \in T_\ell \mid M_\ell^\kappa[t] > \nu\}$, and defining $\mathcal{N} = \{\ell \mid |T_\ell^\kappa| > \tau\}$, the final object count is $|\mathcal{N}|$. We choose $\nu = 0.6$ while κ, τ are optimized for best count performance (see Appendix B.3 for a more comprehensive study).

3.2.4 Metrics for MOT-based counting

Counting in videos using embedded moving cameras is not a common task, and as such it requires a specific evaluation protocol to understand and compare the performance of competing methods. First, not all MOT metrics are relevant, even if some do provide insights to assist evaluation of count performance. Second, considering only raw counts on long videos gives little information on which of the final counts effectively arise from well detected objects.

Count-related MOT metrics

Popular MOT benchmarks usually report several sets of metrics such as ClearMOT ([BS08]) or IDF1 ([Ris+16]) which can account for different components of tracking performance. Recently, [Lui+21] built the so-called HOTA metrics that allow separate evaluation of detection and association using the Jaccard index. The following components of their work are relevant to our task (we provide equation numbers in the original paper for formal definitions).

Detection First, when considering all frames independently, traditional detection recall (DetRe) and precision (DetPr) can be computed to assess the capabilities of the object detector. Denoting with $\mathsf{TP}_n, \mathsf{FP}_n, \mathsf{FN}_n$ the number of true positive, false positive and false negative detections at frame n, respectively, we define $\mathsf{TP} = \sum_t \mathsf{TP}_t$, $\mathsf{FP} = \sum_t \mathsf{FP}_t$ and $\mathsf{FN} = \sum_t \mathsf{FN}_t$, then:

$$\mathsf{DetRe} = \frac{\mathsf{TP}}{\mathsf{TP} + \mathsf{FN}} \ ,$$

$$\mathsf{DetPr} = \frac{\mathsf{TP}}{\mathsf{TP} + \mathsf{FP}} \ .$$

In classical object detection, those metrics are the main target. In our context, as the first step of the system, this framewise performance impacts the difficulty of counting. However, we must keep in mind that these metrics are computed framewise and might not guarantee anything at a video scale. The next points illustrate that remark.

1. If both DetRe and DetPr are very high, objects are detected at nearly all frames and most detections come from actual objects. Therefore, robustness to missing observations is high, but even in this context computing associations may fail if camera movements are nontrivial. 2. For an ideal tracking algorithm which never counts individual objects twice and does not confuse separate objects in a video, a detector capturing each object for only one frame could theoretically be used. Thus, low DetRe could theoretically be compensated with robust tracking. 3. If our approach can rule out faulty tracks which do not follow actual objects, then good counts can still be obtained using a detector generating many false positives. Again, this suggests that low DetPr may allow decent counting performance.

Association. HOTA association metrics are built to measure tracking performance irrespective of the detection capabilities, by comparing predicted tracks against true object trajectories. In our experiments, we compute the Association Recall (AssRe) and the Association Precision (AssPr). Several intermediate quantities are necessary to introduce these final metrics. Following [Lui+21], we denote with prID the ID of a predicted track and gtID the ID of a ground truth track. Given C all couples of prID $-$ gtID found among the true positive detections, and $c \in C$ one of these couples, $\mathsf{TPA}(c)$ is the number of frames where prID is also associated with gtID, $\mathsf{FPA}(c)$ is the number of frames where prID is associated with another ground truth ID or with no ground truth ID, and $\mathsf{FNA}(c)$ is the number of frames where gtID is associated with another predicted ID or with no predicted ID. Then:

$$\mathsf{AssPr} = \frac{1}{\mathsf{TP}} \sum_{c \in C} \frac{\mathsf{TPA}(c)}{\mathsf{TPA}(c) + \mathsf{FPA}(c)} \ ,$$

$$\mathsf{AssRe} = \frac{1}{\mathsf{TP}} \sum_{c \in C} \frac{\mathsf{TPA}(c)}{\mathsf{TPA}(c) + \mathsf{FNA}(c)} \ .$$

See [Lui+21] (fig. 2) for a clear illustration of these quantities.

In brief, a low AssPr implies that several objects are often mingled into only one track, resulting in undercount. A low AssRe implies that single objects are often associated with multiple tracks. If no method is used to discard redundant tracks this results in overcount. Conversely, association precision (AssPr) measures how exclusive tracks are to each object (it decreases whenever a track covers multiple objects). Again, it is useful to reconsider and illustrate the meaning of these metrics in the context of MOT-based counting. Litter items are typically well separated on river banks, thus predicted tracks are not expected to interfere much. This suggests that reaching high AssPr on our footage is not challenging. Contrarily, AssRe is a direct measurement of the capability of the tracker to avoid producing multiple tracks despite missing detections and challenging motion. A high AssRe therefore typically avoids multiple counts for the same object, which is a key aspect of our work.

Nonetheless, association metrics are only computed for predicted tracks which can effectively be matched with ground truth tracks. Consequently, AssRe does not account for tracks predicted from streams of false positive detections generated by the detector (e.g. arising from rocks, water reflections, etc). Since such tracks induce false counts, a tracker which produces the fewest is better, but MOT metrics do not measure it.

Count metrics

Denoting by $\hat{\mathsf{N}}$ and N the respective predicted and ground truth counts for the validation material, the error $\hat{\mathsf{N}} - \mathsf{N}$ is misleading as no information is provided on the quality of the predicted counts. Additionally, results on the original validation footage do not measure the statistical variability of the proposed estimators.

Count decomposition. Define $i \in [\![1, \mathsf{N}]\!]$ and $j \in [\![1, \hat{\mathsf{N}}]\!]$ the labels of the annotated ground truth tracks and the predicted tracks, respectively. At evaluation, we assign each predicted track to either none or at most one ground truth track, writing $j \to \emptyset$ or $j \to i$ for the corresponding assignments. The association is made whenever a predicted track i overlaps with a ground truth track j at any frame, i.e. for a given frame a detection in i is within a

threshold α of an object in j. We compute metrics for 20 values of $\alpha \in [0.05\alpha_{max}, 0.95\alpha_{max}]$, with $\alpha_{max} = 0.1\sqrt{w^2 + h^2}$, then average the results, which is the default method in HOTA to combine results at different thresholds. We keep this default solution, in particular because our results are very consistent accross different thresholds in that range (we only observe a slight decrease in performance for $\alpha = \alpha_{max}$, where occasional false detections probably start to lie below the threshold).

Denote $A_i = \{j \in [\![1, \hat{\mathsf{N}}]\!] \mid j \to i\}$ the set of predicted tracks assigned to the i-th ground truth track. We define:

1. $\hat{\mathsf{N}}_{\mathsf{true}} = \sum_{i=1}^{\mathsf{N}} 1_{|A_i|>0}$ the number of ground truth objects successfully counted.
2. $\hat{\mathsf{N}}_{\mathsf{red}} = \sum_{i=1}^{\mathsf{N}} |A_i| - \hat{\mathsf{N}}_{\mathsf{true}}$ the number of redundant counts per ground truth object.
3. $\hat{\mathsf{N}}_{\mathsf{mis}} = \mathsf{N} - \hat{\mathsf{N}}_{\mathsf{true}}$ the number of ground truth objects that are never effectively counted.
4. $\hat{\mathsf{N}}_{\mathsf{false}} = \sum_{j=1}^{\hat{\mathsf{N}}} 1_{j\to\emptyset}$ the number of counts which cannot be associated with any ground truth object and are therefore considered as false counts.

Using these metrics provides a much better understanding of $\hat{\mathsf{N}}$ as

$$\hat{\mathsf{N}} = \hat{\mathsf{N}}_{\mathsf{true}} + \hat{\mathsf{N}}_{\mathsf{red}} + \hat{\mathsf{N}}_{\mathsf{false}} \ ,$$

while $\hat{\mathsf{N}}_{\mathsf{mis}}$ completely summarises the number of undetected objects.

Conveniently, the quantities can be used to define the count precision (CountPR) and count recall (CountRe) as follows:

$$\mathsf{CountPR} = \frac{\hat{\mathsf{N}}_{\mathsf{true}}}{\hat{\mathsf{N}}_{\mathsf{true}} + \hat{\mathsf{N}}_{\mathsf{red}} + \hat{\mathsf{N}}_{\mathsf{false}}} \ ,$$
$$\mathsf{CountRe} = \frac{\hat{\mathsf{N}}_{\mathsf{true}}}{\hat{\mathsf{N}}_{\mathsf{true}} + \hat{\mathsf{N}}_{\mathsf{mis}}} \ ,$$

which provide good summaries for the overall count quality, letting aside the tracking performance.

Note that these metrics and the associated decomposition are only defined if the previous assignment between predicted and ground truth tracks can be obtained. In our case, predicted tracks never overlap with several ground truth tracks (because true objects are well separated), and therefore this assignment is straightforward. More involved metrics have been studied at the trajectory level (see for example [GRS20] and the references therein), though not specifically tailored to the restricted task of counting. For more complicated data, an adaptation of such contributions into proper counting metrics could be valuable.

Statistics. Since the original validation set comprises only a few unequally long videos, only absolute results are available. Splitting the original sequences into shorter independent sequences of equal length allows to compute basic statistics. For any quantity $\hat{\mathsf{N}}_\bullet$ defined above, we provide $\hat{\sigma}_{\hat{\mathsf{N}}_\bullet}$ the associated empirical standard deviations computed on the set of short sequences.

Footage	DetRe*	DetPr*
S1	37.2	60.7
S2	29.4	38.2
S3	35.1	53.6
All	35.5	55.1

Table 3.1: Detection results

3.3 Experiments

We denote by S_1, S_2 and S_3 the three river sections of the evaluation material and split the associated footage into independent segments of 30 seconds. We further divide this material into two distinct validation (6min30) and test (7min) splits.

To demonstrate the benefits of our work, we select two multi-object trackers and build competing counting systems from them. Our first choice is SORT [Bew+16], which relies on Kalman filtering with velocity updated using the latest past estimates of object positions. Similar to our system, it only relies on image supervision for training, and though DeepSORT [WBP17] is a more recent alternative with better performance, the associated deep appearance network cannot be used without additional video annotations. FairMOT [Zha+21], a more recent alternative, is similarly intended for use with video supervision but allows self-supervised training using only an image dataset. Built as a new baseline for MOT, it combines linear constant-velocity Kalman filtering with visual features computed by an additional network branch and extracted at the position of the estimated object centers, as introduced in CenterTrack [ZKK20]. We choose FairMOT to compare our method to a solution based on deep visual feature extraction.

Similar to our work, FairMOT uses CenterNet for the detection part and the latter is therefore trained as in Section 3.2.1. We train it using hyperparameters from the original paper. The detection outputs are then shared between all counting methods, allowing fair comparison of counting performance given a fixed object detector. We run all experiments at 12fps, an intermediate framerate to capture all objects while reducing the computational burden.

3.3.1 Detection

In the following section, we present the performance of the trained detector. Having annotated all frames of the evaluation videos, we directly compute DetRe and DetPr on those instead of a test split of the image dataset used for training. This allows realistic assessment of the detection quality of our system on true videos that may include blurry frames or artifacts caused by strong motion. We observe low DetRe, suggesting that objects are only captured on a fraction of the frames they appear on. To better focus on count performance in the next sections, we remove segments that do not generate any correct detection: performance on the remaining footage is increased and given by DetRe^* and DetPr^*.

3.3.2 Counts

To fairly compare the three solutions, we calibrate the hyperparameters of our postprocessing block on the validation split and keep the values that minimize the overall count error $\hat{\mathsf{N}}$ for

each of them separately (see Appendix B.3 for more information). All methods are found to work optimally at $\kappa = 7$, but our solution requires $\tau = 8$ instead of $\tau = 9$ for other solutions: this lower level of thresholding suggests that raw output of our tracking system is more reliable.

We report results using the count-related tracking metrics and count decompositions defined in the previous section. To provide a clear but thorough summary of the performance, we report AssRe, CountRe and CountPR as tabled values (the first gives a simple overview of the quality of the predicted tacks while the latter two concisely summarise the count performance). For a more detailed visualisation of the different types of errors, we plot the count error decomposition for all sequences in a separate graph. Note that across all videos and all methods, we find AssPr between 98.6 and 99.2 which shows that this application context is unconcerned with tracks spanning multiple ground truth objects, therefore we do not conduct a more detailed interpretation of AssPr values.

First, the higher values of AssRe confirm the robustness of our solution in assigning consistent tracks to individual objects. This is directly reflected into the count precision performance - with an overall value of CountPR 17.6 points higher than the next best method (SORT) - or even more so in the complete disappearance of orange (redundant) counts in the graph. A key aspect is that these improvements are not counteracted by a lower CountRe: on the contrary, our tracker, which is more stable, also captures more object (albeit still missing most of them, with a CountRe below 50

Footage	Method	AssRe	CountRe	σ(CountRe)	CountPr	σ(CountPr)
	FairMOT	62.0	31.2	25.6	52.6	24.6
Segment 1	Sort	65.6	43.8	26.4	53.8	20.2
	Ours	79.5	50.0	27.9	64.0	23.8
Footage	Method	AssRe	CountRe	σ(CountRe)	CountPr	σ(CountPr)
	FairMOT	8.7	12.5	35.4	50.0	0.0
Segment 2	Sort	20.7	12.5	35.4	33.3	0.0
	Ours	72.7	50.0	0.0	100.0	0.0
Footage	Method	AssRe	CountRe	σ(CountRe)	CountPr	σ(CountPr)
	FairMOT	17.4	25.0	47.1	50.0	50.0
Segment 3	Sort	19.6	25.0	47.1	40.0	50.9
	Ours	24.6	37.5	41.7	60.0	47.9
Footage	Method	AssRe	CountRe	σ(CountRe)	CountPr	σ(CountPr)
	FairMOT	56.7	27.1	31.6	52.0	30.2
Combined	Sort	59.8	35.4	32.7	50.0	30.2
	Ours	76.0	47.9	28.8	67.6	32.0

Table 3.2: Count-related evaluation metrics

3.4 Practical impact and future goals

We successfully tackled video object counting on river banks, in particular issues which could be addressed independently of detection quality. Moreover the methodology developed to assess count quality enables us to precisely highlight the challenges that pertain to video object

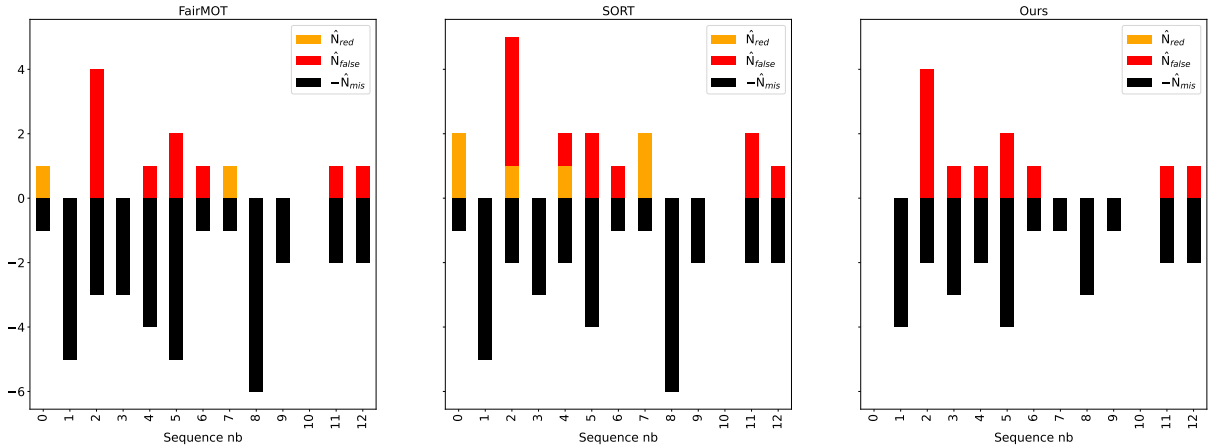


Figure 3.3: Counting results

counting on river banks. Conducted in coordination with Surfrider Foundation Europe, an NGO specialized on water preservation, our work marks an important milestone in a broader campaign for macrolitter monitoring and is already being used in a production version of a monitoring system. That said, large amounts of litter items are still not detected. Solving this problem is largely a question of augmenting the object detector training dataset through crowdsourced images. A specific annotation platform is online, thus the amount of annotated images is expected to continuously increase, while training is provided to volunteers collecting data on the field to ensure data quality. Finally, several expeditions on different rivers are already underway and new video footage is expected to be annotated in the near future for better evaluation. All data is made freely available. Future goals include downsizing the algorithm, a possibility given the architectural simplicity of anchor-free detection and the relatively low computational complexity of EKF. In a citizen science perspective, a fully embedded version for portable devices will allow a larger deployment. The resulting field data will help better understand litter origin, allowing to model and predict litter density in non surveyed areas. Correlations between macro litter density and environmental parameters will be studied (e.g., population density, catchment size, land use and hydromorphology). Finally, our work naturally benefits any extension of macrolitter monitoring in other areas (urban, coastal, etc) that may rely on a similar setup of moving cameras.

Chapter 4

A backward sampling approach for online variational additive smoothing

This chapter is based on the article "A backward sampling approach for online variational additive smoothing" submitted for publication in TMLR, the Transactions of Machine Learning Research, [Cha+].

Notations. In this contribution, no final index is *a priori* defined, as we focus on variational methods that can accomodate continuous streams of observations $\{y_t\}_{t\geq 0}$. Therefore, the letter t does not correspond to a terminal index but rather to any timestep (with s used for indices prior to t). In practice we do consider fixed-length sequences in the experimental section, and here we use T for the final index.

4.1 Introduction

In this contribution, we consider the problem of computing variational approximations of smoothing expectations of the form $\mathbb{E}_{q^\lambda_{0:t}}\left[h_{0:t}(X_{0:t})\right]$ *recursively* on the observations, where $h_{0:t}$ is an additive state functional as defined in (2.8). More specifically, we focus on the challenges that arise in *online* variational smoothing, where the goal is to obtain an approximation $q^\lambda_{0:t} \approx \phi^\theta_{0:t}$ of the smoothing distribution for *every* timestep $t \geq 0$ via updates having a computational cost independent of t. In that regard, we presented in Section 2.3.2 the advantages of the *backward* variational decomposition defined in Equation (2.28) as

$$q^\lambda_{0:t}(\mathrm{d}x_{0:t}) = q^\lambda_t(\mathrm{d}x_t) \prod_{s=1}^{t} q^\lambda_{s-1|s}(x_s, \mathrm{d}x_{s-1}) \ .$$

Indeed, in this setting, recall that additive smoothing derivations from classical SSM literature apply, i.e.

$$\mathbb{E}_{q^\lambda_{0:t}}\left[h_{0:t}(X_{0:t})\right] = \mathbb{E}_{q^\lambda_t}\left[H^\lambda_t(X_t)\right] \ , \tag{4.1}$$

where

$$H^\lambda_t(X_t) = \mathbb{E}_{q^\lambda_{0:t}}\left[h_{0:t}(X_{0:t}) \mid X_t\right] \ . \tag{4.2}$$

This statistic admits the following functional recursion: for all $x_t \in \mathsf{X}$,

$$H^\lambda_t(x_t) = \mathbb{E}_{q^\lambda_{t-1|t}(x_t,\cdot)}\left[H^\lambda_{t-1}(X_{t-1}) + \tilde{h}_t(X_{t-1}, x_t)\right] \ . \tag{4.3}$$

When relying on SMC methods, we explained in Section 2.2.3 how recursions of the same form are readily approximated with a fixed computational cost via particle-based empirical approximations of the true backward kernels, which are used to build discrete approximations of the functions $\{H_t^\lambda\}_{t\geq 0}$ on the finite supports of the approximate filtering distributions $\{\hat{\phi}_t^\theta\}_{t\geq 0}$. This central ingredient allows to deploy online methods for both additive smoothing under $\phi_{0:t}^\theta$ and parameter learning via derivations from recursive MLE (see Section 2.1.2). In backward variational approaches, the availability of same recursions suggests that similar methods could be applied to allow online variational smoothing. However, the quantities involved in variational approaches are very different, which raises several challenges that we briefly describe below.

Deriving tractable approximations of variational backward expectations. First, at t, compared to the discretely supported approximation $\hat{\phi}_t^\theta = \sum_{i=1}^N \bar{\omega}_t^i \delta_{\xi_t^i}$ of the filtering distribution ϕ_t^θ provided in SMC algorithms, the marginal q_t^λ is a parametric distribution defined on the entirety of X. Suppose that we still want to build a Monte Carlo approximation of (4.1) via N draws $\xi_t^i \sim q_t^\lambda$, $1 \leq i \leq N$, the latter distribution being chosen at implementation, it is typically easy to sample from it. In this case, the estimator $N^{-1}\sum_{i=1}^N H_t^\lambda(\xi_t^i)$ requires the evaluation of H_t^λ, which itself is an expectation under $q_{t-1|t}^\lambda$. Again, the variational kernels belong to a chosen family of distributions, and therefore it is easy to obtain samples from them. Given ξ_t^i and M i.i.d samples $\xi_{t-1|t}^{ij} \sim q_{t-1|t}^\lambda(\xi_t^i, \cdot)$, one may therefore consider to approximate $H_t^\lambda(\xi_t^i)$ by $M^{-1}\sum_{j=1}^M H_{t-1}^\lambda(\xi_{t-1|t}^{ij}) + \tilde{h}_t(\xi_{t-1|t}^{ij}, \xi_t^i)$. Unfortunately, the cost of such an approximation scheme grows with t, because the evaluation of the function H_{t-1}^λ on a new sample involves recomputation of all previous conditional expectations $(H_s^\lambda)_{s<t-1}$. Additionally, the backward samples $\{\xi_{t-1|t}^{ij}\}_{j\leq M}$ depend on the draw ξ_t^i which is only known at t, and therefore it is not possible to store evaluations $\{H_{t-1}^\lambda(\xi_{t-1|t}^{ij})\}_{j\leq M}$ at the previous step. In particle-based recursive smoothing, this problem does not appear because at t, for any $x_t \in \mathsf{X}$, the approximate backward distribution $\hat{B}_{t-1|t}^\theta(x_t, \cdot)$ is defined on the support $\{\xi_{t-1}^j\}_{j\leq N}$ of the previous particle approximation of ϕ_{t-1}^θ, which is obtained via particle filtering methods described in Section 2.2.2.

To circumvent this, one solution proposed in [Cam+21] is to replace the true conditional expectations with functional approximations that can be evaluated in $O(1)$ at each timestep. Denoting $\mathcal{F} = \{f : \mathsf{X} \to \mathsf{F}, \mathbb{E}_{q_t}[\|f(X_t)\|_2] < \infty\}$, H_t^λ satisfies, by definition of the conditional expectation,

$$H_t^\lambda = \operatorname*{argmin}_{f\in\mathcal{F}} \mathbb{E}_{q_{t-1:t}^\lambda(X_{t-1},X_t)}\left[\left\|f(X_t) - \left[H_{t-1}^\lambda(X_{t-1}) + \tilde{h}_t(X_{t-1}, X_t)\right]\right\|_2\right].$$

which provides a regression objective to learn an approximation of H_t^λ. In practice authors restrict the minimization problem to a subset of $\mathcal{F}$, a parametric family of functions (typically, a neural network) parameterized by γ, belonging to $\Gamma \subset \mathbb{R}^{d_\gamma}$, and learn this by approximating the expectation with a Monte Carlo method. Namely, the authors propose to estimate H_t^λ by $H_{\hat{\gamma}_t}^\lambda$ where

$$\hat{\gamma}_t = \operatorname*{argmin}_{\gamma\in\Gamma} \frac{1}{N}\sum_{k=1}^N \left\|H_\gamma^\lambda(\xi_t^k) - \left[H_{\hat{\gamma}_{t-1}}^\lambda(\xi_{t-1}^k) + \tilde{h}_t(\xi_{t-1}^k, \xi_t^k)\right]\right\|_2,$$

where $\left\{(\xi_{t-1}^i, \xi_t^i)\right\}_{i=1,\dots,N}$ is an i.i.d. sample under the variational joint distribution of (X_{t-1}, X_t) which has density $q_{t-1:t}^\lambda = q_t^\lambda q_{t-1|t}^\lambda$. Upon convergence, $H_{\hat{\gamma}_t}^\lambda$ is plugged into the final expectation of (4.1).

While this removes the need to compute expectations under $q_{t-1|t}^\lambda$ altogether, a major drawback of this solution is that it requires running an inner optimization on $\gamma \in \Gamma$ until convergence *at every iteration* t, which may be very costly if the parameter space Γ is large. Additionally, the parametric approximations of the conditional expectations introduce a bias which is difficult to analyse and control, especially w.r.t its dependency on t.

Deriving recursions of the ELBO gradient. As mentioned in Section 2.3.2, a convenient aspect of backward variational inference is that the optimization objective (the ELBO) is itself a smoothing expectation:

$$\mathcal{L}_t^{\lambda,\theta} = \mathbb{E}_{q_{0:t}^\lambda}\left[h_{0:t}^{\lambda,\theta}\right] - \mathbb{E}_{q_t^\lambda}\left[\log q_t^\lambda\right] \ , \tag{4.4}$$

with $h_{0:t}^{\lambda,\theta} : x_{0:t} \mapsto \log \ell_{0:t}^\theta(x_{0:t}) - \log q_{0:t}^\lambda(x_{0:t})$ being an additive state functional with components $(h_s^{\lambda,\theta})_{s\leq t}$ defined in Equation (2.29). As such, approximations of the backward expectations as previously described can enable recursive computation of the ELBO. However, in online sequential variational inference, the main goal is to obtain recursive approximations of the ELBO *gradients* w.r.t the variational parameters $(\nabla_\lambda \mathcal{L}_t^{\lambda,\theta})_{t\geq 0}$ to enable stochastic gradient algorithms that update λ at each timestep with a constant computational cost. Unfortunately, since the ELBO involves expectations under distributions that depend on λ, one cannot interchange the integral and derivative operators, and more involved derivations are necessary to relate its gradients with known quantites on which additive smoothing techniques can be applied.

Summary of the contribution. Given the previous challenges, this contribution can be summarized as follows

- We propose a specific definition of the variational backward kernels which allows to defined sample-based approximations of conditional backward expectations similar to those in SMC, but with the conveniency of i.i.d sample under the variational model. This removes the need for costly functional approximations defined in [Cam+21] and allows efficient computation of variational smoothing expectations of additive state functionals. In practice, our approximationsrely self-normalized importance sampling, and as such their bias can easily be related with the number of samples used. Finally, while our approach is still very general, we provide a possible implementation that involves exponential conjugation for fast computation of the variational backward parameters.

- We present a new derivation of the gradient of the ELBO which reduces to multiple additive smoothing problems that can be approximated concurrently via the previous algorithm, therefore allowing online optimization of the ELBO with a fixed computational complexity w.r.t t. Our approach does not rely on reparameterization but naturally comes with a built-in variance reduction technique which allows to obtain high quality gradient estimates.

- We evaluate empirically the quality of our approximations against offline estimates, and demonstrate the reliability of the gradients by introducing our gradients in stochastic gradient algorithms that update the parameter λ at each timestep given streams of observations $\{y_t\}_{t\geq 0}$. We compare the computational times and approximation errors with the approach proposed by [Cam+21].

4.2 A computationally effective approach to online variational additive smoothing

In all that follows, we suppose that we have access to a sequence $(q_t^\lambda)_{t\geq 0}$ of distributions on X whose parameters are obtained recursively in time, and which can be both easily evaluated and sampled from. We consider variational distributions $q_{0:t}^\lambda$ under the backward factorization given by Equation (2.28), where q_t^λ is the marginal of $q_{0:t}^\lambda$ for all $t \geq 0$.

4.2.1 Implicit definition of variational backward kernels using forward potentials

To cope with the aforementioned challenges of computing backward expectations, we propose to introduce additional structure in the backward variational kernels via functions $\psi_t^\lambda : \mathsf{X} \times \mathsf{X} \to \mathbb{R}$ which explicitly relate them to the distributions $(q_t^\lambda)_{t\geq 0}$. Specifically, we prescribe that, for all $t \geq 1$,

$$q_{t-1|t}^\lambda(x_t, x_{t-1}) \propto q_{t-1}^\lambda(x_{t-1})\psi_t^\lambda(x_{t-1}, x_t) \; . \tag{4.5}$$

The functions ψ_t^λ can be made arbitrarily complex, such that, for all t, the backward variational kernel $q_{t-1|t}^\lambda$ has arbitrarily complex dependencies w.r.t x_t. Under this decomposition, given a set $\{\xi_{t-1}^j\}_{j\leq N}$ of N i.i.d samples drawn from q_{t-1}^λ and a measurable function $f : \mathsf{X}\times\mathsf{X} \to \mathbb{R}^{d_f}$, expectations of $f(\cdot, x)$ under $q_{t-1|1}^\lambda(x, \cdot)$ for any $x \in \mathsf{X}$ can be expressed as

$$q_{t-1|t}^\lambda(x, \cdot)[f(\cdot, x)] = \frac{\int q_{t-1}^\lambda(x_{t-1})\psi_t^\lambda(x_{t-1}, x)f(x_{t-1}, x)\mathrm{d}x_{t-1}}{\int q_{t-1}^\lambda(x_{t-1})\psi_t^\lambda(x_{t-1}, x)\mathrm{d}x_{t-1}} \; ,$$

and approximated by $\sum_{j=0}^N \bar{w}_{t-1|t}^{\lambda,j}(x)f(\xi_{t-1}^j, x)$ where, for all $1 \leq j \leq N$

$$\bar{w}_{t-1|t}^{\lambda,j}(x) = \frac{\psi_t^\lambda(\xi_{t-1}^j, x)}{\sum_{k=1}^N \psi_t^\lambda(\xi_{t-1}^k, x)} \; . \tag{4.6}$$

The vector of normalized weights $(\bar{w}_{t-1|t}^{\lambda,j}(x))_{j\leq N}$ is s referred to as the *backward weights* conditionally to x.

4.2.2 Recursive approximations of variational backward conditional expectations

We consider some additive state functional[1] $h_{0:t}$ with components $(\tilde{h}_s)_{s\leq t}$ and aim at computing expectations such as the ones given in equations (4.1)-(4.3). Given elements of the previous

[1]The typical situation of interest being the functional $h_{0:t}^{\lambda,\theta}$ associated to the ELBO $\mathcal{L}_t^{\lambda,\theta}$ of Equation (4.4)

section, at t, for any $x \in \mathsf{X}$, the conditional expectation H_t^λ:

$$H_t^\lambda(x) = \mathbb{E}_{q_{t-1|t}^\lambda(x,\cdot)}\left[H_{t-1}^\lambda(X_{t-1}) + \tilde{h}_t(X_{t-1}, x)\right] ,$$

can be estimated by $\sum_{j=1}^N \bar{w}_{t-1|t}^{\lambda,j}(x)\{H_{t-1}^\lambda(\xi_{t-1}^j) + \tilde{h}_t(\xi_{t-1}^j, x)\}$, where $\{\xi_{t-1}^j\}_{i\leq N}$ are i.i.d samples from q_{t-1}^λ and the weights $\bar{w}_{t-1|t}^{\lambda,j}(x)$ are the backward weights as previously defined. Nesting these approximations over time allows to recursively obtain Monte Carlo estimates $\{\hat{H}_t^i\}_{t\geq 0}^{i\leq N}$ of the conditional expectations $(H_t^\lambda)_{t\geq 0}$ on samples of $(q_t^\lambda)_{t\geq 0}$, i.e. for all $t \geq 0$, $i \leq N$, $\hat{H}_t^i$ approximates $H_t^\lambda(\xi_t^i)$ with $\xi_t^i \overset{\text{i.i.d.}}{\sim} q_t^\lambda$ and

$$\hat{H}_t^i = \sum_{j=1}^N \bar{w}_{t-1|t}^{\lambda,i,j}\{\hat{H}_{t-1}^j + \tilde{h}_t(\xi_{t-1}^j, \xi_t^i)\} , \tag{4.7}$$

where $\bar{w}_{t-1|t}^{\lambda,i,j} = \bar{w}_{t-1|t}^{\lambda,j}(\xi_t^i)$ are the backward weights conditionally to the new samples from the variational distribution. Then, a Monte Carlo estimate of $\mathcal{L}_t^\lambda = \mathbb{E}_{q_t^\lambda}[H_t(X_t)]$ is available in the form of

$$\hat{\mathcal{L}}_t^\lambda = \frac{1}{N}\sum_{i=1}^N \hat{H}_t^i .$$

Crucially, the update of Equation (4.7) only requires having stored the previous set of samples and approximations $\{\xi_{t-1}^j, \hat{H}_{t-1}^j\}_{j\leq N}$, such that the memory and computational cost of computing $\hat{\mathcal{L}}_t^\lambda$ is independant of t.

Relationship with SMC smoothing. Recall from Section 4.2.1 that the true backward kernel is itself such that $b_{t-1|t}^\theta(x_t, x_{t-1}) \propto \phi_{t-1}^\theta(x_{t-1})m_t^\theta(x_{t-1}, x_t)$. In particle methods, the filtering distributions $(\phi_t^\theta)_{t\geq 0}$ are recursively approximated using empirical weighted measures. At t, ϕ_t^θ is estimated by $\sum_{i=1}^N w_t^i \delta_{\xi_t^i}$ with $\sum_{i=1}^N w_t^i = 1$. As a consequence, backward expectations are approximated with the following backward weights

$$\bar{w}_{t-1|t}^{\theta,i,j} = \frac{w_{t-1}^j m_t^\theta(\xi_{t-1}^j, \xi_t^i)}{\sum_{k=1}^N w_{t-1}^k m_t^\theta(\xi_{t-1}^k, \xi_t^i)} .$$

By introducing the structure (4.5), we therefore effectively allow ideas from SMC litterature to be used in the variational context. Nonetheless, the quantities involved and the process to obtain them differ in the following points

- Our samples $\{\xi_t^i\}_{i\leq N}$ are obtained by i.i.d sampling from q_t^λ (whose parameters are deterministically derived) and q_t^λ is directly approximated with the uniformly weighted measure $N^{-1}\sum_{i=1}^N \delta_{\xi_t^i}$. In contrast, particle methods produce new samples by propagating a *selection* of the previous ones with a proposal kernel (the produced samples are therefore not independant), and ϕ_t^θ is approximated with an empirical measure using importance weights.

- In particle smoothing, $(x_{t-1}, x_t) \mapsto m_t^\theta(x_{t-1}, x_t)$ is a kernel in x_t. In contrast, one may choose ψ_t^λ such that $\int \psi_t^\lambda(x_{t-1}, x_t)\mathrm{d}x_t \neq 1$ in the variational counterpart. Therefore, the latter functions may rather be seen as variational forward *potentials* than as kernel densities.

Backward resampling. The cost of computing the weights $\{\bar{w}^{\lambda,i,j}_{t-1|t}\}_{1\leq i,j\leq N}$ - and therefore the update (4.7) - is $O(N^2)$ due to the normalizing constant in Equation (4.6). When considering high dimensional state spaces, increasing the number of samples N may be required to compute the targeted expectations. In this case, the quadratic complexity may be prohibitive. One solution, introduced by [OW+17] in the context of particle smoothing, is to resample, at t, given ξ^i_t, an index $j \in \{1, \dots, N\}$ from the multinomial distribution with weights $\{\bar{w}^{\lambda,i,j}_{t-1|t}\}_{j\leq N}$. Crucially, noting that

$$\bar{w}^{\lambda,i,j}_{t-1|t} \propto_j \psi^\lambda_t(\xi^j_{t-1}, \xi^i_t) \ ,$$

this resampling step can be done via accept-reject methods without having to compute the normalizing constant of the weights. Given a new sample ξ^i_t from q^λ_t, an adaptation of the update of Equation (4.7) can be made by adopting *backward resampling*, i.e. to perform the two following steps:

1. Using accept-reject sampling, draw a set of indices $\mathcal{J}^i_t$ from the multinomial distribution with weights proportional to $\{\psi^\lambda_t(\xi^j_{t-1}, \xi^i_t)\}_{j\leq N}$.

2. Compute

$$\hat{H}^i_t = \frac{1}{M} \sum_{j\in\mathcal{J}^i_t} \left\{ \hat{H}^j_{t-1} + \tilde{h}_{t-1}(\xi^j_{t-1}, \xi^i_t) \right\} \ .$$

In this case, $M = |\mathcal{J}^i_t|$ is generally chosen as $M \ll N$, such that the cost of an update is greatly reduced. In a recent review on the performance of algorithms of this class, [DC23] propose to replace the accept-reject methodology with only a few MCMC steps (typically only M) targetting the distribution with backward weights, and starting from the ancestor particles. This methodology is also possible in our case, however a ξ^i_t does not have a predefined notion of ancestor in $\{\xi^j_{t-1}\}_{j\leq N}$ because of the i.i.d sampling scheme. Algorithm 1 provides a first pseudo-code of this recursive smoothing procedure.

Conjugate potentials for fast inference. One aspect - not visible in the above updates - must be dealt with whenever it is also required to evaluate the p.d.f. of the backward kernels $(q^\lambda_{t-1|t})_{t\geq 0}$. Indeed, recall from Section 4.2.1 that $q^\lambda_{t-1|t}(x, \cdot)$ is only defined up to a normalizing constant $c_t(x)$ given $x \in \mathsf{X}$. For a generic choice of forward potentials ψ^λ_t, the associated integral is intractable and must be approximated to evaluate $q^\lambda_{t-1|t}(x, y)$ on a given $y \in \mathsf{X}$. For the most practical functional $h^{\lambda,\theta}_{0:t}$ of the ELBO, the p.d.f of the backward kernels are part of the components $\tilde{h}^{\lambda,\theta}_t$, such that the recursive smoothing algorithm proposed above requires evaluation of these functions on all couples $(\xi^j_{t-1}, \xi^i_t)_{1\leq i,j\leq N}$ of samples from q^λ_{t-1} and q^λ_t, at each timestep. One simple approximation for the corresponding normalizing constants is to choose $N/\sum_{j=1}^N \psi^\lambda_t(\xi^j_{t-1}, \xi^i_t)$. However other estimators may be preferred to lower the bias in estimating the components. An attractive approach is to choose the forward potentials in a class of functions such that the normalizing constants can be computed analytically. Considering the definition of the backward kernels $q^\lambda_{t-1|t}$ by Equation (4.5), a practical solution is to choose ψ^λ_t such that, given $\psi^\lambda_t(\cdot, x)$ belongs to the probabilistic family of q^λ_{t-1}. When the chosen distributions q^λ_t belong to the exponential family[2], one class of potentials can be built

[2]i.e. when their p.d.f. can be written in the form $q^\lambda_t(x_t) \propto \exp(\eta^\lambda_t \cdot T(x_t))$ where η^λ_t is a natural parameter vector and $T(x)$ is a vector of sufficient statistics.

using ideas from [Joh+16], prescribing that

$$\psi_t^\lambda(x_{t-1}, x_t) = \exp\left(\bar{\eta}_t^\lambda(x_t) \cdot T(x_{t-1})\right) , \tag{4.8}$$

where, for all $x \in \mathsf{X}$, $\bar{\eta}_t^\lambda(x)$ is a vector of natural parameters for the same parametric family (i.e. $x \mapsto \bar{\eta}_t^\lambda$ is a mapping from elements of X to the natural parameter space). As a classical conjugation result, given $x \in \mathsf{X}$, the distribution $q_{t-1|t}^\lambda(x, \cdot)$ belongs to the same parametric family as $q_t^\lambda(x_t)$, and its natural parameter vector $\eta_{t-1|t}^\lambda(x)$ is simply obtained by:

$$\eta_{t-1|t}^\lambda(x) = \eta_{t-1}^\lambda + \bar{\eta}_t^\lambda(x) .$$

In this convenient setting, the backward kernels $q_{t-1|t}^\lambda$ can have arbitrarily complex dependencies in X_t while their p.d.f is analytically derived from the potentials, which effectively removes the need to compute the normalizing constants c_t without reducing the latter kernels to simple transformations / linearisations (e.g. linear-Gaussian kernels).

