ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

MEMORY-BASED APPROACHES TO PROBLEMS IN PROBABILISTIC MODELING

M.Sc. THESIS

Abdullah AKGÜL

Department of Computer Engineering

Computer Engineering Programme

OCTOBER 2022

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

MEMORY-BASED APPROACHES TO PROBLEMS IN PROBABILISTIC MODELING

M.Sc. THESIS

Abdullah AKGÜL
(504201504)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Prof. Dr. Gözde ÜNAL

OCTOBER 2022

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

OLASILIKSAL MODELLEME PROBLEMLERİ İÇİN HAFIZA TABANLI YAKLAŞIMLAR

YÜKSEK LİSANS TEZİ

Abdullah AKGÜL
(504201504)

Bilgisayar Mühendisligi Anabilim Dalı

Bilgisayar Mühendisligi Programı

Tez Danışmanı: Prof. Dr. Gözde ÜNAL

EKİM 2022

Abdullah AKGÜL, a M.Sc. student of ITU Graduate School student ID 504201504 successfully defended the thesis entitled "MEMORY-BASED APPROACHES TO PROBLEMS IN PROBABILISTIC MODELING", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Gözde ÜNAL** Istanbul Technical University	.....
Jury Members :	**Assoc. Prof. Dr. Melih KANDEMİR** University of Southern Denmark	.....
	Assoc. Prof. Dr. Yusuf YASLAN İstanbul Technical University	.....
		.....

Date of Submission : **11 October 2022**
Date of Defense : **25 October 2022**

To my family,

FOREWORD

First, I would like to thank my advisor Prof. Dr. Gözde ÜNAL and Assoc. Prof. Melih Kandemir due to their support, guidance, and collaboration thought this thesis. They always inspire me to be a decent researcher and academician. I want to express my gratitude to my family; my mother Ayşe, my father Durmuş Ali, and my sisters Ayşe Sümeyye, and Erva. They always support me and are there for me. I would like to thank ITU Vision Laboratory and my friends.

I am supported by the 2210A-National Scholarship Programme for MSc Students 2020/2 from the Scientific and Technological Research Council of Turkey (TÜBİTAK) and I am grateful for that.

October 2022

Abdullah AKGÜL
Computer Engineer

TABLE OF CONTENTS

	Page
FOREWORD	**ix**
TABLE OF CONTENTS	**xii**
ABBREVIATIONS	**xiii**
LIST OF TABLES	**xv**
LIST OF FIGURES	**xviii**
SUMMARY	**xix**
ÖZET	**xxi**
1. INTRODUCTION	**1**
1.1 Contributions	2
2. EVIDENTIAL TURING PROCESSES	**5**
2.1 Introduction to Evidential Turing Processes	5
2.2 Problem Statement: Total Calibration	6
2.3 Uncertainty Quantification with Parametric and Evidential Bayesian Models	8
2.4 Complete Bayesian Models: Best of Both Worlds	10
2.5 The Evidential Turing Process: An Effective CBM Realization	12
2.6 Most Important Special Cases of the Evidential Turing Process	13
2.7 Case Study: Classification with Evidential Turing Processes	16
2.8 Experiments	16
2.9 Broad Impect, Limitations and Ethical Concerns	18
2.10 Appendix	19
2.10.1 Further analytical results and derivations	19
2.10.1.1 Kullback-Leibler between two Dirichlet distributions	19
2.10.1.2 The Evidential Deep Learning objective	19
2.10.1.3 Evidential Deep Learning as a latent variable model	20
2.10.1.4 Posterior predictive	21
2.10.2 Experimental Details	21
2.10.2.1 Synthetic 1d two-class classification	21
2.10.2.2 Iris 2d three-class classification	22
2.10.2.3 Experiments on real data	23
3. CONTINUAL LEARNING OF MULTI-MODAL DYNAMICS WITH EXTERNAL MEMORY	**27**
3.1 Introduction to Continual Learning of Multi-Modal Dynamics with External Memory	27
3.2 Problem Setup: Continual Multi-Modal Dynamics Learning	28
3.3 Novel Baseline: Variational Continual Learning (VCL) for Bayesian State-Space Models (BSSM)	30
3.4 Target Model: The Continual Dynamic Dirichlet Process	32
3.5 Related Work	35
3.6 Experiments	37
3.6.1 Data sets	39
3.6.1.1 Synthetic data sets	39
3.6.1.2 Real-world data sets	40
3.6.2 Hyper-parameter selection	40
3.6.2.1 Hyper-parameters on data	41
3.6.2.2 Context length	42
3.6.2.3 Hyper-parameters on architectures	42
3.6.2.4 Hyper-parameters on experimental pipeline	44
3.6.3 Results and ablation study	44
3.7 Broad Impact, Limitations and Ethical Concerns	46
4. CONCLUSION	**49**
4.1 Future Directions	49

REFERENCES **51**
CURRICULUM VITAE **57**

ABBREVIATIONS

ANP : Attentive Neural Process
AUROC : Area Under ROC Curve
BNN : Bayesian Neural Network
BSSM : Bayesian State-Space Model
CBM : Complete Bayesian Model
CDDP : Continual Dynamic Dirichlet Process
CLEVA : Continual Learning EValuation Assessment
CL : Continual Learning
DP : Dirichlet Process
EBM : Evidential Bayesian Model
ECE : Expected Calibration Error
EDL : Evidential Deep Learning
ELBO : Evidence Lower BOund
ENP : Evidential Neural Process
ETP : Evidential Turing Process
EWC : Elastic Weight Consolidation
FMNIST : Fashion MNIST
KL : Kullback-Leibler
LSTM : Long Short-Term Memory
MLP : MultiLayer Perceptron
NLL : Negative Log-Likelihood
NMSE : Normalized Mean Squared Error
NP : Neural Processes
NTM : Neural Turing Machine
PBM : Parametric Bayesian Model
RNN : Recurrent Neural Network
SI : Synaptic Intelligence
SSM : State-Space Model
VCL : Variational Continual Learning

LIST OF TABLES

Page

Table 2.1 : **Ablation Table.** Deactivating the components of ETP one at a time equates it to state-of-the-art methods. We curate ENP as a surrogate for the ANP of [1], which requires test-time context, and an improved NP variant with reduced estimator variance thanks to π. **15**

Table 2.2 : Quantitative results on five data sets showing mean $\pm$ standard deviation across 10 repetitions. Best performing models that overlap within three standard deviations are highlighted in bold. **17**

Table 2.3 : The neural network architectures used for the FMNIST, CIFAR10, and SVHN data set with ReLU activations between the layers. **25**

Table 2.4 : Quantitative results of models on the FMNIST-C, CIFAR10-C and SVHN-C test data set using the LeNet5. The table below reports mean $\pm$ three standard deviations. **26**

Table 3.1 : Summary of data sets. **42**

Table 3.2 : The summary of our main results given as the Area Under Curve (AUC) plotting the change of two performance metrics *NMSE* and *NLL* as a function of the number of learned tasks. The reported numbers are mean $\pm$ standard error over 10 repetitions. The detailed results are provided in Table 3.3. Our CDDP outperforms VCL-BSSM in nearly all cases indicating the benefit of maintaining an external memory of mode descriptors in CL, the central hypothesis of this work. **43**

Table 3.3 : Quantitative results of models on five data sets. The reported numbers are mean $\pm$ standard error over 10 repetitions. The reported scores are taken after training a task and evaluated along with previously seen tasks. **44**

Table 3.4 : Ablation study results given as the Area Under Curve (AUC) on the Sine Waves data set. The reported numbers are mean $\pm$ standard error over 5 repetitions. **46**

LIST OF FIGURES

Page

Figure 2.1 : Plate diagrams of three main Bayesian modeling approaches. Shaded nodes are observed and unshaded ones are latent. The solid direct arrows denote conditional dependencies in the true model, while the bent dashed arrows denote amortization in variational inference. **5**

Figure 2.2 : ETP Plate diagram. The essential design choice is how $p(\pi|Z,w,x)$ is constructed thanks to an external memory M that can be queried with respect to any input x without needing to store context data at test time. The dashed arrow indicates that Z can condition on a context $\mathscr{D}_C$ by updating its parameters M with an explicit function r. **14**

Figure 2.3 : The ETP training routine. **14**

Figure 2.4 : Results averaged across 19 types of corruption (e.g. motion blur, fog, pixelation) applied on the test splits of Fashion MNIST (FMNIST), CIFAR10, and SVHN at five severity levels. **17**

Figure 2.5 : **1D Classification Task.** The upper plot shows the underlying distributions of each of the two classes, as well as the observed data. The lower shows the regularizing evidence the generative model places on each of the two classes depending on location in space, that is, mean $\pm$ one standard deviation over ten samples from $p(Z|M)$. **22**

Figure 2.6 : **2D Classification Task.** The three upper plots visualize the regularizing evidence that the model places on the location in space as the average over ten samples from $p(Z|M)$. The color scales are different across the three plots and vary in overall intensity. The lower plot visualizes a horizontal cut through the middle of these plots, putting them on a common scale. The solid lines in each color indicate the mean memory evidence, and the corresponding shaded areas indicate one standard deviation. Around the separable blue class, the memory strongly emphasizes the correct class with large confidence and suppresses the other two classes depicted in orange and green. While always preferring the correct class, the memory regularizes against overconfidence around the overlapping orange and green classes. **23**

Figure 3.1 : We study CL of dynamical systems, each operate on a non-overlapping set of multiple modes unknown to the learner at each task. We find out that inferring mode descriptors from an observed sequence snipped $(y_{1:C})$, storing them in an external neural episodic memory, retrieving them in the subsequent tasks, and feeding them into the state transition kernel as control input brings improved CL performance. We encourage efficient use of memory space by placing a DP prior on the effective count of encountered modes. **28**

Figure 3.2 : *CDDP Plate Diagram* . Shaded nodes are observed, unshaded nodes are latent, diamonds are deterministic and black nodes are constant. Solid arrows denote the conditional dependencies of the true model and dashed arrows denote the same for the approximate posterior. The first *C* observations, called *the context*, amortize the inference of the initial hidden state, and also determine the content and addressing of the mode embeddings in the external memory. .. **36**

Figure 3.3 : CLEVA Compass. **38**

Figure 3.4 : Sample sequences collected from different modes of the dynamical systems from synthetic data sets. Colors correspond to different modes. For each data set, modes live in the same dynamical system but differ in the free parameters that govern the dynamics. For instance, the Sine Waves differ in magnitude and frequency. **39**

Figure 3.5 : Predictions of CDDP (top row) and VCL-BSSM (middle row) on sample sequences from four tasks of the Character Trajectories data set after CL is over. Bottom row shows the corresponding ground-truth trajectories. The first one-third of each trajectory is considered as context (plotted in red), and the rest is predicted by the models (plotted in blue). Our CDDP remembers the first task much more precisely than VCL-BSSM (first and second columns) after adapting itself accurately to the last task (third and fourth columns). This is achieved by virtue of its external episodic memory. **45**

MEMORY-BASED APPROACHES TO PROBLEMS IN PROBABILISTIC MODELING

SUMMARY

Deep neural networks are an accepted solution for many problems in deep learning; however, the application of deep neural networks to safety-critical areas such as health care is still a hot research topic. To employ deep neural networks in such fields, they are expected to fit the in-domain data set, provide calibrated predictions on problematic regions of the target domain, and separate the out-of-domain queries. Even though these expectancies are studied extensively, these studies are highly fragmented. Therefore, there is no model that is able to fit these requirements simultaneously.

Continual Learning (CL) is a framework that aims to learn numerous tasks in a sequential way. The excellent CL method should adapt to new tasks perfectly without forgetting previous tasks. However, neural networks suffer from catastrophic forgetting which is a performance drop on previously learned tasks caused by the newly learned task. Yet, to get intelligent systems capable of adapting to environmental change, CL is crucial. Because of this, CL is a hot topic but the research on CL is mainly on image classification tasks and there is limited work on time sequence classification tasks. Yet, there is no work on multi-modal dynamics modeling.

In this thesis, we employ an external memory to deal with problems in probabilistic modeling. Our solutions for these problems can be summarized as follows:

i) Evidential Turing Processes (ETP): First, we define total calibration for the first time. After investigating two Bayesian paradigms which are the Bayesian model, and the Evidential Bayesian Model, we introduce the Complete Bayesian Model (CBM) which is a unification of those two paradigms. We develop ETP as an instance of CBMs with neural episodic memory.

We build a pipeline to evaluate the models' performance for total calibration. We compare our solution, the ETP member of CBMs, with state-of-the-art members of other paradigms, and we also provide an ablation study. We investigate the models' performance under five real-world data sets including one time-series classification, and four image classification tasks. Furthermore, we tested the models in the corrupted versions of different data sets. We use four different metrics that are test error as prediction accuracy, Expected Calibration Error as in-domain calibration score, Negative Log-Likelihood (NLL) as model fit, and area under the ROC curve as out-of-domain detection score. We report that only the ETP can excel in all three aspects of total calibration simultaneously.

ii) Continual Dynamic Dirichlet Process (CDDP) for Continual Learning of Multi-modal Dynamics: We introduce a new problem which is CL of multi-modal

dynamics. Since the problem is novel, we create a baseline from the existing ones. For this new problem, we introduce a novel solution that employs an external memory to transfer knowledge between tasks.

We curate a pipeline for this newly introduced problem, and in the pipeline new tasks are coming sequentially and each task has a certain number of different mode samples. Differences in task order may cause different results in CL setups; therefore, we change the task order for each replication. We also generate synthetic data sets and adapt time-series classification data sets to evaluate models' performance in the problem. We compare models' performance with Normalized Mean Squared Error as a measure of prediction accuracy and NLL as a measure of Bayesian model fit that quantifies uncertainty. We reveal that our approach, CDDP, compares favorably to the established parameter transfer approach in CL of multi-modal dynamical systems.

To sum up, in this thesis, by experiments we show that external memory architecture can be used for both calibrations of neural networks to use in safety-critical areas and CL of multi-modal dynamics.

OLASILIKSAL MODELLEME PROBLEMLERİ İÇİN HAFIZA TABANLI YAKLAŞIMLAR

ÖZET

Derin sinir ağları, derin öğrenmedeki birçok problem için kabul edilen bir çözümdür; ancak, derin sinir ağlarının sağlık hizmetleri gibi güvenlik açısından kritik alanlara uygulanması hala çözülmemiş bir araştırma konusudur. Bu tür alanlarda derin sinir ağlarını kullanmak için, etki alanı içi (hedef) veri kümesine uymaları, hedef etki alanının (hedef) sorunlu bölgeleri (karar vermenin zor olduğu alanlar) üzerinde kalibre edilmiş tahminler sağlamaları ve etki alanı dışındaki sorguları etki alanı içi sorgularından ayırmaları beklenir. Bu beklentiler kapsamlı bir şekilde çalışılsa da, bu çalışmalar oldukça parçalıdır yani bu alanlar ayrı ayrı çalışılmış fakat birlikte incelenmemiştir. Bu nedenle, henüz, bu gereksinimleri aynı anda karşılayabilecek bir model yoktur.

Sürekli öğrenme, çok sayıda görevi sıralı bir şekilde öğrenmeyi amaçlayan bir sistemdir. Mükemmel sürekli öğrenme yöntemi, önceki görevleri unutmadan yeni görevlere mükemmel bir şekilde uyum sağlamalıdır. Yine de, sinir ağları, yeni öğrenilen görevin neden olduğu daha önce öğrenilen görevlerde performans düşüşü olan yıkıcı unutmadan muzdariptir. Bununla birlikte, çevresel değişime uyum sağlayabilen akıllı sistemler elde etmek için sürekli öğrenme çok önemlidir ve gereklidir. Bu nedenle, sürekli öğrenme önemli bir araşturma konusudur, fakat sürekli öğrenme üzerine yapılan araştırmalar temel olarak görüntü sınıflandırma görevleri üzerinedir ayrıca zaman dizisi sınıflandırma görevleri üzerinde sınırlı çalışma vardır. Ancak, henüz, çok modlu dinamik modelleme konusunda herhangi bir çalışma bulunmamaktadır.