Algorithm 1 One iteration of our online variational smoothing algorithm

Require: $\{\xi_{t-1}^i, \hat{H}_{t-1}^i\}_{i \leq N}$.
Ensure: $\{\xi_t^i, \hat{H}_t^i\}_{i \leq N}$.
Compute the parameters of q_t^λ
Sample $\{\xi_t^i\}_{i=1}^N$ i.i.d. with distribution q_t^λ.
for $i = 1$ to $i = N$ **do**
for $j = 1$ to $j = M$ **do**
Sample $J_t^{i,j}$ in $\{1, \ldots, N\}$ with probabilities proportional to $\psi_t^\lambda(\xi_{t-1}^\ell, \xi_t^i)$, $1 \leq \ell \leq M$, using accept-reject or MCMC.
end for
Compute

$$\hat{H}_t^i = \frac{1}{M} \sum_{j=1}^{M} \left\{ \hat{H}_{t-1}^{J_t^{i,j}} + \tilde{h}_t(\xi_{t-1}^{J_t^{i,j}}, \xi_t^i) \right\} .$$

end for
Compute

$$\hat{\mathcal{L}}_t^\lambda = \frac{1}{N} \sum_{i=1}^{N} \hat{H}_t^i .$$

4.3 Recursive gradient approximations

We now consider the problem of extending the solutions presented above to the computation of the sequence of gradients $(\nabla_\lambda \mathbb{E}_{q_{0:t}^\lambda}[h_{0:t}])_{t \geq 0}$ recursively to allow online gradient-based optimization in λ. Again, this admits as particular and most interesting case the ELBO $\mathcal{L}_t^{\lambda,\theta}$ with components $\tilde{h}_t^{\lambda,\theta}$.

The first option that comes to mind - when distributions $(q_t^\lambda)_{t \geq 0}$ are continuous in their parameters - is to leverage the reparameterization trick. If $\mathcal{E}$ is a base distribution independent

of λ and $\epsilon \mapsto x_t^\lambda(\epsilon)$ is the function such that $x_t^\lambda(\epsilon) \sim q_t^\lambda$ whenever $\epsilon \sim \mathcal{E}$, then one can write $\nabla_\lambda \mathcal{L}_t^\lambda = \mathbb{E}_{\epsilon_t \sim \mathcal{E}}\left[\left(\nabla_\lambda H_t^\lambda\right)\left(x_t^\lambda(\epsilon_t)\right)\right]$. Then, noting that all steps in the Equation (4.7) are differentiable, it is tempting to apply a similar reparameterization to the approximated updates when considering the gradient of the reparameterized conditional expectation $\epsilon \mapsto \left(\nabla_\lambda H_t^\lambda\right)\left(x_t^\lambda(\epsilon)\right)$. However, this approach is flawed because the approximation of Equation (4.7) is biased due the normalized weights[3]. Building gradients via autodifferentiation of a biased estimator can lead to unexpected behaviour, especially in the case of ELBO maximization which is based on an upper bound. Typically, the autodifferentiation will lead to the parameters that maximize the bias of our approximation. Another issue with this scheme is that it is not compatible with backward resampling steps, which reduce the computational cost of the algorithm, but are not differentiable. In the next sections, we present a methodology which circumvents these issues and allows recursive updates of the gradients without reparameterization.

4.3.1 Gradient recursions based on the score-function estimator

An alternative approach to reparameterization in our context is to first derive new recursions for the gradients of the conditional expectations, then to use the approximations proposed in the previous section on the derived quantities. To this aim, one may use the so-called *score-function estimator* (using terminology from [Moh+20] and as presented in Section 2.3.1). Given a distribution p^λ and a function f^λ both on X and depending on a common parameter λ, under regularity constraints necessary to invert the integral and derivative operators, one can write

$$\nabla_\lambda \mathbb{E}_{p^\lambda}\left[f^\lambda(X)\right] = \mathbb{E}_{p^\lambda}\left[\left\{\nabla_\lambda \log p^\lambda \times f^\lambda\right\}(X) + \nabla_\lambda f^\lambda(X)\right] \ .$$

For functionals $h_{0:t}^\lambda$, this allows the following decomposition of the gradient of the corresponding smoothed expectations:

$$\begin{aligned}\nabla_\lambda \mathcal{L}_t^\lambda &= \nabla_\lambda \mathbb{E}_{q_t^\lambda}\left[H_t^\lambda(X_t)\right] \\ &= \mathbb{E}_{q_t^\lambda}\left[\{\nabla_\lambda \log q_t^\lambda \times H_t^\lambda\}(X_t) + \nabla_\lambda H_t^\lambda(X_t)\right] \ . \end{aligned} \tag{4.9}$$

The first term of the integrand can be readily computed given approximations from section 4.2 i.e. by considering $\{\hat{H}_t^i\}_{i\leq N}$ to approximate H_t^λ on a set of samples $\{\xi_t^i\}_{i\leq N}$ from q_t^λ. Recalling the definition of H_t^λ given by Equation (4.2), denoting $F_t = \nabla_\lambda H_t^\lambda$ and using again the score-function estimator, the second term follows:

$$\begin{aligned}F_t^\lambda(x_t) &= \nabla_\lambda \mathbb{E}_{q_{0:t-1|t}^\lambda(x_t,\cdot)}\left[h_{0:t}^\lambda(X_{0:t-1}, x_t)\right] \\ &= \mathbb{E}_{q_{0:t-1|t}^\lambda(x_t,\cdot)}\left[\nabla_\lambda \log q_{0:t-1|t}^\lambda(x_t, X_{0:t-1}) \times h_{0:t}^\lambda(X_{0:t-1}, x_t) + \nabla_\lambda h_{0:t}^\lambda(X_{0:t-1}, x_t)\right] \ .\end{aligned}$$

The previous equation can be rewritten as

$$F_t^\lambda(x_t) = G_t^\lambda(x_t) + \mathbb{E}_{q_{0:t-1|t}^\lambda(x_t,\cdot)}\left[\nabla_\lambda h_{0:t}^\lambda(X_{0:t-1}, x_t)\right] \ ,$$

where

$$G_t^\lambda : x_t \mapsto \mathbb{E}_{q_{0:t-1|t}^\lambda(x_t,\cdot)}\left[\left(\nabla_\lambda \log q_{0:t-1|t}^\lambda \times h_{0:t}^\lambda\right)(X_{0:t-1}, x_t)\right] \ .$$

The rightmost conditional expectation in F_t may be dealt with using similar approximations as for H_t^λ, since the functional $\nabla_\lambda h_{0:t}^\lambda$ is also additive: we denote with $\{\hat{R}_t^i\}_{i\leq N}$ the corresponding

[3] As it is the case in standard self normalized importance sampling.

approximated terms[4]. Obtaining recursive approximations for $G_t^\lambda(x_t)$ is not straightforward, though. Indeed, under the backward factorization, $\nabla_\lambda \log q^\lambda_{0:t-1|t} = \sum_{s=1}^t \nabla_\lambda \log q^\lambda_{s-1|s}$ is itself an additive functional with components $\nabla_\lambda \log q^\lambda_{s-1|s}$, however the product of functionals in definition of G_t^λ is not. Still, a recursion exists in the form of

$$G_t^\lambda(x_t) = \mathbb{E}_{q^\lambda_{t-1|t}(x_t)}\left[G_{t-1}^\lambda(X_{t-1}) + \nabla_\lambda \log q^\lambda_{t-1|t}(x_t, X_{t-1}) \times \left(H_{t-1}^\lambda(X_{t-1}) + \tilde{h}_t^\lambda(X_{t-1}, x_t)\right)\right] . \tag{4.10}$$

Conveniently, this is again an expectation under the backward kernel, such that the methods proposed in the previous section may be applied to derive a running approximation $\{\hat{G}_t^i\}_{i\leq N}$ of G_t^λ updated recursively, i.e.

$$\hat{G}_t^i = \sum_{i=1}^N \bar{w}^{\lambda,i,j}_{t-1|t}\left\{\hat{G}_{t-1}^j + \nabla_\lambda \log q^\lambda_{t-1|t}(\xi_t^i, \xi_{t-1}^j) \times \left(\hat{H}_{t-1}^j + \tilde{h}_t^\lambda(\xi_{t-1}^j, \xi_t^i)\right)\right\} . \tag{4.11}$$

Then, plugging these expressions into Equation (4.9), we obtain an approximation of $\nabla_\lambda \mathcal{L}_t^\lambda$ as

$$\nabla_\lambda \mathcal{L}_t^\lambda \approx \frac{1}{N}\sum_{i=1}^N \{\hat{F}_t^i + \nabla_\lambda \log q_t^\lambda(\xi_t^i) \times \hat{H}_t^i\} , \tag{4.12}$$

where $\hat{F}_t^i = \hat{G}_t^i + \hat{R}_t^i$. In the specific case of the ELBO, since $\nabla_\lambda h^\lambda_{0:t} = -\nabla_\lambda \log q^\lambda_{0:t}$, it follows that $\mathbb{E}_{q^\lambda_{0:t}}\left[\nabla_\lambda h^\lambda_{0:t}\right] = -\mathbb{E}_{q^\lambda_{0:t}}\left[\nabla_\lambda \log q^\lambda_{0:t}(X_{0:t})\right] = 0$. Therefore the term $\{\hat{R}_t^i\}_{i\leq N}$ vanishes when measured against q_t^λ.
It is worth noting that the updates of $\hat{G}_t^i$ involve the same backward weights as the ones needed to update $\hat{H}_t^i$ in Algorithm 1. Therefore, this new algorithm only requires additional computation of $\nabla_\lambda \log q^\lambda_{t-1|t}(\xi_t^i, \xi_{t-1}^j)$, which is typically obtained by autodifferentiation. Additionally, the backward resampling step is now possible for these gradient recursions because the weights are not differentiated.

4.3.2 Baseline variance reduction

As studied in [Moh+20], direct Monte Carlo estimator the score-function

$$\nabla_\lambda \mathbb{E}_{p^\lambda}[f] = \mathbb{E}_{p^\lambda}\left[\nabla_\lambda \log p^\lambda \times f\right] ,$$

for some functional f, yields high variance and should typically not be used without a proper variance reduction technique. A classical technique to reduce this variance is to design a *control variate*. Recalling that $\mathbb{E}_{p^\lambda}\left[\nabla_\lambda \log p^\lambda\right] = 0$, we remark that

$$\nabla_\lambda \mathbb{E}_{p^\lambda}[f] = \mathbb{E}_{p^\lambda}\left[\nabla_\lambda \log p^\lambda \times \left(f - \mathbb{E}_{p^\lambda}[f]\right)\right] .$$

Therefore, by plugging an estimate of $\mathbb{E}_{p^\lambda}[f]$ (this estimate is called a *baseline* in machine learning literature) into a classical Monte Carlo approximation of the right-hand side expectation, one can obtain an alternative Monte Carlo estimate which could[5] be better in practice. Conveniently, in our setting, baselines for the apprixmation of scores are readily available with the $\{\hat{H}_t^i\}_{i\leq N}$, computed by Algorithm 1 i.e.

[4]This recursive approximation can be obtained in a way completely analogous to that of Equation (4.7), where the $\tilde{h}_t(\cdot,\cdot)$ are replaced by their gradients.

[5]This is actually not ensured when using this direct and naïve approach, see [Moh+20] for a detailled discussion.

- For the expectation of Equation (4.10) and its Monte Carlo approximation (4.11), $\hat{H}_t^i$ is a baseline to approximate $\mathbb{E}_{q_{t-1|t}^{\lambda}(\xi_t^i)}\left[H_{t-1}^{\lambda}(X_{t-1}) + \tilde{h}_t^{\lambda}(X_{t-1}, \xi_t^i)\right]$, and therefore, no additional computation is needed to implement the control variate method.
- Similarly, for the expectation of Equation (4.9) and its Monte Carlo approximation (4.12) $\frac{1}{N}\sum_{i=1}^{N} \hat{H}_t^i$ is a readily avalaible baseline for variance reduction.

Therefore, our methodology comes built-in with variance reduction without having to recompute additional quantities. Algorithm 2 provides the pseudo-code of online optimization algorithm of some objective $\mathbb{E}_{q_{0:t}^{\lambda}}[h_{0:t}]$, which includes as particular case the ELBO with components $\tilde{h}_t^{\lambda,\theta}$.

Algorithm 2 One iteration of the online gradient ascent algorithm

Require: $\{\hat{G}_{t-1}^i, \hat{H}_{t-1}^i\}_{i=1}^N$, λ_{t-1}, γ_t.
Ensure: $\{\hat{G}_t^i, \hat{H}_t^i\}_{i=1}^N$, λ_t.
Compute the parameters of $q_t^{\lambda_{t-1}}$ and sample $\{\xi_t^i\}_{i=1}^N$ i.i.d. with distribution $q_t^{\lambda_{t-1}}$.
for $i = 1$ to $i = N$ **do**
for $j = 1$ to $j = M$ **do**
Sample $J_t^{i,j}$ in $\{1, \ldots, N\}$ with probabilities proportional to $\psi_t^{\lambda_{t-1}}(\xi_{t-1}^{\ell}, \xi_t^i)$, $1 \leq \ell \leq M$.
end for
Compute

$$\hat{H}_t^i = \frac{1}{M}\sum_{j=1}^{M}\left\{\hat{H}_{t-1}^{J_t^{i,j}} + \tilde{h}_t^{\lambda_{t-1}}(\xi_{t-1}^{J_t^{i,j}}, \xi_t^i)\right\},$$

$$\hat{G}_t^i = \frac{1}{M}\sum_{j=1}^{M}\left\{\hat{G}_{t-1}^{J_t^{i,j}} + \tilde{s}_t^{\lambda_{t-1}}(\xi_{t-1}^{J_t^{i,j}}, \xi_t^i)\left(\hat{H}_{t-1}^{J_t^{i,j}} + \tilde{h}_t^{\lambda_{t-1}}(\xi_{t-1}^{J_t^{i,j}}, \xi_t^i) - \hat{H}_t^i\right)\right\}.$$

end for

$$\lambda_t = \lambda_{t-1} + \frac{\gamma_t}{N}\sum_{i=1}^{N} G_t^i + \left(\nabla_\lambda \log q_t\right)^{\lambda_{t-1}}(\xi_t^i)\left(\hat{H}_t^i - \frac{1}{N}\sum_{i=1}^{N}\hat{H}_t^i\right).$$

4.4 Experiments

In all this section, the states are p-dimensional real-valued random variables, and the observations are q-dimensional random variables, i.e. we consider state-spaces with $\mathsf{X} = \mathbb{R}^p$ and $\mathsf{Y} = \mathbb{R}^q$.

4.4.1 Linear-Gaussian HMM

We first evaluate our solution on data for which the smoothing recursions are analytical in θ, such that optimal smoothing is available. This is the case whenever the generative model is

the following Linear-Gaussian HMM with $X_t \in \mathsf{X}$, $Y_t \in \mathsf{Y}$ following

$$X_0 \sim \mathcal{N}(\mu_0^\theta, Q_0^\theta),\ X_t = A^\theta X_{t-1} + \eta\ , t \geq 1\ ,$$
$$Y_t = B^\theta X_t + \epsilon\ , t \geq 0\ ,$$

where μ_0^θ is any vector in X, Q_0^θ is a p-dimensional symetric positive-definite matrix, A^θ is a p-dimensional square matrix with eigenvalues in $]-1, 1[$, B^θ any $p \times q$-dimensional matrix, $\eta \sim \mathcal{N}(0, Q^\theta)$ and $\epsilon \sim \mathcal{N}(0, R^\theta)$, with Q^θ, R^θ respectively p and q-dimensional symmetric positive-definite matrices. In this case the Kalman smoothing recursions yield the best[6] possible estimate of the distribution $\phi_{0:t}^\theta$.
For this experiment, it is possible to choose a variational model parameterizd by λ which gets arbitrarily close to the true posterior by prescribing that $q_{0:t}^\lambda$ is also the smoothing distribution of a Linear-Gaussian HMM. This is a special case of our general setting from Section 4.2.1 where we define a model in λ similar to the true model in θ with parameters

$$\lambda = \left\{\mu_0^\lambda, Q_0^\lambda, A^\lambda, B^\lambda, Q^\lambda, R^\lambda\right\}\ ,$$

such that the $(q_t^\lambda)_{t\geq 0}$ derive from Kalman recursions under this model and $\psi_t^\lambda(x_{t-1}, x_t) \propto q^\lambda(x_t|x_{t-1})$ where the latter term is the p.d.f of the distribution of X_t given X_{t-1} under this model. In this case, the ELBO can also be computed recursively in closed-form because the induced variational backward kernels are linear-Gaussian kernels and the conditional expectations $(H_t^\lambda)_{t\geq 0}$ are quadratic forms.

Learning in an offline setting. We first evaluate our algorithm on a sequence of fixed-length T to evaluate whether the proposed framework indeed enables to perform a gradient ascent algorithm. As an oracle baseline, we can compute the closed-form ELBO and its associated gradient via the reparameterization trick. In all that follows, we call "pathwise" such gradients obtained from reparameterization, following [Moh+20]. In this case where we have access to all observations at once, it is also possible to compute an unbiased Monte Carlo approximation of $\mathcal{L}_T^{\lambda,\theta}$ with $\mathcal{L}_T^{\lambda,\theta} \approx 1/N \sum_{i=1}^N h_{0:T}^{\lambda,\theta}(\xi_{0:T}^i)$ where $(\xi_{0:T}^i)_{i\leq N}$ are i.i.d sequences from $q_{0:T}^\lambda$ sampled via backward sampling starting from q_T^λ then $(q_{t-1|t}^\lambda)_{t\leq T}$. The associated pathwise gradient yields an unbiased Monte Carlo gradient to which we can also compare. Our solution doesn't sample from the same sequences of distributions as the latter, yet in this setting both share a common base measure $\mathcal{N}(0, 1)$, so, for fair comparison, we prescribe a fixed overall sampling budget of N samples per timestep for the two methods.

To compare the three methods at hand, we evaluate our ability to perform gradient-ascent to optimize the ELBO with respect to λ. Figure 4.1 displays the evolution of the ELBO using gradients approximated via all approaches.
For the Monte Carlo approaches, we also plot the evolution of the analytical ELBO computed on the running parameter. In Table 4.1, we report the marginal smoothing root-mean-squared distance with the optimum, averaged over time and dimension, i.e. the distance between the t-th marginal of $\phi_{0:T}^\theta$ obtained with Kalman smoothing on θ and that of $q_{0:T}^\lambda$ at the end of optimization, given by

$$\frac{1}{T}\sum_{t=1}^{T}\sqrt{\frac{1}{p}\sum_{d=1}^{p}\left(\mathbb{E}_{\phi_{0:T}^\theta}\left[X_t^{(d)}\right] - \mathbb{E}_{q_{0:T}^\lambda}\left[X_t^{(d)}\right]\right)^2}\ .$$

[6] In the sense of quadratic loss.

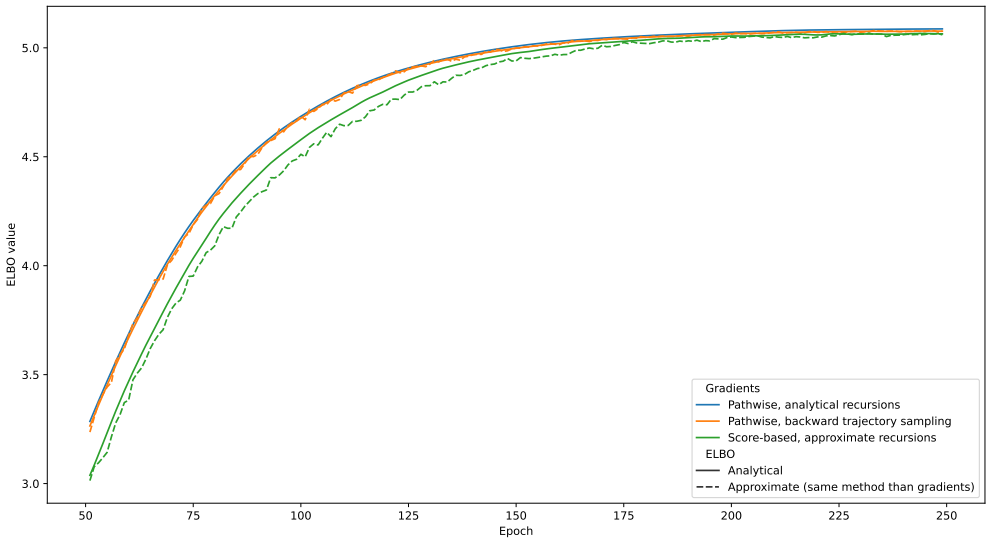


Figure 4.1: Evolution of $\frac{1}{T}\mathcal{L}_T^{\lambda,\theta}$ computed with three differents methods and with three different types of gradients

We run the experiment 10 times and report variability and best run at evaluation (Figure 4.1 only depicts one run for readability). Finally, we provide the computational times per gradient step averaged over all runs. In this experiment, we chose $p = q = 10$, $T = 500$ and $N = 2$ for the Monte Carlo methods.

As expected, the analytical gradients lead to a faster optimization of the ELBO than the Monte Carlo counterparts, which is reflected at evaluation with marginal estimates closest to the optimum. Next, the pathwise trajectory gradients provide accurate estimates with very few samples, the unbiasedness in both ELBO and its gradients is visible with the blue and orange (both dotted and full line) following a similar path in Λ. Using our sample-based approximation of the backward expectations, the bias in the ELBO evaluation is visible with the dotted green line not centered around the full green line, and the biased score-based gradients lead to a slightly poorer convergence.
However, overall, the optimum reached using our solution is very close to the optimum from backward trajectory sampling, which is a benchmark method not amenable to online recursive learning at all. In particular, the best runs of the two methods lead to identical evaluation results. Furthermore, the computational time of our method is in the same order of magnitude, despite the added computational load of propagating gradients via the update of Equation (4.10).

Online learning from streaming data. In a second setting, we keep the same generative and variational models and dimensionality but generate a large sequence of $T = 100000$ observations and simulate the optimization of the joint ELBO $\mathcal{L}_T^{\lambda,\theta}$. The purpose here is to update the variational parameters online, i.e. by discarding already seen data at each step. In the context of stochastic optimization, since $\mathcal{L}_T^{\lambda,\theta} = \sum_{t=0}^{T} \mathcal{L}_t^{\lambda,\theta} - \mathcal{L}_{t-1}^{\lambda,\theta}$ (with the convention $\mathcal{L}_{-1}^{\lambda,\theta} = 0$), the right quantity to optimize becomes $\nabla_\lambda\{\mathcal{L}_t^{\lambda,\theta} - \mathcal{L}_{t-1}^{\lambda,\theta}\}$. In practice we update

Gradients	Marginal smoothing RMSE	Avg. time per grad. step
Pathwise, analytical recursions	0.003 $\pm$ 0.002 (best 0.001)	37.1 ms
Pathwise, backward trajectory sampling	0.005 $\pm$ 0.002 (best 0.004)	50.5 ms
Score-based, approximate recursions	0.007 $\pm$ 0.002 (best 0.004)	89.5 ms

Table 4.1: Marginal errors for the linear models trained on a batch of $T = 500$ observations, against the optimal ones, averaged across dimensions

Gradients	Marginal smoothing RMSE	Avg. time per grad. step
Ours	0.004 $\pm$ 0.001 (best 0.002)	13.1 ms

Table 4.2: Smoothing performance when $q^{\lambda}_{0:t}$ is trained on streaming data

λ_{t+1} by setting:

$$\lambda_{t+1} = \lambda_t + \gamma_{t+1} \left(\nabla_\lambda \mathcal{L}_t^{\lambda,\theta} \Big|_{\lambda_t} - \nabla_\lambda \mathcal{L}_{t-1}^{\lambda,\theta} \Big|_{\lambda_{t-1}} \right) , \tag{4.13}$$

in order to avoid recomputing the previous gradient[7]. This experiment is performed 10 times and we report in Table 4.2 the marginal smoothing errors on the 100000 observations.

4.4.2 Chaotic recurrent neural network.

We now consider the setting first introduced in [Zha+22] and used in [Cam+21] which models latent chaotic dynamics combined with heavy-tailed observation noise as follows:

$$X_0 \sim \mathcal{N}(0, Q), X_t = X_{t-1} + \frac{\Delta}{\tau} \left(\gamma W \tanh \left(X_{t-1} \right) - X_{t-1} \right) + \eta, \ t \geq 1$$
$$Y_t = X_t + \epsilon, \ t \geq 0 \ ,$$

where $\eta \sim \mathcal{N}(0, Q)$ is an isotropic Gaussian distribution and ϵ is a Student-t distribution, these two distributions being mutually independent and time-homogeneous. In practice, we choose the same hyperparameters than [Cam+21] with $\Delta = 0.001, \tau = 0.025, \gamma = 2.5$, 2 degrees of freedom and a scale of 0.1 for the Student-t distribution, and define Q as a diagonal matrix with entries equal to 0.001.

Learning in an offline setting. Again, we start by evaluating the performance of our gradients against the backward trajectory sampling approach run on the same model, for a sequence of fixed length $T = 500$ with state and observation dimension equal to $p = q = 5$. For the variational family, we build a special case of our general framework and rely on the idea of conjugacy as used in [Joh+16] to encode the observations and combine this with the idea of conjugate potentials. Formally, we stay in the Gaussian family, define a linear-Gaussian kernel $q^\lambda(x_t|x_{t-1})$ and prescribe that $\psi_t^\lambda(x_{t-1}, x_t) \propto q^\lambda(x_t|x_{t-1})$ as in the previous section. Then, we define an encoder network e^λ such that $e^\lambda(y_t)$ is a natural parameter for the Gaussian family. We denote $\eta_{t|t-1}$ the natural parameter of the distribution with p.d.f $x_t \mapsto \mathbb{E}_{q^\lambda_{t-1}} \left[q(x_t|X_{t-1}) \right]$. The sequence of natural parameters $(\eta_t^\lambda)_{t\geq 0}$ of the distributions $(q_t^\lambda)_{t\geq 0}$ is then prescribed by the recursion

$$\eta_t^\lambda = \eta_{t|t-1}^\lambda + e^\lambda(y_t) \ .$$

[7] This approximation is typically made in traditional recursive maximum likelihood methods

Gradients	Marginal smoothing RMSE	Avg. time per grad. step
Score-based, approximate recursions	0.135 $\pm$ 0.007 (best 0.122)	173 ms
Pathwise, backward trajectory sampling	0.119 $\pm$ 0.004 (best 0.114)	17 ms

Table 4.3: RMSE between the true states x_t^* and the predicted marginal means $\mathbb{E}_{q_{0:T}^\lambda}[X_t]$

The use of a linear-Gaussian kernel for $q^\lambda(x_t|x_{t-1})$ makes the computation of the natural parameter $\eta_{t|t-1}^\lambda$ analytical, similar to a Kalman predict step. We run gradient-ascent on λ by performing gradient steps using the quantity $\frac{1}{T}\nabla_\lambda \mathcal{L}_T^{\lambda,\theta}$ approximated via backward trajectory sampling and via our score-based method. As before, we use the same hyperparameters and optimization schemes for both methods. Table 4.3 reports the performance against the true states, averaged over dimensions, i.e. the quantity

$$\frac{1}{T}\sum_{t=1}^{T}\sqrt{\frac{1}{p}\sum_{d=1}^{p}\left(x_t^{*(d)} - \mathbb{E}_{q_{0:T}^\lambda}\left[X_t^{(d)}\right]\right)^2}\,.$$

Recursive gradients for faster convergence in the offline setting. Even when we have access to an entire sequence of observations $y_{0:T}$, it can still be beneficial to use the recursive gradients approach for faster convergence. Indeed, when gradients are only available after processing the whole *batch*[8], the best we can do at optimization given a fixed number of observations T is to update the parameter with

$$\lambda^{(k+1)} = \lambda^{(k)} + \gamma_{k+1}\,\nabla_\lambda \mathcal{L}_T^{\lambda,\theta}\Big|_{\lambda^{(k)}}\,, \tag{4.14}$$

where one such update is usually referred to as an "epoch" (one iterate processes the whole set of observations), and $\lambda^{(k)}$ is the value of estimated parameter afer k epochs. Using the recursive gradients, one may perform T intermediate updates within an epoch using

$$\lambda_{t+1}^{(k)} = \lambda_t^{(k)} + \gamma_{t+1}^{(k)}\left\{\nabla_\lambda \mathcal{L}_{t+1}^{\lambda,\theta}\Big|_{\lambda_t^{(k)}} - \nabla_\lambda \mathcal{L}_t^{\lambda,\theta}\Big|_{\lambda_{t-1}^{(k)}}\right\}\,, \tag{4.15}$$

and

$$\lambda_0^{(k+1)} = \lambda_T^{(k)}\,,$$

i.e. inside one epoch we optimize λ recursively on the observations. We compare the two options by optimizing on 10 different sequences of $T = 500$ observations, performing 10 epochs on each, using updates of the form (4.14) for the backward trajectory sampling approach and using updates of the form (4.15) with our score-based approach. Figure 4.2, displays the epoch-wise training curves for each method with $p = q = 5$, where we observe that optimizing with intermediate updates of Equation (4.15) converges faster overall.

Comparison with [Cam+21]. As discussed in Section 4.1, the proposed method of this paper mainly differs from [Cam+21] in the way we approximate the backward statistics H_t^λ by using a recursive sampling approach rather than a regression approach. In order to compare

[8] i.e. the whole set of observation

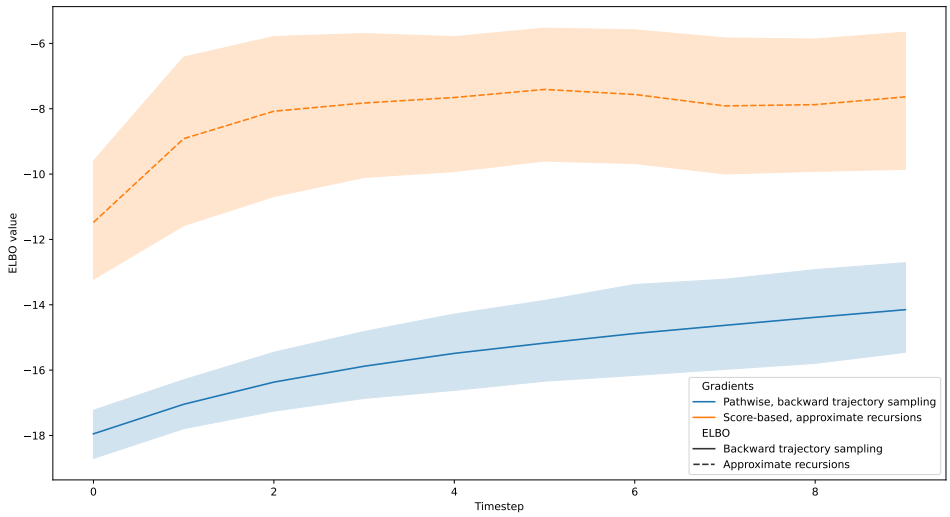


Figure 4.2: Evolution of $\frac{1}{T}\mathcal{L}_T^{\lambda,\theta}$ for $\lambda = \lambda^{(k)}$ when performing with temporal updates inside an epoch (via recursive gradients) or without (via offline pathwise gradients), $k \in \{0, \dots, 10\}$.

the two approaches, we reproduce the experiment of appendix B.2 of [Cam+21], where the authors evaluate their ability to predict the hidden state one step backward (therefore performing 1-step smoothing).

Specifically, we aim at evaluating the quality of our approach to estimate the conditional law of X_{t-1} given $Y_{0:t}$ and of X_{t-1} given $Y_{0:t}$ by evaluating $\mathbb{E}_{q^\lambda_{t-1:t}}\left[X^{(d)}_{t-1}\right]$ and $\mathbb{E}_{q^\lambda_t}\left[X^{(d)}_t\right]$, where λ is learnt using the same setting as [Cam+21]. In their setting, there is no amortization, which means, in this context, that the parameters of $q^\lambda_{t-1:t}$ are given by a set λ_t wich is not related to the parameters λ_{t+1}. Therefore, one can see λ as of a large set $\{\lambda_0, \dots, \lambda_T\}$ whose components can be optimized separately. This framework leads to a large parameter set but has the advantage of allowing to optimize each λ_s separately, which is viable when we do not seek to jointly optimize the parameter λ on a large sequence but rather want the best parameter for the current timestep. To compare our approach with this non-amortized version, we build a non-amortized family where, at t, we directly optimize the parameter $\lambda_t = (\mu_t, \Sigma_t)$ of the distribution $q^\lambda_t \sim \mathcal{N}(\mu_t, \Sigma_t)$ and the parameter $\lambda_{t-1|t}$ of the function $\psi_t^{\lambda_{t-1|t}}$, such that $(\lambda_t, \lambda_{t-1|t})$ is the quantity optimized at the t-th timestep. For this latter function, we match the number of parameters by defining $\psi^\lambda_t(x, y) = \exp\left(\bar{\eta}^\lambda_t(y) \cdot T(x)\right)$ with $\bar{\eta}^\lambda_t(y) = (\bar{\eta}^\lambda_{t,1}(y), \bar{\eta}_{t,2})$ where $y \mapsto \bar{\eta}^\lambda_{t,1}(y)$ is a multi-layer perceptron with 100 neurons from X to X, and $\bar{\eta}_{t,2}$ is a negative definite matrix. We use the same optimization schedules as [Cam+21] with $K = 500$ gradient steps per timestep t.

Table 4.4, reports the average 1-step smoothing errors and filtering errors, i.e. the quantities

$$\frac{1}{T-1}\sum_{t=1}^{T-1}\sqrt{\frac{1}{p}\sum_{d=1}^{p}\left(\mathbb{E}_{q^\lambda_{t-1:t}}\left[X^{(d)}_{t-1}\right] - x^{*(d)}_{t-1}\right)^2}$$

Method	1-step smoothing RMSE	Filtering RMSE	Avg time per grad. step
Ours	0.089 ($\pm$ 0.002)	0.103 ($\pm$ 0.002)	1 ms
[Cam+21]	0.092 ($\pm$ 0.002)	0.103 ($\pm$ 0.002)	4.8 ms

Table 4.4: RMSE between the true states x_t^* and (i) the 1-step smoothing estimates $\mathbb{E}_{q^\lambda_{t-1:t}}\left[X_{t-1}^{(d)}\right]$ and (ii) the filtering estimates $\mathbb{E}_{q_t^\lambda}\left[X_t^{(d)}\right]$

and

$$\frac{1}{T}\sum_{t=1}^{T}\sqrt{\frac{1}{p}\sum_{d=1}^{p}\left(\mathbb{E}_{q_t^\lambda}\left[X_t^{(d)}\right]-x_t^{*(d)}\right)^2}$$

when training our method in these two settings with $p = q = 5$. We also report the errors the computational times for the two methods averaged over 10 runs using 10 different values of θ (hence 10 different sequences).

One can see that for comparable results, our approach based on Monte Carlo for estimating the backward expectation is about 5 times faster thant the regression approach.

Learning on streaming data. Finally, we evaluate the performance in the true online setting when training on a sequence of $T = 300,000$ observations using parameter updates of the form of (4.13). We choose $p = q = 10$ and $N = 200$ particles. To parameterize $q^\lambda_{0:t}$ we use the amortized model presented at the beginning of this section. Table 4.5 provides the smoothing and filtering RMSE against the true states at the end of optimization. We also show the inference performance on new sequences generated under the same parameter θ than the training sequence but with different random seeds. The results clearly highlight that the fitted λ is relevant for new sequences, which makes this scenario particularly appealing when one wants to train a single inference model on a long stream of incoming data, then re-use it for e.g. offline inference on new sequences of arbitrary length.

Sequence	Smoothing RMSE	Filtering RMSE
Training	0.281	0.311
Eval	0.278 ($\pm$ 0.01)	0.305 ($\pm$ 0.014)

Table 4.5: Smoothing and filtering RMSE values for the training sequence and other sequences drawn from the same θ, when λ is learnt online.

4.5 Conclusion

4.5.1 Summary

In this work, we have presented an efficient variational approximation of conditional backward expectations for additive smoothing, and new gradient computations which are based on the latter to enable online learning of variational parameters. At the core of these algorithms, we proposed a decomposition of variational backward kernels which enables to use known ideas from recursive particle-based smoothing. Consequently, the resulting algorithms

are conceptually simpler than previous approaches based on functional approximations, and also more efficient by leveraging recent advances from SMC literature. Despite this, our backward approximation is very flexible and retains the generality of the backward factorization introduced in [Cam+21]. Finally, the averaged statistics of the approximated quantities at any timestep provide straightfoward control variates that can drastically reduce the variance of the gradients, hence competing with reparameterization-based gradients despite the score-function estimator.

To demonstrate the performance of our estimators, we have experimentally compared the convergence of the joint variational objective (the ELBO) against oracles for both Linear-Gaussian and nonlinear non-Gaussian generative models in the offline setting, where we have observed that our biased gradients lead to similar optima than unbiased counterparts (which are not amenable to online learning). Additionally, we have also illustrated the relevance of recursive gradients to accelerate convergence in the batch setting. Finally, and most importantly, we have conducted experiments in the context of true streaming data, where we have demonstrated the stability of our online learning methodology on very long sequences generated with nonlinear models, which had was previously not possible. All in all, we believe that this work is promising to efficiently use the backward variational factorization in online scenarios.

4.5.2 Perspectives

For future research, we identify two directions that could benefit from further investigation. First, we have only implemented the versions of our algorithm that rely on exponentially conjugated potentials, and as such more general parameterizations need to be evaluated. In practice, when the forward potentials $(\psi_t^\lambda)_{t\geq 0}$ are arbitrarily parameterized functions, it is expected that more flexible joint variational approximations can be obtained, and hence better results under complex nonlinear models. Nonetheless, in this case, as mentioned in Section 4.2.2, the normalization constant in the p.d.f. of the variational kernels needs to be estimated, and it remains to understand the impact of this additional approximation on the overall performance.

Then, a more thorough analysis could be conducted to study the proper *stepwise* objective to optimize in situations where parameter updates are performed at every timestep. Indeed, in the context of recursive MLE, we presented in Section 2.1.2 the various decompositions of the log-likelihood can be used in this setting, in particular via the incremental log-likelihood. In this work, we have relied on the decomposition $\mathcal{L}_t^{\lambda,\theta} = \sum_{s=1}^t \mathcal{L}_s^{\lambda,\theta} - \mathcal{L}_{s-1}^{\lambda,\theta}$ as a justification to solve the optimization problem in λ via online stochastic gradient updates which maximize the ELBO over time. However, while the global objective $\mathcal{L}_t^{\lambda,\theta}$ is a lower bound of the log-likelihood l_t^θ at any timestep $t \geq 0$, the differences $\{\mathcal{L}_t^{\lambda,\theta} - \mathcal{L}_{t-1}^{\lambda,\theta}\}_{t\geq 1}$ are a priori not lower bounds. In particular they are not lower bounds of the incremental log-likelihood $r_t^\theta = l_t^\theta - l_{t-1}^\theta$. As such, perfoming multiple parameter updates in the direction of $\nabla_\lambda\{\mathcal{L}_t^{\lambda,\theta} - \mathcal{L}_{t-1}^{\lambda,\theta}\}$ at each timestep is not guaranteed to be stable. As an alternative, [DZP23] have recently explored online variational optimization by deriving explicit lower bounds on r_t^θ, albeit not in the context of smoothing and without relying on the backward factorization, which requires additional assumptions.

Chapter 5

Additive smoothing error in backward variational inference for general state-space models

This chapter is based on the article "Additive smoothing error in backward variational inference for general state-space models" under revision in JMLR, the Journal of Machine Learning Research, [Cha+22].

In this contribution, we establish upper bounds for the error of the variational approximation of additive smoothing in state-space-models (see in particular Proposition 1 and Proposition 3), when the target expectations are approximated by expectations under a variational distribution satisfying the backward factorization of [Cam+21]. The backward factorization of the variational posterior allows the decomposition of the global error into a sum of terms that can be controlled. To the best of our knowledge, these are the first theoretical results providing upper bounds on the state estimation error when using the latter, or in fact any variational posterior approximation (mean field or involving dependencies) in state-space models. This result is obtained in the context of a fixed sized sequence of observations, but leads to open questions in the context of online learning.

These theoretical results are empirically validated with various numerical experiments which also explore several choices of variational kernels. We consider linear and Gaussian state spaces to illustrate the linear growth as the ground truth can be computed in this case. We also use the backward variational approach in the case of nonlinear emission densities and compare it to sequential Monte Carlo smoothers and other state-of-the-art variational estimators. We finally explore the impact of the backward parametrization with nonlinear hidden dynamics and non-Gaussian observation noise in the framework proposed by [Zha+22].

5.1 A control on backward variational additive smoothing

Notations In the following, we consider the state space models as introduced in Section 2.1. The notations for the true model[1] remains the same. The notations for variational quantities remains the same as in Section 2.3.2. As a minor modification, in all this contribution, indices for quantities that depend on time are different that in the rest of this thesis. Here, we use "n" for the final time index given sequences $y_{0:n}$ of length $n+1$, and "k" for all other timesteps. Other than this the previous notations hold.

5.1.1 Assumption and main result

For all $x_k \in \mathbb{R}^d$ and $\theta \in \Theta$, define $\mathbf{L}_k^\theta(x_k, \cdot)$ the kernel with density $\ell_k^\theta(x_k, \cdot)$ with respect to the Lebesgue measure $\mu(\cdot)$:

$$\mathbf{L}_k^\theta(x_k, \mathrm{d}x_{k+1}) = m_k^\theta(x_k, x_{k+1}) g_{k+1}^\theta(x_{k+1}, Y_{k+1}) \mu(\mathrm{d}x_{k+1}) \ .$$

For additive functionals as in (2.8), the error between the target expectation $\phi_{0:n}^\theta h_{0:n}$ and its approximation $q_{0:n}^\lambda h_{0:n}$ can be upper bounded by controlling the bias in the estimation of $\mathbf{L}_k^\theta$ by the approximated model, see for instance [GLO22]. In the context of this paper, as the true model is defined by the forward distributions of X_k given X_{k-1}, and the variational approximation is defined by the backward distributions of X_{k-1} given X_k, we reformulate the discrepancy between the true model and the variational one as follows.

For all sequences of probability densities $\{\tilde{q}_k\}_{0 \le k \le n-1}$ with respect to μ, with the condition $\tilde{q}_n = q_n^\lambda$ with q_n^λ defined in (2.28), let $\tilde{\nu}_{k-1:k}^\lambda$ and $\tilde{\phi}_{k-1:k}^\theta$ be the distributions on $(\mathbb{R}^d \times \mathbb{R}^d, \mathcal{B}(\mathbb{R}^d \times \mathbb{R}^d))$ defined, for all bounded measurable functions h on $\mathbb{R}^d \times \mathbb{R}^d$, by

$$\tilde{\nu}_{k-1:k}^\lambda h = \tilde{q}_k q_{k-1|k}^\lambda h = \int \tilde{q}_k(x_k) q_{k-1|k}^\lambda(x_k, x_{k-1}) h(x_{k-1}, x_k) \mu(\mathrm{d}x_{k-1}, \mathrm{d}x_k) \ ,$$

$$\tilde{\phi}_{k-1:k}^\theta h = \frac{\tilde{q}_{k-1} \mathbf{L}_{k-1}^\theta h}{\tilde{q}_{k-1} \mathbf{L}_{k-1}^\theta \mathbb{1}} = \int \frac{\tilde{q}_{k-1}(x_{k-1}) \ell_{k-1}^\theta(x_{k-1}, x_k) h(x_{k-1}, x_k)}{\int \tilde{q}_{k-1}(u_{k-1}) \ell_{k-1}^\theta(u_{k-1}, u_k) \mu(\mathrm{d}u_{k-1}, \mathrm{d}u_k)} \mu(\mathrm{d}x_{k-1}, \mathrm{d}x_k) \ .$$

The discrepancy between these sequences of joint distributions is then defined with:

$$\tilde{c}_0(\theta) = \left\| \tilde{q}_0 - \phi_0^\theta \right\|_{\mathrm{tv}}, \quad \text{and for all } k \ge 1 \quad \tilde{c}_k(\theta, \lambda) = \left\| \tilde{\phi}_{k-1:k}^\theta - \tilde{\nu}_{k-1:k}^\lambda \right\|_{\mathrm{tv}} \ , \tag{5.1}$$

where $\|\cdot\|_{\mathrm{tv}}$ is the total variation norm, and for all bounded measurable function h,

$$\phi_0^\theta h = \chi^\theta g_0^\theta h / \chi^\theta g_0^\theta \mathbb{1} \ .$$

Note that for $k \ge 1$, $\tilde{c}_k(\theta, \lambda)$ depends on both $\tilde{q}_k$ and $\tilde{q}_{k+1}$.

H1 There exist constants $0 < \sigma_- < \sigma_+ < \infty$ such that for all $k \in \mathbb{N}$, $\theta \in \Theta$, $\lambda \in \Lambda$ and $(x_k, x_{k+1}) \in \mathbb{R}^d \times \mathbb{R}^d$,

$$\sigma_- \le \ell_k^\theta(x_k, x_{k+1}) \le \sigma_+$$

and

$$\sigma_- \le q_{k|k+1}^\lambda(x_{k+1}, x_k) \le \sigma_+.$$

[1] For instance, the smoothing distribution $\phi_{0:n}^\theta$, the transition kernel m_k^θ , the observation density g_{k+1}^θ, etc…

Proposition 1 *Assume that H1 holds. Then, for all $n \in \mathbb{N}$, $\theta \in \Theta$, $\lambda \in \Lambda$, and all additive functionals $h_{0:n}$ as in* (2.8)*, and all probability densities $\tilde{q}_k$, $0 \leq k \leq n-1$, with the condition $\tilde{q}_n = q_n^\lambda$,*

$$\left|q_{0:n}^\lambda h_{0:n} - \phi_{0:n}^\theta h_{0:n}\right| \leq 2\frac{\sigma_+}{\sigma_-} \sum_{k=0}^{n-1} \left\|\tilde{h}_k\right\|_\infty \times \left(\tilde{c}_0(\theta) + \sum_{m=1}^{k} \rho^{k-m+1}\tilde{c}_m(\theta,\lambda) + \tilde{c}_{k+1}(\theta,\lambda) + \sum_{m=k+2}^{n} \rho^{m-k-1}\tilde{c}_m(\theta,\lambda)\right) ,$$

with $\rho = 1 - \sigma_-/\sigma_+$, where σ_- and σ_+ are defined in H1, and $\tilde{c}_0(\theta)$ and $\tilde{c}_m(\theta,\lambda)$, $1 \leq m \leq n$ are defined in (5.1)*.*

Proof The proof is postponed to Appendix C.1. ■

Marginal smoothing distributions are also of utmost importance as they appear in many applications for state estimation problems. These marginal smoothing expectations can be obtained as special cases of expectations of additive functionals, i.e. cases where $\tilde{h}_j = 0$ for all $j \neq k_\star$, for some $0 \leq k_\star \leq n-1$. For this special cas, we have the following corollary.

Corollary 2 *Assume that H1 holds. Then, for all $n \in \mathbb{N}$, $1 \leq k_\star \leq n-1$, $\theta \in \Theta$, $\lambda \in \Lambda$, all bounded measurable functions $\tilde{h}_{k_\star}$ on $\mathbb{R}^d \times \mathbb{R}^d$, and all probability densities $\tilde{q}_k$, $0 \leq k \leq n-1$, with the condition $\tilde{q}_n = q_n^\lambda$,*

$$\left|q_{0:n}^\lambda \bar{h}_{k_\star} - \phi_{0:n}^\theta \bar{h}_{k_\star}\right| \leq 2\frac{\sigma_+}{\sigma_-} \left\|\tilde{h}_{k_\star}\right\|_\infty \times \left(\tilde{c}_0(\theta) + \sum_{m=1}^{k} \rho^{k-m+1}\tilde{c}_m(\theta,\lambda) + \tilde{c}_{k+1}(\theta,\lambda) + \sum_{m=k+2}^{n} \rho^{m-k-1}\tilde{c}_m(\theta,\lambda)\right) ,$$

with $\bar{h}_{k_\star} : x_{0:n} \mapsto \tilde{h}_{k_\star}(x_{k_\star}, x_{k_\star+1})$, $\rho = 1 - \sigma_-/\sigma_+$, where σ_- and σ_+ are defined in H1, and $\tilde{c}_0(\theta)$ and $\tilde{c}_m(\theta,\lambda)$, $1 \leq m \leq n$ are defined in (5.1)*.*

Note that if there exists c_+ such that for all $\theta \in \Theta$, $\lambda \in \Lambda$, $0 \leq m \leq n$, $\tilde{c}_m(\theta,\lambda) \leq c_+(\theta,\lambda)$, by Corollary 2

$$\left|q_{0:n}^\lambda \bar{h}_{k_\star} - \phi_{0:n}^\theta \bar{h}_{k_\star}\right| \leq 4\frac{\sigma_+}{\sigma_-} \left\|\tilde{h}_{k_\star}\right\|_\infty c_+(\theta,\lambda) \left(1 + \frac{\rho}{1-\rho}\right) ,$$

so that the marginal smoothing errors are uniformly bounded in time.

Proof The proof is postponed to Appendix C.1. ■

For all $1 \leq k \leq n$, let $b_{k-1|k}^\theta$ be the backward kernel at time k, defined for all bounded measurable functions h on $\mathbb{R}^d$ and all $x_k \in \mathbb{R}^d$, by

$$b_{k-1|k}^\theta h(x_k) = \frac{\int m_{k-1}^\theta(x_{k-1}, x_k)\phi_{k-1}^\theta(x_{k-1})h(x_{k-1})\mu(\mathrm{d}x_{k-1})}{\int m_{k-1}^\theta(x, x_k)\phi_{k-1}^\theta(x)\mu(\mathrm{d}x)} .$$

When the backward variational kernel is a sharp approximation of the true backward kernel, Proposition 3 provides an explicit control of the smoothing error.

Proposition 3 *Assume that H1 holds. Let $n \in \mathbb{N}$, $\theta \in \Theta$, $\lambda \in \Lambda$. Assume that there exists $\varepsilon > 0$ such that $\left\|q_n^\lambda - \phi_n^\theta\right\|_{\mathrm{tv}} \leq \varepsilon$ and for all $1 \leq k \leq n$, $x_k \in \mathbb{R}^d$, $\left\|q_{k-1|k}^\lambda(x_k, \cdot) - b_{k-1|k}^\theta(x_k, \cdot)\right\|_{\mathrm{tv}} \leq \varepsilon$. Then, for all additive functionals $h_{0:n}$ as in* (2.8)*,*

$$\left|q_{0:n}^\lambda h_{0:n} - \phi_{0:n}^\theta h_{0:n}\right| \leq 4\frac{\sigma_+}{\sigma_-}\left(1 + 2\frac{\rho}{1-\rho}\right)\sum_{k=0}^{n-1}\left\|\tilde{h}_k\right\|_\infty \varepsilon \ ,$$

where $\rho = 1 - \sigma_-/\sigma_+$, with σ_- and σ_+ defined in H1. Therefore, in the case where there exists an upper bound M such that $\sup_{0\leq k\leq n-1}\|\tilde{h}_k\|_\infty \leq M$, then, there exists $c \geq 0$ such that

$$\left|q_{0:n}^\lambda h_{0:n} - \phi_{0:n}^\theta h_{0:n}\right| \leq cn\varepsilon \ .$$

Proof The proof amounts to applying Proposition 1 with for all $0 \leq k \leq n-1$, $\tilde{q}_k = \phi_k^\theta$.

- $\tilde{c}_0(\theta) = \left\|\tilde{q}_0 - \phi_0^\theta\right\|_{\mathrm{tv}} = 0$, as $\tilde{q}_0 = \phi_0^\theta$.
- For all $1 \leq m \leq n-1$,

$$\begin{aligned}
\tilde{c}_m(\theta, \lambda) &= \left\|\tilde{\phi}_{m-1:m}^\theta - \tilde{\nu}_{m-1:m}^\lambda\right\|_{\mathrm{tv}} \ , \\
&= \left\|\frac{\tilde{q}_{m-1}\mathbf{L}_{m-1}^\theta}{\tilde{q}_{m-1}\mathbf{L}_{m-1}^\theta\mathbb{1}} - \tilde{q}_m q_{m-1|m}^\lambda\right\|_{\mathrm{tv}} \ , \\
&\leq \left\|\frac{\phi_{m-1}^\theta\mathbf{L}_{m-1}^\theta}{\phi_{m-1}^\theta\mathbf{L}_{m-1}^\theta\mathbb{1}} - \phi_m^\theta b_{m-1|m}^\theta\right\|_{\mathrm{tv}} + \left\|\phi_m^\theta b_{m-1|m}^\theta - \phi_m^\theta q_{m-1|m}^\lambda\right\|_{\mathrm{tv}} \leq \varepsilon \ ,
\end{aligned}$$

  where the first term in last inequality is zero as $\phi_{m-1}^\theta\mathbf{L}_{m-1}^\theta/\phi_{m-1}^\theta\mathbf{L}_{m-1}^\theta\mathbb{1}$ and $\phi_m^\theta b_{m-1|m}^\theta$ are both equal to the probability density of (X_{m-1}, X_m) given $Y_{0:m}$ under the law of the state-space model parameterized by θ.

- The last term is upper-bounded as follows:

$$\begin{aligned}
\tilde{c}_n(\theta, \lambda) &= \left\|\tilde{\phi}_{n-1:n}^\theta - \tilde{\nu}_{n-1:n}^\lambda\right\|_{\mathrm{tv}} \ , \\
&= \left\|\frac{\tilde{q}_{n-1}\mathbf{L}_{n-1}^\theta}{\tilde{q}_{n-1}\mathbf{L}_{n-1}^\theta\mathbb{1}} - q_n^\lambda q_{n-1|n}^\lambda\right\|_{\mathrm{tv}} \ , \\
&\leq \left\|\frac{\phi_{n-1}^\theta\mathbf{L}_{n-1}^\theta}{\phi_{n-1}^\theta\mathbf{L}_{n-1}^\theta\mathbb{1}} - \phi_n^\theta b_{n-1|n}^\theta\right\|_{\mathrm{tv}} + \left\|\phi_n^\theta b_{n-1|n}^\theta - \phi_n^\theta q_{n-1|n}^\lambda\right\|_{\mathrm{tv}} \\
&\qquad + \left\|\phi_n^\theta q_{n-1|n}^\lambda - q_n^\lambda q_{n-1|n}^\lambda\right\|_{\mathrm{tv}} \leq 2\varepsilon \ ,
\end{aligned}$$

  where the first term in last inequality is zero as $\phi_{n-1}^\theta\mathbf{L}_{n-1}^\theta/\phi_{n-1}^\theta\mathbf{L}_{n-1}^\theta\mathbb{1}$ and $\phi_n^\theta b_{n-1|n}^\theta$ are both equal to the probability density of (X_{n-1}, X_n) given $Y_{0:n}$ under the law of the state-space model parameterized by θ.

■

Remark 4 *By Proposition 1, if there exist h_∞ and c_+ such that for all $0 \leq k \leq n-1$, $\|\tilde{h}_k\|_\infty \leq h_\infty$ and for all $\theta \in \Theta$, $\lambda \in \Lambda$, $0 \leq m \leq n$, $\tilde{c}_m(\theta, \lambda) \leq c_+(\theta, \lambda)$ then*

$$\left|q^\lambda_{0:n} h_{0:n} - \phi^\theta_{0:n} h_{0:n}\right| \leq 4\frac{\sigma_+}{\sigma_-}\left(1 + \frac{\rho}{1-\rho}\right) c_+(\theta, \lambda) h_\infty n \ . \tag{5.2}$$

Remark 5 *Proposition 1 provides a criterion for assessing the sharpness of a variational approximation for $\phi^\theta_{0:n}$. Indeed, for such approximation, write*

$$c_{\inf}(\lambda, \theta) = \inf_{(\tilde{q}_k)_{0\leq k\leq n}} \sum_{k=0}^{n-1} \left(\tilde{c}_0(\theta) + \sum_{m=1}^{k} \rho^{k-m+1} \tilde{c}_m(\theta, \lambda) + \tilde{c}_{k+1}(\theta, \lambda) + \sum_{m=k+2}^{n} \rho^{m-k-1} \tilde{c}_m(\theta, \lambda) \right) \ .$$

Then, if there exist h_∞ that for all $0 \leq k \leq n-1$, $\|\tilde{h}_k\|_\infty \leq h_\infty$, by Proposition 1, we have:

$$\left|q^\lambda_{0:n} h_{0:n} - \phi^\theta_{0:n} h_{0:n}\right| \leq 2\frac{\sigma_+}{\sigma_-} c_{\inf}(\theta, \lambda) h_\infty. \tag{5.3}$$

Although difficult to compute in practice, this criterion might be the focus of future research. An open question here is whether the optimal sequence $(\tilde{q}_k)_{0\leq k\leq n}$ is given by the sequence of true marginal smoothing distributions.

5.1.2 Comments on Proposition 1 and H1

Proposition 1 provides an upper-bound for the smoothing error for additive functionals which is linear in the number of observations. The sharpness of this bound depends on our ability to find a sequence of distributions $(\tilde{q}_k)_{0\leq k\leq n-1}$, so that each $c_k(\theta, \lambda)$, i.e., the total variation distance between $(x_{k-1}, x_k) \mapsto \tilde{q}_k(x_k) q^\lambda_{k-1|k}(x_k, x_{k-1})$ and the probability density proportional to $(x_{k-1}, x_k) \mapsto \tilde{q}_{k-1}(x_{k-1}) \ell^\theta_{k-1}(x_{k-1}, x_k)$, is small.
First, it is worth noting that if q^λ_n is the true filtering distribution at time n and $(q^\lambda_{k-1|k})_{k\geq 1}$ are the true backward distributions, then the unique sequence $(\tilde{q}_k)_{k\geq 1}$ that achieves $\tilde{c}_k(\theta, \lambda) = 0$ for all k is the sequence of true filtering distributions.
However, in generic cases (i.e. non linear gaussian cases), this joint minimization over this sequence of distributions appears to be an open challenge. In Section 5.2.2, we discuss empirically how the backward $q^\lambda_{k-1|k}(x_k, x_{k-1})$ can be parameterized by the user, depending on the form of $\ell^\theta_{k-1}(x_{k-1}, x_k)$ (see the experiments related to the results of Figure 5.2).

Obtaining theoretical guarantees on the variational approximations remains of course an open problem but we believe that Proposition 1 provides a first result in this direction.

About H1. This assumption is rather strong, but typically satisfied in models where the state space is compact. This assumption is classic in the SMC literature in order to obtain quantitative bounds for errors or variance of estimators in the context of smoothing, (see [Dou+11; DL13; OW+17; GLO22]. It is worth noting that in the context of approximating the filtering distributions, weaker assumptions exist (see [CL04; Dou+09]), but the extension of these results to the smoothing context remains an open challenge.

5.2 Numerical experiments

We now present some practical examples of implementations of the backward variational factorization on which we validate our theoretical results.

5.2.1 Linear Gaussian SSMs

A first interesting case is when the variational family *contains* the true model. This is in particular possible when the latter is a linear and Gaussian SSM, i.e. when χ^θ (resp. $m_k^\theta(X_k, \cdot)$ and $g_k^\theta(X_k, \cdot)$) are densities of Gaussian distributions with mean A_0 (resp. AX_k and BX_k) and variance Q_0 (resp. Q and R), such that $\theta = (A_0, Q_0, A, Q, B, R)$. If we define a similar "mirror" model described with another set of parameters $\lambda = (\bar{A}_0, \bar{Q}_0, \bar{A}, \bar{Q}, \bar{B}, \bar{R})$, we can choose $q_n^\lambda \sim \mathcal{N}(\mu_n, \Sigma_n)$ where (μ_n, Σ_n) are provided by the Kalman filtering recursions, and $q_{k-1|k}^\lambda(x_k, x_{k-1}) \sim \mathcal{N}(A_{k-1|k}x_k + b_{k-1|k}, \Sigma_{k-1|k})$ where $(A_{k-1|k}, b_{k-1|k}, \Sigma_{k-1|k})$ are obtained through Kalman smoothing steps. In this case, $q_{0:n}^\lambda$ is of the same form as $\phi_{0:n}^\theta$ and $q_{0:n}^\lambda = \phi_{0:t}^\theta$ when $\lambda = \theta$.

When the latter case is reached, Section 5.1.2 shows that $c_k(\theta, \lambda) = 0$ for all k, suggesting that the additive error vanishes. In this section, we study the case where the parameter θ is known, $d = 5$ and λ is trained on a set of sequences of $n = 50$ observations. The evolution of the ELBO is given in Figure 5.1a. In Figure 5.1b, we depict the controlled term of Proposition 1 in the case of state estimation, i.e. for $h_{0:n} : x_{0:n} \mapsto \sum_{k=0}^n x_k$. This evaluation is performed on $J = 50$ evaluation sequences $(Y_{0:n}^j)_{1 \leq j \leq J}$ of length $n = 500$ sampled from the generative model. Each plot clearly illustrates the linear dependency on the number of observations. We also find that the error rates can vary greatly between parameters $\lambda_1 \neq \lambda_2$, even when $|\mathcal{L}(\theta, \lambda_1) - \mathcal{L}(\theta, \lambda_2)|$ is small. This is observed by computing the errors for different stopping points of the optimization. Additionally, for a given λ, slopes vary across sequences, which highlights the dependency of $(c_k(\theta, \lambda))_{0 \leq k \leq n}$ on the observations.

Appendix C.2.2 provides more implementation details, as well as additional figures for the errors on the marginal distributions.

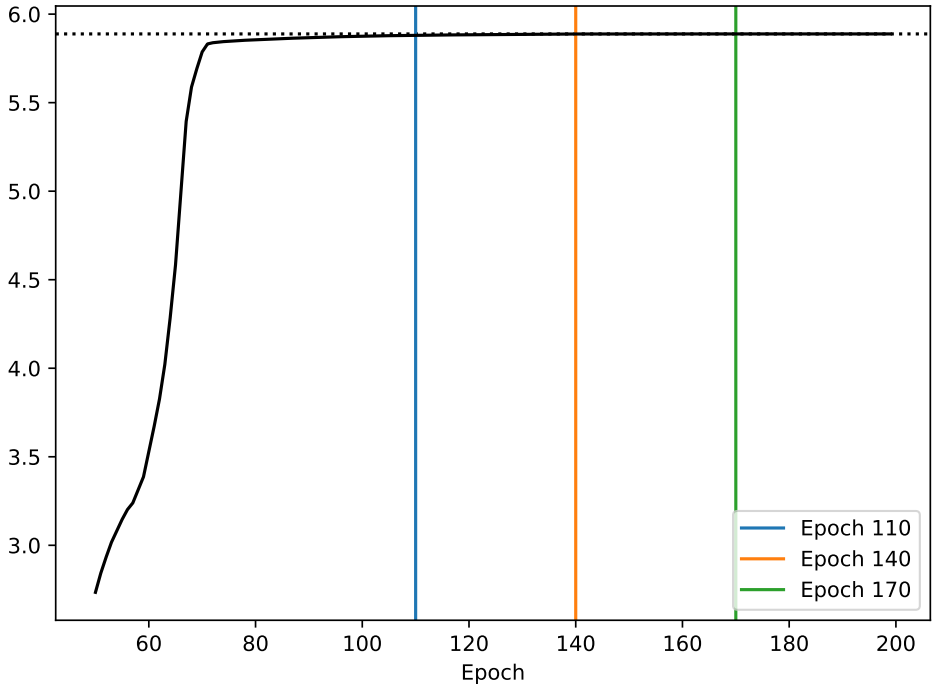
5.2.2 Nonlinear SSMs

The primary motivation to use variational inference is when $\phi_{0:n}^\theta$ cannot be computed analytically, which generally happens when the generative model contains nonlinearities and/or non-Gaussian noises. In this case - contrary to the previous section - there is no obvious choice for the form of the kernels in $q_{0:n}^\lambda$ and many options exist to balance the amount of approximation with the computational complexity. In the next subsections, we revisit some of the literature on sequential variational inference in the backward context to illustrate our theoretical result.

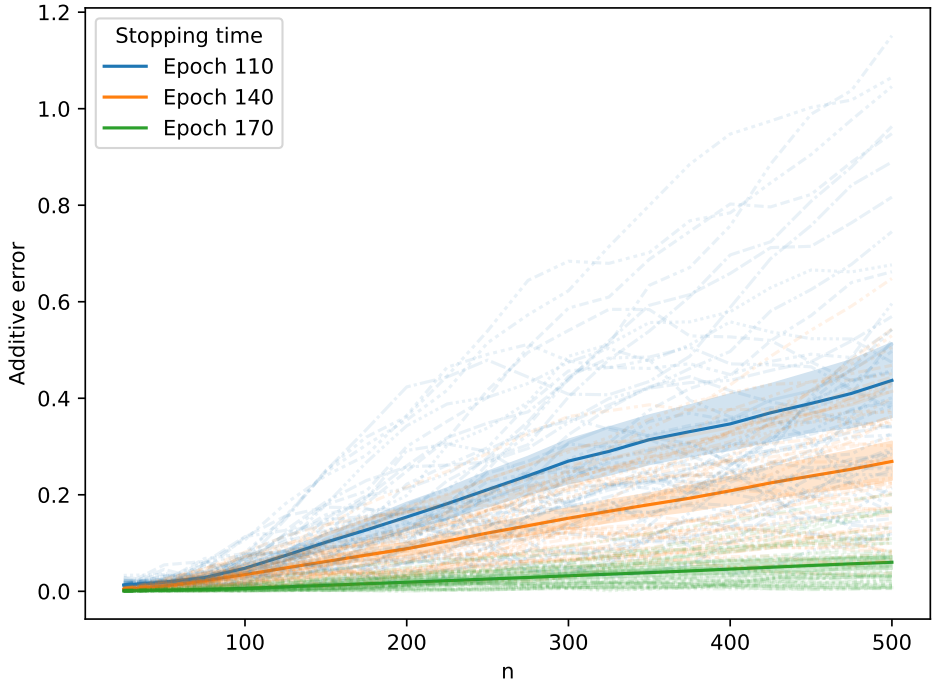
Nonlinearity in the emission distribution

We first consider a generative model where the prior distribution and transition kernels are still linear, but $g_k^\theta(X_k, \cdot)$ is the Gaussian probability density with mean $d^\theta(X_k)$ and variance R, d^θ being a nonlinear mapping commonly referred to as the *decoder*. In this setting, [Häl+21b] showed for the first time that no assumptions are required on d^θ for identifiable state estimation. In particular d^θ need not to be an injective mapping and therefore we use an unconstrained and arbitrary multi layer perceptron (MLP).

In this context, [Häl+21b] obtained promising results via a parameterization of the factors in $q_{0:n}^\lambda$ which relies entirely on Gaussian conjugation and can be analytically marginalized, therefore allowing fast inference. A central element of their approximation is the idea



(a) L_n^θ (dotted line) and $\lambda \mapsto \mathcal{L}(\theta, \lambda)$ over epochs (full line).



(b) $\left|q_{0:n}^\lambda h_{0:n} - \phi_{0:n}^\theta h_{0:n}\right|$ for $\tilde{h}_k(x_k, x_{k+1}) = x_k$. The solid lines display the mean over the 50 independent replicates, the transparent filling is the standard deviation, shaded lines are the all sequences. Values are normalized by the state-space dimension.

Figure 5.1: ELBO during the training of λ (left). Additive smoothing error for a linear Gaussian variational model at successive stopping points of the optimization on 50 different sequences (right)

from [Joh+16], which consists in mapping each observation y_k to a set of valid natural parameters (κ_k, Π_k) for some Gaussian distribution, using an *encoder* network r^λ such that $(\kappa_k, \Pi_k) = r^\lambda(y_k)$. By defining (as in Section 5.2.1) some additional parameters $(\bar{A}_0, \bar{Q}_0, \bar{A}, \bar{Q})$ for kernels $\chi_0^\lambda, m_k^\lambda$ (i.e. similar to the generative model but parameterized by λ) the authors design $q_{0:n}^\lambda$ using forward-backward recursions (see [CMR05, section 3.2.1]) where the forward and backward variables are updated analytically by Gaussian conjugation with the exponential factors $x_k \mapsto e^{\left\langle r^\lambda(y_k), t_{\mathcal{N}}(x_k)\right\rangle}$, $t_{\mathcal{N}}(x_k) = (x_k, x_k x_k^\top)$ being the set of sufficient statistics for a Gaussian distribution in x_k. This algorithm is a special form of two-filter smoothing, which is rather rooted in the alternate *forward* decomposition of the joint smoothing distribution, that is $q_{0:n}^\lambda(x_{0:n}) = q_0^\lambda(x_0) \prod_{k=1}^{n-1} q_{k|k-1}^\lambda(x_{k-1}, x_k)$ where each factor depends on the entire sequence of observations $y_{0:n}$ and is built using the so-called backward *variables* (which are non-normalized quantities distinct to the backward kernels). However, the core idea can be reframed under the backward factorisation very easily by defining a sequence of distributions $(q_k^\lambda)_{k \leq n}$ which are updated from q_{k-1}^λ to q_k^λ via:

- $\bar{q}_k^\lambda(x_k) = \mathbb{E}_{q_{k-1}^\lambda}\left[m_k^\lambda(\cdot, x_k)\right]$ similarly to a Kalman predict step
- $\eta_k = r^\lambda(y_k) + \bar{\eta}_k^\lambda$ where η_k and $\bar{\eta}_k$ are the natural parameters of q_k^λ and $\bar{q}_k^\lambda$, respectively.

and by defining the backward kernels with $q_{k-1|k}^\lambda(x_k, x_{k-1}) \propto q_{k-1}^\lambda(x_{k-1}) m_k^\lambda(x_{k-1}, x_k)$, such that their parameters are derived analytically at each time step from η_{k-1} and the parameters of m_k^λ. We refer to the models of [Joh+16] as the *Conjugate Forward* variational model and to the backward adaptation as the *Conjugate Backward* model.

These solutions are computationally very efficient because they allow closed-form updates of the factors with DNN-predicted encodings which are already Gaussian parameters. Under the backward factorization, more general implementations are possible that still allow analytical marginalisation by keeping the factors in (2.28) conjugated. For example, one may use a recurrent neural network which updates an internal state $(s_k)_{k \leq n}$ from which the backward kernels and the terminal distribution and built analytically via an intermediate linear-Gaussian kernel m_k^λ as before, e.g.

- $s_k = \mathsf{RNN}^\lambda(s_{k-1}, y_k)$ and $q_k^\lambda \sim \mathcal{N}(\mu_k, \Sigma_k)$ where $(\mu_k, \Sigma_k) = \mathsf{MLP}^\lambda(s_k)$
- $q_{k-1|k}^\lambda(x_k, x_{k-1}) \propto q_{k-1}^\lambda(x_{k-1}) m_k^\lambda(x_{k-1}, x_k)$ from which parameters of $q_{k-1|k}^\lambda$ are derived analytically.

We implement such version with a Gated Recurrent Unit (GRU) for the RNN, and refer to it as the *GRU Backward* implementation.

In the nonlinear setting, since the true smoothing distribution $\phi_{0:n}^\theta$ has no analytic form, we use the particle-based Forward Filtering Backward Simulation (FFBSi) algorithm[2] as a surrogate for this ground truth. The FFBSi outputs trajectories approximately sampled from the true target smoothing distributions using sequential importance sampling and resampling steps. This algorithm is also based on a forward-backward decomposition of the smoothing distributions (see [DMS14], Chapter 11, for details). We choose the case $d = 10$, where a high number of particles for the FFBSi (10000 for the bootstrap filtering, 2000 for the backward smoothing) to consider it as a proper ground truth.

[2]Described in Section 2.2.2.

We compare the additive error with respect to the FFBSi (i.e. the left hand term of equation (5.2)) for $h_{0:n} : x_{0:n} \mapsto \sum_{k=0}^{n} x_k$). In appendix, we report the quality of the FFBSi estimator in the form of the sample mean and variance of its error against the true states, which establishes the error made by the oracle reference estimator considered as ground truth.

In Figure 5.2, we plot the evolution of the additive error against this oracle. As predicted by our theoretical result, all backward methods have a linear dependency in the number of observations n. Interestingly, we observe that the *Conjugate Forward* model also shares this property, which suggests that our main theoretical result is also valid for other factorizations. However, while the two-filter formulation brings similar results using the same amount of parameters, it is much less convenient computationally because it requires to compute the entire sequence of backward variables for any new observation.

One hidden aspect of the fully conjugate models is that the natural parameters given by $r^\lambda(y_k)$ implicitly model the distribution of x_k given y_k (unconditionnally on the dynamics), yet this distribution is likely to admit several modes (especially if d^θ is strongly injective on some portions of the support). We observe a slight performance gain for the *GRU Backward* model in this context. In this model, the parameters of the intermediate distributions q_k^λ are updated without any intermediate Gaussian approximation which might explain the better performance.

In Figure 5.3, we provide the marginal errors over time in the same setting. The results coincide with the time-uniform bound presented in Corollary 2.

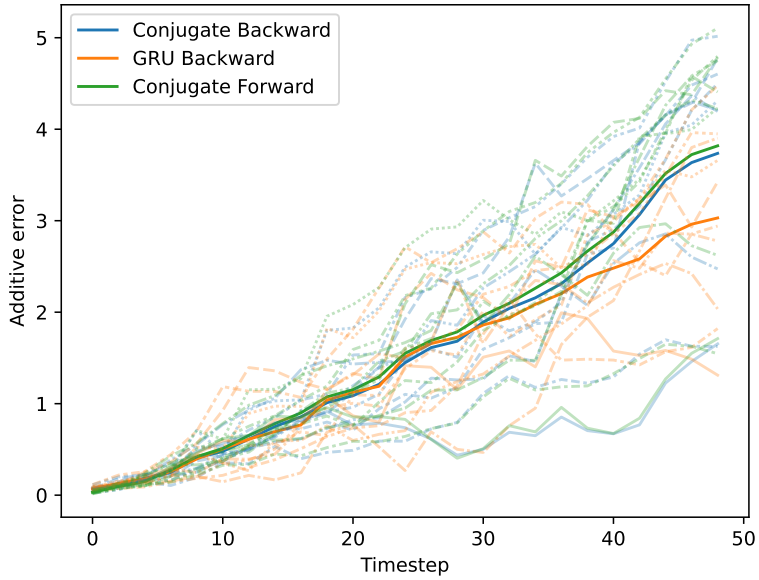


Figure 5.2: Smoothing errors $\left|q_{0:n}^\lambda h_{0:n} - \phi_{0:n}^\theta h_{0:n}\right|$ for $\tilde{h}_k(x_k, x_{k+1}) = x_k$, for the different models with a 10-dimensional latent state in the setting of section 5.2.2 All values are normalized by the dimension of the state space. Experiments were produced on 10 independent sequences. The thick solid lines display the mean over the 10 independent replicates for both approaches, shaded lines are single sequences.

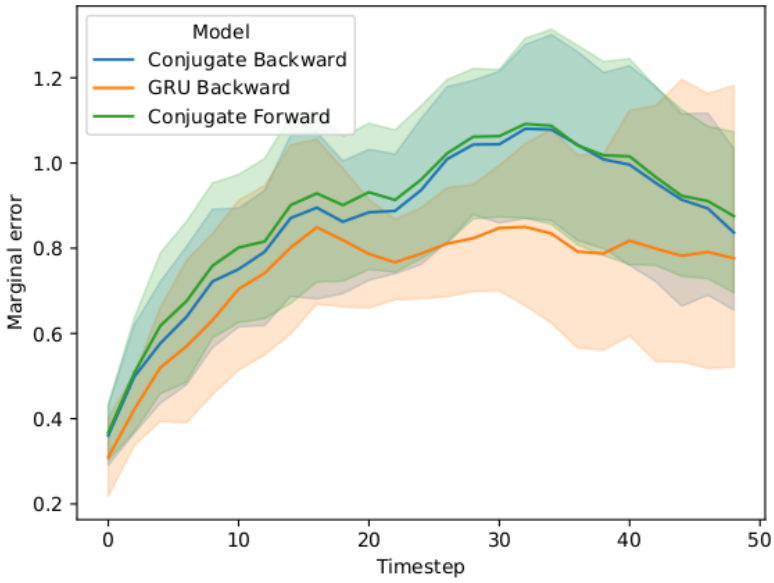


Figure 5.3: Marginal errors $\left(\left|q^{\lambda}_{0:n}h^{m}_{0:n} - \phi^{\theta}_{0:n}h^{m}_{0:n}\right|\right)_{m\leq n}$, i.e. for $\tilde{h}^{m}_{k}(x_k, x_{k+1}) = x_k \mathbb{1}_{k=m}$, in the setting of section 5.2.2 where $\phi^{\theta}_{0:n}$ is obtained by the FFBSi algorithm. All values are normalized by the dimension of the state space. Experiments are produced on 10 independent sequences. The thick solid lines display the mean over the 10 independent replicates for both approaches, the filling is the standard deviation

Nonlinear hidden dynamics with a non-Gaussian observation noise

We now consider a model introduced in [Zha+22], where $m^{\theta}_{k}(x_{k-1}, \cdot)$ is the density of

$$\mathcal{N}\left(x_{k-1} + \delta\left[\gamma W \tanh(x_{k-1}) - x_{k-1}\right]/\tau, Q\right)$$

and g^{θ}_{k} is the density of a Student-t distribution with mean x_k, ν degrees of freedom and scale R. We start by reproducing this *chaotic recurrent neural network* setting as in [Cam+21], Section 5.2. That is, we fit the parameter λ on a given sequence $y_{0:n}$ and we evaluate the performance on the same sequence. To assess the variability of the performance, we train and evaluate on $J = 50$ sequences $(y^{(j)}_{0:n})_{1\leq j\leq J}$, each drawn from a different model with parameter $\theta^{(j)}$, on which we learn a different variational parameter $\lambda^{(j)}$. In Figure 5.5, we plot the evolution of the error with $d = 5$ and $n = 500$ for both the Conjugate Forward and Conjugate Backward models together with the state-of-the-art online backward smoother of [Cam+21]. Once again, all models show a linear dependency on the observations, which supports our main theoretical claim. In Figure 5.4, we provide a more thorough analysis of the additive smoothing performance on other moments for the *Conjugate Backward* model by generating more sequences under a single θ and training for more epochs. Again, in this case, the estimates obtain using the FFBSi considered are considered as ground truth. For all moments, we observe the linearity of the additive smoothing error and the uniform bound on the marginal error. We also observe the dependency of $||h_k||_\infty$ through the increased slopes and higher error values for the additive and marginal errors, respectively.