Bu tez bünyesinde, olasılıksal modellemedeki problemlerle çözebilmek için harici bir bellek kullanımının etkilerini incelenmiştir. Bu sorunlara yönelik çözümlerimizi şu şekilde özetlenebilir:

i) Kanıt Turing Süreçleri: Öncelikle, toplam kalibrasyonu ilk kez tanımlıyoruz. Bayes modeli ve kanıtsal Bayes modeli olmak üzere iki Bayes paradigmasını inceledikten sonra, bu iki paradigmanın birleşimi olan tam Bayes modelini tanıtılmıştır. Nöral epizodik belleğe sahip tam Bayes modellerinin bir örneği olarak Kanıtsal Turing Süreçleri geliştirilmiştir.

Modellerin performansını toplam kalibrasyon üzerinden değerlendirmek için bir deney düzeneği oluşturulmuştur. Tam Bayes modellerinin üyesi olan çözümümüzü, Kanıtsal Turing Süreçleri, diğer paradigmaların en iyi sonuçlarını veren üyeleriyle karşılaştırılmıştır ve ayrıca çözümümüzün kısımlarının etkilerinin daha iyi anlaşılabilmesi için bir ablasyon çalışması yapışmıştır. Modellerin

performansını, bir zaman serisi sınıflandırması ve dört görüntü sınıflandırma görevi dahil olmak üzere beş gerçek dünya veri kümeleri altında araştırılmıştır. Ayrıca modellerin üzerinde oynanmış (bozulmuş) veri kümeleri üzerindeki performansları da incelenmiştir. Zaman serisi sınıflandırma veri kümesi olarak IMDB duygu sınıflandırma veri kümesini, görüntü sınıflandırma veri kümeleri olarak tek kanallı siyah-beyaz bir veri kümesi olan Fashion MNIST (FMNIST) veri kümesini, üç kanallı renkli resimler içeren veri kümeleri olan CIFAR10 ve SVHN veri kümeleri kullanılmıştır. Görüntü üzerine 19 farklı bozulum uygulayarak bozulmuş bir veri kümesi olan CIFAR10-C veri kümesi kullanılmıştır ve aynı sistemi uygulayarak bozulmuş FMNIST-C ve SVHN-C veri kümeleri oluşturulup kullanılmıştır. Tahmin doğruluğu olarak test hatası, etki alanı içi (hedef) kalibrasyon puanı olarak Beklenen Kalibrasyon Hatası , model uyumu olarak Negatif Olabilirlik Olasılığı ve etki alanı dışı örneklerinin tespitinin ölçülmesi için ROC Eğrisi Altında Alan olmak üzere dört farklı metrik kullanılmıştır. Yalnızca bizim önerimiz olan Kanıtsal Turing Sürecinin toplam kalibrasyonun her üç yönünde de aynı anda iyi sonuçlar verebildiği bildirilmiştir.

ii) Multi modlu Dinamiklerin Sürekli Öğrenimi için Sürekli Dinamik Dirichlet Süreci: Öncelikle multi modlu dinamiklerin sürekli öğrenilmesi olan yeni bir problem ortaya konulmuştur. Bu problem yeni olduğu için, mevcut ve bu probleme adapte edilebilecek olan sürekli öğrenme metodlarından bir temel oluşturulmuştur. Ayrıca bu yeni problem için, görevler arasındaki bilgi aktarımını harici bir bellek üzerinden sağlayan yeni bir çözüm sunulmuştur.

Bu yeni tanıtılan problem için bir deney düzeneği tasalanmıştır. Ayrıca bu deney düzeneğinde sırayla yeni görevler gelmesi ve her görev için belirli sayıda farklı mod örneği bulunması sağlanmıştır. Sürekli öğrenme deney düzeneklerinde görev sıralamasındaki bu farklılıklar modellerin öğrenmesine etki edebileceği için farklı sonuçlara sebep olabilir. Bu sebeple, her deney örneği için görev sırasını rastgele olmak üzere değiştirilmiştir. Ayrıca, modellerin bu problemdeki performansını değerlendirmek için üç farklı ve zorlu sentetik veri kümeleri oluşturuyoruz. Sentetik veri kümelerinin haricinde zaman serisi sınıflandırma veri kümelerini bu problem için adapte edilmiştir. Sentetik veri kümeleri bir boyutlu olmak üzere sinüs dalgaları dinamik sistemi, iki boyutlu olan av-avcı denklemleri olarak da bilinen Lotka-Volterra dinamik sistemi ve üç boyutlu olan lineer olmayan kaotik bir dinamik sistem olan Lorenz çekerleri üretilmiştir. Brezilya işaret dilinin oluşturduğu Libras veri kümesi ve tablet üzerinde farklı karakterlerin çizimlerinden oluşan karakter yörüngeleri veri kümeleri bu yeni probleme adapte edilmiştir. Modellerin performansını, tahmin doğruluğunun bir ölçüsü olarak Normalleştirilmiş Ortalama Kareler Hatası ve belirsizliği ölçen bir Bayes modeli uyumu ölçüsü olarak Negatif Olabilirlik Olasılığı ile değerlendirilmiştir. Bizim çözümümüz olan Sürekli Dinamik Dirichlet Süreci'nin, multi modlu dinamik sistemlerin sürekli öğreniminde yerleşik parametre transferi yaklaşımıyla olumlu bir şekilde karşılaştırıldığını bildirilmiştir.

Özetlemek gerekirse, bu tez kapsamında, deneyler ve detaylı ablasyonlar ile, harici bellek mimarisinin hem güvenlik açısından kritik alanlarda kullanılacak sinir

ağlarının kalibrasyonları hem de multi modlu dinamiklerin sürekli öğrenilmesi için kullanılabileceğini gösterilmiştir.

1. INTRODUCTION

Deep neural networks have given outstanding performance in many real-world problems such as classification [2,3], object detection [4,5], and dynamics modeling [6,7]. However, deep neural networks suffer from two important problems which are applicability in safety-critical areas and catastrophic forgetting in Continual Learning (CL). In this thesis, we have addressed these problems with methods based on external memory architectures that have been proven so far for different fields like machine learning.

Firstly, the applicability of deep neural networks to safety-critical use cases such as autonomous driving or medical diagnostics is an active matter of fundamental research [8]. Key challenges in the development of total safety in deep learning are at least three-fold: i) evaluation of model fit, ii) risk assessment in difficult regions of the target domain (e.g. class overlap) based on which safety-preserving fallback functions can be deployed, and iii) rejection of inputs that do not belong to the target domain, such as an image of a cat presented to a character recognition system. The attention of the machine learning community to each of these critical safety elements is in steady increase [9]–[11]. However, the developments often follow isolated directions and the proposed solutions are largely fragmented. Calibration of neural network probabilities focuses primarily on post-hoc adjustment algorithms trained on validation sets from in-domain data [10], ignoring the out-of-domain detection and model fit evaluation aspects. On the other hand, recent advances in out-of-domain detection build on strong penalization of divergence from the probability mass observed in the target domain [12,13], distorting the quality of class probability scores. Such fragmentation of best practices not only hinders their accessibility by real-world applications but also complicates the scientific inquiry of the underlying reasons behind the sub-optimality of neural network uncertainties.

Secondly, CL aims to develop a versatile model that is capable of solving multiple prediction tasks which are presented to the model one task at a time. The model is then expected to learn the latest task as accurately as possible while preserving its excellence at the previous ones. Performance drop caused by the newly learned task is called *catastrophic forgetting*. CL is essential for the development of intelligent agents that can adapt to new environmental conditions not encountered during training. For instance, an autonomous driving controller may improve its policy based on new experiences collected during its customer-side lifetime. There exist a solid body of work that adopts parameter transfer across tasks as the key element of task memorization [14]–[17]. There also appear preliminary studies on building attentive memories to capture tasks [18,19]. Principles that yield memory mechanisms optimal for CL are yet to be discovered. There has been prior work that focuses on CL for Recurrent Neural Networks (RNN) [20] but either classification or instance forecasting for time-series. Complementarily, we study CL in the context of dynamical system identification.

For dynamical system identification and modeling dynamics, Probabilistic State-Space Models (SSM) are the gold standard methodology. SSMs find widespread applicability to forecast socio-economically impactful quantities such as weather, currency exchange, equity prices, and sales trends. SSM research gains significance also in robot learning parallel to the growing interest in model-based reinforcement learning [21]–[23]. SSMs also set the fundamentals of stochastic optimal control [24], enhancing the impact potential of their thorough investigation even further.

1.1 Contributions

In this thesis, we used external memory architecture in probabilistic modeling problems. We summarize our contributions as follows:

For Evidential Turing Process (ETP): We define the first formal definition of total calibration. Then we introduce the Complete Bayesian Model (CBM) and develop the ETP as an optimal realization of CBMs. This chapter is presented at ICLR 2022 as a conference paper [25].

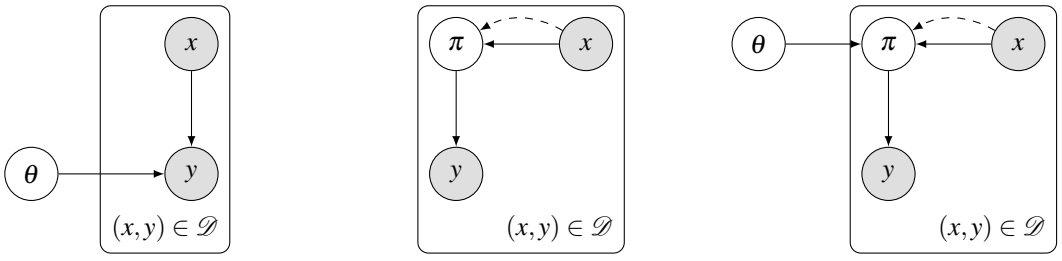
For Continual Dynamic Dirichlet Process (CDDP): We introduce CL of multi-modal dynamical systems; furthermore, we curate a competitive baseline for CL of multi-modal dynamical systems. Then we introduce a novel solution with external memory for CL of multi-modal dynamical systems. This chapter is available at arXiv as a preprint paper [26].

Our experiments demonstrate that external memory architecture can be used for both calibration of neural networks and CL of multi-modal dynamics.

2. EVIDENTIAL TURING PROCESSES

2.1 Introduction to Evidential Turing Processes

We aim to identify the guiding principles for Bayesian modeling that could deliver the most complete set of uncertainty quantification capabilities in one single model. We first characterize Bayesian models with respect to the relationship of their global and local variables: (a) *Parametric Bayesian Models (PBM)* comprise a likelihood function that maps inputs to outputs via probabilistic global parameters, (b) *Evidential Bayesian Models (EBM)* apply an uninformative prior distribution on the parameters of the likelihood function and infer the prior hyperparameters by amortizing on an input observation. Investigating the decomposition of the predictive distribution variance, we find out that the uncertainty characteristics of these two approaches exhibit complementary strengths. We introduce a third approach that employs a prior on the likelihood parameters that both conditions on the input observations on the true model and amortizes on them during inference. The mapping from the input observations to the prior hyperparameters is governed by a random set of global parameters. We refer to the resulting approach as a (c) *CBM*, since its predictive distribution variance combines the favorable properties of Parametric and EBMs. Figure 2.1 gives an overview of the relationship between these three Bayesian modeling approaches.



Parametric Bayesian Model Evidential Bayesian Model Complete Bayesian Model

Figure 2.1 : Plate diagrams of three main Bayesian modeling approaches. Shaded nodes are observed and unshaded ones are latent. The solid direct arrows denote conditional dependencies in the true model, while the bent dashed arrows denote amortization in variational inference.

The advantages of the CBMs come with the challenge of choosing an input-dependent prior. We introduce a new stochastic process construction that addresses this problem by accumulating observations from a context set during training time into the parameters of a global hyperprior variable by an explicitly designed update rule. We design the input-specific prior on the likelihood parameters by conditioning it on both the hyperprior and the input observation. As conditioning via explicit parameter updates amounts to maintaining an external memory that can be queried by the prior distribution, we refer to the eventual stochastic process as a *Turing Process* with inspiration from the earlier work on Neural Turing Machines (NTM) [27] and NP [18]. We arrive at our target Bayesian design that is equipped with the complete set of uncertainty quantification capabilities by incorporating the Turing Process design into a CBM. We refer to the resulting approach as an *ETP*. We observe on five real-world classification tasks that the ETP is the only model that excels simultaneously at model fit, class overlap quantification, and out-of-domain detection.

2.2 Problem Statement: Total Calibration

Consider the following data generating process with K modes

$$y|\pi \sim \mathscr{C}at(y|\pi), \qquad x|y \sim \sum_{k=1}^{K} \mathbb{I}_{y=k} p(x|y=k), \tag{2.1}$$

where $\mathscr{C}at(y|\cdot)$ is a categorical distribution, $\mathbb{I}_u$ is the indicator function for predicate u, and $p(x|y)$ is the class-conditional density function on input observation x. For any prior class distribution π, the classification problem can be cast as identifying the class probabilities for observed patterns $Pr[y|x,\pi] = \mathscr{C}at(y|f^{\pi}_{true}(x))$ where $f^{\pi}_{true}(x)$ is a $K-$dimensional vector with the k-th element

$$f^{\pi,k}_{true}(x) = \pi_k p(x|y=k) \Big/ \sum_{\kappa=1}^{K} \pi_\kappa p(x|y=\kappa). \tag{2.2}$$

In many real-world cases f^{π}_{true} is not known and should be identified from a hypothesis space $h_\pi \in \mathscr{H}_\pi$ via inference given a set of samples $\mathscr{D} = \{(x_n, y_n)|n = 1,\ldots,N\}$ obtained from the true distribution. The risk of the Bayes classifier $\arg\max_y \mathscr{C}at(y|h_\pi(x))$ for a given input x is

$$R(h_\pi(x)) = \sum_{\kappa=1}^{K} \mathbb{I}_{\kappa \neq \arg\max h_\pi(x)} f^{\pi,\kappa}_{true}(x). \tag{2.3}$$

For the whole data distribution, we have $R(h_\pi) = \mathbb{E}_{x|\pi}[R(h_\pi(x))]$. The optimal case is when $h_\pi(x) = f^\pi_{true}(x)$, which brings about the irreducible Bayes risk resulting from class overlap:

$$R_*\left(f^\pi_{true}(x)\right) = \min\Big\{1 - \max f^\pi_{true}(x), \max f^\pi_{true}(x)\Big\}. \tag{2.4}$$

Total Calibration. Given a set of test samples $\mathscr{D}_{ts}$ coming from the true data generating process in Eq. 2.1, we define *Total Calibration* as the capability of a discriminative predictor $h_\pi(x)$ to quantify the following three types of uncertainty simultaneously:

i) Model misfit evaluated by the similarity of $h_\pi(x)$ and $f^\pi_{true}(x)$ measurable directly via

$$\mathrm{KL}\big(\mathscr{C}at(y|f^\pi_{true}(x))p(x) \;||\; \mathscr{C}at(y|h_\pi(x))p(x)\big) = \\ \mathbb{E}_{p(x)}\left[\log \mathscr{C}at(y|f^\pi_{true}(x)\right] + \mathbb{E}_{p(x)}\left[-\log \mathscr{C}at(y|h_\pi(x))\right] \tag{2.5}$$

where $\mathrm{KL}(\cdot||\cdot)$ denotes the Kullback-Leibler (KL) divergence between the two distributions on its arguments. Since $\mathbb{E}_{p(x)}\left[\log \mathscr{C}at(y|f^\pi_{true}(x))\right]$ is a constant with respect to $h_\pi(x)$, an unbiased estimator of the relative goodness of an inferred model is the negative test log-likelihood $NLL(h_\pi(x)) = -1/|\mathscr{D}_{ts}| \sum_{(x,y)\in\mathscr{D}_{ts}} \log \mathscr{C}at(y|h_\pi(x))$.

ii) Class overlap evaluated by $R_*(f^\pi_{true}(x))$. As there is no way to measure this quantity from $\mathscr{D}_{ts}$, it is approximated indirectly via *Expected Calibration Error (ECE)*

$$ECE[h_\pi] = \sum_{m=1}^{M} \frac{|B_m|}{N} \Big| acc(B_m) - conf(B_m) \Big| \tag{2.6}$$

where $B_m = \{(n|h_\pi(x_n) \in [(m-1)/M, m/M]\}$ are M bins that partition the test set $\mathscr{D}_{ts}$ into $\mathscr{D}_{ts} = B_1 \cup \cdots \cup B_M$ and are characterized by their accuracy and confidence scores

$$acc(B_m) = \frac{1}{|B_m|} \sum_{n\in B_m} \mathbb{I}_{y_n = \arg\max h_\pi(x_n)}, \qquad conf(B_m) = \frac{1}{|B_m|} \sum_{n\in B_m} h_\pi(x_n). \tag{2.7}$$

iii) Domain mismatch defines the rarity of an input pattern x_* for the target task, that is $Pr[p(x = x_*) < \varepsilon]$ for small ε. This quantity cannot be measured since $p(x)$ is unknown. It is approximated indirectly as the success of discriminating between samples x_* coming from a different data distribution and those in $\mathscr{D}_{ts}$ by calculating the Area Under ROC Curve (AUROC) curve w.r.t. $h_\pi(x)$.