This experiment also highlights an interesting aspect on the impact of the parameteriza-

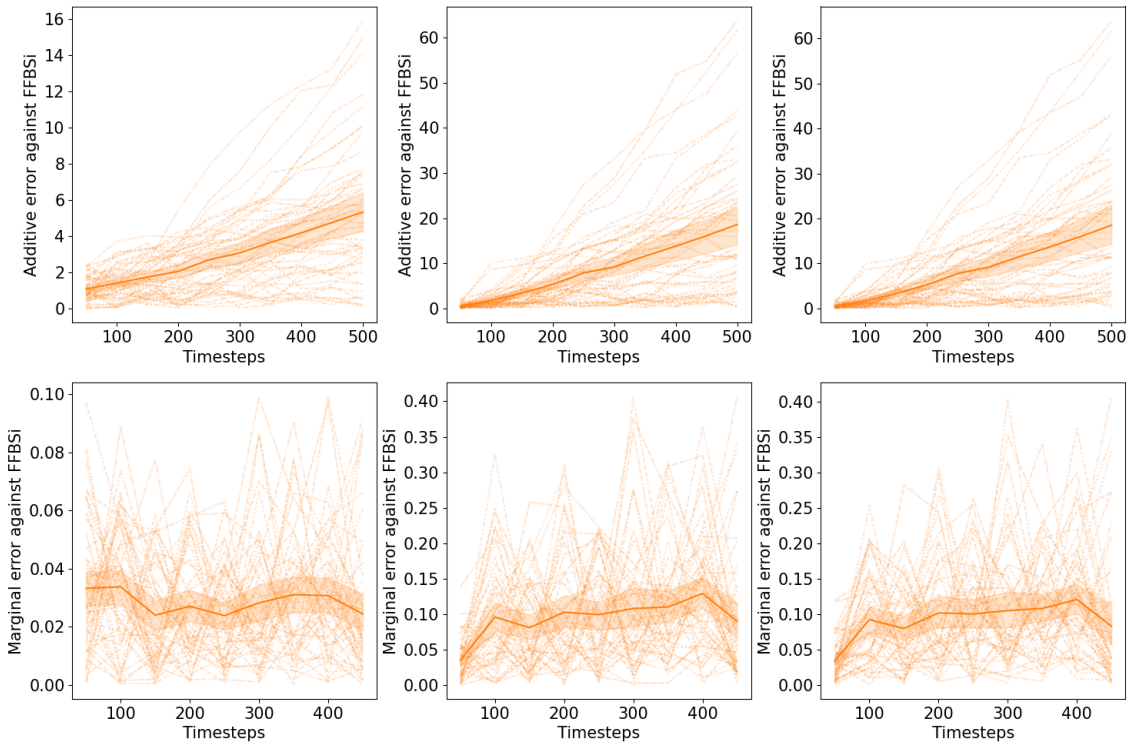


Figure 5.4: Additive (top) and marginal (bottom) errors against FFBSi estimates on the chaotic data with the *Conjugate Backward* model on three types of functionals, from left to right: (i) $\tilde{h}_k(x_{k-1}, x_k) = ||x_k||_1$ (ii) $\tilde{h}_k(x_{k-1}, x_k) = x_k^T x_k$ (iii) $\tilde{h}_k(x_{k-1}, x_k) = x_{k-1}^T x_k$.

tion choices. In the previous sections, training was performed on multiple sequences of fixed length, therefore multiple learning signals are available to learn the terminal distribution q_n^λ (i.e. terminal observations of the sequences in the training set). In the setting of this section, on the contrary, only one data point is available at n. For the offline setting, we therefore do not expect the distributions q_k^λ to be good terminal laws of the subsequences $(y_{0:k})_{k<n}$ under (2.28). Indeed, except for $k = n$, the parameters of these distributions only appear indirectly during optimization (via their relationship with the backward kernels) when optimization of the joint ELBO is performed at a fixed length n. In contrast, the solution of [Cam+21] explicitly performs gradient-descent on a new set of parameters λ_k at each timestep such that $q_k^\lambda = q_k^{\lambda_k}$ is always a good terminal law for $y_{0:k}$. Interestingly, the results for the *Conjugate Forward* and *Conjugate Backward* models - which do not have such regularisation - are only slightly worse than the state-of-the-art, albeit at a much lighter computational cost. Indeed, in practice, Figure 5.5 is obtained simply by using the distributions q_k^λ as terminal laws for $k \leq n$. This suggests that the associated parameterizations may provide good variational *filtering* distributions through the laws q_k^λ as a byproduct of the smoothing objective $q_{0:n}^\lambda$ with no additional regularisation. In section 5.3.3, we discuss more extensively the link between our theoretical results and the choice of parameterizations for the variational kernels.

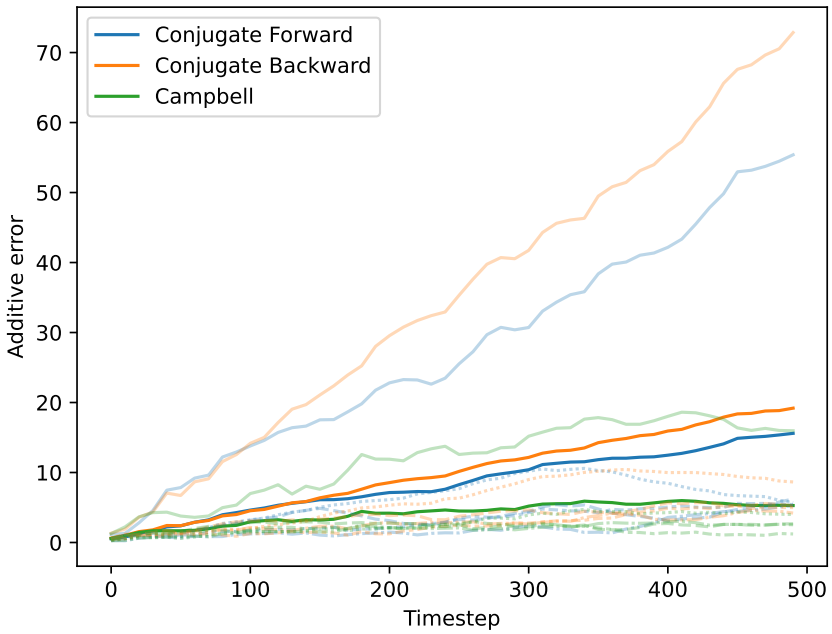


Figure 5.5: $\left|q_{0:n}^\lambda h_{0:n} - \phi_{0:n}^\theta h_{0:n}\right|$ for $\tilde{h}_k(x_k, x_{k+1}) = x_k$ in the setting of section 5.2.2. The solid lines display the mean over the 5 independent replicates which are shown in shaded lines.

On the contrary, the *GRU Backward* model has a different behaviour. In Figure 5.6, the dotted blue curve shows that a good approximation of $q_{0:n}^\lambda$ is obtained by fitting on $y_{0:n}$, however the associated parameter λ does not provide a good approximation of $(q_{0:k}^\lambda)_{k<n}$. If we instead learn λ by computing the gradient of the ELBO for increasingly large subsequences $(y_{0:k})_{k\leq n}$ - i.e. mimicking the training scheme of [Cam+21] - we obtain a different type of approximation, which is suitable for $k < n$, even though this additional constraint results in slightly worse

performance for $k = n$. In this case, the results are comparable with those of [Cam+21].

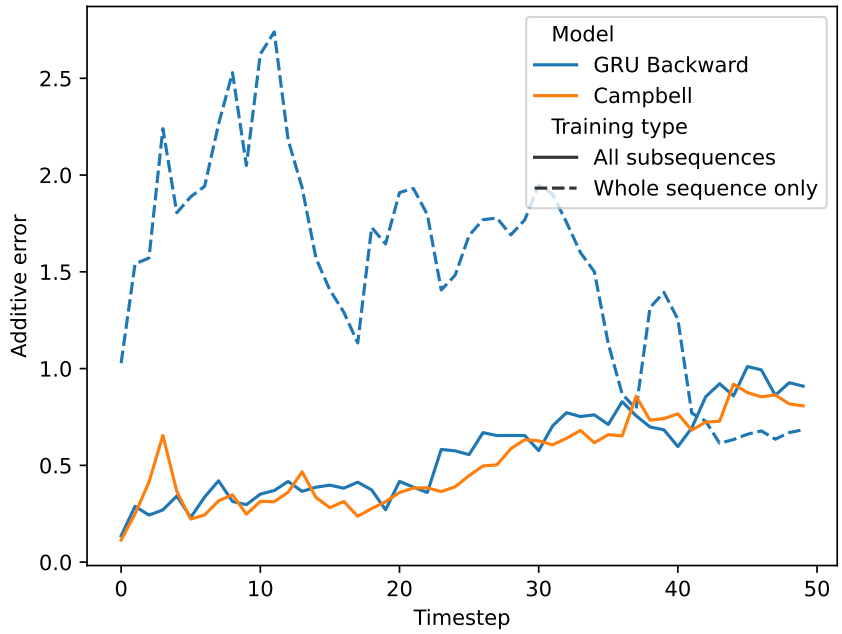


Figure 5.6: $\left|q^{\lambda}_{0:n} h_{0:n} - \phi^{\theta}_{0:n} h_{0:n}\right|$ for $\tilde{h}_k(x_k, x_{k+1}) = x_k$ when training the *GRU Backward* model in two different ways, alongside the solution of [Cam+21]

5.3 Discussion

We have provided the first bound on the additive smoothing error in the context of sequential variational inference using a backward factorization. We have empirically presented cases to illustrate these results. We have also shown that some existing ideas from litterature on message passing or conjugate graphical models can be reframed to be used under the backward factorization. We believe that our theoretical result sheds light on important properties of sequential variational methods and provides perspectives for future research which we detail in this section.

5.3.1 Assumptions

The proposed strong mixing assumptions are classical to obtain theoretical guarantees in non-linear smoothing problems. Weaker assumptions have been proposed in the literature to control filtering distributions. Although these results cannot be extended to smoothing distributions easily, obtaining similar upper bounds as in our contribution with weaker assumptions is an interesting perspective for future works. Our numerical experiments do not restrict to models satisfying these assumptions, suggesting that some relaxations of these classical hypothesis should be investigated.

5.3.2 Additional theoretical guarantees

- Recently, [TY21] proposed a general theoretical framework for analyzing the excess risk associated with empirical Bayes variational Auto Encoders, covering both parametric and nonparametric cases. The authors study the statistical properties of the VAE estimator using M-estimation theory. In our context of time series, extending the M-estimation theory requires to first analyze the asymptotic behavior of the ELBO. We believe this is another appealing property of the backward decomposition of the variational family, as in this case thee ELBO writes

$$\frac{1}{n}\mathcal{L}_n(\theta,\varphi) = \frac{1}{n}\ell_n(\theta) + \frac{1}{n}\mathbb{E}_{q^\lambda_{0:n}}\left[\log\frac{\phi_{\theta,n}(X_n)}{q^\lambda_n(X_n)}\right] + \frac{1}{n}\sum_{s=1}^{n}\mathbb{E}_{q^\lambda_{0:n}}\left[\log\frac{b_{\theta,s-1|s}(X_{s-1},X_s)}{q^\lambda_{s-1|s}(X_{s-1},X_s)}\right],$$

  where $\phi_{\theta,n}$ is the filtering distribution at time n, $(b_{\theta,s-1|s})_{1\le s\le n}$ are the backward kernels of the true model and $\ell_n(\theta)$ is the loglikelihood of the observations. Using this decomposition and additional assumptions, the limiting behavior of the ELBO can be derived to extend the results of [TY21] to state-space models. However, this requires to obtain the asymptotic behavior of various terms which relies on many technicalities and this is therefore left for future work.

  In addition, the backward factorization offers a suitable framework (combined with strong mixing assumptions and regularity conditions on the state-space model) to satisfy Condition A of [TY21]. In an offline learning setting, with fixed n, this provides an interesting perspective to control the total variation distance between the true distribution of the observations and

$$y_{0:n} \mapsto \int\left(\frac{1}{N}\sum_{i=1}^{N} q^\lambda_{0:n}(x_{0:n})\right)\prod_{k=1}^{n} g^\theta_k(x_k,y_k)\mathrm{d}x_{0:n}\,,$$

  where $y^i_{0:n}$, $1\le i\le N$, are i.i.d. sequences with distribution parameterized by θ. These extensions are the focus on the ongoing work [GL23].

- The linear growth with the number of observations matches the results obtained when the true smoothing distributions are replaced by "skewed" or Monte Carlo estimators. Indeed, using for instance [GLO22, Theorem 4.10], we can show that even if the smoothing expectation is computed under the true model but not with the true parameter, the estimation error of the smoothing expectation grows linearly in the number of observations:

$$\left|\phi^{\theta'}_{0:n}h_{0:n} - \phi^{\theta}_{0:n}h_{0:n}\right| \le c(\theta',\theta)n\,.$$

  Therefore, even if the variational family contains the true model, if the minimization of the ELBO does not recover the true parameter, we recover the upper bound linear in the number of observations.

- Obtaining lower bounds for the estimation error of joint smoothing expectation is an open problem, especially in a variational inference framework. This is also an open problem in variational inference for state-space models. We believe that it also relies on important theoretical results which have not been developed yet for the analysis of variational inference of state space models.

5.3.3 Variational kernels parameterization

We do not provide *constructive* assumptions on the variational model, i.e. further works may provide more explicitly the form of the optimal variational factors when the variational kernels belong to a parametric family. Obtaining specific conditions on the variational kernels to optimize the upper bound in Proposition 3 is also an open problem. This leaves a lot of room for implementation choices, even when restricted to the backward factorization. As we did however explore several implementations, we now discuss qualitatively their possible impact on performance and the link with our theoretical results.

Amortization In Section 5.1, we deliberately do not specify explicitly what λ is. In the offline setting with sequences of fixed length n, our results hold in these two cases.

- $\lambda = (\lambda_0, \dots, \lambda_n)$ is directly the set of all parameters of the kernels, where λ_k denotes the parameters of the variational terms involved at k (e.g the parameters for the k-th backward kernel, and for $k = n$, the parameters of the terminal distribution q_n^λ). This corresponds to *non-amortized* inference.
- λ is the global (temporally-shared) parameter of a function f^λ which itself outputs the (local) parameters of the variational kernels from observations, i.e. $f^\lambda(y_{0:n}) = (\lambda_1, \dots, \lambda_n)$. This is usually referred to as *amortized* inference.

One example of non-amortized setting is the implementation of [Cam+21], while both the *Conjugate Backward, Conjugate Forward, GRU Backward* are amortized implementations. While experiments all show the linear behaviour of the additive error, some elements may be discussed with respect to the assumptions involved in the theoretical results. In particular, in Proposition 3, the sharpness of the bound on the additive error is linear in ε, where ε is an upper bound for the error between the variational kernels and their counterparts under the true model. As such, minimizing these local distances with a small ε is key to obtaining a low additive error. In the non-amortized scenario, the parameters of the kernels can be individually tuned during minimization of the joint ELBO and independently of each other. Intuitively, this leaves the highest flexibility to minimize local distances $\|q_n^{\lambda_n} - \phi_n^\theta\|_{\mathrm{tv}}$ and $\|q_{k-1|k}^{\lambda_k}(x_k, \cdot) - b_{k-1|k}^\theta(x_k, \cdot)\|_{\mathrm{tv}}$ for all $k \leq n$, $x_k \in \mathbb{R}^d$, under chosen parameteric families for these kernels. One perspective of this work that remains is to analyse quantitatively how these two types of implementation differ in terms of the local distances recalled above, which is not direct since such distances are not readily available explicitely.

Recursions for parameters of the variational kernels Under the true model, recursions exist that relate the filtering distributions and the backward kernels explicitly, and approximating these recursions is at the core of sequential Bayesian inference algorithms. One question that remains in our study of backward variational methods is whether reproducing similar recursions to build the variational kernels leads to better practical solutions. Again, our results hold irrespective of the dependencies between the parameters of the variational kernels, but experimentally we explored many scenarios. In that respect, experiments of Section 5.2.2 are somehow informative. Indeed, we observe, for example, that the *Conjugate Backward* exhibits the linear additive behaviour for any $k \leq n$ when using the $(q_k^\lambda)_{k \leq n}$ to build the terminal distributions, even when trained on a sequence of fixed length n. Contrarily, the *GRU Backward*

does not. In the former implementation, denoting $\psi_k^\lambda : x_k \mapsto e^{\left\langle r^\lambda(y_k), t_{\mathcal{N}}(x_k)\right\rangle}$, one has, for all $k \leq n$, $q_k^\lambda \propto \psi_k^\lambda \int m_k^\lambda q_{k-1}^\lambda$ and $q_{k-1|k}^\lambda \propto m_k^\lambda q_{k-1}^\lambda$, which is similar to the true model where $\phi_k^\theta(\cdot) \propto g_k^\theta(\cdot) \int m_k^\theta(x_{k-1}, \cdot)\phi_{k-1}^\theta(\mathrm{d}x_{k-1})$ and $b_{k-1|k}^\theta(x_k, \cdot) \propto \phi_{k-1}^\theta(\cdot)m_k^\theta(\cdot, x_k)$. On the contrary, for the *GRU Backward*, no such link can be made.

This discussion is tightly linked to the practical existence and meaning of distributions q_k^λ for $k < n$ in the offline setting that we studied. Indeed, the theoretical study only prescribes implementing explicitly a term for $k = n$. The proof of Proposition 3 suggests that when this terminal distribution q_n^λ is the last term of a sequence $(q_k^\lambda)_{k\leq n}$ where $q_k^\lambda = \phi_k^\theta$ for $k < n$, then it only remains to have the variational backward kernels closest to the true ones to reduce the additive smoothing error. However it is unclear whether this is the optimal scenario in the sense of Proposition 1, i.e. the discrepancies $\tilde{c}_k$ may be lower for some sequence $(\tilde{q}_k)_{k\leq n}$ which is not an approximation of the sequence of true filtering distributions, and an implementation of this optimum might not yield - as is the case for some of our models - good approximations of the latter as a byproduct.

Chapter 6

Conclusion and perspectives

In Chapter 3, a very pragmatic take on video object counting has been proposed via a robust dynamical model combined with an association stage that takes into account uncertainty in both the motion estimates and the detections. Then, we took a step back and studied some methodological aspects behind the promising backward decomposition in sequential variational inference, developing in Chapter 4 a simplified and efficient algorithm to deploy it in recursive settings, and bringing in Chapter 5 new theoretical guarantees that further justify its use as a principled approximation method for high dimensional sequential data. In this chapter, we conclude by presenting what we view as the most promising directions of research from these methodological works in sequential variational inference. In Section 6.1, we discuss some remaining topics that could be explored to strengthen the foundations of backward sequential variational inference. In Section 6.2, we discuss how properly specified dependencies and theoretical guarantees of backward SVI methods could play a key role in novel representation learning solutions when applied to real-world sequential data, in particular regarding the original motivations of this thesis with video object counting.

6.1 Further research in backward SVI

In the context of SVI, we have derived theoretical guidelines that apply to *any* SVI method that respects the backward factorization, and the online algorithm we propose only requires an unconstrained decomposition of the variational backward kernels. As such, our propositions are rather general. As hinted in Section 5.3.3, however, many important questions remain to further justify theoretically some precise implementations in the backward SVI methodology.

6.1.1 Relating the variational factors to the generative model

Since the backward variational decomposition is directly inspired by the Markov factorization of the true smoothing distributions, an important question that remains is whether all implementations (e.g. amortized, non-amortized, etc) and optimization schemes (e.g. online / offline, etc) lead to variational solutions whose individual factors coincide with the true ones. As optimization is performed in the joint space, it is not clear, for example, whether minimization of the ELBO leads to variational parameters for which the variational backward kernel $q^{\lambda}_{t-1|t}$ is a good approximation of the true backward kernel $B^{\theta}_{t-1|t}$ for any $t \geq 1$. Additionally, while in the online setting, minimization of $\mathcal{L}^{\lambda}_{t}$ at all $t \geq 0$ will necesarily enforce the suc-

cessive marginals of $q^\lambda_{0:t}$ to approximate ϕ^θ_t, it is not clear how to derive sequences $(q^\lambda_s)_{s\leq t}$ of distributions such that $q^\lambda_s \approx \phi^\theta_s$ for all $s \leq t$ in the *offline* setting. Additionally, recent works like [DZP23] (which originally only consider filtering objectives and do not target smoothing distributions) raise many questions regarding the possibility of building variational models which explicitly make use of the true transition kernels. Indeed, to derive simple online algorithms that yield sequences $(q^\lambda_t)_{t\geq 0}$ of variational filtering approximations, they introduce "hybrid" versions of the predictive distributions $(\bar{\phi}^\theta_t)_{t\geq 0}$ at each timestep in the form of

$$\bar{q}^{\lambda,\theta}_t = \mathbb{E}_{q^\lambda_{t-1}}\left[m^\theta_t(X_{t-1}, \cdot)\right],$$

which is simply obtained by plugging q^λ_{t-1} in place of ϕ^θ_{t-1} in the definition of the true predictive distribution. As such, they explicitly rely on the true dynamics of the generative model $(m^\theta_t)_{t\geq 1}$ to propagate the variational distributions. As mentionned in Section 2.3.2, filtering objectives derived in this manner are not guaranteed to be stable from an optimization point of view, however, one may imagine variational *backward* kernels defined as

$$q^{\lambda,\theta}_{t-1|t}(x_t, x_{t-1}) \propto q^\lambda_{t-1}(x_{t-1}) m^\theta_t(x_{t-1}, x_t),$$

in which case the normalizing constant is precisely $\bar{q}^{\lambda,\theta}_t(x_t)$ as defined above. In this case, updates of the expectations $(H^\lambda_t)_{t\geq 0}$ associated with the joint ELBO would become

$$H^\lambda_t(x_t) = \mathbb{E}_{\bar{q}^{\lambda,\theta}_{t-1|t}}\left[H^\lambda_{t-1}(X_{t-1}) + \frac{g^\theta_t(x_t)\bar{q}^{\lambda,\theta}_t(x_t)}{q^\lambda_{t-1}(X_{t-1})}\right].$$

Since such definition of the backward kernels still falls in the decomposition (4.5), approximating these recursively would still be possible with the algorithms we proposed. However, with θ known, this methodology would only require parameterizing the flow of distributions $(q^\lambda_t)_{t\geq 0}$, hence removing the need to manually define potentials $(\psi^\lambda_t)_{t\geq 0}$.

6.1.2 Exploring larger variational objectives and families

Then many other directions could be explored to include the backward methodology into more elaborate variational formulations. Indeed, in this work we have only considered the most popular divergence minimization problems related to the reverse KL and the ELBO that derives from it, yet many works have been suggested in VI to derive more expressive variational approximations.

A first observation often made is that the reverse KL typically leads to solutions that underestimate the variance of the targeted distribution by concentrating on its modes. This behaviour, sometimes referred to as the *mode-seeking* property of $\overleftarrow{\mathbb{D}}^\pi_{\mathsf{KL}}$, is opposed the so-called *mass-coverage* behaviour of the *forward* KL defined as $\overrightarrow{\mathbb{D}}^\pi_{\mathsf{KL}} : q \mapsto \mathbb{D}_{\mathsf{KL}}(\pi, q)$. Figure 6.1, illustrates these differences, showing typical solutions under a Gaussian family for a unidimensional target distribution π whose mass is increaglingly spread out into several modes.

Observing this, some works like [NLB20; ZBN22; Kim+22b] focus on building new scalable methods to perform VI using forward KL despite the untractable expectation under the target distribution π. A typical consequence of approaching the minimization problem with $\overrightarrow{\mathbb{D}}^\pi_{\mathsf{KL}}$ is that intricate sampling schemes need to be reintroduced in order to sample from π, e.g. MCMC and Hamiltonian Monte Carlo [Nea+11; HG+14]. In the same vein, some works tackle

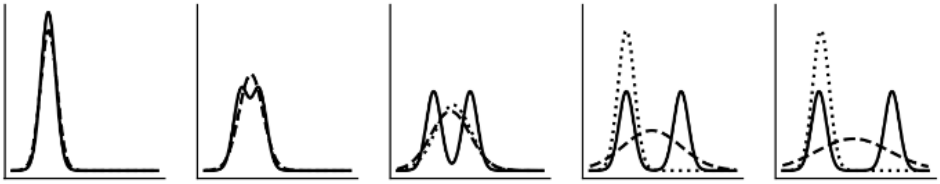


Figure 6.1: Full line: target distribution π, increasingly multimodal from left to right. Dotted line: $\overrightarrow{\mathbb{D}}^{\pi}_{\mathsf{KL}}(q)$. Dashed line: $\overleftarrow{\mathbb{D}}^{\pi}_{\mathsf{KL}}(q)$.

the underestimation of variance by using other divergences, e.g. the Chi divergence [Die+17], or by building optimization schemes specifically for minimization of classes of distances like α-divergences [LT16] or f-divergences [DDP21; DDR23], which are generalizations of $\mathbb{D}_{\mathsf{KL}}$.

In parallel to these adjustements, another class of research focuses on non-parametric families of variational distributions, both to improve flexibility of the solutions in recovering π and to avoid having to choose specific family of distributions based on computational aspects alone. A popular example in recent years [LW16] where the variational distributions are interacting systems of uniformly weighted particles optimized by considering gradients defined directly in $\mathcal{P}(\mathsf{X})$. Such approaches have recently attracted wide interest, notably by recasting sampling algorithms as gradient flows of some underlying distance in probability spaces [KLJ23; Kor+21], which notably introduces strong links between Monte Carlo algorithms and variational inference.

In fact, most of our work attempts to bridge the gap between variational methods and known decompositions or approximations schemes in SSM literature (e.g. SMC), but it is still unclear how far the links between the two literatures go. In that respect, many works [GGT15; NLB20; Nae+18; Zha+22] in SMC propose solutions that somewhat unify the two approaches by viewing VI and minimization of $\mathbb{D}_{\mathsf{KL}}$ over the joint space as a principled methodology to learn sampling proposals which are directly tuned for the smoothing problem. As such, they are conceptually closer to the field of *adaptive SMC* [CMO08], or in the context of smoothing, to elaborate methods that learn globally optimal proposals at each timestep [Hen+17; Law+18], see also [NLS+19, chapter 3] for a general introduction. While such works are insightful to introduce the powerful aspects of amortized inference into SMC, in most cases the discussion on SVI implementations are largely directed by evaluating the quality of the resulting particle approximations that they enable. As such, they do not directly provide theoretical insight for the methods described in this thesis, which aim at replacing particle approximtions altogether. Still, an interesting perspective would be to evaluate the relevance of backward SVI approaches when used as proposals for SMC to mitigate the curse of dimensionality.

6.2 A unifying framework for sequence-wise prediction tasks in videos

As mentioned in Section 1.2.2, a strong motivation behind the study of novel solutions for high-dimensional latent estimation is the recent surge of unsupervised approaches which focus on learning expressive representations of the data from which most predictions tasks can be easily derived. In most of these works, it is generally assumed that latent variables from which observations originate factorize into individual components that are statistically independent,

and that capturing this so-called "disentangled" property of latent representations [Hig+18] is key to facilitate downstream applications. In Figure 6.2, for example, we provide a visual illustration of some powerful aspects of these latents in the context of multi-object discovery.

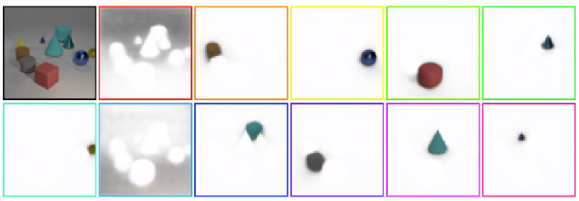


Figure 6.2: Some results from [Kip+22]. An image contains multiple objects, and the goal is to retrieve a latent representation where each component accounts for one of them only. Top-left: reconstruction of an original image from all coordinates of the predicted latent. Other images: reconstruction by isolating a single component of the predicted latent. The role of the illustration is to demonstrate that individual coordinates of the posterior effectively correspond to different objects in the image.

However, up to now, most works have required to enforce the desired structure of the recovered latents, i.e. by explictly introducing additional constraints in the definition of the posteriors [Loc+20; Gre+19; Kab+21; Els+22; SWA22], adding regularisation terms in the MLE objectives [Hig+17], or relying on auxiliary observed data [Kip+22] which further constrains the inference problems. In parallel, the problem of retrieving statistically independent sources of variation from the data has been largely formalized from the point of view of independent component analysis (ICA) [HO00]. Here, a recurring topic is to determine in which settings a set of independent latent variables can be recovered *uniquely* without supervision in the limit of infinite data. In that regard, theoretical results [HP98] and recent empirical studies [Loc+19] essentially characterize latent estimation in *nonlinear* data models as ill-posed in the context of independent observations, which largely hinders representation learning for realistic generative models (e.g. containing complex mappings such as DNNs) when only datasets of separate images are available. Conversely, newer results [GLL20; Khe+20] prove identifiability properties in the context of *dependant* data, and in particular temporal data (which includes videos). From there, recent analyses [HKM23] suggest that recovering independent latents in high-dimensional data can be achieved without any supervision, *provided that the statistical dependencies of the posterior approximations are well specified.*

In practice, most works that have attempted to tackle structured data based on these results [HH20; Häl+21a] have heavily relied on sequential variational approximations, but mostly via decompositions similar to those in Section 2.3.2, which lack the theoretical guarantees developed in this thesis and can hardly be used for long sequences. Consequently, an important perspective of this thesis would be to derive similar solutions using backward decompositions instead. In the context of video object counting, or more generally when targeting global quantities related to entire sequences of observations $y_{0:t}$, the identifiability properties of the sequential setting are especially appealing, because they suggest that formulating prediction in videos as statistical estimates under the smoothing distributions could be a theoretically justified approach to avoid relying on specialized intermediate predictions (such as the point estimates of MOT). For example, given the reliability of backward SVI approximations for additive smoothing, one may imagine a streamlined counting solution which estimates an

object count $\hat{N}$ in a video via

$$\hat{N} = \mathbb{E}_{q^{\lambda}_{0:t}}\left[h_{0:t}(X_{0:t})\right] \approx \mathbb{E}_{\phi^{\theta}_{0:t}}\left[h_{0:t}(X_{0:t})\right] \ ,$$

for some additive state functional $h_{0:t}$ that can extract relevant count information from the recovered latents. Relying on such formalism would be very appealing because it essentially separates the problem into two distinct steps:

1. A generic learning stage to build $q^{\lambda}_{0:t}$ solely from ELBO optimization, where dependencies in the data are captured without any annotation and irrespective of the targeted task.

2. The specification of $h_{0:t}$ depending on the task at hand, and the computation of $q^{\lambda}_{0:t}h_{0:t}$ with λ fixed.

In practice, such framework has several advantages. First, compared to e.g. fully supervised MOT methods which require additional tracking annotations to introduce temporal information in the learning process, the first step is fully unsupervised and only requires correct specification of the dependencies in the posterior. Then, and most importantly, the separation of the representation learning stage from the final prediction stage may allow to rely on much weaker annotations than in common supervised learning settings. Indeed, just as location information is discarded in detection-based object counting in still images, the predicted object locations in MOT-based video object counting are used for temporal association between detections across frames but discarded in the final count. In the previous setting, the temporal coherency of the predictions is expected to be already enforced given that predictions are formulated as expectations under $q^{\lambda}_{0:t}$, and one may therefore derive a "counting" functional $h_{0:t}$ which extracts counts from the smoothed latents, given *only* count supervision.
As an example, suppose that we can annotate, for all timesteps $s \leq t$ in a video $y_{0:t}$, the number of objects N^{+}_{s} appearing in the video at s, but not present for $s' < s$. Then, assuming that an object cannot be visible again after it has left the camera field (e.g. the camera does not move backwards), the total object count in the video is simply given by $N = \sum_{s=0}^{t} N^{+}_{s}$. In practice, such annotations are easy to obtain, because they only require watching the video and marking frames with entering objects (all other frames receive $N^{+}_{s} = 0$), which is considerably less involved than annotating all object locations at all frames. Given this, a counting algorithm may be developed by defining functionals whose components are mappings aimed at estimating $(N^{+}_{s})_{s\leq t}$ from the pairwise latent representations of consecutive frames, given the entire video. Formally, one may define

$$h^{\gamma}_{0:t} : x_{0:t} \mapsto \sum_{s=1}^{t} \tilde{h}^{\gamma}(x_{s-1}, x_s) \ ,$$

where $\tilde{h}^{\gamma}$ is a DNN from $\mathsf{X} \times \mathsf{X}$ to $\mathbb{N}$ parameterized by $\gamma \in \Gamma$, where Γ is a parameter space. To learn γ, one may derive a count penalty function $\mathcal{C}^{\gamma} : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$ (e.g. Poisson regression), and consider the global penalty $\sum_{s=0}^{t} \mathcal{C}^{\gamma}(\hat{N}^{+}_{s}, N^{+}_{s})$ on the video, where

$$\hat{N}^{+}_{s} = \mathbb{E}_{q^{\lambda}_{s-1:s}}[\tilde{h}^{\gamma}(X_{s-1}, X_s)] \ ,$$

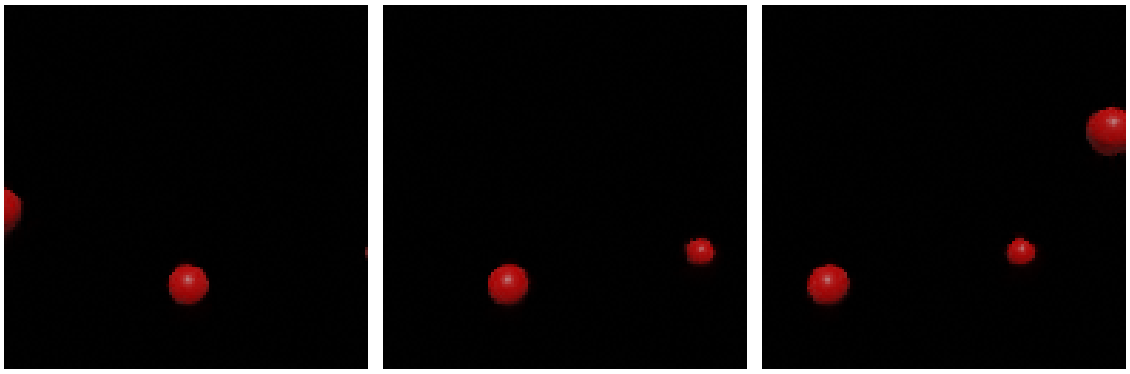
for all $s \leq t$. [1] Upon convergence, an estimate of the global count N would be given by $\hat{N} = \mathbb{E}_{q^{\lambda}_{0:t}}[h^{\gamma}_{0:t}]$. As such, this methodology is very appealing because it does not require annotation of object locations, includes knowledge from *all* frames through $q^{\lambda}_{0:t}$, but still provides a learning signal at individual timesteps (and not only a global count). Regarding this latter point, many other options would be possible to strengthen the "local" supervision, i.e. by defining components which predict the (possibly negative) variation in number of visible objects between $s-1$ and s, or simply predicting the total number of objects N_s visible at any timestep $s \leq t$ and considering $N = \sum_{s=1}^{t} \max(0, N_s - N_{s-1})$ as an alternative definition of the global count[2]. Conversely, the flexility of this setting would also allow, after this first training stage, to finetune γ given global count supervision, i.e. directly using $\mathcal{C}^{\gamma}(\hat{N}, N)$ with $\hat{N} = \mathbb{E}_{q^{\lambda}_{0:t}}[h^{\gamma}_{0:t}]$. Finally, one may also consider retraining λ with γ fixed, using the count penalty to finetune the latent representation for best count performance given a count functional. Conveniently, using Chapter 4, all of these operations could be performed online.

All in all, the expected advantages of this new methodology provide clear answers to the limitations of the work conducted in Chapter 3. Indeed, in the latter contribution, the main axis of improvement was to increase the performance of the object detector via larger datasets of individual images, which requires continuous effort from Surfrider Foundation. Furthermore, as mentioned in introduction, developing a more stable counting solution in the framework of MOT would have required more sophisticated tracking mechanisms, with many aspects not directly related to the final counting task, and with additional steps of hyperparameter tuning. Comparatively, the research conducted in Chapters 4 and 5 paves the way to the development of more streamlined counting solutions. Additionally, they would also be arguably easier to supplement with uncertainty estimates, e.g. by considering confidence intervals based on the variance of the posterior $q^{\lambda}_{0:t}$. While not part of this manuscrit, experiments on synthetic videos of moving objects (see Figure 6.3) are underway to evaluate the relevance of these ideas on real image-based content where clean ground truth data can be obtained, and varying degrees of complexity can be generated.

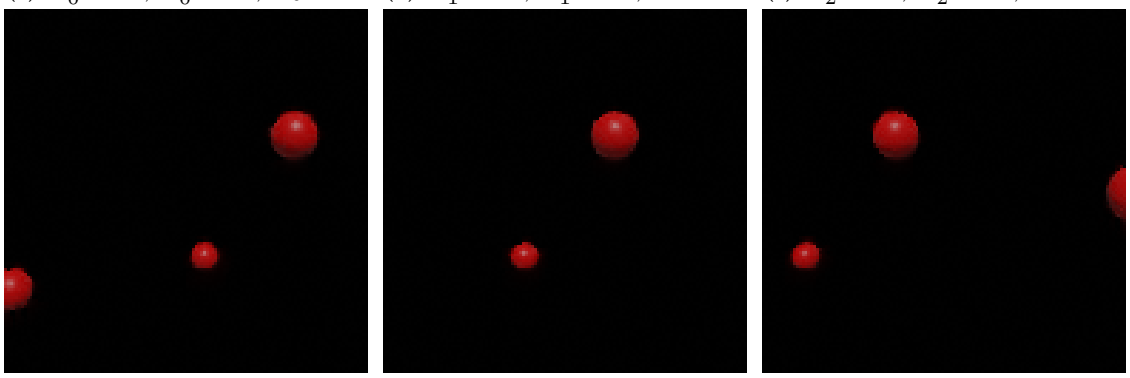
As a final note, because the learning of $q^{\lambda}_{0:t}$ is agnostic to the choice of functional used for the final prediction, the previous framework is highly modular: one may for example learn $q^{\lambda}_{0:t}$ as a prior stage, then perform various predictions using different functionals, keeping the same λ. To illustrate the potential relevance of this aspect, we can consider some minor portion of the observed data of the Plastic Origins project, which exposed the limitations of framing macrolitter pollution monitoring simply as a counting task. For example, situations such as illustrated in Figure 6.4 suggested that approaches which could provide additional forms of predictions (e.g. the surface of the river bank covered by litter) depending on the situations present would be an interesting perspective of research.

[1] Recalling that $q^{\lambda}_{s-1:s}$ is the joint marginal distribution of $q^{\lambda}_{0:t}$ at $s-1$ and s.

[2] Actually, the number of objects visible in any frames can be obtained simply by annotating, on top of N^{+}_{s}, the number of objects N^{-}_{s} leaving the video at all s.



(a) $N_0^+ = 2, N_0^- = 0, N_0 = 2$ (b) $N_1^+ = 1, N_1^- = 1, N_1 = 2$ (c) $N_2^+ = 1, N_2^- = 0, N_2 = 3$



(d) $N_3^+ = 0, N_3^- = 0, N_3 = 3$ (e) $N_4^+ = 0, N_4^- = 1, N_4 = 2$ (f) $N_5^+ = 1, N_5^- = 0, N_5 = 3$

Figure 6.3: An example of synthetic video with objects entering and leaving the camera frame with varying speeds.

Figure 6.4: One example of high-density litter accumulation. Here, individually enumerating objects seems unadapted to measure the level of pollution, e.g. one may rather estimate the surface of the bank that is covered by plastic.

References

[AA22] Jimmy Olsson Alessandro Mastrototaro and Johan Alenlöv. "Fast and Numerically Stable Particle-Based Online Additive Smoothing: The AdaSmooth Algorithm". In: *Journal of the American Statistical Association* 0.0 (2022), pp. 1–12.

[AD03] Christophe Andrieu and Arnaud Doucet. "Online expectation-maximization type algorithms for parameter estimation in general state space models". In: *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).* Vol. 6. IEEE. 2003, pp. VI–69.

[Aga+17] Sergios Agapiou, Omiros Papaspiliopoulos, Daniel Sanz-Alonso, and Andrew M Stuart. "Importance sampling: Intrinsic dimension and computational cost". In: *Statistical Science* (2017), pp. 405–431.

[ALZ16] Carlos Arteta, Victor Lempitsky, and Andrew Zisserman. "Counting in the Wild". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, 2016, pp. 483–498. ISBN: 978-3-319-46478-7.

[And+16] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. "Learning to learn by gradient descent by gradient descent". In: *Advances in neural information processing systems* 29 (2016).

[Arc+15] Evan Archer, Il Memming Park, Lars Buesing, John Cunningham, and Liam Paninski. "Black box variational inference for state space models". In: *arXiv preprint arXiv:1511.07367* (2015).

[Arn+21] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. "Vivit: A video vision transformer". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 6836–6846.

[Bak+19] Steve Bako, Mark Meyer, Tony DeRose, and Pradeep Sen. "Offline Deep Importance Sampling for Monte Carlo Path Tracing". In: *Computer Graphics Forum* 38.7 (2019), pp. 527–542.

[Bar12] Jayme Barbedo. "A Review on Methods for Automatic Counting of Objects in Digital Images". In: *Latin America Transactions, IEEE (Revista IEEE America Latina)* 10 (Sept. 2012), pp. 2112–2124.

[Bay+21] Justin Bayer, Maximilian Soelch, Atanas Mirchev, Baris Kayalibay, and Patrick van der Smagt. "Mind the Gap when Conditioning Amortised Inference in Sequential Latent-Variable Models". In: *International Conference on Learning Representations*. 2021.

[BBL08] Thomas Bengtsson, Peter Bickel, and Bo Li. "Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems". In: *Probability and statistics: Essays in honor of David A. Freedman*. Vol. 2. Institute of Mathematical Statistics, 2008, pp. 316–335.

[BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[Bea+16] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. "What's the Point: Semantic Segmentation with Point Supervision". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, 2016, pp. 549–565. ISBN: 978-3-319-46478-7.

[Bew+16] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. "Simple online and realtime tracking". In: *2016 IEEE international conference on image processing (ICIP)*. IEEE. 2016, pp. 3464–3468.

[BGS15] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. "Importance weighted autoencoders". In: *arXiv preprint arXiv:1509.00519* (2015).