2.3 Uncertainty Quantification with Parametric and Evidential Bayesian Models

Parametric Bayesian Models. A commonplace design choice is to build the hypothesis space by equipping the hypothesis function with a global parameter θ that takes values from a feasible set Θ, that is $\mathscr{H}_\pi = \{h_\pi^\theta(x)|\theta \in \Theta\}$. *PBMs* employ a prior belief $\theta \sim p(\theta)$, which is updated via Bayesian inference on a training set $\mathscr{D}_{tr}$ as $p(\theta|\mathscr{D}_{tr}) = \prod_{(x,y)\in\mathscr{D}_{tr}} \mathscr{C}at(y|h_\pi^\theta(x))p(\theta)/p(\mathscr{D}_{tr})$. Consider the update on the posterior belief

$$p(\theta|\mathscr{D}_{tr} \cup (x_*, y_*)) = \mathscr{C}at(y_*|h_\pi^\theta(x_*))p(\theta|\mathscr{D}_{tr})/Z, \tag{2.8}$$

after a new observation (x_*, y_*) where the normalizer can be expressed as

$$Z = \frac{1}{p(\mathscr{D}_{tr})} \int \mathscr{C}at(y_*|h_\pi^\theta(x_*))L(\theta)p(\theta)d\theta. \tag{2.9}$$

with $L(\theta) = \prod_{(x,y)\in\mathscr{D}_{tr}} \mathscr{C}at(y|h_\pi^\theta(x))$. This can be seen as an inner product between the functions $\mathscr{C}at(y_*|h_\pi^\theta(x_*))$ and $L(\theta)$ on an embedding space modulated by $p(\theta)$. As the sample size grows, the high-density regions of $\mathscr{C}at(y_*|h_\pi^\theta(x_*))$ and $L(\theta)$ will superpose with higher probability, causing Z to increase, making $p(\theta|\mathscr{D}_{tr} \cup (x_*, y_*))$ increasingly more peaked, and consequently we will have

$$\lim_{|\mathscr{D}_{tr}|\to+\infty} Var[\theta|\mathscr{D}_{tr}] = 0. \tag{2.10}$$

This conjecture depends on the assumption that a single random variable θ modulates all samples. The variance of a prediction in favor of a class k decomposes according to the law of total variance

$$Var[y_* = k|x_*, \mathscr{D}] = \underbrace{Var_{\theta\sim p(\theta|\mathscr{D}_{tr})}[h_{\pi,k}^\theta(x_*)]}_{\text{Reducible model uncertainty}} + \underbrace{\mathbb{E}_{\theta\sim p(\theta|\mathscr{D}_{tr})}\left[(1-h_{\pi,k}^\theta(x_*))h_{\pi,k}^\theta(x_*)\right]}_{\text{Data uncertainty}}. \tag{2.11}$$

Due to the Eq. 2.11, the first term on the r.h.s. vanishes in the large sample regime and the second one coincides with the asymptotic solution of the maximum likelihood estimator θ_{MLE}. In cases when $\mathscr{H}_\pi$ contains $f_{true}^\pi(x)$, an ideal inference scheme has the potential to recover it and we get

$$\begin{aligned}\lim_{N\to+\infty} \mathbb{E}_{\theta\sim p(\theta|\mathscr{D}_{tr})}\left[(1-h_{\pi,k}^\theta(x_*))h_{\pi,k}^\theta(x_*)\right] &= (1-h_{\pi,k}^{\theta_{MLE}}(x_*))h_{\pi,k}^{\theta_{MLE}}(x_*) \\ &= (1-f_{true}^\pi(x_*))f_{true}^\pi(x_*).\end{aligned} \tag{2.12}$$

The simple proposition below sheds light on the meaning of this result. It points to the fact that the Bayes risk of a classifier (Eq. 2.4) and the second component of its predictive variance (Eq 2.11) are proportional. This gives a mechanistic explanation for why the second term on the r.h.s. of Eq. 2.11 quantifies class overlap. Despite the commonplace allocation of the wording *data uncertainty* for this term, we are not aware of prior work that derives it formally from basic learning-theoretic concepts.

Proposition. *The following inequality holds for any* $\pi, \pi' \in [0.5, 1]$ *pair*

$$\min(1-\pi, \pi) \geq \min(1-\pi', \pi') \Rightarrow (1-\pi)\pi \geq (1-\pi')\pi'. \tag{2.13}$$

Proof. Define $\pi = 0.5 + \varepsilon$ and $\pi' = 0.5 + \varepsilon'$ for $\varepsilon, \varepsilon' > 0$. As $\min(1-\pi, \pi) \geq \min(1-\pi', \pi') \Rightarrow \varepsilon \leq \varepsilon'$, we get $(1-\pi)\pi = (1-0.5-\varepsilon)(0.5+\varepsilon) = 0.25 - 0.25\varepsilon - \varepsilon^2 \geq 0.25 - 0.25\varepsilon' - \varepsilon'^2 = (1-\pi')\pi' \square$

The conclusion is that $(1 - f^{\pi}_{true}(x)) f^{\pi}_{true}(x) \propto R_*(f^{\pi}_{true}(x))$. Hence, the second term on the r.h.s. of Eq. 2.11 is caused by the class overlap and it can be used to approximate the Bayes risk. Put together, the first term of the decomposition reflects the missing knowledge on the optimal value of θ that can be reduced by increasing the training set size, while the second reflects the uncertainty stemming from the properties of the true data distribution. Hence we refer to the first term as the *reducible model uncertainty* and the second the *data uncertainty*.

Evidential Bayesian Models. In all the analysis above, we assumed a fixed and unknown π that modulates the whole process and cannot be identified by the learning scheme. When we characterize the uncertainty on the class distribution by a prior belief $\pi \sim p(\pi)$, we attain the joint $p(y, \pi|x) = Pr[y|x, \pi] p(\pi|x)$. The variance of the Bayes optimal classifier that averages over this uncertainty $Pr[y|x] = \int Pr[y|x, \pi] p(\pi|x) d\pi$ decomposes as

$$Var[y = k|x] = \underbrace{Var_{\pi \sim p(\pi|x)}[f^{\pi,k}_{true}(x)]}_{\text{Irreducible model uncertainty}} + \underbrace{\mathbb{E}_{\pi \sim p(\pi|x)}\left[(1 - f^{\pi,k}_{true}(x)) f^{\pi,k}_{true}(x)\right]}_{\text{Data uncertainty}}. \tag{2.14}$$

The main source of uncertainty $p(\pi|x)$ is not a posterior this time but an empirical prior on a single observation x, hence its variance will not shrink as $|\mathscr{D}_{tr}| \to +\infty$. Therefore, the first term on the r.h.s. reflects the *irreducible model uncertainty* caused by the

missing knowledge about π. The second term indicates data uncertainty as in PBMs, but accounts for the local uncertainty at x caused by π. *EBMs* [12,13] suggest

$$p(y,\pi|x) = Pr[y|x,\pi]p(\pi|x) \approx Pr[y|\pi]q_\psi(\pi|x) \tag{2.15}$$

where $q_\psi(\pi|x)$ is a density function parameterized by ψ and the dependency of the class-conditional on the input is dropped. Note that the resultant model employs uncertainty on individual data points via π, but it does not have any global random variables. The standard evidential model training is performed by maximizing the expected likelihood subject to a regularizer that penalizes the divergence of $q_\psi(\pi|x)$ from the prior belief

$$\arg\min_{\psi} -\int \log Pr[y|\pi]q_\psi(\pi|x)d\pi + \beta \mathrm{KL}(q_\psi(\pi|x)||p(\pi)). \tag{2.16}$$

Although presented in the original work as as-hoc design choices, such a training objective can alternatively be viewed as β-regularized [28] variational inference of $Pr[y|\pi]p(\pi|x)$ with the approximate posterior $q_\psi(\pi|x)$ [29].

2.4 Complete Bayesian Models: Best of Both Worlds

The uncertainty decomposition characteristics of PBMs and EBMs provide complementary strengths toward the quantification of the three uncertainty types that constitute total calibration.

i) Model misfit: The PBM is favorable since it provides shrinking posterior variance with growing training set size (Eq. 2.10) and the recovery of θ_{MLE} which inherits the consistency guarantees the frequentist approach. Since the uncertainty on the local variable π does not shrink for large samples, Negative Log-Likelihood (NLL) would measure how well ψ can capture the regularities of heteroscedastic noise across individual samples, which is a more difficult task than quantifying the fit of a parametric model.

ii) Class overlap: The EBM is favorable since its data uncertainty term $\mathbb{E}_{\pi\sim p(\pi|x_*)}[(1-h^\theta_{\pi,k}(x_*))h^\theta_{\pi,k}(x_*)]$ is likely to have less estimator variance than $\mathbb{E}_{\theta\sim p(\theta|\mathscr{D}_{tr})}[(1-h^\theta_{\pi,k}(x_*))h^\theta_{\pi,k}(x_*)]$ and calculating $p(\pi|x_*)$ does not require approximate inference as for $p(\theta|\mathscr{D}_{tr})$.

iii) Domain mismatch: The PBM is favorable since the effect of the modulation of a global θ on the variance of Z applies also to the posterior predictive distribution with the only difference that x_* is a test sample. When the test sample is away from the training set, i.e. $\min_{x\in\mathscr{D}_{tr}} ||x_* - x||$ is large, then $p(y_*|x_*,\theta)$ is less likely to superpose with those regions of θ where $L(\theta)$ is pronounced, hence will be flattened out leading to a higher variance posterior predictive for samples coming from another domain. Since EBM has only local random variables, the variance of its posterior is less likely to build a similar discriminative property for domain detection.

Main Hypothesis. *A model that inherits the model uncertainty of PBM and data uncertainty of EBM can simultaneously quantify: i) model misfit, ii) class overlap, and iii) domain mismatch.*

Guided by our main hypothesis, we combine the PBM and EBM designs within a unified framework that inherits the desirable uncertainty quantification properties of each individual approach

$$Pr[y|\pi]p(\pi|\theta,x)p(\theta), \tag{2.17}$$

which we refer to as a *CBM* hinting to its capability to solve the total calibration problem. The model simply introduces a global random variable θ into the empirical prior $p(\pi|\theta,x)$ of the EBM. The variance of the posterior predictive of the resultant model decomposes as

$$\begin{aligned} Var[y|x] = &\underbrace{Var_{p(\theta|\mathscr{D})}\Big[\mathbb{E}_{p(\pi|x,\theta)}[\mathbb{E}[y|\pi,x]]\Big]}_{\text{Reducible Model Uncertainty}} + \underbrace{\mathbb{E}_{p(\theta|\mathscr{D})}\Big[Var_{p(\pi|\theta,x)}[\mathbb{E}[y|\pi]]\Big]}_{\text{Irreducible Model Uncertainty}} \\ &+ \underbrace{\mathbb{E}_{p(\theta|\mathscr{D})}\Big[\mathbb{E}_{p(\pi|x,\theta)}[Var[y|\pi]]\Big]}_{\text{Data Uncertainty}} \end{aligned} \tag{2.18}$$

where the r.h.s. of the decomposition has the following desirable properties: i) The first term recovers the form of the reducible model uncertainty term of PBM when π is marginalized $Var_{p(\theta|\mathscr{D})}[\mathbb{E}_{p(\pi|x,\theta)}[\mathbb{E}[y|\pi,x]]] = Var_{p(\theta|\mathscr{D})}[\mathbb{E}[y|\theta,x]]$; ii) The second term is the irreducible model uncertainty term of EBM averaged over the uncertainty on the newly introduced global variable θ. It quantifies the portion of uncertainty stemming from heteroscedastic noise, which is not explicit in the decomposition of PBM; iii) The third term recovers the data uncertainty term of EBM when θ is marginalized: $\mathbb{E}_{p(\theta|\mathscr{D})}[\mathbb{E}_{p(\pi|x,\theta)}[Var[y|\pi]] = \mathbb{E}_{p(\pi|x)}[Var[y|\pi]]$.

2.5 The Evidential Turing Process: An Effective CBM Realization

Equipping the empirical prior $p(\pi|\theta,x)$ of CBM with the capability of *learning* to generate accurate prior beliefs on individual samples is the key to total calibration. We adopt the following two guiding principles to obtain highly expressive empirical priors: i) The existence of a global random variable θ can be exploited to express complex reducible model uncertainties using the NP [18] framework, ii) The conditioning of $p(\pi|\theta,x)$ on a global variable θ and an individual observation x can be exploited with an attention mechanism where θ is a memory and x is a query. Dependency on a context set during test time can be lifted using the NTM [27] design, which maintains an external memory that is updated by a rule detached from the optimization scheme of the main loss function. We extend the stochastic process derivation of the NP by marginalizing out a local variable from a PBM with an external memory that follows the NTM design and arrive at a new family of stochastic processes called a *Turing Process,* formally defined as below.

Definition 1. *A Turing Process is a collection of random variables $Y = \{y_i | i \in \mathscr{I}\}$ for an index set $\mathscr{I} = \{1,\dots,K\}$ with arbitrary $K \in \mathbb{N}^+$ that satisfy the two properties*

$$\textit{(i) } p_M(Y) = \int \prod_{y_i \in Y} p(y_i|\theta) p_M(\theta) d\theta, \quad \textit{(ii) } p_{M'}(Y|Y') = \int \prod_{y_i \in Y} p(y_i|\theta) p_{M'}(\theta|Y') d\theta, \tag{2.19}$$

for another random variable collection Y' of arbitrary size that lives in the same probability space as Y, a probability measure $p_M(\theta)$ with free parameters M, and some function $M' = r(M,Y')$.

The Turing Process can express all stochastic processes since property (i) is sufficient to satisfy Kolmogorov's Extension Theorem [30] and choosing r simply to be an identity mapping would revert us back to the generic definition of a stochastic process. The Turing Process goes further by permitting to explicitly specify how information accumulates within the prior. This is a different approach from the Neural Process that performs conditioning $p(Y|Y')$ by applying an amortized posterior $q(\theta|Y')$ on a plain stochastic process that satisfies only Property (i), hence maintains a data-agnostic prior $p(\theta)$ and depends on a context set also at the prediction time.

Given a disjoint partitioning of the data $\mathscr{D} = \mathscr{D}_C \cup \mathscr{D}_T$ into a context set $(x', y') \in \mathscr{D}_C$ and a target set $(x, y) \in \mathscr{D}_T$, we propose the data generating process on the target set below

$$p(y, \mathscr{D}_T, \pi, \theta) = p(w) \underbrace{p_M(Z)}_{\substack{\text{External} \\ \text{memory}}} \prod_{(x,y)\in\mathscr{D}_T} \Big[p(y|\pi) \underbrace{p(\pi|Z,w,x)}_{\substack{\text{Input-specific} \\ \text{prior}}} \Big]. \tag{2.20}$$

We decouple the global parameters $\theta = \{w, Z\}$ into a w that parameterizes a map from the input to the hyperparameter space and an external memory $Z = \{z_1, \ldots, z_R\}$ governed by the distribution $p_M(Z)$ parameterized by M that embeds the context data during training. The memory variable Z updates its belief conditioned on the context set $\mathscr{D}_C$ by updating its parameters with an explicit rule $p_M(Z|\mathscr{D}_C) \leftarrow p_{\mathbb{E}_{Z\sim p_M(Z)}[r(Z,\mathscr{D}_C)]}(Z)$. The hyperpriors of $p(\pi|Z,w,x)$ are then determined by querying the memory Z for each input observation x using an attention mechanism, for instance a transformer network [31]. Choosing the variational distribution as $q_\lambda(\theta, \pi|x) = q_\lambda(w) p_M(Z) q(\pi|Z,w,x)$, we attain the variational free energy formula for our target model

$$\mathscr{F}_{ETP}(\lambda) = \tag{2.21}$$
$$\mathbb{E}_{q_\lambda(w)} \left[-\sum_{(x,y)\in\mathscr{D}} \mathbb{E}_{q_\lambda(\pi|Z,w,x)} \left[\mathbb{E}_{p_M(Z)} \left[\log p(y|\pi) + \log \frac{q_\lambda(\pi|Z,w,x)}{p(\pi|Z,w,x)} \right] \right] + \log \frac{q_\lambda(w)}{p(w)} \right].$$

The learning procedure will then alternate between updating the variational parameters λ via gradient-descent on $\mathscr{F}_{ETP}(\lambda)$ and updating the parameters of the memory variable Z via an external update rule as summarized in Figure 2.2 and Figure 2.3. ETP can predict a test sample (x_*, y_*) by only querying the input x_* on the memory Z and evaluating the prior $p(\pi_*|Z,w,x_*)$ without need for any context set as

$$p(y_*|\mathscr{D}_{tr}, x_*) \approx \iint p(y_*|\pi_*) q(\pi_*|Z,w,x_*) p_M(Z) q_\lambda(w) dZ dw. \tag{2.22}$$

2.6 Most Important Special Cases of the Evidential Turing Process

As we build the design of the ETP on the CBM framework that over-arches the prevalent parametric and evidential approaches, its ablation amounts to recovering many state-of-the-art modeling approaches as listed in Table 2.1 and detailed below.

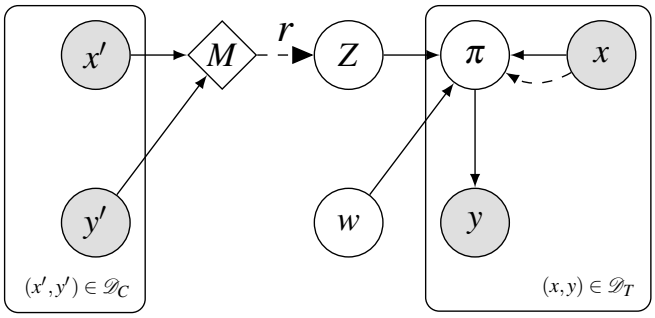


Figure 2.2 : ETP Plate diagram. The essential design choice is how $p(\pi|Z,w,x)$ is constructed thanks to an external memory M that can be queried with respect to any input x without needing to store context data at test time. The dashed arrow indicates that Z can condition on a context $\mathscr{D}_C$ by updating its parameters M with an explicit function r.

---

while *Model not converged* **do**
  for $\mathscr{D}_{batch} \subset \mathscr{D}$ **do**
    Choose $\mathscr{D}_C \subset \mathscr{D}_{batch}$
    $M \leftarrow \mathbb{E}_{Z\sim p_M(Z)}\left[r(Z,\mathscr{D}_C)\right]$
    $\lambda \leftarrow \lambda - \nabla_\lambda \mathscr{F}_{ETP}(\lambda)$
  end
end

Figure 2.3 : The ETP training routine.

Bayesian Neural Networks (BNN) [32]–[34] is the most characteristic PBM example that parameterize a likelihood $p(y|\theta,x)$ with a neural network $f_\theta(\cdot)$ whose weights follow a distribution $\theta \sim p(\theta)$. In the experiments we assume as a prior $\theta \sim \mathscr{N}(\theta|0,\beta^{-1}I)$ and infer its posterior by mean-field variational posterior $q_\lambda(\theta)$ applying the reparameterization trick [35,36].

Evidential Deep Learning (EDL) [12] introduces the characteristic example of EBMs. EDL employs an uninformative Dirichlet prior on the class probabilities, which then set the mean of a normal distribution on the one-hot-coded class labels $y, \pi \sim \mathscr{D}ir(\pi|1,\ldots,1)\mathscr{N}(y|\pi,0.5I_K)$. EDL links the input observations to the distribution of class probabilities by performing amortized variational inference with the approximate distribution $q_\lambda(\pi;x) = \mathscr{D}ir(\pi|\alpha_\lambda(x))$.

Neural Processes (NP) [18,37] is a PBM used to generate stochastic processes from neural networks by integrating out the global parameters θ as in Property (i) of

Definition 1. NPs differ from PBMs by amortizing an arbitrary sized context set $\mathscr{D}_C$ on the global parameters $q(\theta|\mathscr{D}_C)$ via an aggregator network during inference. NPs can be viewed as Turing Processes with an identity map r. Follow-up work equipped NPs with attention [1], translation equivariance [38,39], and sequential prediction [40,41]. All of these NP variants assume prediction-time access to context, making them suitable for interpolation tasks such as image in-painting but not for generic prediction problems.

Evidential Neural Process (ENP) is the EBM variant of a neural process we introduce to bring together best practices of EDL and NPs and make the strongest possible baseline for ETP, defined as

$$p(y,\pi,Z|\mathscr{D}_C) = \mathscr{C}at(y|\pi)\mathscr{D}ir\big(\pi|\alpha_\theta(Z)\big)\mathscr{N}\big(Z|(\mu,\sigma^2) = r(\{h_\theta(x_j,y_j) \mid (x_j,y_j) \in \mathscr{D}_C\})\big), \tag{2.23}$$

with an aggregation rule $r(\cdot)$, an encoder net $h_\theta(\cdot,\cdot)$, and a neural net $\alpha_\theta(\cdot)$ mapping the global variable Z to the class probability simplex. We further improve upon ENP with an Attentive Neural Process (ANP) approach [1] by replacing the fixed aggregation rule r with an attention mechanism, thereby allowing the model to switch focus depending on the target. At the prediction time, we use $Z \sim \mathscr{N}(Z|\mathbf{1},\kappa^2 I)$ for small κ to induce an uninformative prior on π in the absence of a context set.

Table 2.1 : Ablation Table. Deactivating the components of ETP one at a time equates it to state-of-the-art methods. We curate ENP as a surrogate for the ANP of [1], which requires test-time context, and an improved NP variant with reduced estimator variance thanks to π.

Model	π	Z	M	r	$\mathscr{D}_c^{(test)}$	Compare
ETP (Target)	✓	✓	✓	✓	×	
ANP [1]	✓	✓	✓	×	✓	No
ENP (Surrogate)	✓	✓	×	×	×	Yes
NP [18]	×	✓	×	×	×	No
EDL [12]	✓	×	×	×	×	Yes
BNN [36]	×	×	×	×	×	Yes

✓: Component active ×: Component inactive
$\mathscr{D}_c^{(test)}$: Demand for context data at test time

2.7 Case Study: Classification with Evidential Turing Processes

We demonstrate a realization of an ETP to a classification problem below

$$y, \pi, w, Z | M \sim \mathscr{C}at(y|\pi)\mathscr{D}ir\big(\pi|\exp(a(v_w(x);Z))\big)\mathscr{N}(w|0,\beta^{-1}I)\prod_{r=1}^{R}\mathscr{N}(z_r|m_r,\kappa^2 I). \tag{2.24}$$

The global variable Z is parameterized by the memory $M = (m_1, \ldots, m_R)$ consisting of R cells $m_r \in \mathbb{R}^K$.[1] The input data x is mapped to the same space via an encoding neural net $v_w(\cdot)$ parameterizing a Dirichlet distribution over the class probabilities π via an attention mechanism $a(v_w, z) = \sum_{z' \in Z} \phi(v_w(x), z')z$, where $\phi(v_w(x), z) = \text{softmax}(\{k_\psi(z)^\top v_w(x)/\sqrt{K} | z \in Z\})$. The function $k_\psi(\cdot)$ is the key generator network operating on the embedding space. We update the memory cells as a weighted average between remembering and updating

$$m \leftarrow \mathbb{E}_{Z\sim p(Z|M)}\left[\tanh\left(\gamma m + (1-\gamma)\sum_{(x,y)\in\mathscr{D}_C}\phi(v_w(x),z)[\text{onehot}(y) + \text{soft}(v_w(x))]\right)\right] \tag{2.25}$$

where $\gamma \in (0,1)$ is a fixed scaling factor controlling the relative importance. The second term adds new information to the cell as a weighted sum over all pairs (x,y), taking both the true label (via onehot(y)) as well as the uncertainty in the prediction (via soft$(v_w(x))$) into account. The final $\tanh(\cdot)$ transformation ensures that the memory content remains in a fixed range across updates, as practised in Long Short-Term Memories (LSTM) [6].

2.8 Experiments

We benchmark ETP against the state of the art as addressed in the ablation plan in Table 2.1 according to the total calibration criteria developed in Sec. 2.2 on five real-world data sets. See the Appendix 2.10.2 for full details on the experimental setup in each case and for the hyper-parameters used throughout. We provide a reference implementation of the proposed model and the experimental pipeline.[2]

[1] In the experiments we fix $\kappa^2 = 0.1$. Preliminary results showed stability as well for larger/smaller values.

[2] `https://github.com/ituvisionlab/EvidentialTuringProcess`

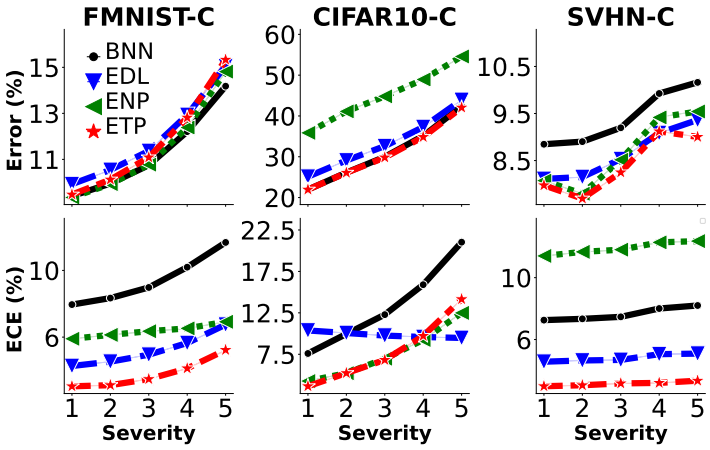


Figure 2.4 : Results averaged across 19 types of corruption (e.g. motion blur, fog, pixelation) applied on the test splits of Fashion MNIST (FMNIST), CIFAR10, and SVHN at five severity levels.

Table 2.2 : Quantitative results on five data sets showing mean ± standard deviation across 10 repetitions. Best performing models that overlap within three standard deviations are highlighted in bold.

Domain Data (Architecture)	**IMDB** (LSTM)	**Fashion** (LeNet5)	**SVHN** (LeNet5)	**CIFAR10** (LeNet5)	**CIFAR100** (ResNet18)
Prediction accuracy as % test error					
BNN	**16.4** ±0.6	**7.9**±0.1	7.9±0.1	**15.3**±0.3	30.2 ±0.3
EDL	38.3 ±13.3	8.6±0.1	7.3±0.1	18.5±0.2	45.2 ±0.4
ENP	50.0 ±0.0	**7.9**±0.2	**6.7**±0.1	**14.8**±0.2	39.0 ±0.3
ETP (Target)	**15.8** ±1.3	**7.9**±0.2	**6.9**±0.1	**15.3**±0.2	**29.2** ±0.3
In-domain calibration as % Expected Calibration Error (ECE)					
BNN	14.4 ±0.4	6.7±0.	6.5±0.	5.5±0.3	15.2±0.0
EDL	41.1 ±2.6	3.7±0.2	4.0±0.1	9.0±0.2	**5.3**±0.4
ENP	**0.8** ±1.6	6.0±0.2	10.7±0.2	7.2±0.3	39.7±0.4
ETP (Target)	**3.1** ±0.4	**2.6**±0.2	**2.6**±0.1	**2.7**±0.1	6.6±0.1
Model fit as negative test log-likelihood					
BNN	0.47 ±0.0	0.65±0.0	0.71±0.0	0.50±0.0	1.78±0.0
EDL	0.66 ±0.1	0.37±0.0	0.34±0.0	0.72±0.0	2.24±0.0
ENP	0.69 ±0.0	0.34±0.0	0.33±0.0	0.50±0.0	2.52±0.0
ETP (Target)	**0.37** ±0.0	**0.29**±0.0	**0.26**±0.0	**0.46**±0.0	**1.36**±0.0
Out-of-domain detection as % Area Under ROC Curve					
OOD Data	Random	MNIST	CIFAR100	SVHN	TinyImageNet
BNN	**60.9** ±4.2	75.9±2.3	86.2±0.5	**84.1**±1.3	97.2±0.5
EDL	**55.1** ±5.1	77.5±2.0	90.9±0.3	79.2±0.7	89.6±0.3
ENP	**53.7** ±5.7	**88.9**±1.0	**92.4**±0.4	**81.4**±0.8	**100.0**±0.1
ETP (Target)	**59.1** ±5.1	**90.0**±0.9	90.0±0.4	**82.1**±0.6	99.6±0.1

Total calibration. Table 2.2 reports the prediction error and the three performance scores introduced in Sec. 2.2 that constitute total calibration. We perform the out-of-domain detection task as in [13] and classify the test split of the target domain and a data set from another domain based on the predictive entropy. ETP is the only method that consistently ranks among top performers in all data sets with respect to all performance scores, which supports our main hypothesis that CBM should outperform PBM and EBM in total calibration.

Response to gradual domain shift. In order to assess how well models can cope with a gradual transition from their native domain, we evaluate their ECE performance on data perturbed by 19 types of corruption [42] at five different severity levels. Figure 2.4 depicts the performance of models averaged across all corruptions. ETP gives the best performance nearly in all data sets and distortion severity levels (see Appendix 2.10.2.3 Table 2.4 for a tabular version of these results).

Computational cost. We measured the average wall-clock time per epoch in CIFAR10 to be 10.8 ± 0.1 seconds for ETP, 8.2 ± 0.1 seconds for BNN, 8.6 ± 0.1 seconds for EDL, and 9.5 ± 0.2 seconds for ENP. The relative standings of the models remain unchanged in all the other data sets.

2.9 Broad Impect, Limitations and Ethical Concerns

Broad impact. Our work delivers evidence for the claim that epistemic awareness of a Bayesian model is indeed a capability learnable only from in-domain data, as appears in biological intelligence via closed-loop interactions of neuronal systems with stimuli. The implications of our work could follow up in interdisciplinary venues, with focusing on the relation of associative memory and attention to the neuroscientific roots of epistemic awareness [43].

Limitations and ethical concerns. While we observed ETP outperforms BNN variants in our experiments, whether BNNs could bridge the gap after further improvements in the approximate inference remains an open question. The attention mechanism of ETP could also be improved, for instance to transformer nets [31]. The explainability of the memory content of ETP deserves further investigation. The

generalizability of our results to very deep architectures, such as ResNet 152 or DenseNet 161 could be a topic of a separate large-scale empirical study. ETP does not improve the explainability and fairness of the decisions made by the underlying design choices, such as the architecture of the used neural nets in the pipeline. Potential negative societal impacts of deep neural net classifiers stemming from these two factors need to be circumvented separately before the real-world deployment of our work.

2.10 Appendix

2.10.1 Further analytical results and derivations

2.10.1.1 Kullback-Leibler between two Dirichlet distributions

As both our ETP and and the EDL baseline use the KL divergence between two Dirichlet distributions, we give the full details of its calculation here. For two distributions over a K-dimensional probability π, $\mathscr{D}ir(\pi|\alpha)$ and $\mathscr{D}ir(\pi|\beta)$, parameterized by $\alpha, \beta \in \mathbb{R}_+^K$, the following identity holds

$$\begin{aligned}\mathrm{KL}\left(\mathscr{D}ir(\pi|\alpha) \parallel \mathscr{D}ir(\pi|\beta)\right) = \log\left(\frac{\Gamma(\sum_k \alpha_k)\prod_k \Gamma(\beta_k)}{\Gamma(\sum_k \beta_k)\prod_k \Gamma(\alpha_k)}\right) \\ + \sum_k (\alpha_k - \beta_k)\left(\psi(\alpha_k) - \psi(\sum_k \alpha_k)\right),\end{aligned} \tag{2.26}$$

where $\Gamma(\cdot)$ and $\psi(a) := \frac{d}{da}\log\Gamma(a)$ are the *gamma* and *digamma* functions [44].