[BKM17] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. "Variational Inference: A Review for Statisticians". In: *Journal of the American Statistical Association* 112.518 (2017), pp. 859–877.

[BL20] Guillem Brasó and Laura Leal-Taixé. "Learning a neural solver for multiple object tracking". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 6247–6257.

[BN06] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[Bru+18] Antoine Bruge, Cristina Barreau, Jérémy Carlot, Hélène Collin, Clément Moreno, and Philippe Maison. "Monitoring litter inputs from the Adour river (southwest France) to the marine environment". In: *Journal of Marine Science and Engineering* 6.1 (2018). ISSN: 20771312.

[BS08] Keni Bernardin and Rainer Stiefelhagen. "Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics". In: *EURASIP Journal on Image and Video Processing* 2008 (2008), pp. 1–10.

[BS96] Dimitri Bertsekas and Steven E Shreve. *Stochastic optimal control: the discrete-time case*. Vol. 5. Athena Scientific, 1996.

[CA18] Badr-Eddine Chérief-Abdellatif and Pierre Alquier. "Consistency of variational Bayes inference for estimation and model selection in mixtures". In: (2018).

[Cae+20] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. "nuscenes: A multimodal dataset for autonomous driving". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11621–11631.

[Cam+21] Andrew Campbell, Yuyang Shi, Thomas Rainforth, and Arnaud Doucet. "Online variational filtering and parameter learning". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 18633–18645.

[Car+20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. "End-to-end object detection with transformers". In: *European conference on computer vision*. Springer. 2020, pp. 213–229.

[Cha+] Mathis Chagneux, Pierre Gloaguen, Sylvain Le Corff, and Jimmy Olsson. "A backward sampling approach for online variational additive smoothing". In: *arXiv* ().

[Cha+17] Prithvijit Chattopadhyay, Ramakrishna Vedantam, Ramprasaath R Selvaraju, Dhruv Batra, and Devi Parikh. "Counting Everyday Objects in Everyday Scenes". In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. Vol. 2017-Janua. 2017, pp. 4428–4437. ISBN: 978-1-5386-0457-1.

[Cha+22] Mathis Chagneux, Élisabeth Gassiat, Pierre Gloaguen, and Sylvain Le Corff. "Amortized backward variational inference in nonlinear state-space models". In: *arXiv:2206.00319* (2022).

[Cha+23] Mathis Chagneux, Sylvain Le Corff, Pierre Gloaguen, Charles Ollion, Océane Lepâtre, and Antoine Bruge. "Macrolitter Video Counting on Riverbanks Using State Space Models and Moving Cameras". In: *Computo* (Feb. 16, 2023), undefined. ISSN: 2824-7795.

[Che+20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PMLR. 2020, pp. 1597–1607.

[Che+21] Leiyu Chen, Shaobo Li, Qiang Bai, Jing Yang, Sanlong Jiang, and Yanming Miao. "Review of Image Classification Algorithms Based on Convolutional Neural Networks". In: *Remote Sensing* 13.22 (2021). ISSN: 2072-4292.

[Che19] Badr-Eddine Cherief-Abdellatif. "Consistency of ELBO maximization for model selection". In: *Proceedings of The 1st Symposium on Advances in Approximate Bayesian Inference*. Ed. by Francisco Ruiz, Cheng Zhang, Dawen Liang, and Thang Bui. Vol. 96. Proceedings of Machine Learning Research. PMLR, 2019, pp. 11–31.

[Chu+15] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. "A recurrent latent variable model for sequential data". In: *Advances in neural information processing systems* 28 (2015).

[CL04] Pavel Chigansky and Robert Liptser. "Stability of nonlinear filters in nonmixing case". In: *The Annals of Applied Probability* 14.4 (2004), pp. 2038–2056.

[CLD18] Chris Cremer, Xuechen Li, and David Duvenaud. "Inference suboptimality in variational autoencoders". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1078–1086.

[CMO08] Julien Cornebise, Éric Moulines, and Jimmy Olsson. "Adaptive methods for sequential importance sampling with application to state space models". In: *Statistics and Computing* 18 (2008), pp. 461–480.

[CMR05] O. Cappé, É. Moulines, and T. Rydén. *Inference in Hidden Markov Models*. Springer, 2005.

[Cor+21] Adrien Corenflos, James Thornton, George Deligiannidis, and Arnaud Doucet. "Differentiable particle filtering via entropy-regularized optimal transport". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 2100–2111.

[CP+20] Nicolas Chopin, Omiros Papaspiliopoulos, et al. *An introduction to sequential Monte Carlo*. Springer, 2020.

[Csi84] Imre Csiszár. "Information geometry and alternating minimization procedures". In: *Statistics and Decisions, Dedewicz* 1 (1984), pp. 205–237.

[DC23] Hai-Dang Dau and Nicolas Chopin. *On Backward Smoothing Algorithms*. Mar. 2023.

[DDP21] Kamélia Daudel, Randal Douc, and François Portier. "Infinite-dimensional gradient-based descent for alpha-divergence minimisation". In: *The Annals of Statistics* 49.4 (2021), pp. 2250–2270.

[DDR23] Kamélia Daudel, Randal Douc, and François Roueff. "Monotonic alpha-divergence minimisation for variational inference". In: *Journal of Machine Learning Research* 24.62 (2023), pp. 1–76.

[DDS10a] Pierre Del Moral, Arnaud Doucet, and Sumeetpal Singh. "Forward smoothing using sequential Monte Carlo". In: *arXiv preprint arXiv:1012.5390* (2010).

[DDS10b] Pierre Del Moral, Arnaud Doucet, and Sumeetpal S Singh. "A backward particle interpretation of Feynman-Kac formulae". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 44.5 (2010), pp. 947–975.

[DDS10c] Pierre Del Moral, Arnaud Doucet, and Sumeetpal S. Singh. "A backward particle interpretation of Feynman-Kac formulae". en. In: *ESAIM: Mathematical Modelling and Numerical Analysis* 44.5 (2010), pp. 947–975.

[Del04] Pierre Del Moral. *Feynman-kac formulae*. Springer, 2004.

[Den+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[Dev+19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *ArXiv* abs/1810.04805 (2019).

[DFG13] Arnaud Doucet, Nando de Freitas, and Neil Gordon. *Sequential Monte Carlo methods in practice*. Springer Science & Business Media, 2013.

[DG01] Pierre Del Moral and Alice Guionnet. "On the stability of interacting processes with applications to filtering and genetic algorithms". In: *Annales de l'Institut Henri Poincare (B) Probability and Statistics* 37.2 (2001), pp. 155–194. ISSN: 0246-0203.

[Die+17] Adji Bousso Dieng, Dustin Tran, Rajesh Ranganath, John Paisley, and David Blei. "Variational Inference via Chi Upper Bound Minimization". In: *Advances in Neural Information Processing Systems* 30 (2017).

[DJ09] Arnaud Doucet and Adam Johansen. "A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later". In: *Handbook of Nonlinear Filtering* 12 (Jan. 2009).

[DL13] Cyrille Dubarry and Sylvain Le Corff. "Non-asymptotic deviation inequalities for smoothed additive functionals in nonlinear state-space models". In: *Bernoulli* 19.5B (2013), pp. 2222–2249.

[DLR77] A.P. Dempster, N.M. Laird, and D.B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm (with discussion)". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39 (1977), pp. 1–38.

[DMS14] Randal Douc, Eric Moulines, and David Stoffer. *Nonlinear time series. Theory, methods and applications with R examples*. CRC Press, Jan. 2014. ISBN: 9780429112638.

[Dos+15] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. "FlowNet: Learning Optical Flow with Convolutional Networks". In: *2015 IEEE International Conference on Computer Vision (ICCV)* (2015), pp. 2758–2766.

[Dou+09] Randal Douc, Gersande Fort, Eric Moulines, and Pierre Priouret. "Forgetting the initial distribution for Hidden Markov Models". In: *Stochastic Processes and their Applications* 119.4 (2009), pp. 1235–1256. ISSN: 0304-4149.

[Dou+11] Randal Douc, Aurélien Garivier, Eric Moulines, and Jimmy Olsson. "Sequential Monte Carlo smoothing for general state space hidden Markov models". In: *The Annals of Applied Probability* 21.6 (2011), pp. 2109–2145.

[DZP23] Matthew Dowling, Yuan Zhao, and Il Memming Park. *Real-Time Variational Method for Learning Neural Trajectory and its Dynamics*. 2023.

[Els+22] Gamaleldin Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael C Mozer, and Thomas Kipf. "Savi++: Towards end-to-end object-centric learning from real-world videos". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 28940–28954.

[EM21] Vıéctor Elvira and Luca Martino. "Advances in importance sampling". In: *arXiv preprint arXiv:2102.05407* (2021).

[Emm+19] Tim van Emmerik, Romain Tramoy, Caroline van Calcar, Soline Alligant, Robin Treilles, Bruno Tassin, and Johnny Gasperi. "Seine Plastic Debris Transport Tenfolded During Increased River Discharge". In: *Frontiers in Marine Science* 6.October (2019), pp. 1–7. ISSN: 22967745.

[ES20] Tim van Emmerik and Anna Schwarz. "Plastic debris in rivers". In: *WIREs Water* 7.1 (2020), e1398.

[Far03] Gunnar Farnebäck. "Two-frame motion estimation based on polynomial expansion". In: *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*. Springer. 2003, pp. 363–370.

[Fei20] Christoph Feichtenhofer. "X3d: Expanding architectures for efficient video recognition". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 203–213.

[FPW17] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. "Spatiotemporal multiplier networks for video action recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4768–4777.

[Fra+16] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. "Sequential neural models with stochastic layers". In: *Advances in neural information processing systems* 29 (2016).

[Gas+14] Johnny Gasperi, Rachid Dris, Tiffany Bonin, Vincent Rocher, and Bruno Tassin. "Assessment of floating plastic debris in surface water along the Seine River". In: *Environmental Pollution* 195 (2014), pp. 163–166. ISSN: 0269-7491.

[GGT15] Shixiang Shane Gu, Zoubin Ghahramani, and Richard E Turner. "Neural adaptive sequential monte carlo". In: *Advances in neural information processing systems* 28 (2015).

[GJL17a] Roland Geyer, Jenna Jambeck, and Kara Law. "Production, use, and fate of all plastics ever made". In: *Science Advances* 3 (July 2017), e1700782.

[GJL17b] Pieralberto Guarniero, Adam M. Johansen, and Anthony Lee. "The Iterated Auxiliary Particle Filter". In: *Journal of the American Statistical Association* 112.520 (2017), pp. 1636–1647.

[GL23] Elisabeth Gassiat and Sylvain Le Corff. "Variational Autoencoder excess risk bound for state space models". In: *Work in progress* (2023).

[GLL20] Elisabeth Gassiat, Sylvain Le Corff, and Luc Lehéricy. "Identifiability and consistent estimation of nonparametric translation hidden Markov models with general state space". In: *The Journal of Machine Learning Research* 21.1 (2020), pp. 4589–4628.

[GLO22] Pierre Gloaguen, Sylvain Le Corff, and Jimmy Olsson. "A pseudo-marginal sequential Monte Carlo online smoothing algorithm". In: *Bernoulli* 28.4 (2022), pp. 2606–2633.

[GMH13] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. "Speech recognition with deep recurrent neural networks". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013), pp. 6645–6649.

[Gon+21] Daniel González-Fernández, Andrés Cózar, Georg Hanke, Josué Viejo, Carmen Morales-Caselles, Rigers Bakiu, Damià Barceló, Filipa Bessa, Antoine Bruge, Maríéa Cabrera, et al. "Floating macrolitter leaked from Europe into the ocean". In: *Nature Sustainability* 4.6 (2021), pp. 474–483.

[Gre+19] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. "Multi-object representation learning with iterative variational inference". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2424–2433.

[GRS20] Ángel F. García-Fernández, Abu Sajana Rahmathullah, and Lennart Svensson. "A metric on the space of finite sets of trajectories for evaluation of multi-target tracking algorithms". In: *IEEE Transactions on Signal Processing* 68 (2020), pp. 3917–3928. ISSN: 1053-587X, 1941-0476.

[GS20] Thushari Gamage and J.D.M. Senevirathna. "Plastic pollution in the marine environment". In: *Heliyon* 6 (Aug. 2020), e04709.

[Häl+21a] Hermanni Hälvä, Sylvain Le Corff, Luc Lehéricy, Jonathan So, Yongjie Zhu, Elisabeth Gassiat, and Aapo Hyvarinen. "Disentangling identifiable features from noisy data with structured nonlinear ICA". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 1624–1633.

[Häl+21b] Hermanni Hälvä, Sylvain Le Corff, Luc Lehéricy, Jonathan So, Yongjie Zhu, Élisabeth Gassiat, and Aapo Hyvärinen. "Disentangling Identifiable Features from Noisy Data with Structured Nonlinear ICA". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 34. 2021.

[HB15] Matthew D Hoffman and David M Blei. "Structured stochastic variational inference". In: *Artificial Intelligence and Statistics*. 2015, pp. 361–369.

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[He+20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. "Momentum contrast for unsupervised visual representation learning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9729–9738.

[He+22] Leiying He, Fangdong Wu, Xiaoqiang Du, and Guofeng Zhang. "Cascade-SORT: A robust fruit counting approach using multiple features cascade matching". In: *Computers and Electronics in Agriculture* 200 (2022), p. 107223. ISSN: 0168-1699.

[Hen+17] Jeremy Heng, Adrian Bishop, George Deligiannidis, and Arnaud Doucet. "Controlled Sequential Monte Carlo". In: *Annals of Statistics* 48 (Aug. 2017).

[HG+14] Matthew D Hoffman, Andrew Gelman, et al. "The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo." In: *J. Mach. Learn. Res.* 15.1 (2014), pp. 1593–1623.

[HH20] Hermanni Hälvä and Aapo Hyvarinen. "Hidden markov nonlinear ica: Unsupervised learning from nonstationary time series". In: *Conference on Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 939–948.

[Hig+17] Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: *ICLR*. 2017.

[Hig+18] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. "Towards a definition of disentangled representations". In: *arXiv preprint arXiv:1812.02230* (2018).

[HKM23] Aapo Hyvärinen, Ilyes Khemakhem, and Hiroshi Morioka. "Nonlinear Independent Component Analysis for Principled Disentanglement in Unsupervised Deep Learning". In: *ArXiv* abs/2303.16535 (2023).

[HO00] Aapo Hyvärinen and Erkki Oja. "Independent component analysis: algorithms and applications". In: *Neural networks* 13.4-5 (2000), pp. 411–430.

[Hof+13] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. "Stochastic variational inference". In: *Journal of Machine Learning Research* (2013).

[How+17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[HP98] A. Hyvarinen and P. Pajunen. "On existence and uniqueness of solutions in nonlinear independent component analysis". In: *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36227)*. Vol. 2. 1998, 1350–1355 vol.2.

[HS81] Berthold K.P. Horn and Brian G. Schunck. "Determining optical flow". In: *Artificial Intelligence* 17.1 (1981), pp. 185–203. ISSN: 0004-3702.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9 (1997), pp. 1735–1780.

[Jam+15] Jenna Jambeck, Roland Geyer, Chris Wilcox, Theodore Siegler, Miriam Perryman, Anthony Andrady, Ramani Narayan, and Kara Law. "Marine pollution. Plastic waste inputs from land into the ocean". In: *Science (New York, N.Y.)* 347 (Feb. 2015), pp. 768–771.

[Joh+16] Matthew J Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. "Composing graphical models with neural networks for structured representations and fast inference". In: *Advances in neural information processing systems (NeurIPS)* 29 (2016).

[JRB18] Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. "Differentiable particle filters: End-to-end learning with algorithmic priors". In: *arXiv preprint arXiv:1805.11122* (2018).

[Kab+21] Rishabh Kabra, Daniel Zoran, Goker Erdogan, Loic Matthey, Antonia Creswell, Matt Botvinick, Alexander Lerchner, and Chris Burgess. "Simone: View-invariant, temporally-abstracted object representations via unsupervised video decomposition". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 20146–20159.

[Kal60] Rudolph Emil Kalman. "A New Approach to Linear Filtering and Prediction Problems". In: *Transactions of the ASME–Journal of Basic Engineering* 82.Series D (1960), pp. 35–45.

[Kan+09] Nicholas Kantas, Arnaud Doucet, Sumeetpal Sindhu Singh, and Jan Marian Maciejowski. "An overview of sequential Monte Carlo methods for parameter estimation in general state-space models". In: *IFAC Proceedings Volumes* 42.10 (2009), pp. 774–785.

[Kar+14] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. "Large-Scale Video Classification with Convolutional Neural Networks". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1725–1732.

[Khe+20] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. "Variational autoencoders and nonlinear ica: A unifying framework". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, pp. 2207–2217.

[Kim+20] Geon-Hyeong Kim, Youngsoo Jang, Hongseok Yang, and Kee-Eung Kim. "Variational Inference for Sequential Data with Future Likelihood Estimates". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 5296–5305.

[Kim+22a] Jonggwan Kim, Yooil Suh, Junhee Lee, Heechan Chae, Hanse Ahn, Yongwha Chung, and Daihee Park. "EmbeddedPigCount: Pig Counting with Video Object Detection and Tracking on an Embedded Board". In: *Sensors* 22.7 (2022). ISSN: 1424-8220.

[Kim+22b] Kyurae Kim, Jisu Oh, Jacob R. Gardner, Adji Bousso Dieng, and Hongseok Kim. "Markov Chain Score Ascent: A Unifying Framework of Variational Inference with Markovian Gradients". In: *NeurIPS*. 2022.

[Kip+22] Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. "Conditional Object-Centric Learning from Video". In: *International Conference on Learning Representations (ICLR)*. 2022.

[KL17] Mohammad Khan and Wu Lin. "Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models". In: *Artificial Intelligence and Statistics*. PMLR. 2017, pp. 878–887.

[KLH18] Rahul Krishnan, Dawen Liang, and Matthew Hoffman. "On the challenges of learning with inference networks on sparse, high-dimensional data". In: *International conference on artificial intelligence and statistics*. PMLR. 2018, pp. 143–151.

[KLJ23] Juan Kuntz, Jen Ning Lim, and Adam M. Johansen. "Particle algorithms for maximum likelihood training of latent variable models". In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. Vol. 206. Proceedings of Machine Learning Research. 2023, pp. 5134–5180.

[KM53] Herman Kahn and Andy W Marshall. "Methods of reducing sample size in Monte Carlo computations". In: *Journal of the Operations Research Society of America* 1.5 (1953), pp. 263–278.

[Kor+21] Anna Korba, Pierre-Cyril Aubin-Frankowski, Szymon Majewski, and Pierre Ablin. "Kernel stein discrepancy descent". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 5719–5730.

[KSS15] Rahul G Krishnan, Uri Shalit, and David Sontag. "Deep kalman filters". In: *arXiv preprint arXiv:1511.05121* (2015).

[KSS17] Rahul Krishnan, Uri Shalit, and David Sontag. "Structured inference networks for nonlinear state space models". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1. 2017.

[KTG92] Daniel Kahneman, Anne Treisman, and Brian J Gibbs. "The reviewing of object files: Object-specific integration of information". In: *Cognitive Psychology* 24.2 (1992), pp. 175–219. ISSN: 0010-0285.

[Kuh55] H. W. Kuhn. "The Hungarian method for the assignment problem". In: *Naval Research Logistics Quarterly* 2.1-2 (1955), pp. 83–97.

[KW14] Diederik Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: Dec. 2014.

[Lar+18] Issam H Laradji, Negar Rostamzadeh, Pedro O Pinheiro, David Vazquez, and Mark Schmidt. "Where are the blobs: Counting by localization with point supervision". In: *Proceedings of the european conference on computer vision (ECCV)*. 2018, pp. 547–562.

[Law+18] Dieterich Lawson, George Tucker, Christian A Naesseth, Chris Maddison, Ryan P Adams, and Yee Whye Teh. "Twisted variational sequential monte carlo". In: *Third workshop on Bayesian Deep Learning (NeurIPS)*. 2018.

[LB+95] Yann LeCun, Yoshua Bengio, et al. "Convolutional networks for images, speech, and time series". In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

[LGJ20] Jiahe Li, Xu Gao, and Tingting Jiang. "Graph Networks for Multiple Object Tracking". In: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2020, pp. 708–717.

[Li+22] Ximing Li, Zeyong Zhao, Jingyi Wu, Yongding Huang, Jiayong Wen, Shikai Sun, Huanlong Xie, Jian Sun, and Yuefang Gao. "Y-BGD: Broiler counting based on multi-object tracking". In: *Computers and Electronics in Agriculture* 202 (2022), p. 107347. ISSN: 0168-1699.

[Lie+20] Colin Lieshout, Kees Oeveren, Tim van Emmerik, and Eric Postma. "Automated River Plastic Monitoring Using Deep Learning and Cameras". In: *Earth and Space Science* 7 (Aug. 2020), e2019EA000960.

[Lin+14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755.

[Liu+18] Xu Liu, Steven W Chen, Shreyas Aditya, Nivedha Sivakumar, Sandeep Dcunha, Chao Qu, Camillo J Taylor, Jnaneshwar Das, and Vijay Kumar. "Robust fruit counting: Combining deep learning, tracking, and structure from motion". In: *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2018, pp. 1045–1052.

[Liu+19] Xu Liu, Steven W Chen, Chenhao Liu, Shreyas S Shivakumar, Jnaneshwar Das, Camillo J Taylor, James Underwood, and Vijay Kumar. "Monocular camera based fruit counting and mapping with semantic data association". In: *IEEE Robotics and Automation Letters* 4.3 (2019), pp. 2296–2303.

[LKH18] Wu Lin, Mohammad Emtiyaz Khan, and Nicolas Hubacher. "Variational Message Passing with Structured Inference Networks". In: *International Conference on Learning Representations*. 2018.

[LM97] François LeGland and Laurent Mével. "Recursive estimation in hidden Markov models". In: *Proceedings of the 36th IEEE Conference on Decision and Control*. Vol. 4. IEEE. 1997, pp. 3468–3473.

[Loc+19] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. "Challenging common assumptions in the unsupervised learning of disentangled representations". In: *international conference on machine learning*. PMLR. 2019, pp. 4114–4124.

[Loc+20] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. "Object-centric learning with slot attention". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 11525–11538.

[Low99] David G Lowe. "Object recognition from local scale-invariant features". In: *Proceedings of the seventh IEEE international conference on computer vision*. Vol. 2. Ieee. 1999, pp. 1150–1157.

[LSV20] Anthony Lee, Sumeetpal Singh, and Matti Vihola. "Coupled conditional backward sampling particle filter". In: *Annals of Statistics* 48 (Oct. 2020), pp. 3066–3089.

[LT16] Yingzhen Li and Richard E Turner. "Rényi divergence variational inference". In: *Advances in neural information processing systems* 29 (2016).

[Lui+21] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. "Hota: A higher order metric for evaluating multi-object tracking". In: *International journal of computer vision* 129 (2021), pp. 548–578.

[Luo+21] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. "Multiple object tracking: A literature review". In: *Artificial intelligence* 293 (2021), p. 103448.

[LV06] Friedrich Liese and Igor Vajda. "On divergences and informations in statistics and information theory". In: *IEEE Transactions on Information Theory* 52.10 (2006), pp. 4394–4412.

[LW16] Qiang Liu and Dilin Wang. "Stein variational gradient descent: A general purpose bayesian inference algorithm". In: *Advances in neural information processing systems* 29 (2016).

[LZ10] Victor Lempitsky and Andrew Zisserman. "Learning to Count Objects in Images". In: *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*. 2010. ISBN: 978-1-61782-380-0.

[Mah03] Ronald PS Mahler. "Multitarget Bayes filtering via first-order multitarget moments". In: *IEEE Transactions on Aerospace and Electronic systems* 39.4 (2003), pp. 1152–1178.

[MCY18] Joseph Marino, Milan Cvitkovic, and Yisong Yue. "A General Method for Amortizing Variational Filtering". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018.

[Min+21] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. "Image segmentation using deep learning: A survey". In: *IEEE transactions on pattern analysis and machine intelligence* 44.7 (2021), pp. 3523–3542.

[Min01] Thomas P. Minka. "Expectation Propagation for Approximate Bayesian Inference". In: *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*. UAI'01. Seattle, Washington: Morgan Kaufmann Publishers Inc., 2001, pp. 362–369. ISBN: 1558608001.

[Moh+20] Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. "Monte carlo gradient estimation in machine learning". In: *The Journal of Machine Learning Research* 21.1 (2020), pp. 5183–5244.

[Mor+14] David Morritt, Paris V. Stefanoudis, Dave Pearce, Oliver A. Crimmen, and Paul F. Clark. "Plastic in the Thames: A river r through it". In: *Marine Pollution Bulletin* 78.1 (2014), pp. 196–200. ISSN: 0025-326X.

[Mor19] Bahman Moraffah. "Inference for Multiple Object Tracking: A Bayesian Nonparametric Approach". In: *arXiv* (2019). ISSN: 23318422.

[MYM18] Joe Marino, Yisong Yue, and Stephan Mandt. "Iterative amortized inference". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 3403–3412.

[Nae+18] Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. "Variational sequential monte carlo". In: *International conference on artificial intelligence and statistics*. PMLR. 2018, pp. 968–977.

[Nea+11] Radford M Neal et al. "MCMC using Hamiltonian dynamics". In: *Handbook of markov chain monte carlo* 2.11 (2011), p. 2.

[NH98] Radford M. Neal and Geoffrey E. Hinton. "A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants". In: *Learning in Graphical Models*. Ed. by Michael I. Jordan. Dordrecht: Springer Netherlands, 1998, pp. 355–368. ISBN: 978-94-011-5014-9.

[NLB20] Christian Naesseth, Fredrik Lindsten, and David Blei. "Markovian score climbing: Variational inference with KL (p|| q)". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 15499–15510.

[NLS+19] Christian A Naesseth, Fredrik Lindsten, Thomas B Schön, et al. "Elements of sequential monte carlo". In: *Foundations and Trends® in Machine Learning* 12.3 (2019), pp. 307–392.

[OW+17] Jimmy Olsson, Johan Westerborn, et al. "Efficient particle-based online smoothing in general hidden Markov models: the PaRIS algorithm". In: *Bernoulli* 23.3 (2017), pp. 1951–1996.

[Owe13] Art B. Owen. *Monte Carlo theory, methods and examples*. 2013.

[Pha+15] Viet-Quoc Pham, Tatsuo Kozakaya, Osamu Yamaguchi, and Ryuzo Okada. "COUNT Forest: CO-Voting Uncertain Number of Targets Using Random Forest for Crowd Density Estimation". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 3253–3261.

[PR23] Alfred Wegener Institute Helmholtz Centre for Polar and Marine Research. *LITTERBASE:Online portal for marine litter*. 2023. URL: https://litterbase.awi.de/.

[PS20] Pedro F Proença and Pedro Simões. *TACO: Trash Annotations in Context for Litter Detection*. 2020.

[PS99] Michael K. Pitt and Neil Shephard. "Filtering via Simulation: Auxiliary Particle Filters". In: *Journal of the American Statistical Association* 94.446 (1999), pp. 590–599. ISSN: 01621459.

[Rab89] L.R. Rabiner. "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[RCC99] Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*. Vol. 2. Springer, 1999.

[Red+16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[Ren+15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015).

[Reu+14] Stephan Reuter, Ba-Tuong Vo, Ba-Ngu Vo, and Klaus Dietmayer. "The Labeled Multi-Bernoulli Filter". In: *IEEE Transactions on Signal Processing* 62.12 (2014), pp. 3246–3260.

[RGB14] Rajesh Ranganath, Sean Gerrish, and David Blei. "Black Box Variational Inference". In: *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*. Ed. by Samuel Kaski and Jukka Corander. Vol. 33. Proceedings of Machine Learning Research. Reykjavik, Iceland: PMLR, 2014, pp. 814–822.

[Ris+16] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. "Performance measures and a data set for multi-target, multi-camera tracking". In: *European conference on computer vision*. Springer. 2016, pp. 17–35.

[Roc+16] Chelsea Rochman, Anthony Andrady, Sarah Dudas, Joan Fabres, François Galgani, Denise lead, Valeria Hidalgo-Ruz, Sunny Hong, Peter Kershaw, Laurent Lebreton, Amy Lusher, Ramani Narayan, Sabine Pahl, James Potemra, Chelsea Rochman, Sheck Sherif, Joni Seager, Won Shim, Paula Sobral, and Linda Amaral-Zettler. "Sources, fate and effects of microplastics in the marine environment: Part 2 of a global assessment". In: *GESAMP Reports and Studies No.90* (Dec. 2016).

[RTB16] Rajesh Ranganath, Dustin Tran, and David Blei. "Hierarchical variational models". In: *International conference on machine learning*. PMLR. 2016, pp. 324–333.

[Rv15] Patrick Rebeschini and Ramon van Handel. "Can Local Particle Filters Beat the Curse of Dimensionality?" In: *The Annals of Applied Probability* 25.5 (Oct. 2015). ISSN: 1050-5164.

[Sär13] S. Särkkä. *Bayesian Filtering and Smoothing*. New York, NY, USA: Cambridge University Press, 2013.

[SBM15] Chris Snyder, Thomas Bengtsson, and Mathias Morzfeld. "Performance bounds for particle filters using the optimal proposal". In: *Monthly Weather Review* 143.11 (2015), pp. 4750–4761.

[SG20] Tanmay Shankar and Abhinav Gupta. "Learning robot skills with temporal variational inference". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 8624–8633.

[Sin+23] Angad Singh, Omar Makhlouf, Maximilian Igl, Joao Messias, Arnaud Doucet, and Shimon Whiteson. "Particle-Based Score Estimation for State Space Model Learning in Autonomous Driving". In: *Conference on Robot Learning*. PMLR. 2023, pp. 1168–1177.

[Sny+08] Chris Snyder, Thomas Bengtsson, Peter Bickel, and Jeff Anderson. "Obstacles to High-Dimensional Particle Filtering". In: *Monthly Weather Review* 136.12 (2008), pp. 4629–4640.

[Sny11] Chris Snyder. "Particle filters, the "optimal" proposal and high-dimensional systems". In: *Proceedings of the ECMWF Seminar on Data Assimilation for atmosphere and ocean*. Citeseer. 2011, pp. 1–10.

[SSB17] Deepak Babu Sam, Shiv Surya, and R. Venkatesh Babu. "Switching Convolutional Neural Network for Crowd Counting". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 4031–4039.

[SWA22] Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. "Simple unsupervised object-centric learning for complex and naturalistic videos". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 18181–18196.

[SZ15] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations*. 2015.

[Tia+19] Mengxiao Tian, Hao Guo, Hong Chen, Qing Wang, Chengjiang Long, and Yuhao Ma. "Automated Pig Counting Using Deep Learning". In: *Computers and Electronics in Agriculture* 163 (2019). ISSN: 01681699.

[TPL20] Mingxing Tan, Ruoming Pang, and Quoc V Le. "Efficientdet: Scalable and efficient object detection". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 10781–10790.

[Tra+15] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. "Learning spatiotemporal features with 3d convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 4489–4497.

[TY21] Rong Tang and Yun Yang. "On Empirical Bayes Variational Autoencoder: An Excess Risk Bound". In: *Proceedings of Thirty Fourth Conference on Learning Theory*. Ed. by Mikhail Belkin and Samory Kpotufe. Vol. 134. Proceedings of Machine Learning Research. PMLR, 2021, pp. 4068–4125.

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[VVH17] Ba-Ngu Vo, Ba-Tuong Vo, and Hung Gia Hoang. "An Efficient Implementation of the Generalized Labeled Multi-Bernoulli Filter". In: *IEEE Transactions on Signal Processing* 65.8 (2017), pp. 1975–1987.

[Wan+20] Zhongdao Wang, Liang Zheng, Yixuan Liu, Yali Li, and Shengjin Wang. "Towards real-time multi-object tracking". In: *European Conference on Computer Vision*. Springer. 2020, pp. 107–122.

[WBJ05] John Winn, Christopher M Bishop, and Tommi Jaakkola. "Variational message passing." In: *Journal of Machine Learning Research* 6.4 (2005).

[WBP17] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. "Simple online and realtime tracking with a deep association metric". In: *2017 IEEE international conference on image processing (ICIP)*. IEEE. 2017, pp. 3645–3649.

[Web+15] Theophane Weber, Nicolas Heess, Ali Eslami, John Schulman, David Wingate, and David Silver. "Reinforced variational inference". In: *Advances in Neural Information Processing Systems (NIPS) Workshops*. 2015.

[Wei+21] Marissa A Weis, Kashyap Chitta, Yash Sharma, Wieland Brendel, Matthias Bethge, Andreas Geiger, and Alexander S Ecker. "Benchmarking unsupervised object representations for video sequences". In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 8253–8313.

[Wel20] Natalie A. Welden. "Chapter 8 - The environmental impacts of plastic pollution". In: *Plastic Waste and Recycling*. Ed. by Trevor M. Letcher. Academic Press, 2020, pp. 195–222. ISBN: 978-0-12-817880-5.

[Wol+20] Mattis Wolf, Katelijn van den Berg, Shungudzemwoyo P Garaba, Nina Gnann, Klaus Sattler, Frederic Stahl, and Oliver Zielinski. "Machine learning for aquatic plastic litter detection, classification and quantification (APLASTIC-Q)". In: *Environmental Research Letters* 15.11 (Nov. 2020), p. 114042.

[Wu+20] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. "A comprehensive survey on graph neural networks". In: *IEEE transactions on neural networks and learning systems* 32.1 (2020), pp. 4–24.

[WV00] Eric A Wan and Rudolph Van Der Merwe. "The unscented Kalman filter for nonlinear estimation". In: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. Ieee. 2000, pp. 153–158.

[XSY17] Feng Xiong, Xingjian Shi, and Dit-Yan Yeung. "Spatiotemporal Modeling for Crowd Counting in Videos". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 5161–5169.

[Xu+20] Jingsong Xu, Litao Yu, Jian Zhang, and Qiang Wu. "Automatic Sheep Counting by Multi-object Tracking". In: *2020 IEEE International Conference on Visual Communications and Image Processing (VCIP)*. 2020, pp. 257–257.

[Yu+18] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. "Deep layer aggregation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2403–2412.

[ZBN22] Liyi Zhang, David M Blei, and Christian A Naesseth. "Transport score climbing: Variational inference using forward KL and adaptive neural transport". In: *arXiv preprint arXiv:2202.01841* (2022).

[Zha+15] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. "Cross-scene crowd counting via deep convolutional neural networks". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 833–841.

[Zha+16] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 589–597.

[Zha+18] Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. "Advances in variational inference". In: *IEEE transactions on pattern analysis and machine intelligence* 41.8 (2018), pp. 2008–2026.

[Zha+19] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. "Object detection with deep learning: A review". In: *IEEE transactions on neural networks and learning systems* 30.11 (2019), pp. 3212–3232.

[Zha+21] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. "Fairmot: On the fairness of detection and re-identification in multiple object tracking". In: *International Journal of Computer Vision* 129 (2021), pp. 3069–3087.

[Zha+22] Yuan Zhao, Josue Nassar, Ian Jordan, Mónica Bugallo, and Il Memming Park. "Streaming variational monte carlo". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.1 (2022), pp. 1150–1161.

[Zhu+17] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. "Flow-Guided Feature Aggregation for Video Object Detection". In: *Proceedings of the IEEE International Conference on Computer Vision*. Vol. 2017-Octob. 2017, pp. 408–417. ISBN: 978-1-5386-1032-9.

[ZKK20] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. "Tracking objects as points". In: *European conference on computer vision*. Springer. 2020, pp. 474–490.

[ZP20] Yuan Zhao and Il Memming Park. "Variational online learning of neural dynamics". In: *Frontiers in computational neuroscience* 14 (2020), p. 71.

[ZWK19] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. "Objects as points". In: *arXiv preprint arXiv:1904.07850* (2019).

Appendix A

Further technical background

A.1 Additional elements on deep learning, computer vision and MOT

We consider videos as sequences of T digital images $V = (I_t)_{t \leq T}$, where each image is an element of $\mathcal{I} = [0,1]^{W \times H \times C}$, W and H being integers respectively for the width and height of the image in number of pixels and C the number of color channels (with $C = 3$ in RGB images). As such, the first two dimensions are specifically referred to as the *spatial* dimensions. In practice, the range of possible values per pixel is fixed, such that the segment $[0,1]$ is further discretized (e.g. 256 values for 8-bit images). We use the notation $I(x,y,c)$ to denote an individual component of an image at a given position (i.e. a pixel) and simply $I(x,y)$ when the operations involved can be understood independently on the channel dimension.

A.1.1 Prediction and feature extraction on images with DNNs

In modern computer vision methods, the most common approach to build algorithms for prediction in images is supervised learning. In short, given a space $\mathcal{Y}$ of high-level attributes in the images, a dataset $\mathcal{D} = \{(I_k, Y_k)\}_{k \leq K}$ of elements in $\mathcal{I} \times \mathcal{Y}$ is assembled and a differentiable parametric function $\mathcal{F}_\gamma : \mathcal{I} \to \mathcal{Z}$ is defined, where $\mathcal{Z}$ is referred to as the prediction space. Given a penalty function $\mathcal{L} : \mathcal{Z} \times \mathcal{Y} \to \mathbb{R}$ differentiable w.r.t its left input, the main challenge is to find

$$\gamma^* = \arg\max_{\gamma \in \Gamma} \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}\left(\mathcal{F}_\gamma(I_k), Y_k\right)$$

where Γ is the parameter space. In general, the optimization problem is solved via stochastic gradient descent and autodifferentiation. In this context, some central questions to build efficient computer vision algorithms are the following:

- Which labels are best suited for a given task ?
- Which classes of functions $\{\mathcal{F}_\gamma\}_{\gamma \in \Gamma}$ are best suited for image data ?

Some examples of popular tasks on images and the possible labels associated to them are the following:

- Object detection and localization with labels as coordinates $Y = \{(x_i, y_i, w_i, h_i)\}_{i \leq I}$ for the rectangular areas covered by the objects of interest, known as bounding-box supervision
- Classification, where $\mathcal{Y} = \{\mathcal{C}_i\}_{i \leq I}$ is a set of possible attributes to describe the content of an image
- Segmentation, e.g. $\mathcal{Y} = \{0, 1\}^{W \times H}$ such that $Y(x, y) = 1$ if the pixel at (x, y) is covered by an object of interest, 0 otherwise.

Network architectures and image features In classical computer vision methodology, most algorithms first transform input images into a set of informative descriptions, called images *features*, then perform the prediction from these features (which usually involves some form of thresholding). For example, a basic ad-hoc solution to object segmentation is to predict the convex hull of the coordinates of object contours of input images. To obtain these contours, one may extract discrete gradients $\nabla_{x,y} I$ w.r.t the spatial dimensions (features), then return the coordinates having the highest values (thresholding).

One of the most important elements in the shift towards learning-based methods for computer vision is the developement of network architectures that define classes of functions $\mathcal{F}_\gamma$ in a similar way. Indeed, in most deep learning-based methods, the latter mapping can usually be understood as performing the following operations

- From a raw image $I \in \mathcal{I}$, a first mapping $F_{\gamma_b} : \mathcal{I} \to \mathcal{H}$ produces a compact representation $\hat{H} = F_{\gamma_b}(I)$ via a sequence of operations built from inductive biases on images.
- A generic layer $f_{\gamma_h} : \mathcal{H} \to \mathcal{Z}$ - usually a fully-connected DNN - converts these features into a valid prediction for the required task.

In computer vision, the first step is often referred to as *feature extraction*, and the function F_{γ_b} is called the *backbone*, while the function f_{γ_h} may be described as the *head*.