2.10.1.2 The Evidential Deep Learning objective

The analytical expression of the EDL [12] loss as defined in the main paper can be computed as follows. For a single K-dimensional observation y, dropping the n from the notation and instead using the index for the dimensionality throughout the

following equations, we have

$$\mathbb{E}_{p(\pi|x)}\left[||y-\pi||^2\right] = \mathbb{E}_{p(\pi|x)}\left[(y-\pi)^\top(y-\pi)\right] \tag{2.27}$$

$$= \sum_{k=1}^{K} \mathbb{E}_{p(\pi|x)}\left[(y_k-\pi_k)^2\right] \tag{2.28}$$

$$= \sum_{k=1}^{K} \mathbb{E}_{p(\pi|x)}\left[(y_k-\mathbb{E}_{p(\pi|x)}[\pi_k]+\mathbb{E}_{p(\pi|x)}[\pi_k]-\pi_k)^2\right] \tag{2.29}$$

$$= \sum_{k=1}^{K} (y_k-\mathbb{E}_{p(\pi|x)}[\pi_k])^2 + \mathrm{var}_{p(\pi|x)}[\pi_k], \tag{2.30}$$

with the tractable expectation and variance of a Dirichlet distributed π. The KL term is between two Dirichlet distributions and given as

$$\begin{aligned}&\mathrm{KL}\left(\mathscr{D}ir(\pi|\alpha_\theta(x)) \parallel \mathscr{D}ir(\pi|1,\ldots,1)\right)\\ &= \log\left(\frac{\Gamma\left(\sum_k \alpha_\theta(x)_k\right)}{\Gamma(K)\prod_k \Gamma(\alpha_\theta(x)_k)}\right) + \sum_{k=1}^{K}(\alpha_\theta(x)_k-1)\Big(\psi(\alpha_\theta(x)_k)-\psi\big(\textstyle\sum_k \alpha_\theta(x)_k\big)\Big).\end{aligned} \tag{2.31}$$

2.10.1.3 Evidential Deep Learning as a latent variable model

As discussed in the main paper for a set of N observations $\mathscr{D} = \{(x_1,y_1),\ldots,(x_N,y_N)\}$, the EDL objective minimizes is given as

$$\mathscr{L}_{\mathrm{EDL}} = \sum_{n=1}^{N} \mathbb{E}_{p(\pi_n|x_n)}\left[||y_n-\pi_n||_2^2\right] + \lambda \mathrm{KL}\left(p(\pi_n|x_n) \parallel \mathscr{D}ir(\pi_n|1,\ldots,1)\right), \tag{2.32}$$

where $p(\pi_n|x_n) = \mathscr{D}ir(\pi|\alpha_\theta(x_n))$. We can instead assume the following generative model

$$\pi_n \sim \mathscr{D}ir(\pi_n|1,\ldots,1) \qquad \forall n \tag{2.33}$$

$$y_n \sim \mathscr{N}(y_n|\pi_n, 0.5I_K) \qquad \forall n, \tag{2.34}$$

i.e. latent variables π_n and K-dimensional observations y_n following a multivariate normal prior. Approximating the intractable posterior $p(\pi|\mathbf{y})$, where $\pi = (\pi_1,\ldots,\pi_n)$, $\mathbf{y} = (y_1,...,y_n)$, with an amortized variational posterior $q(\pi_n;x_n) = \mathscr{D}ir(\pi_n|\alpha_\theta(x_n))$, where $\alpha_\theta(\cdot)$ is the same architecture as in the EDL model, we have as the *Evidence*

Lower BOund (ELBO) to be maximized

$$\sum_{n=1}^{N} \mathbb{E}_{q(\pi_n;x_n)} \left[\log p(y_n|\pi_n)\right] - \mathrm{KL}\left(q(\pi_n;x_n) \parallel p(\pi_n)\right) \tag{2.35}$$

$$= \sum_{n=1}^{N} \mathbb{E}_{q(\pi_n;x_n)} \left[-\frac{1}{2}\log(2\pi) + \frac{K}{2}\log(2) - (y_n - \pi_n)^\top (y_n - \pi_n)\right] - \mathrm{KL}\left(q(\pi_n;x_n) \parallel p(\pi_n)\right) \tag{2.36}$$

$$= \text{const} - \sum_{n=1}^{N} \mathbb{E}_{q(\pi_n;x_n)} \left[(y_n - \pi_n)^\top (y_n - \pi_n)\right] - \mathrm{KL}\left(q(\pi_n;x_n) \parallel p(\pi_n)\right) \tag{2.37}$$

$$= \text{const} - \mathscr{L}_{\mathrm{EDL}}, \tag{2.38}$$

that is the negative ELBO is equal to the EDL loss up to an additive constant.

2.10.1.4 Posterior predictive

For a new input observation x^*, the corresponding posterior predictive distribution on the output y^* can be computed as

$$p(y^* = k|x^*, \mathscr{D}) \approx \mathbb{E}_{p(Z|M)} \left[\mathbb{E}_{q_\lambda(\pi|x^*,Z)} \left[p(y^* = k|\pi)\right]\right] = \mathbb{E}_{p(Z|M)} \left[h_\theta^k \big/ \textstyle\sum_{k'} h_\theta^{k'}\right], \tag{2.39}$$

where $h_\theta^k = h_\theta\left(v_\theta(x^*), a(v_\theta(x^*); Z)\right)_k$. That is, we can compute it analytically up to a sampling-based evaluation of the expectation over $p(Z|M)$.

2.10.2 Experimental Details

2.10.2.1 Synthetic 1d two-class classification

We illustrate the qualitative behaviour of ETP on a synthetic 1d two-class classification task. The data consist of observations (20 per class) from two highly overlapping Gaussian distributions. The neural net, a simple one hidden layer architecture, can learn the task with high accuracy. Figure 2.5 shows the raw data and underlying distributions as well as the learned memory evidence. The model learns to place more evidence on the correct class further away from the decision boundary while also becoming more varied with increasing distance. Within the high-density region, the asymmetry in how the specific observations are spread out leads to an asymmetry in the memory evidence. Below zero, the blue points are clearly separated, leading

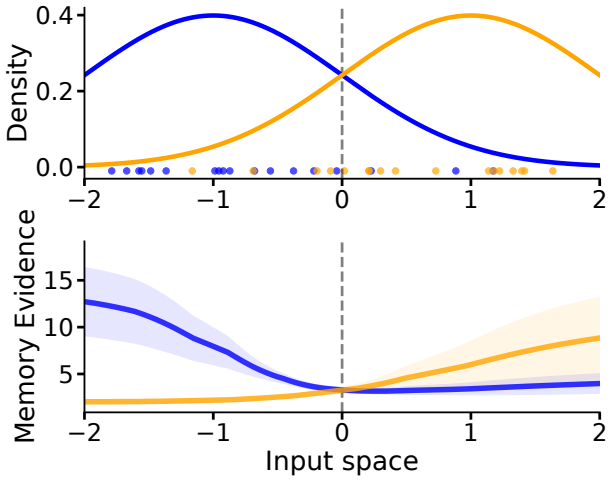


Figure 2.5 : 1D Classification Task. The upper plot shows the underlying distributions of each of the two classes, as well as the observed data. The lower shows the regularizing evidence the generative model places on each of the two classes depending on location in space, that is, mean $\pm$ one standard deviation over ten samples from $p(Z|M)$.

to a quick emphasis on that class as we move toward the left. Above zero, however, some blue observations in the orange region prevent the model from rapidly developing over-confidence.

This synthetic data set consists of two classes with 20 observations each sampled from standard normal distributions centered at ± 1 respectively. The encoding function $v_\phi(\cdot)$ consists of a multi-layer perceptron with a single hidden layer consisting of 32 neurons and a ReLU activation function. It is optimized for 400 epochs with the Adam optimizer [45] using the PyTorch [46] default parameters and a learning rate of 0.001. In order to keep the model as simple as possible, the key generator function $k_\lambda(\cdot)$ is constrained to being the identity function, while $h_\xi\left(v(x), a(v(x);Z)\right) = v(x) + \tanh(a(v(x);Z)$. The memory update is simplified to dropping the $\tanh(\cdot)$ rescaling.

2.10.2.2 Iris 2d three-class classification

The data set used in this experiment is the classical Iris data set[3] consisting of 150 samples of three iris species (50 per class), with four features measured per sample. We first map the data to the first two principal components for visualization and

[3]Originally due to Fisher (1936) and Anderson (1935). We rely on the version provided by the `scikit learn` library, see `https://scikit-learn.org/stable/modules/classes.html\#module-sklearn.datasets` (accessed in March 2021).

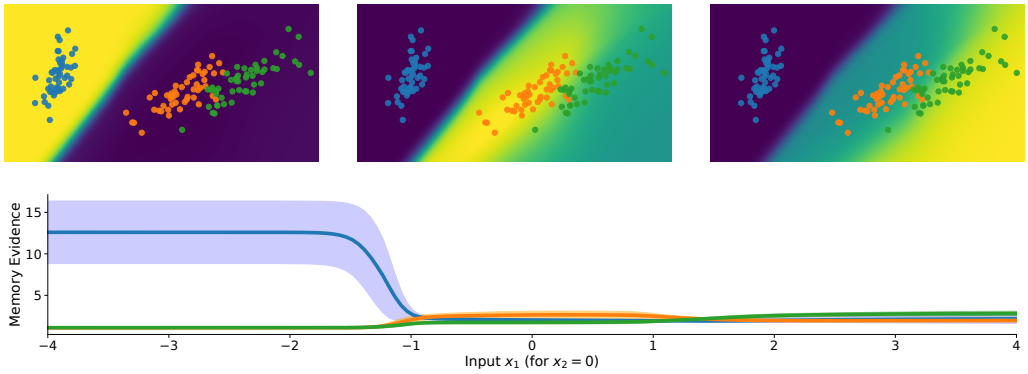


Figure 2.6 : 2D Classification Task. The three upper plots visualize the regularizing evidence that the model places on the location in space as the average over ten samples from $p(Z|M)$. The color scales are different across the three plots and vary in overall intensity. The lower plot visualizes a horizontal cut through the middle of these plots, putting them on a common scale. The solid lines in each color indicate the mean memory evidence, and the corresponding shaded areas indicate one standard deviation. Around the separable blue class, the memory strongly emphasizes the correct class with large confidence and suppresses the other two classes depicted in orange and green. While always preferring the correct class, the memory regularizes against overconfidence around the overlapping orange and green classes.

also use this modified version as the training data. This gives an interpretable toy learning setup where one class is clearly separated from the other two overlapping classes. The encoding function $v_\phi(\cdot)$ consists of an MultiLayer Perceptron (MLP) with two hidden layers of 32 neurons each and ReLU activation functions. We train it for 400 epochs with the Adam optimizer using the PyTorch default parameters and a learning rate of 0.001. In order to keep the model as simple as possible, we constrain the key generator function $k_\lambda(\cdot)$ to the identity function while setting $h_\xi\big(v(x), a(v(x); Z)\big) = v(x) + \tanh(a(v(x); Z)$. We simplify the memory update by dropping the $\tanh(\cdot)$ rescaling. Figure 2.6 shows the varying importance the model assigns to different parts of the input space depending on the class under consideration. For the separate blue class, the model is significantly more confident in relative terms and absolute values.

2.10.2.3 Experiments on real data

All experiments are implemented in PyTorch [46] version 1.7.1 and trained on a TITAN RTX. They have been replicated ten times over random initial seeds.

FMNIST. We use a LeNet5-sized architecture (see Table 2.3). The out-of-distribution data is the MNIST [47] data set. We z-score normalize all data sets with the in-domain mean and standard deviations. We train each model for 50 epochs with the Adam optimizer [45] using a learning rate of 0.001.

CIFAR10. We use a LeNet5-sized architecture (see Table 2.3). The out-of-distribution data is the SVHN [48]. We z-score normalize all data sets with the in-domain mean and standard deviations. We further rely on data augmentation for the in-domain data using random crops with 32 pixels and four pixel padding as well as horizontal flips. We train each model for 100 epochs using the Adam optimizer [45] with a learning rate of 0.0001.

SVHN. We use a LeNet5-sized architecture (see Table 2.3). The out-of-distribution data is the CIFAR10 data set. We z-score normalize all data sets with the in-domain mean and standard deviations. We train each model for 100 epochs using the Adam optimizer [45] with a learning rate of 0.0001.

IMDB Sentiment Classification. We use an LSTM architecture with 64 embedding dimensions and 2 layers with 256 hidden dimensions. We generate the out-of-distribution data by sampling values from the $[1, 1000]$interval uniformly at random. We train each model for 20 epochs using the SGD optimizer [49] with a learning rate of 0.05 and 0.9 momentum. For data preparation, we follow the source code of the original work [4].

Shared Details and Observations. In all experiments, we train the BNN models with a cross-entropy loss using softmax as the squashing function to calculate class probabilities. The EDL model is trained with the original loss [12]. We make our predictions with the mean of the Dirichlet distribution as in the original work. For all models, we use entropy as the criterion for detecting out-of-distribution instances.

Robustness against perturbations. We use the CIFAR10-C data set in the same setup as in original work [42]. As FMNIST-C and SVHN-C are not available, we

[4] `https://www.kaggle.com/arunmohan003/sentiment-analysis-using-lstm-pytorch` (accessed in March 2021).

Table 2.3 : The neural network architectures used for the FMNIST, CIFAR10, and SVHN data set with ReLU activations between the layers.

FMNIST	CIFAR10/SVHN
Convolution (5×5) with 20 channels	Convolution (5×5) with 192 channels
MaxPooling (2×2) with stride 2	
Convolution (5×5) with 50 channels	Convolution (5×5) with 192 channels
MaxPooling (2×2) with stride 2	
Linear with 500 neurons	Linear with 1000 neurons
Linear with 10 neurons	

CIFAR100	
Layer Name	ResNet-18
Conv1	$7 \times 7, 64$, stride 2
	3×3 max pool, stride 2
Conv2_x	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
Conv3_x	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
Conv4_x	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
Conv5_x	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
Average Pool	7×7 average pool
Linear	100 neurons

IMDB	
Layer Name	LSTM
Embedding	$1001 \rightarrow 64$
LSTM	2 layers with 256 hidden size
Linear	2 neurons

Table 2.4 : Quantitative results of models on the FMNIST-C, CIFAR10-C and SVHN-C test data set using the LeNet5. The table below reports mean ± three standard deviations.

		FMNIST-C		CIFAR10-C		SVHN-C	
Severity	Method	Err (%) (↓)	ECE (%) (↓)	(↓)	ECE (%) (↓)	(↓)	ECE (%) (↓)
1	BNN	**9.4**±2.8	8.0±2.4	**21.9**±4.5	**7.6**±2.9	**8.8**±1.2	7.3±1.0
	EDL	**9.9**±2.6	4.3±1.0	**25.1**±4.6	10.4±1.3	**8.1**±1.2	4.6±0.6
	ENP	**9.3**±2.7	5.9±0.5	35.9±6.0	**4.3**±1.7	**8.1**±1.3	11.4±1.0
	ETP	**9.5**±2.8	**3.0**±1.0	**21.9**±4.6	**3.6**±2.2	**8.0**±1.2	**3.0**±0.4
2	BNN	**9.9**±2.3	8.3±1.9	**26.1**±5.3	**10.0**±3.7	**8.9**±1.2	7.3±0.9
	EDL	**10.5**±2.1	**4.5**±0.8	**29.1**±5.1	**10.1**±1.6	**8.2**±1.1	4.6±0.6
	ENP	**9.9**±2.3	6.1±1.0	**41.1**±6.6	**5.2**±2.1	**7.8**±1.2	11.7±1.1
	ETP	**10.1**±2.4	**3.1**±0.8	**26.1**±5.3	**5.2**±2.8	**7.7**±1.1	**3.0**±0.5
3	BNN	**10.7**±2.3	9.0±1.9	**30.0**±7.6	**12.3**±5.3	**9.2**±1.6	7.5±1.2
	EDL	**11.4**±2.1	**4.9**±0.9	**32.6**±7.0	**9.8**±1.5	**8.5**±1.5	**4.7**±0.7
	ENP	**10.8**±2.3	6.4±1.6	**44.8**±9.0	**7.0**±3.4	**8.5**±1.7	11.8±1.5
	ETP	**11.1**±2.4	**3.5**±0.9	**29.8**±7.2	**6.8**±3.8	**8.2**±1.5	**3.2**±0.6
4	BNN	**12.2**±4.2	10.2±3.6	**35.1**±10.9	**15.9**±8.3	**9.9**±2.5	8.0±1.9
	EDL	**12.9**±4.2	**5.6**±1.8	**37.3**±10.0	**9.6**±1.4	**9.1**±2.4	**5.1**±1.0
	ENP	**12.4**±4.3	**6.5**±2.2	**48.9**±11.2	**9.2**±5.4	**9.4**±2.8	12.3±2.2
	ETP	**12.8**±4.6	**4.1**±1.8	**34.8**±10.3	**9.7**±6.4	**9.1**±2.4	**3.2**±1.0
5	BNN	**14.2**±5.4	**11.7**±4.6	**42.4**±13.7	21.0±10.4	**10.2**±2.5	8.2±1.9
	EDL	**15.1**±5.8	**6.7**±3.1	**43.9**±12.5	**9.5**±2.4	**9.4**±2.4	**5.1**±1.1
	ENP	**14.8**±6.5	**6.9**±3.1	**54.6**±12.7	**12.5**±7.0	**9.5**±2.8	12.4±2.3
	ETP	**15.3**±7.1	**5.2**±3.4	**42.1**±13.1	**14.1**±8.6	**9.0**±2.5	**3.3**±1.0

create them following the same procedure as in the original work. For FMNIST we adapt the procedure to gray-scale images by replicating the image three times for each channel. After applying the distortions, we map it back to the gray scale. Table 2.4 gives the numerical results used to create Figure 2.4 in the main paper.