While many options are possible for the feature extraction step, convolutional neural networks (CNNs) have shown to be particularly suited for images by exploiting translational invariance to produce compact representations with fewer weights. To define the mappings F_{γ_b}, the core component of CNNs are parameterized spatial convolutions followed by non-linearities, i.e. given $H \in \mathbb{R}^{w \times h \times c}$ and $\omega \in \mathbb{R}^{(2\delta_x+1) \times (2\delta_y+1) \times c}$, transformations of the form $\sigma(\omega * H)$ where σ is a nonlinear function (e.g. a sigmoid activation function) and

$$(\omega * H)(x, y, c) = \sum_{u=-\delta_x}^{\delta_x} \sum_{v=-\delta_y}^{\delta_y} \sum_{w=0}^{c} \omega(u, v, w) H(i - u, j - v, k - w)$$

A distinctive element of these transformations is the notion of weight-sharing, i.e. in the previous equation all components of the output involve the entire weight ω distributed across all dimensions of the input. Though many variants exist, CNNs then additionally involve spatial subsampling: the result of the previous operation is reshaped by keeping only the maximal components amongst subgroups of values. Repeatedly performing these operations with multiple weights per step yields a sequence of complex transformations $(H_k)_{k \leq K}$ from an input image $I = F_0$ to a set of features $H = H_K$ (where the layers $(H_k)_{k<K}$ are called *activations*). In deep learning, the precise organization of the previous transformations is usually referred to

as an *architecture*, and building efficient CNN architectures is a very active area of research, as the latter often concentrate the main computational load in image-based prediction tasks. In many popular architectures built for maximum flexibility like [SZ15; He+16], the total number of parameters $|\gamma_b|$ of the backbones is in the order of several millions, while some works like [How+17; TPL20] specifically focus on computational efficiency with clever implementations and weight distribution. Nonetheless, one aspect making all CNNs particularly appealing computationally - and which essentially enabled their massive use in most image-based tasks - is the easy parallelization of the operations involved. In Figure A.1, we provide a visual illustration of a deep learning architecture involving a CNN, and in Figure A.2 we show one popular architecture used as a backbone layer for various prediction tasks. In these illustrations, ReLU stands for the function $x \mapsto \max(0, x)$ applied unitwise to components of the outputs, which is another popular choice of nonlinear activation function.

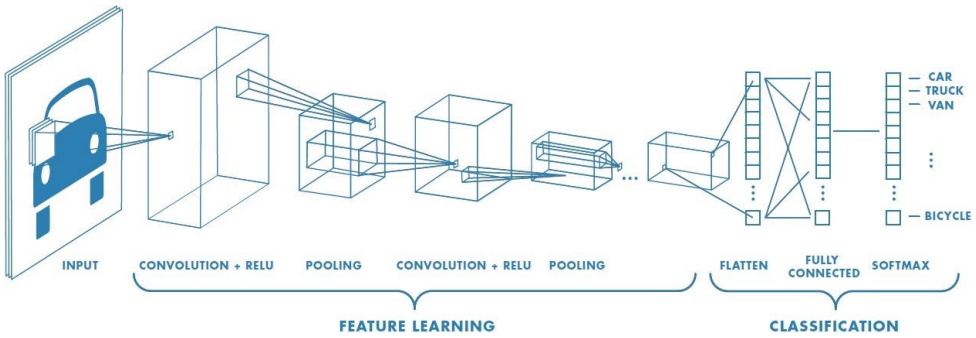


Figure A.1: Overview of a CNN-based architecture for classification

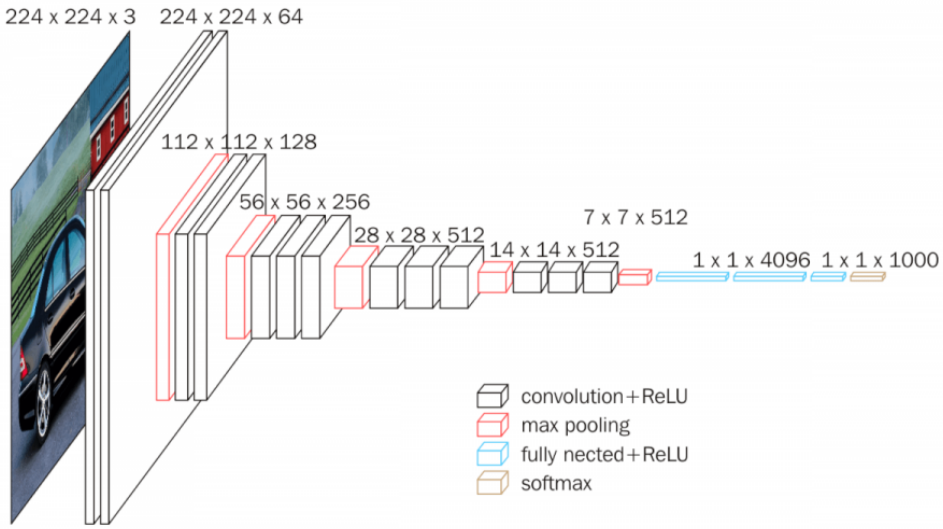


Figure A.2: The VGG16 architecture

In most predictions tasks, the parameters of the backbone layers and the prediction head are trained jointly by gradient descent w.r.t to $\gamma = [\gamma_b, \gamma_f]$ and the composite function $\mathcal{F}_\gamma = f_{\gamma_h} \circ F_{\gamma_b}$. During this process, all layers of the DNN will be tuned for a specific task given the supervision. However, a common practise in computer vision is to view the output of the backbone layers $F_{\gamma_b}(I)$ as a representation of the input image $I \in \mathcal{I}$ which may be relevant

for several tasks if the original training procedure targeted a sufficiently high-level prediction problem. As a consequence, a popular procedure is *pre-training*. First, a set of parameters γ_b is learnt via an intermediate task (e.g. classification) and associated head on a very large dataset. Then, a new head f_{γ_h} and labels are chosen for a more specific task, and parameters are tuned by considering derivatives $\nabla_{\gamma_h}\mathcal{F}$ instead of $\nabla_{\gamma}\mathcal{F}$, keeping the parameters γ_b fixed. This last step is known as *finetuning*.

In computer vision with deep learning methods, a central topic is that of finding the most efficient training procedures to yield strong image features which can be used in a variety of tasks, as any improvement in this regard benefits multiple applications at once.

A.1.2 Recurrent Neural Networks

A Recurrent Neural Network (RNN) is a type of artificial neural network designed for sequential data processing and time-series prediction. It is a class of deep learning models that can capture patterns and dependencies in sequences of data. Unlike feedforward neural networks, RNNs have a feedback loop that allows information to persist and be passed from one step in the sequence to the next. This enables RNNs to maintain a memory of previous inputs, making them capable of handling sequences of varying lengths.

The basic building block of an RNN is called a "cell" which can be thought of as a small neural network that takes an observation and hidden state and produces a new hidden state. The hidden state is like the memory of the cell, and it gets updated at each time step in the sequence. This hidden state allows RNNs to capture information from previous time steps and use it to influence the predictions at the current time step. Formally, one may define a RNN as a mapping $A^\gamma : \mathsf{A} \times \mathsf{Y} \to \mathsf{A}$ where A is the space of hidden states, Y is the observation space, and γ is a parameter in a parameter space Γ. In this context, given a sequence of observations $y_0, \ldots, y_t$ and an initial state a_{-1} (which can be a learnable parameter), RNNs propgate the hidden state $a_s \in \mathsf{A}$ for $s \leq t$ via updates of the form

$$a_s = A^\gamma(a_{s-1}, y_s) \ .$$

A.1.3 Additional topics in multi-object tracking

In this Appendix, we describe in more detail some new state-of-the-art MOT methods briefly mentioned in introduction.

GNN-based assignments

In MOT, many efforts are made to design models which take into account as much information as possible from the videos. However complex these models may be, a choice has to be made to select which quantity can be derived from them to define a cost (e.g. maximum a posteriori estimates in probabilistic models, similarity values for appearance models, etc). To avoid choosing these costs from heuristics, another class of models further abstracts the association problem by using graphical neural networks which are trained to derive optimal cost values from track supervision. In Figure A.3, we show an illustration from [LGJ20] where separate estimates from motion and appearance models are fed into separate graphical models which generate corresponding costs that are combined in final cost matrix for the assignements. Some work, like [BL20], even push this idea further and directly provide CNN features

extracted from images as input for the graph neural networks, then perform the association as a binary classification task on the edges of the graphs, were positive predictions correspond to matchings. We illustrate this in Figure A.4.

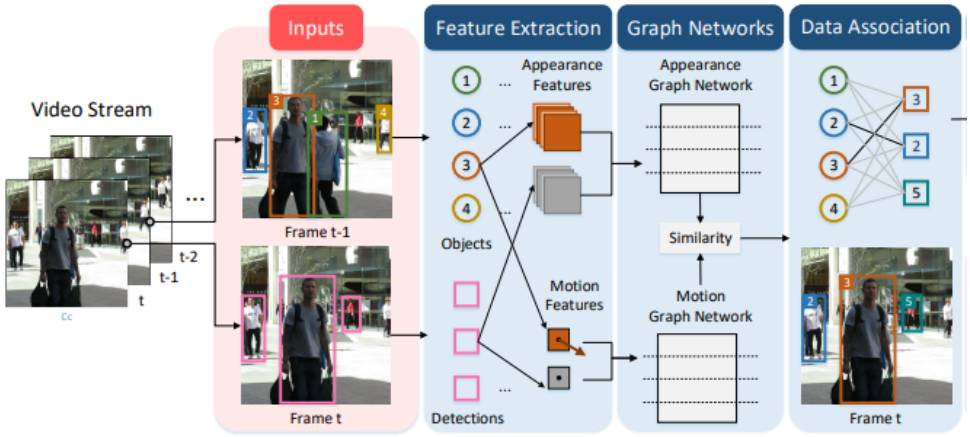


Figure A.3: Tracking architecture from [LGJ20] to illustrate the GNN approach to tracking-by-detection MOT

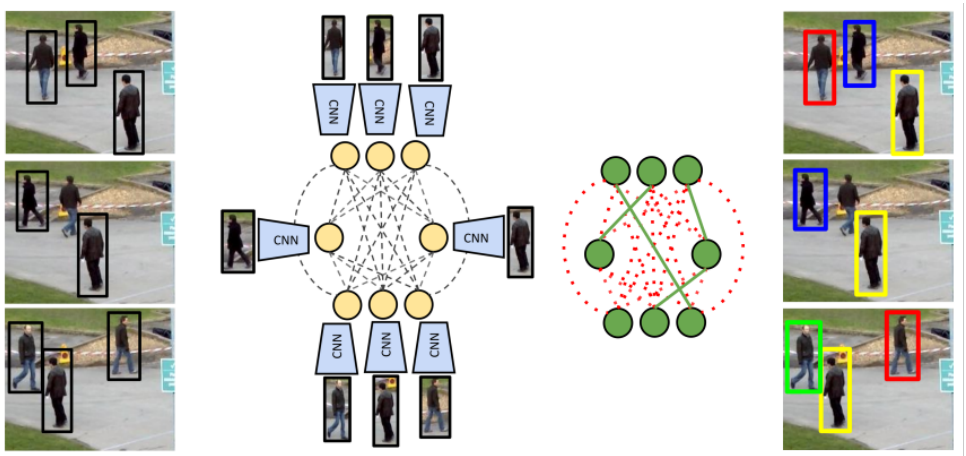


Figure A.4: Architecture from [BL20] where the association procedure is formulated as binary classification on edges of a GNN

In all these methods, dense datasets of annotated videos with corresponding tracks are required to train the networks involved.

Model-free and single network methods

In an attempt to streamline the previous approaches and simplify MOT predictions, new methods have appeared that blur the distinction between the detection and association stages.

Estimating tracking associations simultaneously with detections In a first line of work, new methods supplement detection networks with additional outputs that provide auxilary predictions which are used specifically for tracking. In [ZKK20], illustrated in Figure A.5, a

single network is provided with consecutive pairs of frames and the existing set of tracked objects at the corresponding timesteps, and new detections are greedily associated using an prediction on the displacement of the previous detections. In [Wan+20] appearance features for the detected objects are directly output along with the positions of the objects, and in [Zha+21], a single network infers from each image the set of object positions and bounding-boxes, predicted motion of their centers and the corresponding appearance features, which further facilitates the greedy association process.

Figure A.5: The single network approach from [ZKK20]

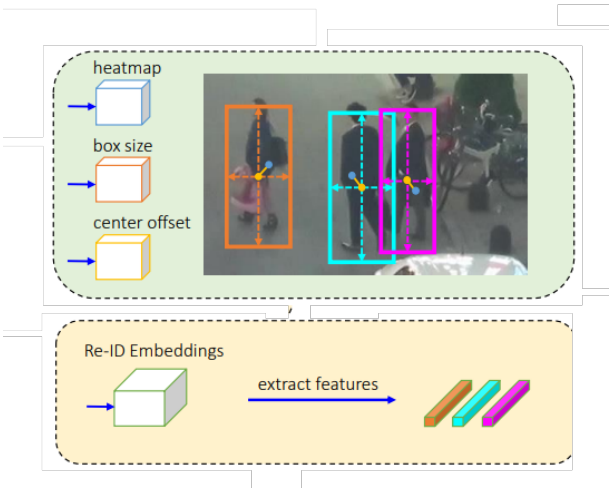


Figure A.6: The single network approach from [Zha+21], where a single network predicts both the location, spatial extent, predicted motion and appearance features of the detected objects in each frames

Attention-based approaches Following advances in transformer-based methods, some works directly tackle the MOT task by formulating it as a set prediction approach on the space of spatial coordinates and labels. At each frame, the set of features of the previous detections and their assigned labels are fed into a transformer network which leverages attention mechanisms to associate the new detections with the previous tracks. This is illustrated in Fig A.7.

A.2 Additional topics on state-space models

A.2.1 Alternate smoothing decompositions

The forward-backward algorithm One of the earliest methods for inference on SSMs [Rab89], which targets *only* the marginal distributions $(\phi^{\theta}_{s|t})_{s\leq t}$, is the *forward-backward* algorithm. In the latter, the marginals are expressed in the following way

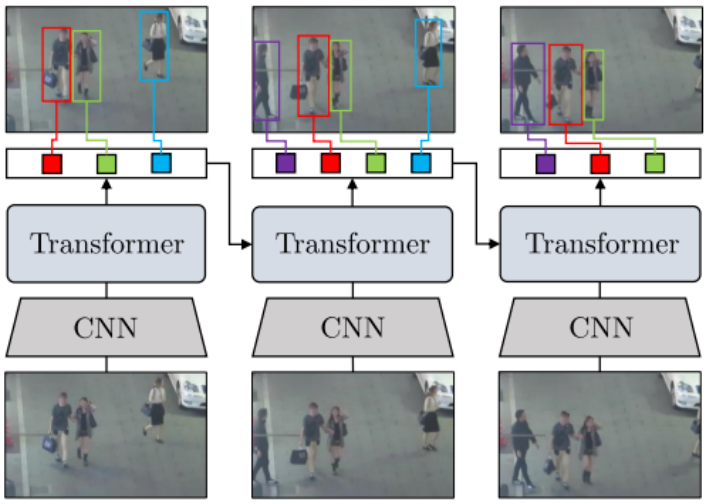


Figure A.7: Tracking by attention

$$\phi^\theta_{s|t} h \propto \int_{\mathsf{X}} h(x_s) \alpha^\theta_s(\mathrm{d}x_s) \beta^\theta_{s|t}(x_s)$$

where

- $(\alpha^\gamma_s)_{s \leq t}$ is a sequence on un-normalized *measures* such that each α^θ_s only depends on $y_{0:s}$.
- $(\beta^\theta_{s|t})_{s \leq t}$ is a sequence of *functions* such that $\beta^\theta_{s|t}$ only depends on $y_{s+1:t}$.

in practice, the computation of the measures $(\alpha^\theta_s)_{s \leq t}$ (referred to as the forward measures) is similar to the forward filtering recursions described in 2.1.1. More precisely,

$$\phi^\theta_{t|t} h = \frac{\alpha^\theta_s h}{\alpha^\theta_s \mathbb{1}_{\mathsf{X}}}$$

i.e. the forward measures are un-normalized versions of the filtering distributions. The backward functions, however, cannot be related to existing distributions. With the convention $\beta^\theta_{t|t} = \mathbb{1}_{\mathsf{X}}$, they are defined as

$$\beta^\theta_{s|t} : x_s \mapsto \int_{\mathsf{X}} m^\theta_{s+1}(x_s, \mathrm{d}x_{s+1}) g^\theta_{s+1}(x_{s+1}) \beta_{s+1|t}(x_{s+1})$$

which is not a normalized quantity in x_s. When further defining

$$\bar{\beta}^\theta_{s|t} = \frac{\alpha^\theta_s \mathbb{1}^\theta_{\mathsf{X}}}{\alpha^\theta_s \beta^\theta_{s|t}} \beta^\theta_{s|t}$$

the marginal smoothing distribution at s can be expressed directly as

$$\phi^\theta_{s|t} h = \int_{\mathsf{X}} \phi^\theta_{t|t} \bar{\beta}^\theta_{s|t} h \tag{A.1}$$

It turns out that separate recursions can be found for the quantities $(\bar{\beta}^\theta_{s|t})_{s \leq t}$ without explicitly involving the forward measures. In this case, the normalized forward-backward algorithm consists in

1. Running the forward filtering recursions up to s.
2. Computing the backward recursion for the $\bar{\beta}^{\theta}_{s|t}$ from t down to s.

Then, marginal smoothing estimates are simply given from A.1. Intuitively, the forward-backward algorithm aggregates information from past and future observations (around a central timestep s) which are combined in a final step. In discrete SSMs (i.e. those where the state space X is discrete), the forward measures correspond to vectors of probabilities corresponding to $\mathbb{P}(Y_0 = y_0, \dots, Y_s = y_s, X_s = x_s)$, and the backward functions are viewed as the conditional probabilities $\mathbb{P}(Y_{s+1} = y_s, \dots, Y_t = y_t | X_s = x_s)$. Note finally that the methodology behind the forward-backward algorithm is rooted in message-passing / belief propagation approaches, which are not specifically bound to SSMs but can be extended to any data viewed as a factor graph. Finally, in some works, the forward-backward algorithm is referred to as the *two-filter* algorithm, as the backward variables are re-defined to correspond to filtering distributions obtained by considering the reverse sequence of observations $y_{t:s+1}$. In SSM literature, the use of the forward-backward algorithm is mostly restricted to situations where only the marginal smoothing estimates are required, but in practice the bi-variate marginal distributions of (X_{s-1}, X_s) given $Y_{0:t}$ are also easily derived from the forward and backward quantities. While this algorithm does not play a central role in the contributions of this thesis, many related works that we will mention tackle inference on structured data via similar "message-passing" routines, i.e. by aggregating information all observations into un-normalized quantities updated using the factor graph defined by the model.

Forward factorization We present here another possible factorization of the joint smoothing distribution, which is based on the Markov property of the forward process $(X_s)_{s \geq 0}$ given $Y_{0:t} = y_{0:t}$. For $1 \leq s \leq t$, we denote $F^{\theta}_{s|t}$ the corresponding Markov transition kernel such that $F^{\theta}_{s|t}(X_{s-1}, \cdot)$ is the conditional distribution of X_s given $(X_{s-1}, Y_{0:t})$. In this setting, the *forward factorization* of the joint smoothing distribution is then given, for any measurable function $h : \mathsf{X}^{t+1} \to \mathsf{H}$, as

$$\phi^{\theta}_{0:t|t} h = \int_{\mathsf{X}^{t+1}} h(x_{0:t}) \left\{ \phi^{\theta}_{0|t}(\mathrm{d}x_0) \prod_{s=1}^{t} F^{\theta}_{s|t}(x_{s-1}, \mathrm{d}x_s) \right\} \tag{A.2}$$

Considering the dependency graph of SSMs, one may observe that the distribution of X_s given $(X_{s-1}, Y_{0:t})$ does not depend on $Y_{0:s-1}$ for $1 \leq s \leq t$. As such, the previous Markov kernels are in practice only parameterized given future observations, and in fact their definition is best understood using the backward variables introduced in the forward-backward algorithm. For $1 \leq s \leq t$, any $x_{s-1} \in \mathsf{X}$ and any measurable function $h : \mathsf{X} \to \mathsf{H}$,

$$F_{s|t}(x_{s-1}, \cdot) h = \frac{\ell^{\theta}_s(x_{s-1}, \cdot) \left(\beta^{\theta}_{s|t} \times h \right)}{\beta^{\theta}_{s-1|t}(x_{s-1})}$$

While the existence of factorization A.2 plays a central role in proving ergodic properties of the conditional process $(X_s)_{s \geq 0}$ given on $Y_{0:t} = y_{0:t}$, its practical use for joint smoothing inference is not very much explored in classical SSM literature, notably because the corresponding implementations require to complete passes through the data: a reverse pass to build the backward functions $(\beta^{\theta}_{s|t})_{s \leq t}$ and a forward pass to build the Markov kernels $(F^{\theta}_{s|t})_{s \leq t}$. Nonetheless,

a convenient aspect of the forward factorization is the direct availability of joint smoothing distributions that go beyond the observed timesteps, i.e. for $t' > t$ and any measurable function $h : \mathsf{X}^{t'+1} \to \mathsf{H}$, $\phi^\theta_{0:t'|t} h$ can be obtained by extending decomposition A.2 by defining the kernels $F^\theta_{s|t} = m^\theta_s$ for $t < s \leq t'$ without any additional renormalization step.

A.2.2 Detailed Kalman filtering and smoothing computations

With $\mathsf{X} = \mathbb{R}^p$ and $\mathsf{Y} = \mathbb{R}^q$, denote $\chi^\theta \sim \mathcal{N}(\mu_0, Q_0)$ and for all $1 \leq s \leq t$,

$$m^\theta_s(X_{s-1}, \cdot) \sim \mathcal{N}(A_s X_{s-1} + a_s, Q_s), \; g^\theta_s(X_s, \cdot) \sim \mathcal{N}(B_s X_s + b_s, R_s)$$

where $A_s \in \mathbb{R}^{p \times p}$, $B_s \in \mathbb{R}^{p \times q}$ and Q_s, R_s are positive definite matrices in $\mathbb{R}^{p \times p}$ and $\mathbb{R}^{q \times q}$, respectively. At s, suppose that $\phi^\theta_{s-1} \sim \mathcal{N}(\mu^\theta_{s-1}, \Sigma^\theta_{s-1})$, we briefly present the analytical computations that correspond to the Kalman versions of the forward filtering recursions and the construction of the backward kernels.

- The Kalman predict step (which corresponds to 2.6) computes $\bar{\phi}^\theta_s \sim \mathcal{N}(\bar{\mu}^\theta_s, \bar{\Sigma}^\theta_s)$ with

$$\bar{\mu}^\theta_s = A_s \mu^\theta_{s-1} + a_s, \; \bar{\Sigma}^\theta_s = A_s \Sigma^\theta_{s-1} A_s^T + Q_s$$

- The Kalman update step (which corresponds to 2.7) computes $\phi^\theta_s \sim \mathcal{N}(\mu^\theta_s, \Sigma^\theta_s)$ with

$$\mu^\theta_s = \bar{\mu}^\theta_s + K^\theta_s \left[y_s - \left(B_s \bar{\mu}^\theta_s + b_s \right) \right], \; \Sigma^\theta_s = \left(I - K^\theta_s B_s \right) \bar{\Sigma}^\theta_s$$

  with $K_s = \bar{\Sigma}^\theta_s B_s^T \left(B_s \bar{\Sigma}^\theta_s B_s^T + R_s \right)^{-1}$

Given that $B^\theta_{s-1}(X_s, \cdot) \propto \phi^\theta_{s-1} m^\theta_s(\cdot, X_s)$, B^θ_{s-1} is a linear and Gaussina kernel such that $B^\theta_{s-1}(X_s, \cdot) \sim \mathcal{N}(\overleftarrow{A}^\theta_{s-1} X_s + \overleftarrow{a}^\theta_s, \overleftarrow{\Sigma}^\theta_s)$, with

$$\overleftarrow{A}^\theta_{s-1} = \overleftarrow{K}^\theta_{s-1}, \; \overleftarrow{a}^\theta_{s-1} = \overleftarrow{C}_{s-1} \mu^\theta_{s-1} - \overleftarrow{K}^\theta_s a_s, \; \overleftarrow{\Sigma}^\theta_{s-1} = \overleftarrow{C}^\theta_{s-1} \Sigma^\theta_{s-1}$$

where $\overleftarrow{K}^\theta_{s-1} = \Sigma^\theta_{s-1} A_s^T \left(A_s \Sigma^\theta_{s-1} A_s^T + Q_s \right)^{-1}$ and $\overleftarrow{C}^\theta_{s-1} = I - \overleftarrow{K}^\theta_{s-1} A_s$

Appendix B

Appendix for Chapter 3

B.1 Categories

In this work, we do not seek to precisely predict the proportions of the different types of counted litter. However, we build our dataset to allow classification tasks. Though litter classifications built by experts already exist, most are based on semantic rather than visual features and do not particularly consider the problem of class imbalance, which makes statistical learning more delicate. In conjunction with water pollution experts, we therefore define a custom macrolitter taxonomy which balances annotation ease and pragmatic decisions for computer vision applications. This classification, depicted in Figure B.1 can be understood as follows.

1. We define a set of frequently observed classes that annotateors can choose from, divided into:

    - Classes for rigid and easily recognisable items which are often observed and have definite shapes
    - Classes for fragmented objects which are often found along river banks but whose aspects are more varied

2. We define two supplementary categories used whenever the annotater cannot classify the item they are observing in an image using classes given in 1.

    - A first category is used whenever the item is clearly identifiable but its class is not proposed. This will ensure that our classification can be improved in the future, as images with items in this category will be checked regularly to decide whether a new class needs to be created.
    - Another category is used whenever the annotater does not understand the item they are seeing. Images containing items denoted as such will not be used for applications involving classification.

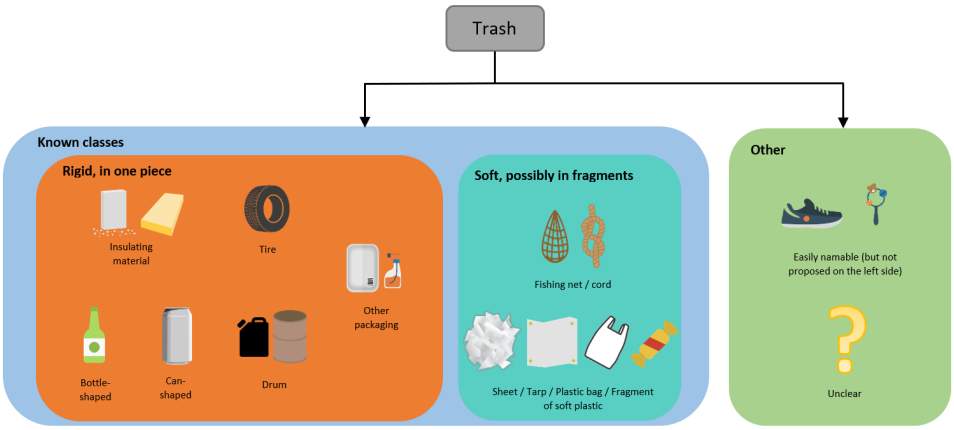


Figure B.1: Trash categories defined to facilitate porting to a counting system that allows trash identification

B.2 Details on the evaluation videos

B.2.1 River segments

In this section, we provide further details on the evaluation material. Figure B.2 shows the setup and positioning of the three river segments S_1, S_2 and S_3 used to evaluate the methods. The segments differ in the following aspects.

- Segment 1: Medium current, high and dense vegetation not obstructing vision of the right riverbank from watercrafts, extra objects installed before the field experiment.
- Segment 2: High current, low and dense vegetation obstructing vision of the right riverbank from watercrafts.
- Segment 3: Medium current, high and little vegetation not obstructing vision of the left riverbank from watercrafts.

B.2.2 Track annotation protocol

To annotate tracks on the evaluation sequences, we used the online tool "CVAT" which allows to locate bounding boxes on video frames and propagate them in time. The following items provide further details on the exact annotation process.

- Object tracks start whenever a litter item becomes fully visible and identifiable by the naked eye.
- Positions and sizes of objects are given at nearly every second of the video with automatic interpolation for frames in-between: this yields clean tracks with precise positions at 24fps.
- We do not provide inferred locations when an object is fully occluded, but tracks restart with the same identity whenever the object becomes visible again.
- Tracks stop whenever an object becomes indistinguishable and will not reappear again.

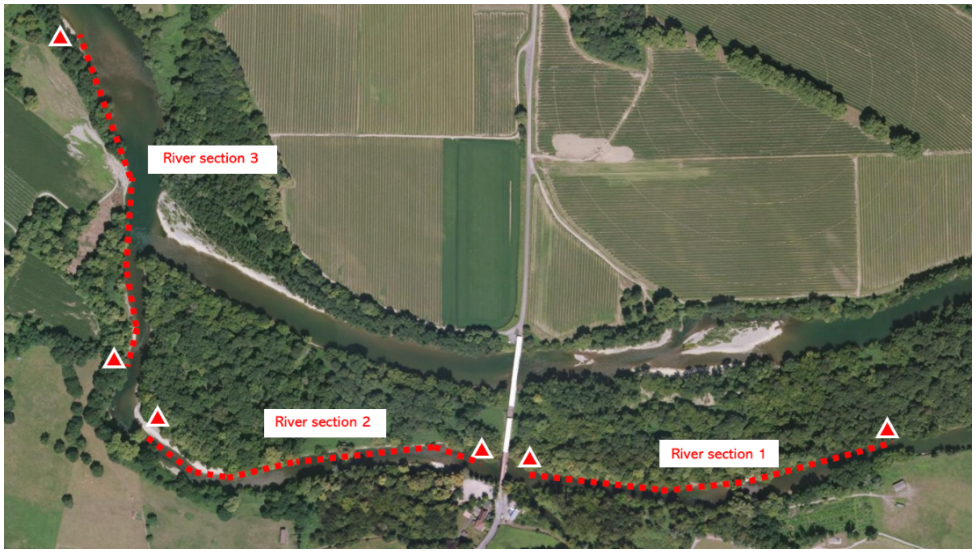


Figure B.2: Aerial view of the three river segments of the evaluation material

B.3 Implementation details for the tracking module

Covariance matrices for state and observation noises In our state space model, Q models the noise associated with the movement model we posit in 3.2.2 involving optical flow estimates, while R models the noise associated with the observation of the true position via our object detector. An attempt to estimate the diagonal values of these matrices was the following.

- To estimate R, we computed a mean L_2 error between the known positions of objects and the associated predictions by the object detector, for images in our training dataset.
- To estimate Q, we built a small synthetic dataset of consecutive frames taken from videos, where positions of objects in two consecutive frames are known.

We computed a mean L_2 error between the known positions in the second frame and the positions estimated by shifting the positions in the first frame with the estimated optical flow values.

This led to $R_{00} = R_{11} = 1.1$, $Q_{00} = 4.7$ and $Q_{11} = 0.9$, for grids of dimensions $\lfloor w/p \rfloor \times \lfloor h/p \rfloor = 480 \times 270$. All other coefficients were not estimated and supposed to be 0.

An important remark is that though we use these values in practice, we found that tracking results are largely unaffected by small variations of R and Q. As long as values are meaningful relative to the image dimensions and the size of the objects, most noise levels show relatively similar performance.

Influence of τ and κ An understanding of κ, τ and ν can be stated as follows. For any track, given a value for κ and ν, an observation at time n is only kept if there are also $\nu \cdot \kappa$ observations in the temporal window of size κ that surrounds n (windows are centered around n except at the start and end of the track). The track is only counted if the remaining number of observations is strictly higher than τ. At a given $\nu > 0.5$, κ and τ should ideally be chosen

to jointly decrease $\hat{\mathsf{N}}_{\mathsf{false}}$ and $\hat{\mathsf{N}}_{\mathsf{red}}$ as much as possible without increasing $\hat{\mathsf{N}}_{\mathsf{mis}}$ (true objects become uncounted if tracks are discarded too easily).

In Figure B.3, we plot the error decomposition of the counts for several values of κ and τ with $\nu = 0.6$ for the outputs of the three different trackers. We choose $\nu = 0.7$ and compute the optimal point as the one which minimizes the overall count error $\hat{\mathsf{N}}(= \hat{\mathsf{N}}_{\mathsf{mis}} + \hat{\mathsf{N}}_{\mathsf{red}} + \hat{\mathsf{N}}_{\mathsf{false}})$.

Bayesian filtering Considering a state space model with $(Y_t, X_t)_{t\geq 0}$ the random processes for the states and observations, respectively, the filtering recursions are given by:

- The predict step: $p(x_{t+1}|y_{1:t}) = \int p(x_{t+1}|x_t)p(y_t|x_{1:t})\mathrm{d}x_t$.
- The update step: $p(x_{t+1}|y_{1:t+1}) \propto p(y_{t+1}|x_{t+1})p(x_{t+1}|y_{1:t})$.

The recursions are intractable in most cases, but when the model is linear and Gaussian, i.e. such that:

$$X_t = A_t X_{t-1} + a_t + \eta_t$$
$$Y_t = B_t X_t + b_t + \epsilon_t$$

with $\eta_t \sim \mathcal{N}(0, Q_t)$ and $\epsilon_t \sim \mathcal{N}(0, R_t)$, then the distribution of X_t given $Z_{1:t}$ is a Gaussian $\mathcal{N}(\mu_t, \Sigma_t)$ following:

- $\mu_{t|t-1} = A_t\mu_{t-1} + a_t$ and $\Sigma_{t|t-1} = A_t\Sigma_{t-1}A_t^T + Q_t$ (Kalman predict step),
- $\mu_t = \mu_{t|t-1} + K_t\left[Z_t - (B_t\mu_{t|t-1} + b_t)\right]$ and $\Sigma_t = (I - K_tB_t)\Sigma_{t|t-1}$ (Kalman update step),

where $K_t = \Sigma_{t|t-1}B_t^T(B_t\Sigma_{t|t-1}B_t^T + R_t)^{-1}$.
In the case of the linearized model in 3.2.2, EKF consists in applying these updates with:

$$A_t = I + \partial_X\Delta_t(\lfloor\mu_{t-1}\rfloor),$$
$$a_t = \Delta_t(\lfloor\mu_{t-1}\rfloor) - \partial_X\Delta_t(\lfloor\mu_{t-1}\rfloor)\mu_{t-1},$$
$$Q_t = Q, R_t = R,$$
$$B_t = I, b_t = 0.$$

Computing the confidence regions In words, $P(i, \ell)$ is the mass in $V_\delta(y_t^i) \subset \mathbb{R}^2$ of the probability distribution of Y_t^ℓ given $Y_{1:n-1}^\ell$. It is related to the filtering distribution at the previous timestep via

$$p(y_t|y_{1:t-1}) = \int\int p(y_t|x_t)p(x_t|x_{t-1})p(x_{t-1}|y_{1:t-1})\mathrm{d}x_t\mathrm{d}x_{t-1}$$

When using EKF, this distribution is a multivariate Gaussian whose moments can be analytically obtained from the filtering mean and variance and the parameters of the linear model, i.e.

$$\mathbb{E}\left[Y_t^\ell|Y_{1:t-1}^\ell\right] = B_t(A_t\mu_{t-1} + a_t) + b_t$$

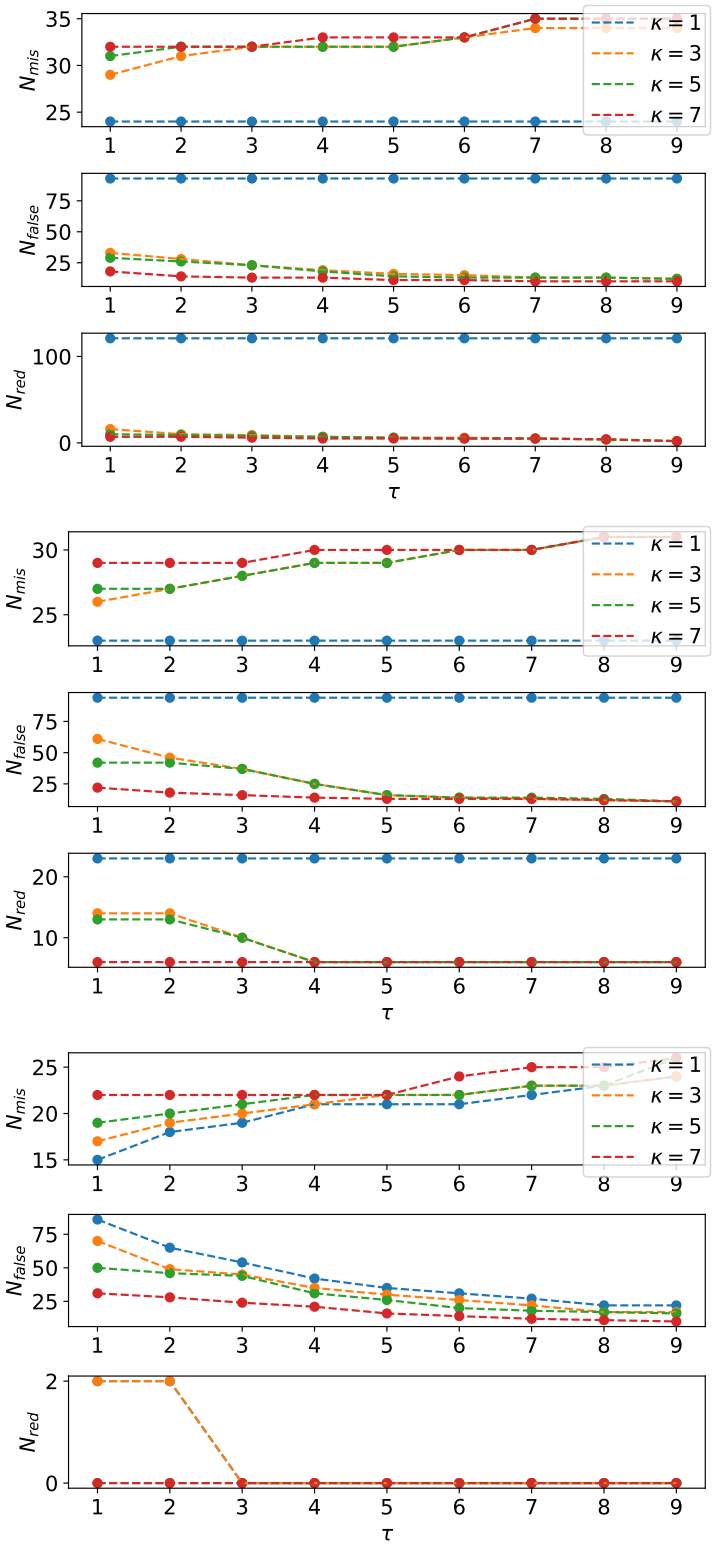


Figure B.3: Calibration of κ and τ for the three competing methods. From top to bottom: FairMOT (best $\kappa = 7, \tau = 9$), SORT (best $\kappa = 7, \tau = 9$), Ours (best $\kappa = 7, \tau = 8$)

and

$$\mathbb{V}\left[Y_t^\ell | Y_{1:t-1}^\ell\right] = B_t(A_t\Sigma_t A_t^T + Q_t)B_t^T + R_t$$

following the previously introduced notation. Note that given the values of A_t, B_t, a_t, b_t in our model these equations are simplified in practice, e.g. $B_t = I, b_t = 0$ and $A_t\mu_{t-1} + a_t = \mu_{t-1} + \Delta_t(\lfloor \mu_{t-1} \rfloor)$. In $\mathbb{R}^2$, values of the cumulative distribution function (cdf) of a multivariate Gaussian distribution are easy to compute. Denote with F_t^ℓ the cdf of $\mathbb{L}_t^\ell$. If $V_\delta(u)$ is a squared neighborhood of size δ and centered on $u = (x, y) \in \mathbb{R}^2$, then, denoting with $\mathbb{L}_t^\ell$ the distribution of Y_t^ℓ given $Y_{1:t-1}^\ell$:

$$\mathbb{L}_t^\ell(V_\delta(u)) = F_t^\ell(x+\delta, y+\delta) + F_t^\ell(x-\delta, y-\delta) - \left[F_t^\ell(x+\delta, y-\delta) + F_t^\ell(x-\delta, y+\delta)\right]$$

This allows easy computation of $P(k, \ell) = \mathbb{L}_t^\ell(V_\delta(u_t^i))$.

Impact of the filtering algorithm An advantage of the data association method proposed in 3.2.3 is that it is very generic and does not constrain the tracking solution to any particular choice of filtering algorithm. As for EKF, UKF implementations are already available to compute the distribution of Y_t given $Y_{1:t-1}$ and the corresponding confidence regions (see B.3 above). We propose a solution to compute this distribution when SMC is used, and performance comparisons between the EKF, UKF and SMC versions of our trackers are discussed.