3. CONTINUAL LEARNING OF MULTI-MODAL DYNAMICS WITH EXTERNAL MEMORY

3.1 Introduction to Continual Learning of Multi-Modal Dynamics with External Memory

We introduce a novel problem setup where dynamical system modeling tasks emerge sequentially and a probabilistic SSM is expected to learn them cumulatively. We further assume each task to follow multi-modal dynamics, where each individual sequence of a task follows one of the possible modes that describe the task. Successful CL in such a setup presupposes maximally efficient encoding of tasks into a long-term memory and their accurate retrieval. We curate a novel CL model tailored for addressing this challenging problem. Our model captures the characteristics of unknown modes of sequences of the present task into fixed-sized vectors, called *mode descriptors*, stores these descriptors in an external neural episodic memory addressable via a learnable attention mechanism. Our model represents multiple task modes by feeding these mode descriptors into the state transition kernel as an additional input. We place a Dirichlet Process (DP) prior on the attention weights of the memory to encourage the explanation of the data with minimum number of modes. Figure 3.1 illustrates our model with external memory and the problem with two tasks and four modes. Our resulting Bayesian model can be efficiently trained using a straightforward adaptation of existing variational SSM inference techniques.

We evaluate our model in two real-world time series prediction data sets and three synthetic data sets generated from three challenging nonlinear multi-modal dynamical systems. Our method exhibits consistent performance improvement over the established parameter transfer approach, verifying the importance of parsimonious use of neural episodic memory in efficient within-task knowledge acquisition and effective cross-tasks knowledge transfer.

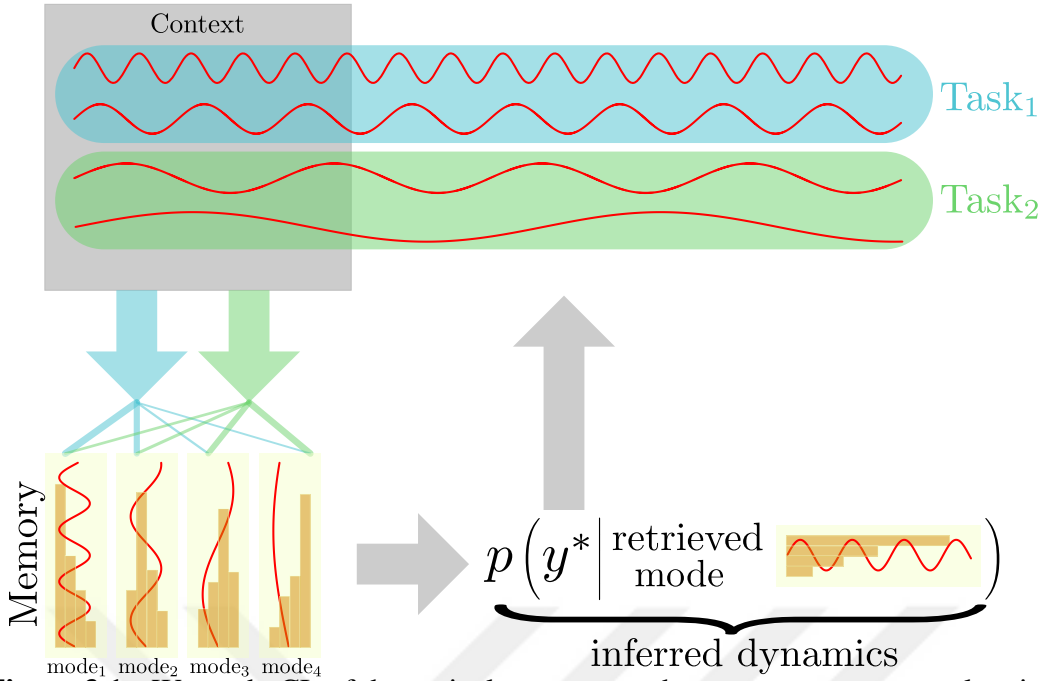


Figure 3.1 : We study CL of dynamical systems, each operate on a non-overlapping set of multiple modes unknown to the learner at each task. We find out that inferring mode descriptors from an observed sequence snipped ($y_{1:C}$), storing them in an external neural episodic memory, retrieving them in the subsequent tasks, and feeding them into the state transition kernel as control input brings improved CL performance. We encourage efficient use of memory space by placing a DP prior on the effective count of encountered modes.

3.2 Problem Setup: Continual Multi-Modal Dynamics Learning

We assume a learning agent that observes a dynamical environment via sequences $y_{1:T} = \{y_1, \ldots, y_T\}$ consisting of time-indexed measurements $y_t \in \mathscr{Y}$ living in a measurable space $\mathscr{Y}$. We denote a snippet of a sequence $y_{t:t'} = \{y_t, \ldots, y_{t'}\}$ for an arbitrary time interval $[t, t']$. Modes refer to general distinguishable properties of a dynamic system such as states with different characteristics of sine wave (amplitude, frequency) or semantically different dynamics such as different character trajectories. We refer all these factors as modes. We define the *mode* of a sequence $y_{1:T}$ as a fixed-sized vector and m as an element of a $K-$dimensional embedding space $\mathscr{X}$. A dynamical system can potentially operate within a large number of modes. Making an analogy to the real world, an autonomous vehicle accounts for different environmental characteristics when planning and control for different weather conditions, countries, and times of a day. Only few of the modes are active for a particular time point and each mode instantly activates and deactivates for limited time periods.

We search for a learning algorithm that enables the agent to fit to a dynamical environment which has perpetually changing global characteristics. *We cast the corresponding learning problem as CL of multi-modal dynamical systems, where each task is defined as a group of modes that modulate a specific dynamical system.* In formal terms, a task $\mathscr{T}_i$ is defined as a group of mode descriptors sampled from a mode generating oracle $P(\mathscr{X})$, that is

$$\mathscr{T}_i = \{m_r^i | m_r \sim P(\mathscr{X}), r = 1, \cdots, R\}, \quad i = 1, \cdots U, \tag{3.1}$$

where R is number of modes per task and U is number of tasks. We assume that each sequence of a task follows dynamics modulated by a mode sampled from $\mathscr{T}_i$

$$\mathscr{D}_{\mathscr{T}_i} = \left\{ y_{1:T}^n \middle| m_n \sim P(\mathscr{T}_i), y_{1:T}^n \sim p(y_{1:T}|m_n), n = 1, \cdots, N \right\}, \tag{3.2}$$

where $P(\mathscr{T}_i)$ is a probability mass function defined on the modes of task $\mathscr{T}_i$ and $p(y_{1:T}|m_r)$ is the probability measure that describes the true behavior of the environment dynamics under mode m_r within a time interval $[1,T]$. The marginal distribution of a task with respect to its modes is given as

$$p(y_{1:T}|\mathscr{T}_i) = \sum_{r=1}^{R} p(y_{1:T}|m_r) P(m_r|\mathscr{T}_i). \tag{3.3}$$

The agent observes a task via a data set $\mathscr{D}_{\mathscr{T}_i}$ that contains only the sequences $y_{1:T}^n$ *but not their corresponding modes*.

We are interested in learning a model that minimizes the true CL risk functional below

$$R_{\mathbb{L}}^{CL}(h_\theta) = \lim_{U \to +\infty} \sum_{i=1}^{U} \mathbb{E}_{\mathscr{T}_i \sim P(\mathscr{X})} \left[\mathbb{E}_{y_{1:T} \sim p(y_{1:T}|\mathscr{T}_i)} \left[\mathbb{L}(h_\theta(\cdot|y_{1:C}), y_{C+1:T}) \right] \right], \tag{3.4}$$

which amounts to the limit of the cumulative risk of individual tasks as they appear one at a time. Above, $h_\theta : \mathscr{Y}^C \to \mathscr{Y}^{T-C}$ is a stochastic process that can map an observed sequence of an arbitrary length C to the subsequent $T-C$ time steps. We call the first C observations in the sequence $y_{1:C}$ as the *context* and define $\mathbb{L}$ as a sequence-specific loss function defined on the future values of the sequence $y_{C+1:T}$. We evaluate the performance of predictions $\widehat{y}$ via two scores:

Normalized Mean Squared Error (NMSE):

$$\mathbb{L}_{NMSE}\left(h_\theta(\cdot|y_{1:C}), y_{C+1:T}\right) = \mathbb{E}_{\widehat{y}_{C+1:T} \sim h_\theta(\cdot|y_{1:C})} \left[\frac{||\widehat{y}_{C+1:T} - y_{C+1:T}||_2^2}{||y_{C+1:T}||_2^2} \right] \tag{3.5}$$

as a measure of the prediction accuracy of h_θ when used as a Gibbs predictor, and

NLL:

$$\mathbb{L}_{NLL}\big(h_\theta(\cdot|y_{1:C}), y_{C+1:T}\big) = \log h_\theta(\widehat{y}_{C+1:T}|y_{C+1:T}) \tag{3.6}$$

as a measure of Bayesian model fit that quantifies the model's own assessment on the uncertainty of its assumptions.

We approximate the true CL risk by its empirical counterpart

$$\widehat{\mathbb{R}}_{\mathbb{L}}^{CL} = \frac{1}{N}\sum_{i=1}^{U}\sum_{n=1}^{N}\mathbb{L}\Big(h_\theta(y^n_{C+1:T}|y^n_{1:C})\Big|\mathscr{T}_i\Big) \tag{3.7}$$

for a finite number U of tasks presented to h_θ one at a time.

3.3 Novel Baseline: Variational Continual Learning (VCL) for Bayesian State-Space Models (BSSM)

We build our target model on a Bayesian treatment of state-space modeling, which is proven to be effective in learning under high uncertainty and knowledge transfer across tasks, as practiced in the seminal prior art of CL [14,15]. We perform approximate inference using variational Bayes due to its multiple successful applications to SSMs [50]–[52] and its favorable computational properties. *As there does not exist any prior work tailored specifically towards CL for dynamical systems, we curate our own baseline*. We adopt the established practice of setting the posterior of the learned parameters of the previous task as the prior of the next one and determine VCL [15] as the state-of-the-art representative of this approach. Elastic Weight Consolidation (EWC) [14] also follows the same approach but uses a simpler posterior inference scheme.

Bayesian State-Space Models: are characterized by the data generating process below

$$\theta \sim p(\theta), \tag{3.8}$$

$$x_0 \sim p(x_0), \tag{3.9}$$

$$x_t|x_{t-1}, \theta \sim p(x_t|x_{t-1}, \theta), \tag{3.10}$$

$$y_t|x_t \sim p(y_t|x_t), \tag{3.11}$$

where x_t and y_t correspond to the latent and observed state variables for time step t, respectively. The system dynamics are modeled by the first-order Markovian transition kernel $p(x_t|x_{t-1},\theta)$ parameterized by θ that in turn follows a prior distribution $p(\theta)$. The latent states are mapped to the observation space via a probabilistic observation model $p(y_t|x_t)$. We formulate the initial latent state x_0 as another random variable that follows the prior distribution $p(x_0)$.

Variational Inference is required to approximate the posterior $p(x_{0:T},\theta|y_{1:T})$, which will be intractable for many choices of distribution families for the data generating process in Eq. 3.10. Following [53], we choose the variational distribution to be mean-field across the parameters of the dynamics and the latent states

$$q_{\xi,\psi}(x_{0:T},\theta|y_{1:T}) = q_\xi(x_0|y_{1:C})\prod_{t=1}^{T} p(x_t|x_{t-1},\theta)q_\psi(\theta), \tag{3.12}$$

where $q_{\xi,\psi}$ is an approximation to the true posterior $p(x_{0:T},\theta|y_{1:T})$ with variational free parameters (ξ,ψ). This formulation has multiple advantages. Firstly, modeling the marginal posterior on the initial latent state x_0 by amortizing on the context observations $y_{1:C}$ makes the ELBO calculation invariant to the context length. Secondly, adopting the prior transition kernel avoids duplicate learning of environment dynamics with twice as many free parameters and prevents training from instabilities caused by the inconsistencies between prior and posterior dynamics. Applying Jensen's inequality in the conventional way, the corresponding ELBO will be

$$\begin{aligned}
&\log p(y_{1:T}) \\
&= \log \int_{X,\theta} p(y_{1:T}|x_{1:T})p(x_{1:T}|\theta,x_0)p(x_0)p(\theta) && (3.13)\\
&= \log \int_{X,\theta} \left\{ \frac{p(y_{1:T}|x_{1:T})p(x_{1:T}|\theta,x_0)p(x_0)p(\theta)}{p(x_{1:T}|\theta,x_0)q_\xi(x_0|y_{1:C})q_\psi(\theta)} \right. && (3.14)\\
&\qquad\qquad \left. p(x_{1:T}|\theta,x_0)q_\xi(x_0|y_{1:C})q_\psi(\theta) \right\} \\
&\geq \mathbb{E}_{p(x_{1:T}|\theta,x_0)q_\xi(x_0|y_{1:C})q_\psi(\theta)}[\log p(y_{1:T}|x_{1:T})] && (3.15)\\
&\qquad - KL(q_\xi(x_0|y_{1:C})||p(x_0)) - KL(q_\psi(\theta)||p(\theta)),
\end{aligned}$$

where $X = \{x_0,\cdots,x_T\}$.

VCL for BSSMs can be implemented as follows. Having fitted the ELBO (Eq. 3.16) ($\mathscr{L}$) on the data set for the first task which has observed sequences $\mathscr{D}_{\mathscr{T}_1}$, we attain $(\xi_1^*, \psi_1^*) = \text{argmax}_{\xi,\psi}\mathscr{L}(\xi, \psi, \mathscr{D}_{\mathscr{T}_1})$. When the next task arrives with data $\mathscr{D}_{\mathscr{T}_2}$, we assign $p(\theta) \leftarrow q_{\psi_1^*}(\theta)$ and maximize the ELBO again $(\xi_2^*, \psi_2^*) = \text{argmax}_{\xi,\psi}\mathscr{L}(\xi, \psi, \mathscr{D}_{\mathscr{T}_2})$. We repeat this process continually for every new coming task. We refer to this newly curated baseline in the rest of the paper as *VCL-BSSM*. We neglect the coreset extension of VCL since its application to BSSMs is tedious and its advantage is not demonstrated with sufficient significance in the static prediction tasks studied in the original work.

3.4 Target Model: The Continual Dynamic Dirichlet Process

The commonplace Bayesian approach to CL transfers knowledge across tasks by assigning the learned posterior on the parameters of the previous task as the prior on the parameters of the current task. This is an effective approach when the subject of transfer is a feed-forward model, such as a classifier in supervised learning setup [15] or a policy network in reinforcement learning [14]. *We conjecture that the existing parameter transfer Ansatz would not be sufficient for the transfer of more complex task properties such as modes of dynamical systems.* We address this problem by tailoring a novel CL approach from an original combination of an aged statistical machine learning tool, the DP, with modern neural episodic memory and attention mechanisms.

Dirichlet Processes are stochastic processes defined on a countably infinite number of categorical outcomes, every finite subset of which follows a multinomial distribution drawn from a Dirichlet prior [54]. A DP follows a Griffiths-Engen-McCloskey (GEM) distribution [55]

$$\pi'_r|\alpha_0 \sim \text{Beta}(1, \alpha_0), \tag{3.16}$$

$$\pi_r = \pi'_r \prod_{j=1}^{r-1}(1 - \pi'_j), \tag{3.17}$$

$$\pi = [\pi_1, \ldots, \pi_R], \tag{3.18}$$

which we denote in short hand as $\pi \sim \text{GEM}(\alpha_0, R)$. The GEM distribution can also be viewed as a stick-breaking process [56,57] where $\alpha_0 > 0$ is a scalar hyper-parameter.

The data generation process below is called a DP for a base measure $G_0(\mathscr{X})$ defined on a $\sigma-$algebra $\mathbb{B}$ of $\mathscr{X}$

$$m_r \sim G_0(\mathscr{X}), \tag{3.19}$$

$$\pi \sim \text{GEM}(\alpha_0, R), \tag{3.20}$$

$$G|\pi = \sum_{r=1}^{R} \pi_r \delta_{m_r}, \tag{3.21}$$

where $\delta_x(A)$ is Dirac delta measure that takes value 1 if $x \in A$ and 0 otherwise for any measurable set $A \in \mathbb{B}$ and G is a categorical distribution with parameters π.

Neural Episodic Memory. We assume that the environment dynamics can be expressed within a measurable latent embedding space $x_t \in \mathscr{X}$, and a sequence encoder $e_\lambda(x_{t:t'})$ parameterized by λ for $t \leq t'$ that maps a sequence of latent embeddings into a fixed dimensional vector, as well as a neural memory $M = \{m_1, \cdots, m_R\}$ consisting of R elements $m_r \in \mathscr{X}$ that live in the same space as latent embeddings x_t. We can construct a probability measure for $\mathbb{B}$ from the memory M by updating

$$m_r \leftarrow (1 - w_r(y_{1:C}, m_r))m_r + w_r(y_{1:C}, m_r)e_\lambda(y_{1:C}) \tag{3.22}$$

for each sequence where

$$w_r(y_{1:C}, m_r) = \frac{e^{\langle m_r, e_\lambda(y_{1:C})\rangle}}{\sum_{j=1}^{R} e^{\langle m_j, e_\lambda(y_{1:C})\rangle}} \tag{3.23}$$

for some similarity function $\langle \cdot, \cdot \rangle : \mathscr{X} \times \mathscr{X} \to \mathbb{R}^+$. This construction imposes a memory attention mechanism, where the encoded mode descriptors attend to the memory elements. Here we make the fair assumption that the modality of a sequence can be identified also from the observation space, while we need to infer the latent representations accurately to model the mode dynamics in detail. We choose an uninformative base measure that assigns equal prior probabilities to memory elements $G_0(M) = \sum_{k=1}^{R} \frac{1}{R} \delta_{m_r}$.

The full model. We complement the BSSM in Eq. 3.10 with an external neural episodic memory M that is updated for each observed sequence with the rule in Eq. 3.22. We place a DP prior on the retrieval of mode descriptors $m_r \in M$ to encourage the model to generate a minimum number of modes. We also feed the retrieved mode

descriptor into the transition kernel $p(x_t|x_{t-1},m,\theta)$ as control input. The resultant model, which we call as the *CDDP*, follows the generative process

$$\pi \sim GEM(\alpha_0, R), \tag{3.24}$$

$$m|\pi \sim \sum_{r=1}^{R} \pi_r \delta_{m_r}, \tag{3.25}$$

$$\theta \sim p(\theta), \tag{3.26}$$

$$x_0 \sim p(x_0), \tag{3.27}$$

$$x_t|x_{t-1},m,\theta \sim p(x_t|x_{t-1},m,\theta), \tag{3.28}$$

$$y_t|x_t \sim p(y_t|x_t), \tag{3.29}$$

where the memory capacity R is set to a bigger number than the expected upper limit of the mode count. Figure 3.2 summarizes the CDDP model design for a single sequence.

Inference. Since $p(x_{0:T},\theta|y_{1:T})$ is intractable, we approximate it by variational inference. We inherit the advantages of the BSSM inference scheme by choosing the variational distribution as

$$q_{\xi,\psi}(x_{0:T},\theta,\pi|y_{1:T}) = \tag{3.30}$$
$$q_\xi(x_0|y_{1:C},\pi)\prod_{t=1}^{T}\sum_{r=1}^{R} q_\xi(\pi=r|y_{1:C})p(x_t|x_{t-1},m_r,\theta)q_\psi(\theta),$$

where

$$q_\psi(\theta) = \mathcal{N}(\theta|\mu,\Sigma), \tag{3.31}$$

$$q_\xi(\pi|y_{1:C}) = Cat(w_1(m_1,y_{1:C}),\ldots,w_K(m_R,y_{1:C})), \tag{3.32}$$

$$q_\xi(x_0|y_{1:C},\pi) = \tag{3.33}$$
$$\sum_{r=1}^{R} \pi_r \mathcal{N}\left(x_0 \middle| v_1(m_r,e_\lambda(y_{1:C})), v_2(m_r,e_\lambda(y_{1:C}))\right)$$

with $x_0 \sim \mathcal{N}(0,I)$. In the expressions above, v_1 and v_2 refer to dense layers. The corresponding ELBO is then calculated following similar lines to Eq. 3.16

$$\log p(y_{1:T}) \tag{3.34}$$

$$= \log \int_{X,\theta} \sum_{r=1}^{R} \pi_r p(y_{1:T}|x_{1:T}) p(x_{1:T}|\theta, x_0, m_r) p(x_0) p(\theta) \tag{3.35}$$

$$= \log \int_{X,\theta} \sum_{r=1}^{R} \pi_r \tag{3.36}$$

$$\left\{ \frac{p(y_{1:T}|x_{1:T}) p(x_{1:T}|\theta, x_0, m_r) p(x_0) p(\theta)}{p(x_{1:T}|\theta, x_0, m_r) q_\xi(x_0|y_{1:C}) q_\xi(\pi = r|y_{1:C}) q_\psi(\theta)} \right.$$

$$\left. p(x_{1:T}|\theta, x_0, m_r) q_\xi(x_0|y_{1:C}) q_\xi(\pi = r|y_{1:C}) q_\psi(\theta) \right\}$$

$$\geq \sum_{r=1}^{R} q_\xi(\pi = r|y_{1:C}) \tag{3.37}$$

$$\mathbb{E}_{p(x_{1:T}|\theta, x_0, m_r) q_\xi(x_0|y_{1:C}) q_\psi(\theta)} [\log p(y_{1:T}|x_{1:T})]$$

$$- KL(q_\xi(x_0|y_{1:C})||p(x_0)) - KL(q_\psi(\theta)||p(\theta))$$

$$- \sum_{r=1}^{R} w_r(m_r, y_{1:C}) \log(w_r(m_r, y_{1:C})/\pi_r).$$

Prediction. Having trained the model on the latest task $\mathscr{T}_i$, its posterior predictive distribution for a new sequence Y^* and corresponding latent embeddings $X = \{x_0, \cdots, x_T\}$ can be computed as

$$p(Y^*_{C+1:T}|\mathscr{D}_{\mathscr{T}_i}, Y^*_{1:C}) = \sum_{r=1}^{R} q(\pi = r|Y^*_{1:C}) \tag{3.38}$$

$$\int_{X,\theta} q_\xi(x_0|Y^*_{1:C}, \pi = r) q_\psi(\theta) \left[\right.$$

$$\left. \prod_{t=C+1}^{T} p(y^*_t|x_t) p(x_t|x_{t-1}, \theta, m_r) \right].$$

3.5 Related Work

State Space Models. Deterministic SSMs, such as the LSTM [58], the Gated Recurrent Unit [7], and Transformer Nets [31] are in frequent use for sequential predictions. There exist a considerable body of work on Bayesian versions of SSMs that employ Gaussian Process as transition kernels [50]–[52] and perform variational inference. Another vein of work named Recurrent SSMs [19,22,23] models the

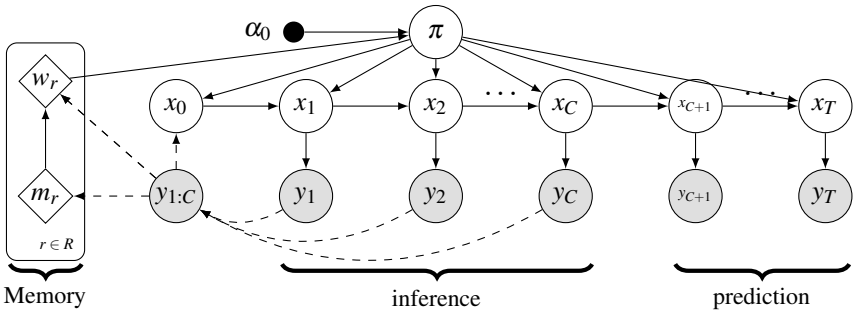


Figure 3.2 : *CDDP Plate Diagram* . Shaded nodes are observed, unshaded nodes are latent, diamonds are deterministic and black nodes are constant. Solid arrows denote the conditional dependencies of the true model and dashed arrows denote the same for the approximate posterior. The first C observations, called *the context*, amortize the inference of the initial hidden state, and also determine the content and addressing of the mode embeddings in the external memory.

transition dynamics as RNN that map a state to the next time step deterministically, while admitting a random state variable from the previous time step as input and feeding its output to the distribution of this variable at the subsequent time step.

Attention and Memory in Neural Nets. NTMs [27] and Differential Neural Computers [59] are first examples of attention-based neural episodic memory use with external updates. Transformer Networks [31] introduce the notion of self attention and effectively use it in conjunction with cross attention to perform sequence-to-sequence prediction for long horizons. The ANP [1] employs attention network to build a neural stochastic process that is consistent over the observed predictions. It has been shown that an attention network can also be viewed as a Hopfield Network [60], which explains its success at memorization of complex patterns. The ETP [25] maintains an external memory that learns to feed a Dirichlet prior on the class distribution with informative concentration parameters inferred during minibatch training updates.

Continual Learning is an instance of meta learning [61]–[63] where new tasks are introduced one at a time and a base model is expected to learn the newest task without forgetting the previous ones. Unlike biological intelligence, artificial intelligence suffers from catastrophic forgetting in continual/lifelong learning. Early approaches to CL such as Synaptic Intelligence (SI) [17] and EWC [14] transfer knowledge by the transfer of either deterministic parameters or their inferred distribution. SI calculates a plasticity measure by comparing the change rate of the gradients of the

loss and the change rate of the synapses, whereas EWC uses Fisher information. VCL [15] improves on EWC with a more comprehensive inference scheme and an additional feedback channel via a core set of most characteristic observations of past tasks. Generalized VCL [64] maximizes the same ELBO as VCL but using β-VAE to prevent the training instability caused by the dominance of the KL-divergence term. We do not use β-VAE since our setup does not have this problem. Sequential Neural Processes [16] captures the evolution of parameters across tasks via a recurrent SSM that determines the sufficient statistics of a NP prior.

[65] explains the superior performance of memorization-based CL algorithms based on experience replay, core set, and episodic memory [15,66,67] over regularization-based algorithms [14,17]. For the first time, our CDDP studies probabilistic multi-modal dynamics in a CL setting by knowledge transfer via learned mode descriptors maintained in external memory.

Continual Learning with Episodic Memory. Closed-loop memory replay GAN [68] builds a memory by maximum sample diversity in order to reduce catastrophic forgetting by remembering samples of previous tasks. Generative memory net [69] creates a memory with two GANs in order to generate adversarial samples for previous tasks to be used in the new task. In [70], a memory keeps sample random examples from previous tasks and these samples are used in the training of the new tasks. There are other works [66,67,71] that uses memory for CL however, all of them build the memory with the help of data points provided with the ground truth but in our setup mode labels are not provided. Hence our work is not comparable to theirs. As neither of them studies dynamics modeling, their adaption to our setup is not straightforward.

3.6 Experiments

We evaluate the performance of CDDP rigorously on five challenging applications of CL to time series forecasting. We provide a PyTorch [46] implementation of our full experiment pipeline that contains studied models as supplement to our submission, data generation process of the used synthetic environments, as well as the performance evaluation procedures.

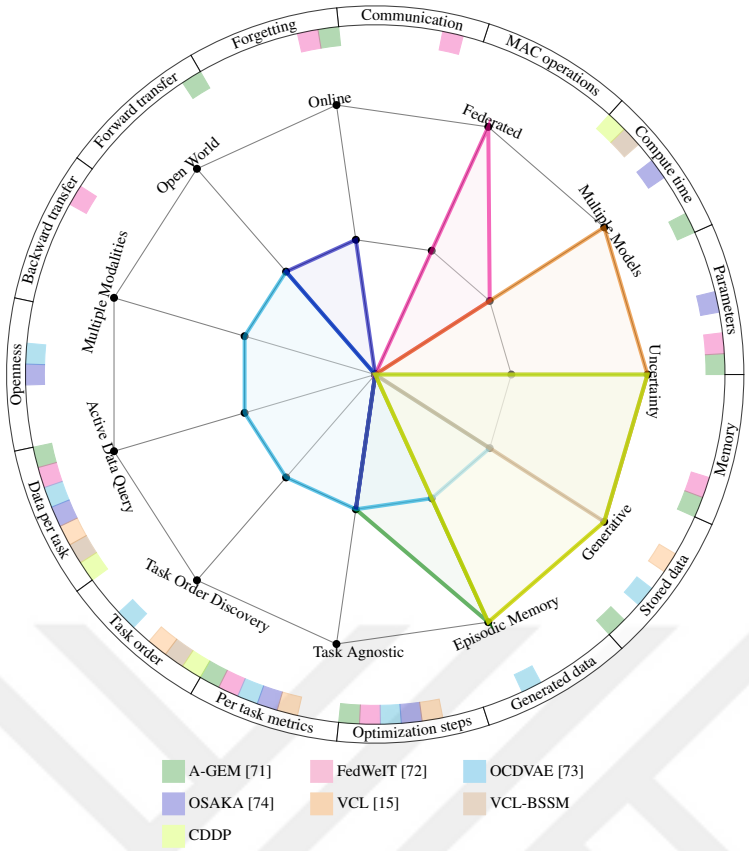


Figure 3.3 : CLEVA Compass.

Baseline Selection. As we are the first to investigate CL for BSSM inference, there is no prior work we can take as a baseline without significant adaptation. We determine knowledge transfer across tasks by setting the parameter posterior of the previous task as the prior of the next as the state of the art approach in CL. We adapt VCL, the most established variant of this approach, to BSSMs in Section 3.3 as a baseline that can maximally challenge our CDDP. To justify our baseline selection, we use the Continual Learning EValuation Assessment (CLEVA) Compass [75] which provides a visual indication for comparison of CL methods. In Figure 3.3, we show CLEVA Compass with five existing CL methods: OSAKA [74], FedWeIT [72], A-GEM [71], VCL [15], and OCDVAE [73]; with proposed novel baseline (VCL-BSSM) and our solution CDDP. Baseline methods must be generative due to the nature of the dynamic modeling problem. Furthermore, as stated in the problem setup (Section 3.2), the correct mode labels are not provided therefore baseline methods need to be unsupervised generative. No method meets these requirements but we adapt the VCL method.