SMC-based tracking Denote ϕ_t the filtering distribution (ie. that of X_t given $Y_{1:t}$) for the HMM $(X_t, Y_t)_{t\geq 1}$ (omitting the dependency on the observations for notation ease). Using a set of samples $\{\xi_t^i\}_{1\leq i\leq N}$ and importance weights $\{\bar{\omega}_t^i\}_{1\leq i\leq N}$, SMC methods build an approximation of the following form:

$$\hat{\phi}_t^{SMC}(\mathrm{d}x_t) = \sum_{i=1}^N \bar{\omega}_t^i \delta_{\xi_t^i}(\mathrm{d}x_t)\,.$$

Contrary to EKF and UKF, the distribution $\mathbb{L}_t$ of Y_t given $Y_{1:t-1}$ is not directly available but can be obtained via an additional Monte Carlo sampling step. Marginalizing over (X_{t-1}, X_t) and using the conditional independence properties of HMMs, we decompose $\mathbb{L}_t$ using the conditional state transition $\mathbb{M}_t(x, \mathrm{d}x')$ and the likelihood of Y_t given X_t, denoted by $\mathbb{G}_t(x, \mathrm{d}y)$:

$$\mathbb{L}_t(\mathrm{d}y_t) = \int\int \mathbb{G}_t(x_t, \mathrm{d}y_t)\mathbb{M}_t(x_{t-1}, \mathrm{d}x_t)\phi_{t-1}(\mathrm{d}x_{t-1})\,.$$

Replacing ϕ_{t-1} with $\hat{\phi}_{t-1}^{SMC}$ into the previous equation yields

$$\widehat{\mathbb{L}}_t^{SMC}(\mathrm{d}y_k) = \sum_{i=1}^N \bar{\omega}_{t-1}^i \int \mathbb{G}_t(x_t, \mathrm{d}y_k)\mathbb{M}_t(\xi_{t-1}^i, \mathrm{d}x_t)\,.$$

In our model, the state transition is Gaussian and therefore easy to sample from. Thus an approximated predictive distribution $\widehat{\mathbb{L}}_t$ can be obtained using Monte Carlo estimates built from random samples $\{\xi_t^{i,j}\}_{1\leq i\leq N}^{1\leq j\leq M}$ drawn from $\mathbb{M}_t(\xi_{t-1}^i, \mathrm{d}x_k)$. This leads to

$$\widehat{\mathbb{L}}_t(\mathrm{d}y_t) = \sum_{i=1}^N \sum_{j=1}^M \bar{\omega}_{t-1}^i \mathbb{G}_t(\xi_t^{i,j}, \mathrm{d}y_t)\,.$$

Performance comparison In theory, sampling-based methods like UKF and SMC are better suited for nonlinear state space models like the one we propose in **??**. However, we observe very few differences in count results when upgrading from EKF to UKF to SMC. In practise, there is no difference at all between our EKF and UKF implementations, which show strictly identical values for $\hat{\mathsf{N}}_{\mathsf{true}}$, $\hat{\mathsf{N}}_{\mathsf{false}}$ and $\hat{\mathsf{N}}_{\mathsf{red}}$. For the SMC version, values for $\hat{\mathsf{N}}_{\mathsf{false}}$ and $\hat{\mathsf{N}}_{\mathsf{red}}$ improve by a very small amount (2 and 1, respectively), but $\hat{\mathsf{N}}_{\mathsf{mis}}$ is slightly worse (one more object missed), and these results depend loosely on the number of samples used to approximate the filtering distributions and the number of samples for the Monte Carlo scheme. Therefore, our motion estimates via the optical flow Δ_n prove very reliable in our application context, so much that EKF, though suboptimal, brings equivalent results. This comforts us into keeping it as a faster and computationally simpler option. That said, this conclusion might not hold in scenarios where camera motion is even stronger, which was our main motivation to develop a flexible tracking solution and to provide implementations of UKF and SMC versions. This allows easier extension of our work to more challenging data.

Appendix C

Appendix for Chapter 5

C.1 Proofs of the main results

C.1.1 Proof of Proposition 1

Following [GLO22], write

$$q^{\lambda}_{0:n} h_n - \phi^{\theta}_{0:n} h_n = \sum_{k=0}^{n-1} \left(q^{\lambda}_{0:n} \bar{h}_{k|n} - \phi^{\theta}_{0:n} \bar{h}_{k|n} \right), \tag{C.1}$$

where, for each $k \in \{0, n-1\}$, $\bar{h}_{k|n}$ is defined on $(\mathbb{R}^d)^{n+1}$ by

$$\bar{h}_{k|n} : x_{0:n} \mapsto \tilde{h}_k(x_k, x_{k+1}) \,. \tag{C.2}$$

Define, for each $n \in \mathbb{N}$ and $m \in \{0, n\}$, the kernel

$$\mathbf{L}^{\theta}_{m,n}(x'_{0:m}, \mathrm{d}x_{0:n}) := \delta_{x'_{0:m}}(\mathrm{d}x_{0:m}) \prod_{\ell=m}^{n-1} \mathbf{L}^{\theta}_{\ell}(x_\ell, \mathrm{d}x_{\ell+1}) \tag{C.3}$$

on $(\mathbb{R}^d)^{n+1} \times \mathcal{B}((\mathbb{R}^d)^{n+1})$, with the convention $\prod_{\ell=n}^{n-1} f(\ell) = 1$. We have the following decomposition:

$$q^{\lambda}_{0:n} \bar{h}_{k|n} - \phi^{\theta}_{0:n} \bar{h}_{k|n} = \sum_{m=1}^{n} \left(\frac{\tilde{q}_{0:m} \mathbf{L}^{\theta}_{m,n} \bar{h}_{k|n}}{\tilde{q}_{0:m} \mathbf{L}^{\theta}_{m,n} \mathbb{1}} - \frac{\tilde{q}_{0:m-1} \mathbf{L}^{\theta}_{m-1,n} \bar{h}_{k|n}}{\tilde{q}_{0:m-1} \mathbf{L}^{\theta}_{m-1,n} \mathbb{1}} \right) + \frac{\tilde{q}_0 \mathbf{L}^{\theta}_{0,n} \bar{h}_{k|n}}{\tilde{q}_0 \mathbf{L}^{\theta}_{0,n} \mathbb{1}} - \frac{\chi^{\theta} g^{\theta}_0 \mathbf{L}^{\theta}_{0,n} \bar{h}_{k|n}}{\chi^{\theta} g^{\theta}_0 \mathbf{L}^{\theta}_{0,n} \mathbb{1}},$$

where for all $1 \leq m \leq n$, $\tilde{q}_{0:m} = \tilde{q}_m \prod_{k=1}^{m} q^{\lambda}_{k-1|k}$, $\tilde{q}_{0:0} = \tilde{q}_0$, and since $\chi^{\theta} g^{\theta}_0 \mathbf{L}^{\theta}_{0,n} \bar{h}_{k|n} / \chi^{\theta} g^{\theta}_0 \mathbf{L}^{\theta}_{0,n} \mathbb{1} = \phi^{\theta}_{0:n} \bar{h}_{k|n}$. For each $n \in \mathbb{N}$, define $\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{0,n}(x'_0, \mathrm{d}x_{0:n}) := \delta_{x'_0}(\mathrm{d}x_0) \prod_{\ell=0}^{n-1} \mathbf{L}^{\theta}_{\ell}(x_\ell, \mathrm{d}x_{\ell+1})$ and for $m \in \{1, n\}$,

$$\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n}(x'_m, \mathrm{d}x_{0:n}) := \delta_{x'_m}(\mathrm{d}x_m) \prod_{\ell=0}^{m-1} q^{\lambda}_{k|k+1}(x_{\ell+1}, \mathrm{d}x_\ell) \prod_{\ell=m}^{n-1} \mathbf{L}^{\theta}_{\ell}(x_\ell, \mathrm{d}x_{\ell+1}), \tag{C.4}$$

on $\mathbb{R}^d \times \mathcal{B}((\mathbb{R}^d)^{n+1})$. As for all $m \in \{1, n\}$ and measurable function h, $\tilde{q}_{0:m}\mathbf{L}^{\theta}_{m,n}h = \tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n} h$,

$$\frac{\tilde{q}_{0:m}\mathbf{L}^{\theta}_{m,n}\bar{h}_{k|n}}{\tilde{q}_{0:m}\mathbf{L}^{\theta}_{m,n}\mathbb{1}} - \frac{\tilde{q}_{0:m-1}\mathbf{L}^{\theta}_{m-1,n}\bar{h}_{k|n}}{\tilde{q}_{0:m-1}\mathbf{L}^{\theta}_{m-1,n}\mathbb{1}} = \frac{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}}{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n}\mathbb{1}} - \frac{\tilde{q}_{m-1}\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n}\bar{h}_{k|n}}{\tilde{q}_{m-1}\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n}\mathbb{1}}.$$

Therefore,

$$q^{\lambda}_{0:n}\bar{h}_{k|n} - \phi^{\theta}_{0:n}\bar{h}_{k|n} = \sum_{m=1}^{n}\left(\frac{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}}{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n}\mathbb{1}} - \frac{\tilde{q}_{m-1}\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n}\bar{h}_{k|n}}{\tilde{q}_{m-1}\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n}\mathbb{1}}\right) + \frac{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k|n}}{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}} - \frac{\chi^{\theta}g^{\theta}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k|n}}{\chi^{\theta}g^{\theta}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}}. \quad \text{(C.5)}$$

By Lemma 6,

$$\left|\frac{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k|n}}{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}} - \frac{\phi^{\theta}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k|n}}{\phi^{\theta}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}}\right| \leq 2\left\|\tilde{q}_0 - \phi^{\theta}_0\right\|_{\mathrm{tv}}\frac{\sigma_+}{\sigma_-}\|\tilde{h}_k\|_{\infty}\,.$$

Consider now the error term at time $m > 0$ in (C.5). Define the kernel

$$\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}(x'_{m-1}, x'_m, \mathrm{d}x_{0:n}) := \delta_{x'_{m-1}}(\mathrm{d}x_{m-1})\prod_{\ell=0}^{m-2} q^{\lambda}_{\ell|\ell+1}(x_{\ell+1}, \mathrm{d}x_\ell)\delta_{x'_m}(\mathrm{d}x_m)\prod_{\ell=m}^{n-1}\mathbf{L}^{\theta}_{\ell}(x_\ell, \mathrm{d}x_{\ell+1}), \quad \text{(C.6)}$$

on $(\mathbb{R}^d)^2 \times \mathcal{B}((\mathbb{R}^d)^{n+1})$ so that for all $x_{m-1}, x_m \in \mathbb{R}^d$,

$$\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}(x_{m-1}, x_m) = \begin{cases} q^{\lambda}_{m-2|m-1}\cdots q^{\lambda}_{k|k+1}\tilde{h}_k(x_{m-1})\mathbf{L}^{\theta}_{m,n}\mathbb{1}(x_m) & \text{if } k \leq m-2\,, \\ \tilde{h}_k(x_{m-1}, x_m)\mathbf{L}^{\theta}_{m,n}\mathbb{1}(x_m) & \text{if } k = m-1\,, \\ \mathbf{L}^{\theta}_{m,n}\tilde{h}_k(x_m) & \text{if } k \geq m\,. \end{cases}$$

Then, write

$$\frac{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}}{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n}\mathbb{1}} - \frac{\tilde{q}_{m-1}\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n}\bar{h}_{k|n}}{\tilde{q}_{m-1}\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n}\mathbb{1}} = \frac{\tilde{q}_m q^{\lambda}_{m-1|m}\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}}{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n}\mathbb{1}} - \frac{\tilde{q}_{m-1}\mathbf{L}^{\theta}_{m-1}\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}}{\tilde{q}_{m-1}\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n}\mathbb{1}}\,.$$

Let $1 \leq m \leq n$ and x^*_{m-1} and x^*_m be arbitrary elements in $\mathbb{R}^d$. For $k \neq m-1$, define

$$\begin{aligned}\mathcal{L}^{*,\lambda,\theta}_{m,n}\bar{h}_{k|n}(x_{m-1}, x_m) &= \frac{\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}(x_{m-1}, x_m)}{\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\mathbb{1}(x_{m-1}, x_m)} - \frac{\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}(x^*_{m-1}, x^*_m)}{\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\mathbb{1}(x^*_{m-1}, x^*_m)}\,, \\ &= \frac{\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}(x_{m-1}, x_m)}{\mathbf{L}^{\theta}_{m,n}\mathbb{1}(x_m)} - \frac{\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}(x^*_{m-1}, x^*_m)}{\mathbf{L}^{\theta}_{m,n}\mathbb{1}(x^*_m)}\end{aligned} \quad \text{(C.7)}$$

and for $k = m-1$, $\mathcal{L}^{*,\lambda,\theta}_{m,n}\bar{h}_{k|n}(x_{m-1}, x_m) = \tilde{h}_k(x_{m-1}, x_m)$. By Lemma 7, $\left\|\mathcal{L}^{*,\lambda,\theta}_{m,n}\bar{h}_{k|n}\right\|_{\infty}$ can be upper bounded and note that

$$\begin{aligned}&\frac{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}}{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n}\mathbb{1}} - \frac{\tilde{q}_{m-1}\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n}\bar{h}_{k|n}}{\tilde{q}_{m-1}\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n}\mathbb{1}} \\ &\qquad = \frac{\tilde{q}_m q^{\lambda}_{m-1|m}\left\{\mathcal{L}^{*,\lambda,\theta}_{m,n}\bar{h}_{k|n}\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\mathbb{1}\right\}}{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n}\mathbb{1}} - \frac{\tilde{q}_{m-1}\mathbf{L}^{\theta}_{m-1}\left\{\mathcal{L}^{*,\lambda,\theta}_{m,n}\bar{h}_{k|n}\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\mathbb{1}\right\}}{\tilde{q}_{m-1}\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n}\mathbb{1}}\,.\end{aligned}$$

By definition of the normalized measure $\tilde{\phi}^{\theta}_{m-1:m}$,

$$\begin{aligned}
&\frac{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n} \bar{h}_{k|n}}{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}} - \frac{\tilde{q}_{m-1} \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n} \bar{h}_{k|n}}{\tilde{q}_{m-1} \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n} \mathbb{1}} \\
&\qquad = \frac{\tilde{q}_m q^{\lambda}_{m-1|m} \left\{ \mathcal{L}^{*,\lambda,\theta}_{m,n} \bar{h}_{k|n} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1} \right\}}{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}} - \frac{\tilde{\phi}^{\theta}_{m-1:m} \left\{ \mathcal{L}^{*,\lambda,\theta}_{m,n} \bar{h}_{k|n} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1} \right\}}{\tilde{\phi}^{\theta}_{m-1:m} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}} \\
&\qquad = \frac{\tilde{q}_m q^{\lambda}_{m-1|m} \left\{ \mathcal{L}^{*,\lambda,\theta}_{m,n} \bar{h}_{k|n} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1} \right\} - \tilde{\phi}^{\theta}_{m-1:m} \left\{ \mathcal{L}^{*,\lambda,\theta}_{m,n} \bar{h}_{k|n} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1} \right\}}{\tilde{\phi}^{\theta}_{m-1:m} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}} \\
&\qquad\qquad + \frac{\tilde{q}_m q^{\lambda}_{m-1|m} \left\{ \mathcal{L}^{*,\lambda,\theta}_{m,n} \bar{h}_{k|n} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1} \right\}}{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}} \left(\frac{\tilde{\phi}^{\theta}_{m-1:m} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1} - \tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}}{\tilde{\phi}^{\theta}_{m-1:m} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}} \right) .
\end{aligned}$$

Then, using that

$$\left| \frac{\tilde{q}_m q^{\lambda}_{m-1|m} \left\{ \mathcal{L}^{*,\lambda,\theta}_{m,n} \bar{h}_{k|n} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1} \right\}}{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}} \right| \leq \left\| \mathcal{L}^{*,\lambda,\theta}_{m,n} \bar{h}_{k|n} \right\|_{\infty} ,$$

and the fact that $\tilde{\nu}^{\lambda}_{m-1:m} = \tilde{q}_m q^{\lambda}_{m-1|m}$,

$$\left| \frac{\tilde{\phi}^{\theta}_{m-1:m} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1} - \tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}}{\tilde{\phi}^{\theta}_{m-1:m} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}} \right| \leq \left\| \tilde{\phi}^{\theta}_{m-1:m} - \tilde{\nu}^{\lambda}_{m-1:m} \right\|_{\mathrm{tv}} \frac{\left\| \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1} \right\|_{\infty}}{\tilde{\phi}^{\theta}_{m-1:m} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}} ,$$

and

$$\begin{aligned}
&\left| \frac{\tilde{q}_m q^{\lambda}_{m-1|m} \left\{ \mathcal{L}^{*,\lambda,\theta}_{m,n} \bar{h}_{k|n} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1} \right\} - \tilde{\phi}^{\theta}_{m-1:m} \left\{ \mathcal{L}^{*,\lambda,\theta}_{m,n} \bar{h}_{k|n} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1} \right\}}{\tilde{\phi}^{\theta}_{m-1:m} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}} \right| \\
&\qquad\qquad \leq \left\| \tilde{\phi}^{\theta}_{m-1:m} - \tilde{\nu}^{\lambda}_{m-1:m} \right\|_{\mathrm{tv}} \frac{\left\| \mathcal{L}^{*,\lambda,\theta}_{m,n} \bar{h}_{k|n} \right\|_{\infty} \left\| \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1} \right\|_{\infty}}{\tilde{\phi}^{\theta}_{m-1:m} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}} ,
\end{aligned}$$

yields

$$\left| \frac{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n} \bar{h}_{k|n}}{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}} - \frac{\tilde{q}_{m-1} \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n} \bar{h}_{k|n}}{\tilde{q}_{m-1} \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n} \mathbb{1}} \right| \leq 2 \left\| \tilde{\phi}^{\theta}_{m-1:m} - \tilde{\nu}^{\lambda}_{m-1:m} \right\|_{\mathrm{tv}} \frac{\left\| \mathcal{L}^{*,\lambda,\theta}_{m,n} \bar{h}_{k|n} \right\|_{\infty} \left\| \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1} \right\|_{\infty}}{\tilde{\phi}^{\lambda}_{m} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}} .$$

Note also that by H1,

$$\tilde{\phi}^{\theta}_{m-1:m} \tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1} \geq \sigma_{-} \mu \mathbf{L}^{\theta}_{m+1,n-1} \mathbb{1} ,$$

and for all $x_m \in \mathbb{R}^d$,

$$\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}(x_m) \leq \sigma_{+} \mu \mathbf{L}^{\theta}_{m+1,n-1} \mathbb{1} .$$

Therefore,

$$\left| \frac{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n} \bar{h}_{k|n}}{\tilde{q}_m \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n} \mathbb{1}} - \frac{\tilde{q}_{m-1} \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n} \bar{h}_{k|n}}{\tilde{q}_{m-1} \boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n} \mathbb{1}} \right| \leq 2 \frac{\sigma_{+}}{\sigma_{-}} \left\| \tilde{\phi}^{\theta}_{m-1:m} - \tilde{\nu}^{\lambda}_{m-1:m} \right\|_{\mathrm{tv}} \left\| \mathcal{L}^{*,\lambda,\theta}_{m,n} \bar{h}_{k|n} \right\|_{\infty} .$$

The proof is completed using Lemma 7.

C.1.2 Proof of Corollary 2

It is enough to introduce the same decomposition as the one used in Proposition 1:

$$q^{\lambda}_{0:n}\bar{h}_{k_\star|n} - \phi^{\theta}_{0:n}\bar{h}_{k_\star|n} = \sum_{m=1}^{n}\left(\frac{\tilde{q}_m\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k_\star|n}}{\tilde{q}_m\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n}\mathbb{1}} - \frac{\tilde{q}_{m-1}\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n}\bar{h}_{k_\star|n}}{\tilde{q}_{m-1}\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n}\mathbb{1}}\right) + \frac{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k_\star|n}}{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}} - \frac{\chi^{\theta}g^{\theta}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k_\star|n}}{\chi^{\theta}g^{\theta}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}}.$$

Each term is then controlled similarly as in the proof of Proposition 1. By Lemma 6,

$$\left|\frac{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k_\star|n}}{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}} - \frac{\phi^{\theta}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k_\star|n}}{\phi^{\theta}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}}\right| \leq 2c_0(\gamma)\frac{\sigma_+}{\sigma_-}\|\tilde{h}_k\|_\infty \ .$$

On the other hand, the error term at time $m > 0$ is upper bounded by

$$\left|\frac{\tilde{q}_m\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}}{\tilde{q}_m\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m,n}\mathbb{1}} - \frac{\tilde{q}_{m-1}\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n}\bar{h}_{k|n}}{\tilde{q}_{m-1}\boldsymbol{\mathcal{L}}^{\lambda,\theta}_{m-1,n}\mathbb{1}}\right| \leq 2\frac{\sigma_+}{\sigma_-}c_m(\theta,\lambda)\left\|\mathcal{L}^{*,\lambda,\theta}_{m,n}\bar{h}_{k|n}\right\|_\infty \ .$$

The proof is completed using Lemma 7.

C.2 Technical results

Lemma 6 *Assume that H1 holds. Then for all, $\theta \in \Theta$, $\lambda \in \Lambda$, $n \geq 1$, $k \in \{0, n-1\}$, bounded and measurable function $\tilde{h}_k$,*

$$\left|\frac{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k|n}}{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}} - \frac{\chi^{\theta}g^{\theta}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k|n}}{\chi^{\theta}g^{\theta}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}}\right| \leq 2\left\|\tilde{q}_0 - \phi^{\theta}_0\right\|_{\mathrm{tv}}\frac{\sigma_+}{\sigma_-}\|\tilde{h}_k\|_\infty \ ,$$

where $\bar{h}_{k|n}$ is defined in (C.2).

Proof Consider the following decomposition of the first term:

$$\begin{aligned}\frac{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k|n}}{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}} - \frac{\chi^{\theta}g^{\theta}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k|n}}{\chi^{\theta}g^{\theta}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}} &= \frac{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k|n}}{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}} - \frac{\phi^{\theta}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k|n}}{\phi^{\theta}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}} \ , \\ &= \frac{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k|n} - \phi^{\theta}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k|n}}{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}} \\ &\quad + \frac{\phi^{\theta}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k|n}}{\phi^{\theta}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}}\frac{\phi^{\theta}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1} - \tilde{q}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}}{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}} \ ,\end{aligned}$$

where ϕ^{θ}_0 the filtering distribution at time 0, i.e the law defined as $\phi^{\theta}_0 h = \chi^{\theta}g^{\theta}_0 h/\chi^{\theta}g^{\theta}_0$. Note that

$$\left|\frac{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k|n} - \phi^{\theta}_0\mathbf{L}^{\theta}_{0,n}\bar{h}_{k|n}}{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}}\right| \leq \left\|\tilde{q}_0 - \phi^{\theta}_0\right\|_{\mathrm{tv}}\frac{\|\mathbf{L}^{\theta}_{0,n}\bar{h}_{k|n}\|_\infty}{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}} \leq \left\|\tilde{q}_0 - \phi^{\theta}_0\right\|_{\mathrm{tv}}\frac{\|\mathbf{L}^{\theta}_{0,n}\mathbb{1}\|_\infty\|\bar{h}_{k|n}\|_\infty}{\tilde{q}_0\mathbf{L}^{\theta}_{0,n}\mathbb{1}}$$

and, using that $\phi_0^\theta \mathbf{L}_{0,n}^\theta \bar{h}_{k|n} / \phi_0^\theta \mathbf{L}_{0,n}^\theta \mathbb{1} \leq \|\bar{h}_{k|n}\|_\infty$,

$$\left| \frac{\phi_0^\theta \mathbf{L}_{0,n}^\theta \bar{h}_{k|n}}{\phi_0^\theta \mathbf{L}_{0,n}^\theta \mathbb{1}} \frac{\phi_0^\theta \mathbf{L}_{0,n}^\theta \mathbb{1} - \tilde{q}_0 \mathbf{L}_{0,n}^\theta \mathbb{1}}{\tilde{q}_0 \mathbf{L}_{0,n}^\theta \mathbb{1}} \right| \leq \left\| \tilde{q}_0 - \phi_0^\theta \right\|_{\mathrm{tv}} \frac{\|\mathbf{L}_{0,n}^\theta \mathbb{1}\|_\infty \|\bar{h}_{k|n}\|_\infty}{\tilde{q}_0 \mathbf{L}_{0,n}^\theta \mathbb{1}}$$

Then,

$$\left| \frac{\tilde{q}_0 \mathbf{L}_{0,n}^\theta \bar{h}_{k|n}}{\tilde{q}_0 \mathbf{L}_{0,n}^\theta \mathbb{1}} - \frac{\phi_0^\theta \mathbf{L}_{0,n}^\theta \bar{h}_{k|n}}{\phi_0^\theta \mathbf{L}_{0,n}^\theta \mathbb{1}} \right| \leq 2 \left\| \tilde{q}_0 - \phi_0^\theta \right\|_{\mathrm{tv}} \frac{\|\mathbf{L}_{0,n}^\theta \mathbb{1}\|_\infty \|\bar{h}_{k|n}\|_\infty}{\tilde{q}_0 \mathbf{L}_{0,n}^\theta \mathbb{1}} \, .$$

By H1, for all $x_0 \in \mathbb{R}^d$,

$$\mathbf{L}_{0,n}^\theta \mathbb{1}(x_0) = \int \ell_{0,\theta}(x_0, x_1) \mu(\mathrm{d}x_1) \mathbf{L}_{1,n}^\theta \mathbb{1}(x_1) \leq \sigma_+ \int \mu(\mathrm{d}x_1) \mathbf{L}_{1,n}^\theta \mathbb{1}(x_1)$$

and

$$\tilde{q}_0 \mathbf{L}_{0,n}^\theta \mathbb{1} = \int \tilde{q}_0(\mathrm{d}x_0) \ell_{0,\theta}(x_0, x_1) \mu(\mathrm{d}x_1) \mathbf{L}_{1,n}^\theta \mathbb{1}(x_1) \geq \sigma_- \int \mu(\mathrm{d}x_1) \mathbf{L}_{1,n}^\theta \mathbb{1}(x_1) \, ,$$

which yields

$$\left| \frac{\tilde{q}_0 \mathbf{L}_{0,n}^\theta \bar{h}_{k|n}}{\tilde{q}_0 \mathbf{L}_{0,n}^\theta \mathbb{1}} - \frac{\phi_0^\theta \mathbf{L}_{0,n}^\theta \bar{h}_{k|n}}{\phi_0^\theta \mathbf{L}_{0,n}^\theta \mathbb{1}} \right| \leq 2 \left\| \tilde{q}_0 - \phi_0^\theta \right\|_{\mathrm{tv}} \frac{\sigma_+}{\sigma_-} \|\tilde{h}_k\|_\infty \, .$$

■

Lemma 7 *Assume that H1 holds. Then for all $n \in \mathbb{N}$, $\theta \in \Theta$, $\lambda \in \Lambda$, $m \in \{1, n\}$, $k \in \{0, n-1\}$, $x_{m-1}, x_m, x_{m-1}^*, x_m^*$ in $\mathbb{R}^d$, bounded and measurable function $\tilde{h}_k$,*

$$\left| \mathcal{L}_{m,n}^{*,\lambda,\theta} \bar{h}_{k|n}(x_{m-1}, x_m) \right| \leq \begin{cases} \|\tilde{h}_k\|_\infty \rho^{m-k-1} & \text{if } k \leq m-2 \, , \\ \|\tilde{h}_k\|_\infty & \text{if } k = m-1 \, , \\ \|\tilde{h}_k\|_\infty \rho^{k-m+1} & \text{if } k \geq m \, . \end{cases}$$

where $\rho = 1 - \sigma_-/\sigma_+$ and $\bar{h}_{k|n}$ is defined in (C.2) *and $\mathcal{L}_{m,n}^{*,\lambda,\theta} \bar{h}_{k|n}$ is defined in* (C.7)*.*

Proof The proof is adapted from [GLO22, Lemma D.3] and given here for completeness. Assume first that $k \leq m-2$. Then,

$$\frac{\tilde{\mathcal{L}}_{m,n}^{\lambda,\theta} \bar{h}_{k|n}(x_{m-1}, x_m)}{\mathbf{L}_{m,n}^\theta \mathbb{1}(x_m)} = q_{m-2|m-1}^\lambda \cdots q_{k|k+1}^\lambda \tilde{h}_k(x_{m-1})$$

Therefore,

$$\frac{\tilde{\mathcal{L}}_{m,n}^{\lambda,\theta} \bar{h}_{k|n}(x_{m-1}, x_m)}{\mathbf{L}_{m,n}^\theta \mathbb{1}(x_m)} - \frac{\tilde{\mathcal{L}}_{m,n}^{\lambda,\theta} \bar{h}_{k|n}(x_{m-1}^*, x_m^*)}{\mathbf{L}_{m,n}^\theta \mathbb{1}(x_m^*)} = (\delta_{x_{m-1}} - \delta_{x_{m-1}^*}) q_{m-2|m-1}^\lambda \cdots q_{k|k+1}^\lambda \tilde{h}_k \, .$$

By H1, the Dobrushin coefficient of the variational backward kernels is upper-bounded by $1 - \sigma_-/\sigma_+$ so that

$$\left| \frac{\tilde{\mathcal{L}}_{m,n}^{\lambda,\theta} \bar{h}_{k|n}(x_{m-1}, x_m)}{\mathbf{L}_{m,n}^\theta \mathbb{1}(x_m)} - \frac{\tilde{\mathcal{L}}_{m,n}^{\lambda,\theta} \bar{h}_{k|n}(x_{m-1}^*, x_m^*)}{\mathbf{L}_{m,n}^\theta \mathbb{1}(x_m^*)} \right| \leq \left(1 - \frac{\sigma_-}{\sigma_+}\right)^{m-k-1} \left\| \tilde{h}_k \right\|_\infty \, .$$

In the case where $k = m - 1$,

$$\frac{\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}(x_{m-1}, x_m)}{\mathbf{L}^{\theta}_{m,n}\mathbb{1}(x_m)} = \tilde{h}_k(x_k, x_{k+1}) \ ,$$

so that the result is straightforward. Assume now first that $k \geq m$. Note that

$$\frac{\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}(x_{m-1}, x_m)}{\mathbf{L}^{\theta}_{m,n}\mathbb{1}(x_m)} = \frac{\mathbf{L}^{\theta}_{m,n}\bar{h}_{k|n}(x_{m-1}, x_m)}{\mathbf{L}^{\theta}_{m,n}\mathbb{1}(x_m)} = \frac{F^{\theta}_{m|n} \dots F^{\theta}_{k|n}\bar{h}_{k|n}(x_m) \cdot \mathbf{L}^{\theta}_{m,n}\mathbb{1}(x_m)}{\mathbf{L}^{\theta}_{m,n}\mathbb{1}(x_m)} \ ,$$

where the forward kernel $\mathsf{F}^{\theta}_{\ell|n}$ is given by

$$\mathsf{F}^{\theta}_{\ell|n}h(x_\ell) = \frac{\mathbf{L}^{\theta}_{\ell}(h\mathbf{L}^{\theta}_{\ell+1,n-1}\mathbb{1})(x_\ell)}{\mathbf{L}^{\theta}_{\ell,n-1}\mathbb{1}(x_\ell)} \ .$$

By H1,

$$\mathsf{F}^{\theta}_{\ell|n}h(x_\ell) \geq \frac{\sigma_-}{\sigma_+}\mu_{\ell|n}h \ ,$$

with $\mu_{\ell|n}h = \mu(h\mathbf{L}^{\theta}_{\ell+1,n-1}\mathbb{1})(x_\ell)/\mu\mathbf{L}^{\theta}_{\ell+1,n-1}\mathbb{1}$. Therefore, the Dobrushin coefficients of the kernels $F^{\theta}_{\ell|n}$ are also upper-bounded by $1 - \sigma_-/\sigma_+$. On the other hand,

$$\frac{\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}(x_{m-1}, x_m)}{\mathbf{L}^{\theta}_{m,n}\mathbb{1}(x_m)} - \frac{\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}(x^*_{m-1}, x^*_m)}{\mathbf{L}^{\theta}_{m,n}\mathbb{1}(x^*_m)} = (\lambda_{m|n} - \lambda'_{m|n})\mathsf{F}^{\theta}_{m|n} \dots \mathsf{F}^{\theta}_{k|n}\bar{h}_{k|n},$$

where $\lambda_{m|n}h = \delta_{x_m}h\mathbf{L}^{\theta}_{m,n}\mathbb{1}/\delta_{x_m}\mathbf{L}^{\theta}_{m,n}\mathbb{1}$ and $\lambda'_{m|n}h = \delta_{x'_m}h\mathbf{L}^{\theta}_{m,n}\mathbb{1}/\delta_{x'_m}\mathbf{L}^{\theta}_{m,n}\mathbb{1}$. This yields

$$\left|\frac{\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}(x_{m-1}, x_m)}{\mathbf{L}^{\theta}_{m,n}\mathbb{1}(x_m)} - \frac{\tilde{\mathcal{L}}^{\lambda,\theta}_{m,n}\bar{h}_{k|n}(x^*_{m-1}, x^*_m)}{\mathbf{L}^{\theta}_{m,n}\mathbb{1}(x^*_m)}\right| \leq \left(1 - \frac{\sigma_-}{\sigma_+}\right)^{k-m+1} \left\|\tilde{h}_k\right\|_\infty \ ,$$

which concludes the proof.

■

C.2.1 Hardware configuration

We ran all experiments on a machine with the following specifications.

- CPUs: 4x Intel(R) Xeon(R) Gold 6154 (total 72 cores, 144 threads).
- RAM: 260 Go.

No GPU was used.

C.2.2 Linear Gaussian models

We provide here additional figures for the experiments of Section 5.2.1. Figure C.1 shows the accuracy of the optimal Kalman smoothing (with true parameters γ) w.r.t the true states, as well as the numerical values for the smoothing errors at the three stopping points of the optimization. We also provide examples of smoothed states for the fully fitted models against the ground truth Kalman smoother which uses the true parameters γ.

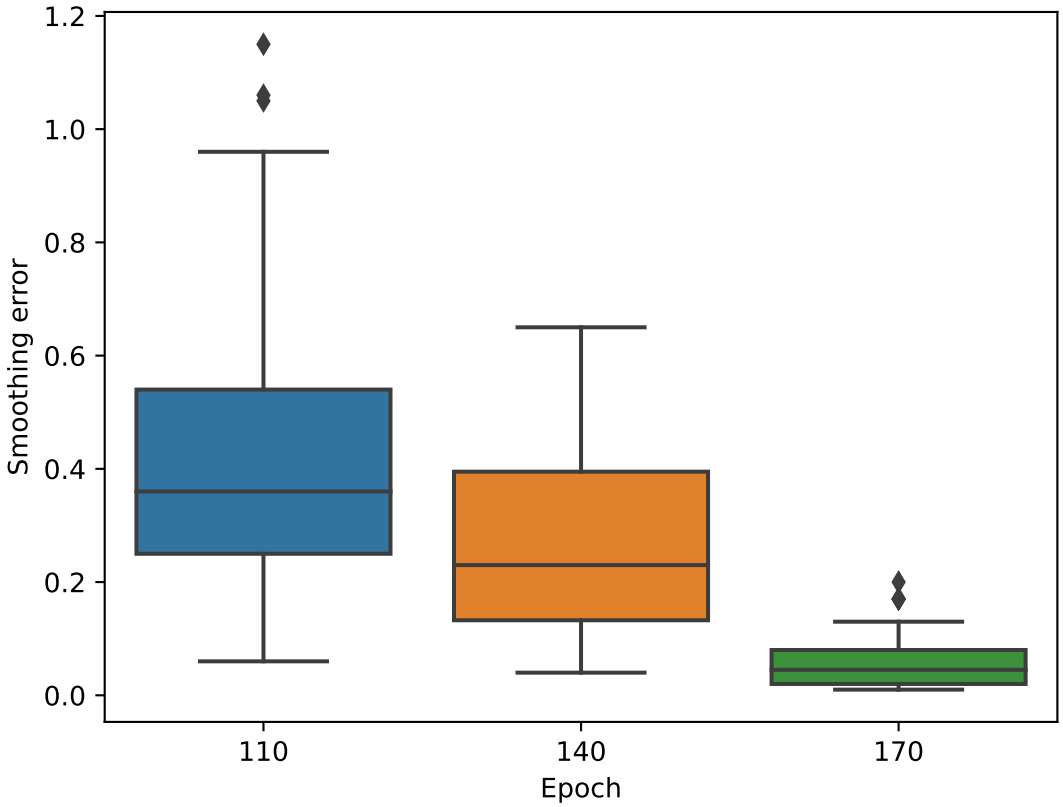


Figure C.1: Smoothing errors $\left|q^{\lambda}_{0:n}h_{0:n} - \phi^{\theta}_{0:n}h_{0:n}\right|$ for $\tilde{h}_k(x_k, x_{k+1}) = x_k$ at $n = 500$, when $\phi^{\theta}_{0:n}$ is given via Kalman smoothing with the true parameters γ and $q^{\lambda}_{0:n}$ is given via Kalman smoothing with parameters λ selected at epochs 110,140 and 170. Each plot is generated from the $J = 50$ sequences $(Y^j_{0:n})_{1\leq j\leq J}$ drawn from p^{γ}

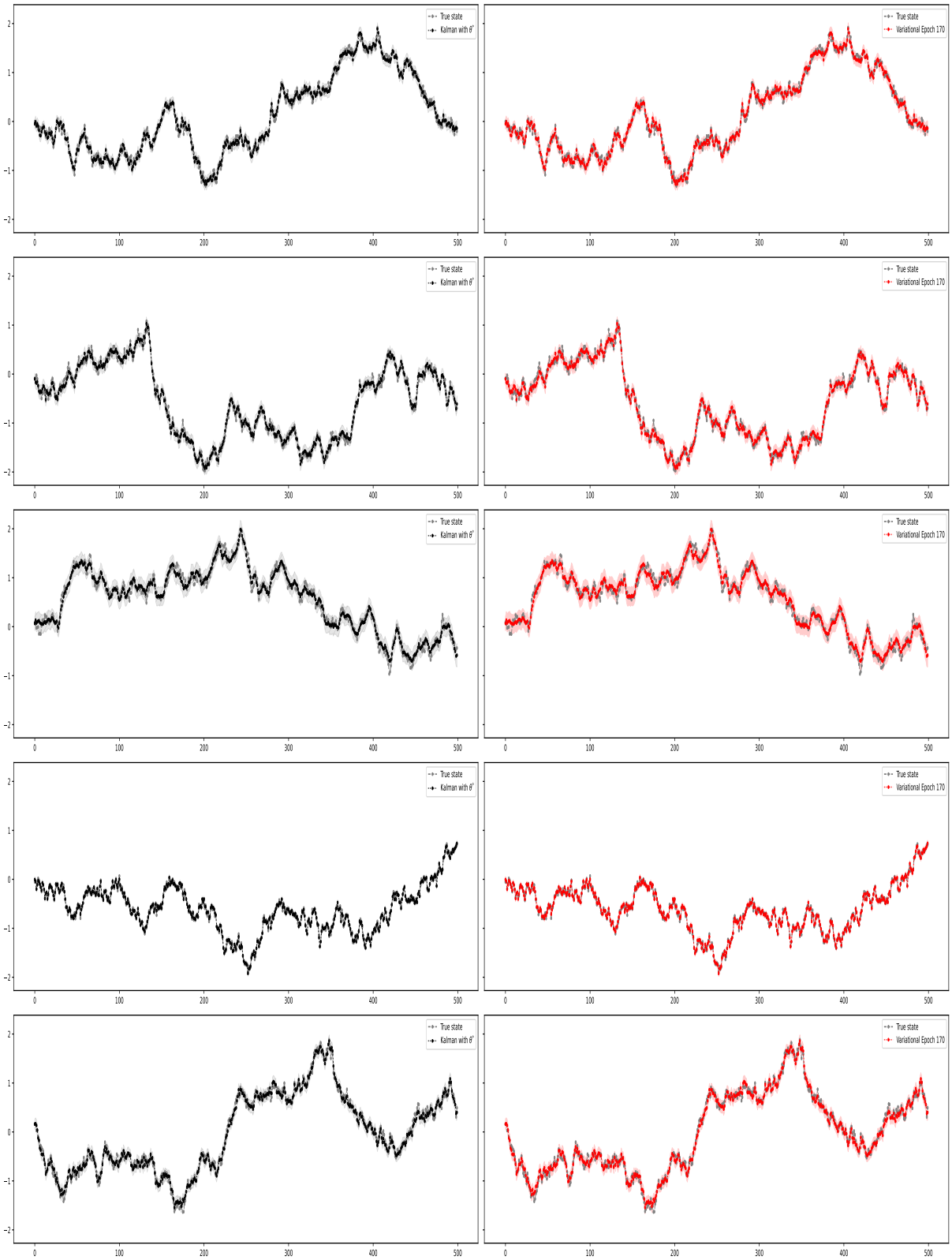


Figure C.2: Example of smoothed states when the dimension of the state space is 5 and the observations is 5. Left column: component-wise (from top to bottom) smoothed states with true parameters γ. Right column: same thing with learnt parameters λ. The dashed fillings are the 95% confidence intervals. The horizontal axis is the time axis.

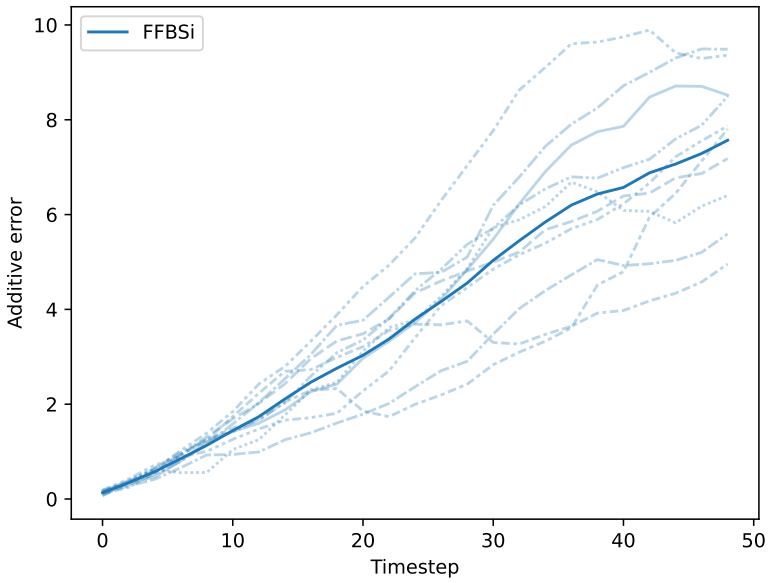


Figure C.3: Smoothing errors $\left|h_{0:n}(x^*_{0:n}) - \phi^\theta_{0:n} h_{0:n}\right|$ for $\tilde{h}_k(x_k, x_{k+1}) = x_k$, where $x^*_{0:n}$ is the true sequence of hidden states and $\phi^\theta_{0:n}$ is obtained by the FFBsi algorithm. All values are normalized by the dimension of the state space. Experiments are produced on 10 independent sequences. The thick solid lines display the mean over the 10 independent replicates for both approaches, shaded lines are single sequences.

C.2.3 Nonlinear models

Here we provide additional details on the experiments of section 5.2.2.

- For the nonlinear emission function d^θ of the data model, we used a single-layer perceptron with a ReLU activation function (which induces non-injectivty on some portions of the support).
- For the *Conjugate Forward* and *Conjugate Backward* methods, the encoder r^λ is a multilayer perceptron (MLP) and a $\tanh$ activation function. The activation function is not applied to the output layer to ensure that the values can exceed values outside the range $[-1, 1]$, being natural parameters of Gaussian distributions. The output of the network is split into two natural parameters η_1 and η_2, the latter being constrained to strictly negative values by applying the softplus function $x \mapsto -\log(1 + e^x)$. We use Xavier initialization for the matrix parameters, and random normal initialisation for the bias parameters.
- For GRU Backward model, H^λ is a Deep GRU as implemented in the Haiku library from the JAX ecosystem.

For the experiments of section 5.2.2, we use small networks with two hidden layers of size 8 (both for r^λ and the GRU in the corresponding models). For the experiments of section 5.2.2, we use configurations similar to that of [Cam+21] for fair comparison, i.e. neural networks with a single hidden layer of size 100.

In Figure C.3, we plot the evolution of the additive error of the FFBSi oracle against the true states.

Appendix D

Contexte, contributions et perspectives en français

D.1 Comptage automatique de macrodéchets à partir de vidéos

Les travaux présentés dans ce manuscrit combinent des idées issues de multiples domaines, allant de l'ingénierie mathématique, en particulier la littérature sur le suivi de multiples objets et la reconnaissance d'objets basée sur l'apprentissage profond, à de nouvelles approches d'inférence séquentielle via des méthodes variationnelles. Les contributions présentées sont cependant motivées par une application spécifique qui a été à l'origine du projet de thèse : une collaboration avec Surfrider Foundation Europe pour étudier et développer de nouvelles solutions de comptage automatisé des macrodéchets déposés sur les berges des rivières françaises. Nous présentons ici directement cette application (en termes non techniques), les problèmes méthodologiques qu'elle sous-tend et comment elle a motivé les directions de recherche de cette thèse.

La pollution par les déchets concerne toutes les régions du globe. Chaque année, près de dix milliards de tonnes de déchets plastiques sont produites, dont 80 échouent dans les décharges ou dans la nature [GJL17a], menaçant notamment tous les océans, mers et environnements aquatiques du monde [Wel20; GS20]. On sait que la pollution plastique impacte déjà plus de 3763 espèces marines dans le monde (voir par exemple [PR23] pour une analyse détaillée) avec un risque de prolifération tout au long de la chaîne alimentaire. Cette accumulation de déchets est le point final du cheminement largement mal compris des déchets, provenant principalement de sources terrestres [Roc+16], cependant les rivières ont été identifiées comme une voie majeure d'introduction de déchets dans les environnements marins [Jam+15]. Par conséquent, des données de terrain sur les rivières et une surveillance sont absolument nécessaires pour évaluer l'impact des mesures pouvant être prises. L'analyse de ces données de terrain au fil du temps est essentielle pour comprendre l'efficacité des actions mises en œuvre telles que le choix d'alternatives zéro déchet au plastique, la conception de nouveaux produits durables ou réutilisables, l'introduction de politiques visant à réduire le suremballage.

Différentes méthodes ont déjà été testées pour surveiller les déchets dans les rivières : collecte et tri des déchets sur les berges [Bru+18], comptage visuel des déchets dérivants à partir de ponts [Gon+21], barrages flottants [Gas+14] et filets [Mor+14]. Toutes sont utiles pour comprendre l'origine et la typologie de la pollution par les déchets, mais sont difficilement

compatibles avec une surveillance à long terme à l'échelle nationale. Les outils de surveillance doivent être fiables, faciles à mettre en place sur différents types de rivières et doivent donner un aperçu de la pollution plastique lors des pics de débit pour aider à localiser les points clés et à fournir des tendances aux décideurs. Des études plus récentes suggèrent que le transport des débris plastiques pourrait être mieux compris en comptant les déchets piégés sur les berges des rivières, fournissant ainsi une bonne indication de la pollution locale par les macrodéchets, en particulier après l'augmentation du débit de la rivière [Emm+19; ES20].

D.1.1 Contexte

Dans ce contexte, Surfrider Foundation Europe a créé le projet *Plastic Origins* dont l'un des objectifs est de développer des solutions efficaces de suivi automatisé du comptage des macroplastiques sur les berges des rivières. Les données obtenues dans le cadre de ce projet (qui sont présentées plus en détail dans le chapitre 3) peuvent être résumées de la façon suivante.

1. Plusieurs milliers d'images de déchets annotées indépendantes, plus précisément des déchets photographiés sur les berges des rivières avec leur position et leur étendue dans l'image identifiés par des boites rectangulaires.

2. Des dizaines de vidéos haute résolution non annotées de berges de rivières contenant des déchets, filmées à partir de caméras portables dans des bateaux en mouvement, d'une durée de quelques secondes à plusieurs minutes.

3. Plusieurs expéditions de collecte de données où les bénévoles sont invités à fournir des estimations visuelles du nombre de déchets sur certaines des sections de rivière couvertes par les séquences vidéo décrites ci-dessus.

Dans la Figure D.1, nous montrons quelques exemples d'ensembles de données annotés d'images de déchets statiques, où des boites englobantes sont superposées pour visualiser les annotations. Dans la figure D.2, nous montrons deux ensembles d'images provenant d'une des vidéos (sur deux sections différentes de l'expédition fluviale associée). De tels exemples illustrent les caractéristiques typiques du cadre imposé par les données.

- Dans les images comme dans les vidéos, les objets à détecter se présentent sous une grande variété de formes et de couleurs. Ils sont capturés sous différents angles et distances. Les arrière-plans, l'éclairage et l'encombrement visuel des scènes varient considérablement.

- Dans les vidéos, les berges des rivières sont filmées à partir d'une caméra qui filme principalement perpendiculairement à la direction du mouvement. La caméra se déplace globalement le long de la rivière, mais le mouvement peut être très non linéaire, par exemple avec des variations de vitesse et des rotations non triviales. Pendant le processus de prise de vue, un objet donné sera visible pendant une durée variable, principalement en fonction des occultations et de la vitesse de la caméra. Plusieurs angles du même objet peuvent être visibles lorsque la caméra se déplace, de sorte que son aspect visuel peut légèrement changer avec le temps.

Figure D.1: 12 exemples du jeu de données d'images annotées

D.1.2 Spécificités du projet

Chez Surfrider, il a été établi très tôt que toutes les campagnes de collecte de données (images statiques annotées et vidéos) seraient menées mais que la solution la plus pratique pour le suivi des déchets était de travailler directement sur du matériel vidéo, car filmer les expéditions fluviales est la solution la plus simple pour suivre régulièrement les déchets et recueillir des

Figure D.2: Deux groupes (un par colonne) de 4 images d'une vidéo

données sur place tout au long de l'année.

Par conséquent, l'accent a été mis sur les solutions capables de prédire automatiquement un nombre total d'éléments d'intérêt visibles dans les vidéos. Cette dernière tâche, que nous appellons *comptage d'objets vidéo* dans le document, se situe à l'intersection des domaines de la *vision par ordinateur* et de l'*analyse de données temporelles*. Cette tâche est particulière pour différentes raisons.

1. Pour une vidéo donnée, chaque objet peut être visible dans plusieurs images mais ne doit être compté qu'une seule fois.

2. L'emplacement des objets individuels n'est pas nécessairement requis dans la prédiction finale.

Le premier aspect rend la tâche de comptage dans des vidéos très différente de celle de comptage dans des images statiques indépendantes. Dans la littérature existante, la réidentification d'objets sur plusieurs images est un sujet central dans la recherche sur le *suivi multi-objets* (en anglais *multi-object tracking MOT*), qui vise à prédire les trajectoires individuelles des objets d'intérêt dans les séquences vidéo, c'est-à-dire détecter et localiser ces objets dans chaque image et en leur attribuant un identifiant cohérent dans le temps.

Mais le deuxième point est progressivement devenu une particularité de ce projet. D'une part, le résultat demandé est plus restreint que pour le MOT traditionnel et la plupart des applications vidéo, car des prédictions précises image par image ne sont pas requises. D'un autre côté, les directions de recherche possibles sont plus larges, car on peut envisager des solutions qui ne s'appuient pas explicitement sur la détection d'objets par image comme quantité intermédiaire pour produire des décomptes vidéo globaux. Dans le paragraphe suivant, nous décrivons succinctement certains autres aspects importants et spécifiques à la tâche à accomplir.

Formats d'annotation. Le contenu vidéo du projet ne comporte aucune forme d'annotation détaillée sur les vidéos, c'est-à-dire que l'on n'a pas accès à des exemples de séquences vidéo avec des objets localisés et identifiés dans chaque image. Les données vidéo sont soit des séquences simples sans annotations, soit des segments vidéo où la vérité terrain est un décompte global pour le segment. En vision par ordinateur basée sur la vidéo, ceci est un exemple de données *faiblement* annotées. Au contraire, l'ensemble de données d'images est densément annoté avec des emplacements d'objets précis, mais les images sont acquises de manière indépendante et ne présentent donc pas les dépendances temporelles des vidéos sur lesquelles la tâche finale doit être effectuée. Par conséquent, plusieurs formes de données sont disponibles, et il n'est pas clair comment les combiner pour créer une solution efficace qui utilise au mieux chacune d'entre elles. [1]

Ressources informatiques. Un autre élément de ce projet concerne les limitations informatiques spécifiées par Surfrider Foundation. Dès le début, il a été annoncé qu'une solution facilement portable serait préférable pour des configurations embarquées avec une puissance de traitement limitée, idéalement une solution qui pourrait fonctionner directement sur les

[1] Dans le chapitre 3, nous proposons un algorithme qui ne nécessite qu'une supervision sous forme d'images annotées indépendantes pour entraîner un détecteur d'objet. Dans les chapitres 4 et 5, nous étudions des méthodes génériques d'inférence dans des modèles séquentiels basés sur des objectifs d'optimisation *non supervisés*.

smartphones utilisés pour filmer les vidéos afin d'éviter d'envoyer les données vers un appareil secondaire. Il a également été suggéré en cours de route que les méthodes permettant de traiter les données à la volée seraient préférables, car le stockage et le traitement de toutes les images simultanément peuvent être fastidieux sur des appareils embarqués. [2]

Variabilité et fiabilité de la vérité terrain. Parmi les décomptes globaux fournis avec les vidéos, une variabilité a été observée lorsqu'il a été demandé à plusieurs personnes d'identifier les déchets sur les mêmes sections de rivière. La figure 1.3 illustre cela à l'aide des décomptes rapportés par 20 bénévoles sur trois emplacements distincts couverts par des vidéos de l'ensemble de données. Cette variabilité dans les estimations de la vérité terrain suggère que la tâche automatisée de comptage des macrodéchets pourrait bénéficier d'estimations d'incertitude ainsi que des décomptes prédits. De telles estimations d'incertitude rendraient en outre la solution finale plus fiable pour la surveillance de la pollution, par exemple en permettant d'écarter les mauvaises prédictions. [3]

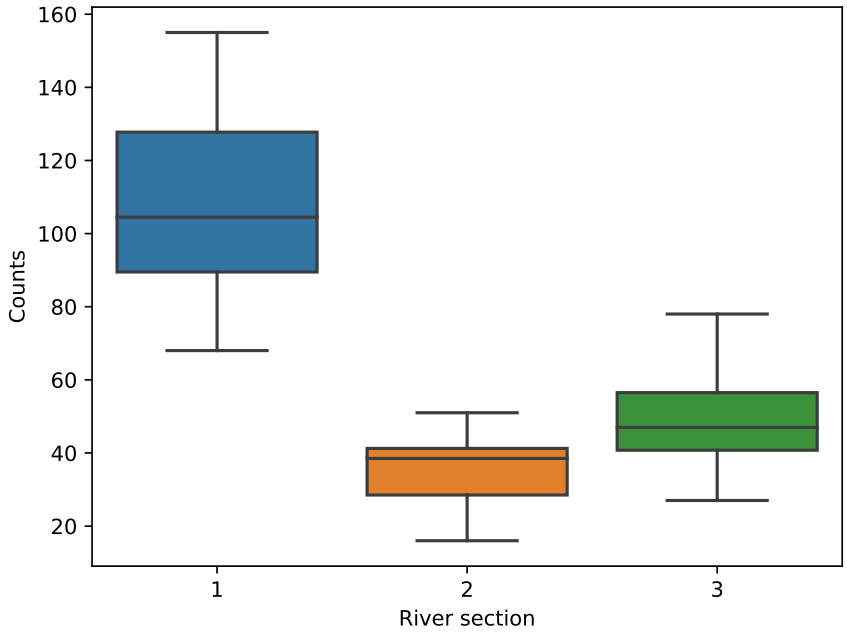


Figure D.3: Variabilité du comptage visuel parmi 20 bénévoles pour trois sections de rivière

Observations en grande dimension. Du point de vue de l'inférence pour les données séquentielles, les observations qui composent les séquences d'intérêt (les vidéos) sont des im-

[2]Dans le chapitre 3, un effort a été mis pour choisir des solutions efficaces issues de la vision par ordinateur et développer des approximations connues pour bien s'adapter à la tâche ciblée. De plus, la plupart des calculs de l'algorithme que nous proposons peuvent être effectués en ligne. Dans le chapitre 4 nous proposons spécifiquement un algorithme en ligne pour construire des approximations génériques dans le contexte variationnel.

[3]Dans le chapitre 3, la solution de suivi que nous proposons tient naturellement compte de l'incertitude dans le mouvement de la vidéo pour générer des décomptes, tandis que dans les chapitres 4 et 5, le formalisme bayésien des solutions que nous étudions peut naturellement être utilisé pour obtenir des intervalles de crédibilité autour des estimations statistiques.

ages en haute résolution. Ainsi, la dimensionnalité des données à chaque pas de temps est de l'ordre de plusieurs millions, ce qui rend impossible l'application des méthodes classiques d'inférence séquentielle directement dans l'espace des vidéos originales. [4]

D.2 Présentation des contributions

Compte tenu des défis pratiques liés au comptage d'objets, et des approches plus générales en inférence variationnelle séquentielle, le travail de thèse est divisé en deux ensembles de contributions.

- Une contribution technique qui aborde directement le comptage d'objets dans des videos contenant des macrodéchets.

- Deux contributions méthodologiques liées à l'inférence variationnelle qui exploitent des idées de SMC pour développer une compréhension théorique des méthodes existantes et améliorer leurs propriétés computationelles.

Comptage vidéo de macrodéchets sur les rives des rivières à l'aide de modèles d'espace d'état et de caméras mobiles, *Mathis Chagneux, Sylvain Le Corff, Pierre Gloaguen, Charles Ollion, Océane Lepâtre, et Antoine Bruge. Publié (avec le code source) dans Computo, 2023.*

Dans le chapitre 3, nous présentons une nouvelle méthode pour compter les macrodéchets dans des vidéos des berges des rivières filmées depuis des caméras embarquées sur des bateaux. Ici, nous nous appuyons sur le suivi multi-objets (MOT) mais nous nous concentrons sur les problèmes clés liés aux comptages erronés et redondants qui surviennent dans les scénarios de faibles performances de détection. Notre système ne nécessite qu'une supervision préalable sous forme d'images indépendantes, et effectue une filtrage bayésien via un modèle d'espace d'état basé sur le flot optique. Nous présentons un nouvel ensemble de données d'images recueillies grâce à une campagne de crowdsourcing et utilisées pour entrainer un détecteur d'objets particulièrement adapté à la tâche de comptage. Dans le cadre de ce travail, des vidéos réalistes capturées par des experts en surveillance de l'eau ont été annotées et utilisées pour l'évaluation. Des améliorations de la qualité du comptage sont démontrées par rapport aux systèmes construits à partir des suiveurs multi-objets de pointe partageant les mêmes capacités de détection. Une décomposition précise des erreurs permet une analyse claire et met en évidence les questions restant ouvertes. Cette première contribution a été menée en étroite collaboration avec Surfrider Foundation Europe, et fournit un outil initial qui a depuis été largement mis en production dans le cadre du projet Plastic Origins, bénéficiant d'un soutien régulier et mis à jour de manière incrémentielle.

Une approche d'échantillonnage *backward* pour le lissage additif variationnel en ligne, *Mathis Chagneux, Pierre Gloaguen, Sylvain Le Corff, Jimmy Olsson. Soumis pour publication dans Transactions on Machine Learning Research (TMLR), 2023.*

[4] Dans les chapitres 5 et 4, nous étudions en profondeur les solutions variationnelles comme solution alternative pour résoudre les problèmes de passage à l'échelle des approximations classiques telles que les Méthodes de Monte Carlo.

Dans le chapitre 4, nous exploitons des idées issues des approches de lissage récursif développées dans la littérature SMC pour obtenir un algorithme en ligne efficace dans le contexte de l'inférence variationnelle *backward*. Dans ce travail, nous proposons une décomposition spécifique des distributions variationnelles qui mime celle de la loi *a posteriori* ciblée et permet de reproduire des schémas d'approximation connus des espérances conditionnelles impliquées dans les récursions. En conséquence, cela élimine le besoin d'approximations fonctionnelles précédemment nécessaires pour le calcul récursif des espérances de lissage de fonctionnelles d'état additives sous des approximations variationnelles. Ensuite, nous proposons une nouvelle décomposition du gradient de la fonction objectif de l'optimisation variationnelle basée sur l'estimateur de la fonction score, ce qui permet l'apprentissage récursif des paramètres variationnels. Numériquement, la qualité des gradients est démontrée par rapport à d'autres estimateurs hors ligne, et la pertinence et l'efficacité de l'approche proposée sont illustrées sur de longues séquences d'observations.

Erreur de lissage additif dans l'inférence variationnelle inverse *backward* pour les modèles d'espace d'état généraux, *Mathis Chagneux, Élisabeth Gassiat, Pierre Gloaguen, Sylvain Le Corff. En révision majeure en vue d'une publication dans Journal of Machine Learning Research (JMLR), 2023.*

Dans le chapitre 5, nous étudions les propriétés théoriques de la décomposition variationnelle *backward* (ou inverse), où nous établissons sous des hypothèses de mélange que l'approximation variationnelle des espérances de fonctionnelles d'état additives induit une erreur qui croît au plus linéairement avec le nombre d'observations. Cette garantie est cohérente avec les bornes supérieures connues pour l'approximation des distributions de lissage en utilisant des méthodes de Monte Carlo standard. Nous illustrons notre résultat théorique avec des solutions variationnelles de pointe basées à la fois sur la paramétrisation inverse et sur des alternatives utilisant d'autres décompositions. Cette étude numérique propose des lignes directrices pour l'inférence variationnelle basée sur les réseaux de neurones dans les modèles à espace d'état.

D.3 Perspective: un cadre unificateur pour la prédiction séquentielle dans les vidéos

Comme mentionné dans la Section 1.2.2, l'une des principales motivations de l'étude de nouvelles solutions pour l'estimation latente en grande dimension est la récente popularité d'approches non supervisées qui se concentrent sur l'apprentissage de représentations expressives des données, à partir desquelles la plupart des tâches de prédiction peuvent être facilement obtenues. Dans la plupart de ces travaux, on suppose généralement que les variables latentes à partir desquelles les observations proviennent se factorisent en composants statistiquement indépendants, et que la capture de cette propriété dite de "désentrelacement" des représentations latentes [Hig+18] est essentielle pour faciliter les applications ultérieures. Sur la Figure 6.2, par exemple, nous présentons une illustration visuelle de certains aspects utiles de ces espaces latents dans le contexte de la découverte multi-objet.

Cependant, jusqu'à présent, la plupart des travaux ont nécessité d'imposer la structure souhaitée, par exemple en introduisant explicitement des contraintes supplémentaires dans la définition des lois a posteriori [Loc+20; Gre+19; Kab+21; Els+22], en ajoutant des termes de

régularisation dans les objectifs MLE [Hig+17], ou en s'appuyant sur des données observées auxiliaires [Kip+22] qui contraignent davantage les problèmes d'inférence. Parallèlement, le problème non supervisé d'identification de signaux indépendants à partir des données à dépendances complexes a largement été formalisé du point de vue de l'analyse en composantes indépendantes (ICA) [HO00]. Dans ce cadre, un thème récurrent est de déterminer dans quelles conditions un ensemble de variables latentes indépendantes peut être récupéré de manière unique sans supervision dans la limite d'une infinité de données. À cet égard, les résultats théoriques [HP98] et les études empiriques récentes [Loc+19] caractérisent essentiellement le problème d'estimation latente dans les modèles de données *non linéaires* comme mal posé dans le contexte d'observations indépendantes, ce qui entrave considérablement l'apprentissage de représentations pour les modèles génératifs réalistes (par exemple, contenant des fonctions complexes tels que les réseaux de neurones profonds) lorsque seuls des ensembles de données d'images indépendantes sont disponibles. À l'inverse, de nouveaux résultats [GLL20; Khe+20] prouvent des propriétés d'identifiabilité dans le contexte de données *dépendantes*, et en particulier de données temporelles (ce qui inclut les vidéos). À partir de ces résultats, des analyses récentes [HKM23] suggèrent que la récupération d'états latents indépendants dans des données non indépendantes en grande dimension peut être réalisée sans aucune supervision, à condition que les dépendances statistiques des approximations des lois a posteriori soient bien spécifiées.

En pratique, la plupart des travaux qui ont tenté de traiter des données structurées sur la base de ces résultats [HH20; Häl+21a] se sont fortement appuyés sur des approximations variationnelles séquentielles, mais principalement via des décompositions similaires à celles de la Section 2.3.2, qui manquent de garanties théoriques développées dans cette thèse et qui peuvent difficilement être utilisées pour de longues séquences. Par conséquent, une perspective importante de cette thèse serait de d'obtenir des solutions similaires en utilisant les décompositions "backward". Dans le contexte du comptage d'objets dans des vidéos, ou plus généralement lorsque l'on vise des quantités globales liées à des séquences entières d'observations $y_{0:t}$, les propriétés d'identifiabilité du cadre séquentiel sont particulièrement attrayantes, car elles suggèrent que la formulation de tâches de prédiction dans les vidéos en tant qu'estimations de statistiques sous les distributions de lissage pourrait être une approche théoriquement justifiée pour éviter de s'appuyer sur des prédictions intermédiaires spécialisées (telles que les estimations ponctuelles de MOT).

Par exemple, étant donné la fiabilité des approximations SVI "backward" pour le lissage additif, on peut imaginer une solution qui estime un décompte d'objets $\hat{N}$ dans une vidéo via

$$\hat{N} = \mathbb{E}_{q^{\lambda}_{0:t}}\left[h_{0:t}(X_{0:t})\right] \approx \mathbb{E}_{\phi^{\theta}_{0:t}}\left[h_{0:t}(X_{0:t})\right] \ ,$$

pour une certaine fonctionnelle d'état additive $h_{0:t}$ qui peut extraire des informations de compte pertinentes à partir des états latents récupérés. S'appuyer sur un tel formalisme serait très intéressant car il sépare essentiellement le problème en deux étapes distinctes.

1. Une étape d'apprentissage générique pour construire $q^{\lambda}_{0:t}$ uniquement à partir de l'optimisation de l'ELBO, où les dépendances dans les données sont capturées sans aucune annotation et indépendamment de la tâche ciblée.

2. La spécification de $h_{0:t}$ en fonction de la tâche en cours et le calcul de $q^{\lambda}_{0:t}h_{0:t}$ avec λ fixé.

En pratique, un tel cadre présente plusieurs avantages. Tout d'abord, par rapport aux méthodes de MOT entièrement supervisées, qui nécessitent des annotations de suivi supplémentaires

pour introduire des informations temporelles dans le processus d'apprentissage, la première étape est entièrement non supervisée et ne nécessite que la spécification correcte des dépendances dans la loi a posteriori. Ensuite, et surtout, la séparation de l'étape d'apprentissage de la représentation de l'étape de prédiction finale peut permettre de s'appuyer sur des annotations beaucoup moins lourdes que dans les configurations d'apprentissage supervisé classiques. En effet, tout comme l'information de localisation est ignorée dans le comptage d'objets basé sur la détection dans les images fixes, les emplacements d'objets prédits dans le comptage d'objets vidéo basé sur le MOT sont utilisés pour l'association temporelle des détections entre images, mais sont ignorés dans le comptage final. Dans le cadre précédent, la cohérence temporelle des prédictions devrait déjà être imposée étant donné que les prédictions sont formulées comme des espérances sous $q^{\lambda}_{0:t}$, et l'on peut donc obtenir une fonctionnelle de "comptage" $h_{0:t}$ qui extrait les comptes à partir des états latents lissés, nécessitant *uniquement* une supervision de comptage.

À titre d'exemple, supposons que nous puissions annoter, pour tous les pas de temps $s \leq t$ dans une vidéo $y_{0:t}$, le nombre d'objets N^+_s apparaissant dans la vidéo à s, mais absents pour $s' < s$. Ensuite, en supposant qu'un objet ne puisse plus être visible après avoir quitté le champ de la caméra (par exemple, la caméra ne recule pas), le nombre total d'objets dans la vidéo est simplement donné par $N = \sum_{s=0}^{t} N^+_s$. En pratique, de telles annotations sont faciles à obtenir, car elles nécessitent seulement de regarder la vidéo et de marquer les images ayant des objets entrants (toutes les autres images reçoivent $N^+_s = 0$), ce qui est beaucoup moins contraignant que d'annoter les emplacements de tous les objets à toutes les images. Un algorithme de comptage peut alors être développé en définissant des fonctionnelles dont les composantes sont des fonctions visant à estimer $(N^+_s)_{s\leq t}$ à partir des représentations latentes de paires d'images consécutives, étant donné l'ensemble de la vidéo. Formellement, on peut définir

$$h^{\gamma}_{0:t} : x_{0:t} \mapsto \sum_{s=1}^{t} \tilde{h}^{\gamma}(x_{s-1}, x_s) \ ,$$

où $\tilde{h}^{\gamma}$ est un réseau de neurones de $\mathsf{X} \times \mathsf{X}$ à $\mathbb{N}$ paramétré par $\gamma \in \Gamma$, où Γ est un espace de paramètres. Pour apprendre γ, on peut choisir une fonction de pénalisation de comptage $\mathcal{C}^{\gamma} : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$ (par exemple, régression de Poisson), et considérer la pénalisation globale $\sum_{s=0}^{t} \mathcal{C}^{\gamma}(\hat{N}^+_s, N^+_s)$ sur la vidéo, où

$$\hat{N}^+_s = \mathbb{E}_{q^{\lambda}_{s-1:s}}[\tilde{h}^{\gamma}(X_{s-1}, X_s)] \ ,$$

pour tous les $s \leq t$. [5] À convergence, une estimation du compte global N serait donnée par $\hat{N} = \mathbb{E}_{q^{\lambda}_{0:t}}[h^{\gamma}_{0:t}]$.

En tant que tel, cette méthodologie est très attrayante car elle ne nécessite pas d'annoter la localisation des objets, inclut des connaissances de *toutes* les images grâce à $q^{\lambda}_{0:t}$, mais fournit toujours un signal d'apprentissage aux pas de temps individuels (et non seulement un compte global). En ce qui concerne ce dernier point, de nombreuses autres options seraient possibles pour renforcer la supervision "locale", par exemple en définissant des composantes qui prédisent la variation (éventuellement négative) du nombre d'objets visibles entre $s-1$ et s, ou en prédisant simplement le nombre total d'objets N_s visibles à tout instant $s \leq t$ et en considérant $N = \sum_{s=1}^{t} \max(0, N_s - N_{s-1})$ comme une définition alternative du compte

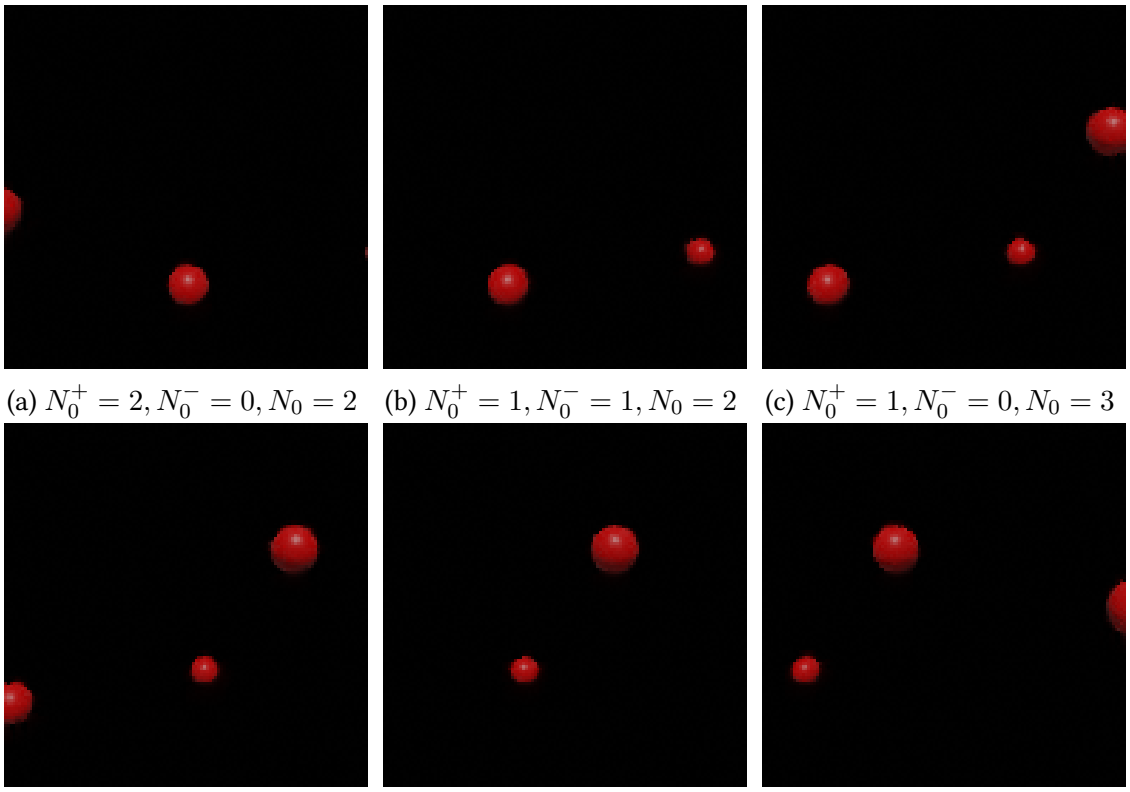
[5] Rappelons que $q^{\lambda}_{s-1:s}$ est la distribution marginale conjointe de $q^{\lambda}_{0:t}$ à $s-1$ et s.

global[6]. À l'inverse, la flexibilité de ce cadre permettrait également, après cette première étape d'entraînement, d'affiner γ avec une annotation de comptage global, c'est-à-dire en utilisant directement $\mathcal{C}^\gamma(\hat{N}, N)$ avec $\hat{N} = \mathbb{E}_{q^\lambda_{0:t}}[h^\gamma_{0:t}]$. Enfin, on peut également envisager de réentraîner λ avec γ fixé, en utilisant la pénalisation de comptage pour affiner la représentation latente en vue d'une meilleure performance. En pratique, en utilisant le Chapitre 4, toutes ces opérations pourraient être effectuées en ligne.

Dans l'ensemble, les avantages attendus de cette nouvelle méthodologie apportent des réponses claires aux limitations du travail mené dans le Chapitre 3. En effet, dans cette contribution, l'axe principal d'amélioration consistait à augmenter les performances du détecteur d'objets grâce à des ensembles de données plus importants d'images individuelles, ce qui nécessite un effort continu de la part de la Fondation Surfrider. De plus, comme mentionné dans l'introduction, le développement d'une solution de comptage plus stable dans le cadre du MOT aurait nécessité des mécanismes de suivi plus sophistiqués, avec de nombreux aspects non directement liés à la tâche de comptage final, et avec des étapes supplémentaires d'ajustement des hyperparamètres. Comparativement, le travail mené dans les Chapitres 4 et 5 ouvre la voie au développement de solutions de comptage plus simples. De plus, elles seraient également plus facilement complétées par des estimations d'incertitude, par exemple en considérant des intervalles de confiance basés sur la variance de la loi a posteriori $q^\lambda_{0:t}$. Bien que cela ne fasse pas partie de ce manuscrit, des expériences sur des vidéos synthétiques d'objets en mouvement (voir Figure D.4) sont en cours pour évaluer la pertinence de ces idées sur des contenus réels à base d'images où une vérité terrain peut être obtenue facilement, et où différents degrés de complexité peuvent être générés.

Enfin, il convient de noter que, étant donné que l'apprentissage de $q^\lambda_{0:t}$ est indépendant du choix de la fonctionnelle utilisée pour la prédiction finale, le cadre décrit plus haut est très modulaire: on peut par exemple apprendre $q^\lambda_{0:t}$ dans une étape préalable, puis effectuer diverses prédictions en utilisant différentes fonctionnelles, à λ fixé. Pour illustrer la pertinence de cet aspect, on peut s'intéresser à une petite partie des données observées dans le cadre du projet Plastic Origins, qui a révélé les limites de la modélisation de la surveillance de la pollution par les macrodéchets simplement comme une tâche de comptage. Par exemple, des situations comme celles illustrées dans la Figure D.5 suggèrent que des approches capables de fournir des prédictions sous d'autres formes (par exemple, la surface de la berge de la rivière couverte de déchets) en fonction des situations présentes sur le terrain constitueraient une perspective de recherche intéressante.

[6]En pratique, le nombre d'objets visibles dans n'importe quelle image peut être obtenu simplement en annotant, en plus de N^+_s, le nombre d'objets N^-_s quittant la vidéo à tous les s.



(a) $N_0^+ = 2, N_0^- = 0, N_0 = 2$ (b) $N_0^+ = 1, N_0^- = 1, N_0 = 2$ (c) $N_0^+ = 1, N_0^- = 0, N_0 = 3$

(d) $N_0^+ = 0, N_0^- = 0, N_0 = 3$ (e) $N_0^+ = 0, N_0^- = 1, N_0 = 2$ (f) $N_0^+ = 1, N_0^- = 0, N_0 = 3$

Figure D.4: Un exemple de vidéo synthétique avec des objets entrant et sortant du champ de la caméra à des vitesses variables.

Figure D.5: Un exemple de situation présentant une très grande densité de déchets. Dans ce contexte, énumérer individuellement les objets semble inadapté pour mesurer le niveau de pollution, par exemple, on peut plutôt estimer la surface de la berge couverte de plastique.

ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

Titre : Poursuite multi-cibles et nouvelles approches variationnelles pour les données séquentielles en grande dimension: application au comptage d'objets dans les vidéos.

Mots clés : Poursuite multi-cibles, inférence variationnelle, données séquentielles, vision par ordinateur, modèles à espace d'état.

Résumé : Dans cette thèse, nous nous intéressons à de nouvelles approaches pour les problèmes de prédiction associés à des données séquentielles en grande dimension, en s'intéressant particulièrement aux cas de données faiblement annotées et à la réduction du coût de calcul. Dans le cadre d'un problème spécifique de comptage de macrodéchets à partir de vidéos, nous développons d'abord une solution robuste basée sur une approche de poursuite multi-cibles, qui combine l'apprentissage profond et l'estimation bayésienne récursive classique. Etant donné les récentes formulations de problèmes similaires sous forme d'estimateurs statistiques dans des modèles à données latentes de grande dimension, nous nous concentrons sur les extensions récentes des méthodes d'inférence variationelles au cadre séquentiel. Dans ce cadre, nous étudions les aspects théoriques et calculatoires de la factorization dite *backward*, qui est une paramétrisation prometteuse pour les approximations variationnelles des distributions de lissage dans les modèles à espaces d'états. En particular, sous ces approximations *backward*, nous obtenons une borne théorique pour l'erreur d'approximation d'espérances de fonctionnelles additives, et nous développons un algorithme efficace pour l'inférence d'état recursive et l'apprentissage en ligne des paramètres variationels. Ces contributions renforcent la pertinence des approches variationelles en tant qu'alternatives au méthodes Monte Carlo pour les données séquentielles, et ouvrent la voie à leur adoption en tant qu'estimateurs génériques pour la prévision non supervisée dans les données temporelles en grande dimension, comme pour le comptage dans les vidéos.

Title : Multi-target tracking and novel sequential variational approaches for high-dimensional sequential data: an application to object counting in videos

Keywords : Multi-object tracking, variational inference, sequential data, computer vision, state-space models.

Abstract : This thesis studies novel approaches for prediction in high-dimensional sequential data, with a particular focus on scalability and weakly annotated settings. Motivated by the specific task of macrolitter counting in videos, we first derive a robust solution based on the multi-target tracking methodology, which combines deep learning and classical recursive Bayesian estimation. Then, we focus on the recent extensions of variational inference methods to the sequential setting, motivated by recent formulations of multi-target problems as statistical estimates in high-dimensional latent data models. Here, we study theoretical and computational aspects of the so-called backward factorization as a promising parameterization for variational approximations of the smoothing distributions in general state-space models. In particular, under such backward variational approximations, we derive a theoretical bound for the approximation error when considering expectations of additive state functionals, and develop an efficient algorithm for recursive latent estimation and online learning of the variational parameters. These contributions strengthen the relevance of variational approaches as alternatives to Monte Carlo methods in sequential settings, and pave the way to their adoption as generic solutions for unsupervised prediction in high-dimensional temporal data, such as for video object counting.

Institut Polytechnique de Paris
91120 Palaiseau, France