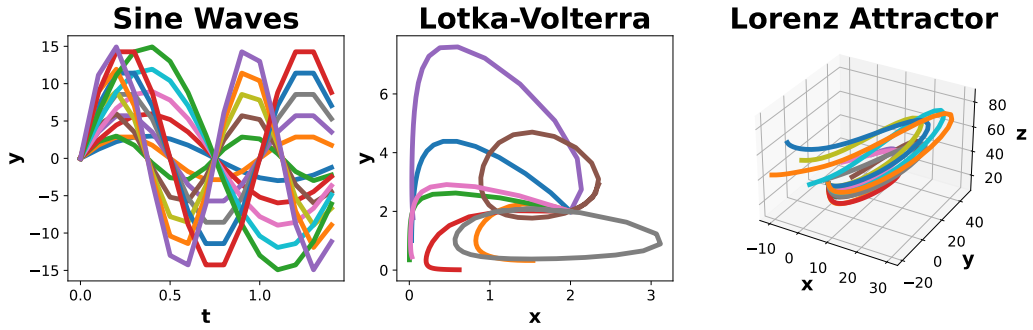


Figure 3.4 : Sample sequences collected from different modes of the dynamical systems from synthetic data sets. Colors correspond to different modes. For each data set, modes live in the same dynamical system but differ in the free parameters that govern the dynamics. For instance, the Sine Waves differ in magnitude and frequency.

3.6.1 Data sets

3.6.1.1 Synthetic data sets

We evaluate our CDDP on three nonlinear dynamical systems, where forecasting is challenging, while ground-truth task similarity is controllable. Too much task similarity would make CL unnecessary, while too much task difference would make it infeasible. When tasks share a reasonable degree of similarity, a successful CL algorithm is expected to capture, encode, and memorize these similarities and discard their differences. We perturb all three dynamical systems with Gaussian white noise.

We generate modes that share similar dynamical properties, as their time evolution follows the same set of differential equations. However, modes differ from each other in the choice of the free parameters that govern the dynamical system. **i)Sine Waves** data set consists of signals grouped into modes described by different magnitudes and frequencies. Sine waves are created from the function $A\sin(2\pi ft)$ where A is the magnitude, f is the frequency, and t is time. We generate different modes from five different choices for magnitudes and three frequency levels. **ii) Lotka-Volterra** equations stem from modeling the dynamics of predator and prey populations and are defined as the nonlinear differential equations: $dx_t/dt = \alpha x_t - \beta x_t y_t, dy_t/dt = \delta x_t y_t - \gamma y_t$. We generate different modes by setting the free parameter pairs (α, δ) and (β, γ) to different values. **iii) Lorenz Attractor** is a three-dimensional nonlinear dynamical system developed originally to model atmospheric convection. As the

Lorenz attractor is chaos-theoretic, hence not solvable with predictable consistency using the existing numerical methods, it is often used as a challenging environment to evaluate probabilistic dynamical models [76,77]. The Lorenz system is described by $dx_t/dt = \sigma(y_t - x_t), dy_t/dt = x_t(\rho - z_t) - y_t, dz_t/dt = x_t y_t - \beta z_t$. We determine our modes by settings the free parameters σ, ρ, and β of the Lorenz system to different values.

Figure 3.4 presents a sample visualization of modes that constitute the CL tasks on each of the three dynamical systems.

3.6.1.2 Real-world data sets

We evaluate our CDDP in two real-world time series classification data sets from [78]. **i) Libras Movement Data Set** is the official Brazilian sign language and acronym of the Portuguese name "LÍngua BRAsileira de Sinais". The data set consists of sequences of (x, y) coordinates of hand movements from 15 different classes. **ii) Character Trajectories** data set consists of sequences of velocities of (x, y) coordinates and pen force collected from hand-writings of 20 English alphabet characters, which can be written by a single pen-down segment. Treating each class as a mode, we generate multimodal time series forecasting tasks from these two data sets that satisfy the learning setup in Eq. 3.4.

3.6.2 Hyper-parameter selection

Unlike many commonplace CL benchmark setups that do few-way low-dimensional image classification, BSSM inference of even a single dynamical system necessitates careful tuning of hyper-parameters. Thus, we set latent state dimensionality, learning rate, and the hidden layer sizes of the encoder, transition kernel, and decoder networks to values that maximize the performance of the BSSM on the first task. Notably, the selection favors VCL and CDDP equally.

Each experiment repetition presents the tasks to the CL models at random order.

3.6.2.1 Hyper-parameters on data

i) Synthetic Data Sets: Each synthetic data set has 12 train sequences and 6 test sequences per mode. There are 15 modes 5 tasks in Sine Waves, 8 modes 4 tasks in Lotka-Volterra and 12 modes 4 tasks in Lorenz Attractor. We set the sequence lengths and time steps $(T, \Delta t)$ to $(15, 0.1)$ for Sine Waves, $(25, 0.4)$ for Lotka-Volterra , and $(50, 0.01)$ for the Lorenz Attractor. For each mode in the whole data set, we generate a long sequence and divide the whole sequence into subsequences of length T with a certain space between them. For all synthetic data sets, fixed jump space between subsequences is 5 time-point. By adding additional Gaussian noise to the train split, we construct train and test splits with these subsequences.

Sine Waves) We generate different modes from five different choices for $A \in [3, 6, 9, 12, 15]$, and three frequency levels for $f \in \left[\frac{2}{3}, 1, \frac{4}{3}\right]$. We add Gaussian noise from $\mathcal{N}(0, \sigma)$ and set σ to $\frac{A}{100}$ for each mode to the train split.

Lotka-Volterra) We generate different modes by setting the free parameter pairs (α, δ) and (β, γ) to different values. Starting with a fixed point $(x_0, y_0) = (2, 2)$, we select levels as $\alpha, \beta, \gamma \in [0.25, 0.75]$, and a constant $\delta = 0.5$. We add a Gaussian noise from $\mathcal{N}(0, 0.001)$ to the train split.

Lorenz Attractor) We determine our modes by settings the free parameters σ, ρ, and β of the Lorenz system to different values. We start with initial state $(x_0, y_0, z_0) = (1, 1, 28)$ and generate a long sequence. We generate modes with different but partially overlapping dynamical characteristics by setting ρ to three and σ and β to two different values. We select the three free parameters levels as $\rho \in [28, 42, 56], \sigma \in [8, 12]$, and $\beta \in \left[\frac{5}{3}, \frac{13}{3}\right]$. We add Gaussian noise from $\mathcal{N}(0, 0.01)$ to the train split.

ii) Libras: The data set has 12 train sequences and 12 test sequences per mode. There are 15 modes 5 tasks. The sequence lengths and time step $(T, \Delta t)$ are (45, $\frac{7}{45}$). There are no fixed Δ for this data set but to find it we divide the approximate seconds taken for a sample to the number of frame count.

Table 3.1 : Summary of data sets.

Type	Data Set	# of Tasks	Total # of Modes	# of Modes per Task	Total # of Sequences Train/Test	Sequence Length (T)	# of Attributes
Synthetic	Sine Waves	5	15	3	180/90	15	1
	Lotka-Volterra	4	8	2	96/48	25	2
	Lorenz Attractor	4	12	3	144/72	50	3
Real World	Libras	5	15	3	180/180	45	2
	Character Trajectories	5	20	4	1422/1436	109	3

iii) Character Trajectories: The data set has different number of instances per modes therefore, we divide each instances samples to two halves then first half is placed in the train split and rest of them are placed in the test split. Hence, train split has 1422 samples, and test split with 1436 samples In the data set, each character is created with a different sequence length, to adapt the data set to our pipeline we subsample the sequences with the smallest sequence length which is 109. The sequence lengths and time step $(T, \Delta t)$ are (109, 0.05). There are 20 modes (characters) and 5 tasks.

Data set details are summarized in Table 3.1.

3.6.2.2 Context length

We select context length empirically as one-third of the sequence length amounting to five for Sine Waves, eight for Lotka-Volterra, 16 for the Lorenz Attractor, 15 for Libras, and 35 for Character Trajectories. The reason for that is, generally, one-third of the sequence captures the general characteristic of the dynamics and gives indications about modes.

3.6.2.3 Hyper-parameters on architectures

Our CDDP has four main architectural elements:

i) Encoder: The sequence encoder $e_\lambda(x_{t:t'})$ governs the mean of our normal distributed recognition model $q_\psi(x_0|y_{1:C}, \pi)$. We feed C observations as a stacked set of values into the encoder. The sequence encoder is a single dense layer for Sine Waves, Lotka-Volterra, Libras; and a multi-layer perceptron for the Lorenz Attractor, and Character Trajectories with two hidden layers of size 90. The perceptron uses the $tanh(\cdot)$ activation function followed by layer normalization.

Table 3.2 : The summary of our main results given as the Area Under Curve (AUC) plotting the change of two performance metrics *NMSE* and *NLL* as a function of the number of learned tasks. The reported numbers are mean ± standard error over 10 repetitions. The detailed results are provided in Table 3.3. Our CDDP outperforms VCL-BSSM in nearly all cases indicating the benefit of maintaining an external memory of mode descriptors in CL, the central hypothesis of this work.

	AUC NMSE		**AUC NLL**	
Data Set	**VCL-BSSM**	**CDDP**	**VCL-BSSM**	**CDDP**
Sine Waves	1.00 ± 0.04	$\mathbf{0.91 \pm 0.03}$	3.57 ± 0.09	$\mathbf{3.50 \pm 0.09}$
Lotka-Volterra	$\mathbf{0.58 \pm 0.04}$	0.60 ± 0.06	1.50 ± 0.05	$\mathbf{1.32 \pm 0.08}$
Lorenz Attractor	0.26 ± 0.00	$\mathbf{0.24 \pm 0.01}$	4.42 ± 0.04	$\mathbf{4.35 \pm 0.06}$
Libras	$\mathbf{0.14 \pm 0.00}$	$\mathbf{0.14 \pm 0.00}$	-0.37 ± 0.02	$\mathbf{-0.39 \pm 0.04}$
Character Trajectories	0.87 ± 0.04	$\mathbf{0.64 \pm 0.01}$	0.14 ± 0.02	$\mathbf{-0.19 \pm 0.03}$

ii) Decoder: The likelihood function $p(y_t|x_t)$ of the base model serves as a probabilistic decoder that maps the latent state x_t to the observed state y_t. We choose the emission distribution to be normal with mean governed by a single dense layer for Sine Waves, Lotka-Volterra, Libras; and a MLP with two hidden layers of size 90 for the Lorenz Attractor, and Character Trajectories. The perceptron uses the $tanh(\cdot)$ activation function followed by layer normalization.

iii) Transition Kernel: We choose the transition kernel $p(x_t|x_{t-1}, m, \theta)$ of CDDP to be a normal distribution with mean governed by a plain RNN that receives a concatenation of the previous hidden state and the mode descriptor as input. All covariates are set to a fixed variance of 0.1. The RNN on the mean is a MLP with one hidden layer of size 40 for Sine Waves, Lotka-Volterra, and Libras; and 90 for the Lorenz Attractor, Character Trajectories. The perceptron uses the $tanh(\cdot)$ activation function followed by layer normalization. The transition kernel of the base model of VCL $p(x_t|x_{t-1}, \theta)$ follows the same RNN architecture except that its input does not contain a mode descriptor.

iv) External Memory: We set the memory size to 20 for the Sine Wave environment, 10 for Lotka-Volterra, 15 for the Lorenz Attractor, 20 for Libras, and 30 for Character Trajectories.

Table 3.3 : Quantitative results of models on five data sets. The reported numbers are mean ± standard error over 10 repetitions. The reported scores are taken after training a task and evaluated along with previously seen tasks.

Data Set	Model	Score	1 Task	2 Tasks	3 Tasks	4 Tasks	5 Tasks
Sine Waves	VCL-BSSM	NMSE	$\mathbf{0.13 \pm 0.02}$	0.97 ± 0.14	1.22 ± 0.15	1.27 ± 0.09	1.39 ± 0.08
		NLL	$\mathbf{1.76 \pm 0.22}$	3.75 ± 0.24	3.84 ± 0.13	4.20 ± 0.15	4.29 ± 0.10
	CDDP	NMSE	0.16 ± 0.02	$\mathbf{0.88 \pm 0.11}$	$\mathbf{1.07 \pm 0.15}$	$\mathbf{1.12 \pm 0.14}$	$\mathbf{1.31 \pm 0.11}$
		NLL	2.09 ± 0.27	$\mathbf{3.43 \pm 0.22}$	$\mathbf{3.77 \pm 0.17}$	$\mathbf{4.05 \pm 0.14}$	$\mathbf{4.15 \pm 0.10}$
Lotka-Volterra	VCL-BSSM	NMSE	0.17 ± 0.03	$\mathbf{0.71 \pm 0.08}$	$\mathbf{0.92 \pm 0.15}$	$\mathbf{0.53 \pm 0.03}$	
		NLL	0.56 ± 0.20	1.97 ± 0.28	$\mathbf{2.02 \pm 0.19}$	$\mathbf{1.46 \pm 0.12}$	
	CDDP	NMSE	$\mathbf{0.13 \pm 0.03}$	0.72 ± 0.20	0.96 ± 0.18	0.60 ± 0.05	
		NLL	$\mathbf{0.32 \pm 0.18}$	$\mathbf{1.35 \pm 0.21}$	2.05 ± 0.15	1.54 ± 0.21	
Lorenz Attractor	VCL-BSSM	NMSE	0.22 ± 0.02	0.25 ± 0.03	0.27 ± 0.02	0.28 ± 0.01	
		NLL	4.22 ± 0.16	4.41 ± 0.08	4.55 ± 0.09	4.52 ± 0.04	
	CDDP	NMSE	$\mathbf{0.21 \pm 0.02}$	$\mathbf{0.24 \pm 0.02}$	$\mathbf{0.25 \pm 0.02}$	$\mathbf{0.26 \pm 0.02}$	
		NLL	$\mathbf{4.06 \pm 0.19}$	$\mathbf{4.33 \pm 0.02}$	$\mathbf{4.53 \pm 0.06}$	$\mathbf{4.46 \pm 0.03}$	
Libras	VCL-BSSM	NMSE	$\mathbf{0.13 \pm 0.01}$	0.14 ± 0.00	$\mathbf{0.13 \pm 0.01}$	0.15 ± 0.00	$\mathbf{0.14 \pm 0.00}$
		NLL	-0.41 ± 0.04	-0.35 ± 0.03	$\mathbf{-0.42 \pm 0.04}$	-0.32 ± 0.03	-0.36 ± 0.03
	CDDP	NMSE	0.14 ± 0.00	$\mathbf{0.13 \pm 0.01}$	0.14 ± 0.01	$\mathbf{0.14 \pm 0.01}$	$\mathbf{0.14 \pm 0.01}$
		NLL	$\mathbf{-0.42 \pm 0.03}$	$\mathbf{-0.40 \pm 0.04}$	-0.38 ± 0.05	$\mathbf{-0.40 \pm 0.03}$	$\mathbf{-0.37 \pm 0.04}$
Character Trajecto-ries	VCL-BSSM	NMSE	$\mathbf{0.42 \pm 0.04}$	0.80 ± 0.04	0.90 ± 0.02	1.11 ± 0.08	1.11 ± 0.10
		NLL	$\mathbf{-0.23 \pm 0.06}$	0.06 ± 0.05	0.20 ± 0.05	0.31 ± 0.08	0.35 ± 0.09
	CDDP	NMSE	0.52 ± 0.05	$\mathbf{0.56 \pm 0.02}$	$\mathbf{0.69 \pm 0.04}$	$\mathbf{0.71 \pm 0.03}$	$\mathbf{0.71 \pm 0.02}$
		NLL	-0.05 ± 0.05	$\mathbf{-0.26 \pm 0.04}$	$\mathbf{-0.17 \pm 0.03}$	$\mathbf{-0.23 \pm 0.04}$	$\mathbf{-0.26 \pm 0.04}$

3.6.2.4 Hyper-parameters on experimental pipeline

We select Adam [45] as optimizer with learning rates of 0.005 for Sine Wawes, 0.001 for Lotka-Volterra, 0.0005 for Lorenz Attractor, 0.001 for Libras, and 0.001 for Character Trajectories. For all data sets, the batch size is equal to 9 except for Character Trajectories which is equal to 64 due to high number of instances. We train the models 300 for Sine Waves, 750 for Lotka-Volterra, 500 for Lorenz Attractor, 300 for Libras, and 2000 for Character Trajectories epochs per task.

We performed our experiments on Intel Xeon CPU E5-2650 v3 with 10 cores and Intel Core(TM) i7-6700K CPU with 4 cores.

3.6.3 Results and ablation study

Main Results. Table 3.2 summarizes model performance throughout the whole CL process as the area under learning curve. Our CDDP outperforms the parameter transfer based VCL-BSSM baseline consistently in nearly all cases. Storing mode

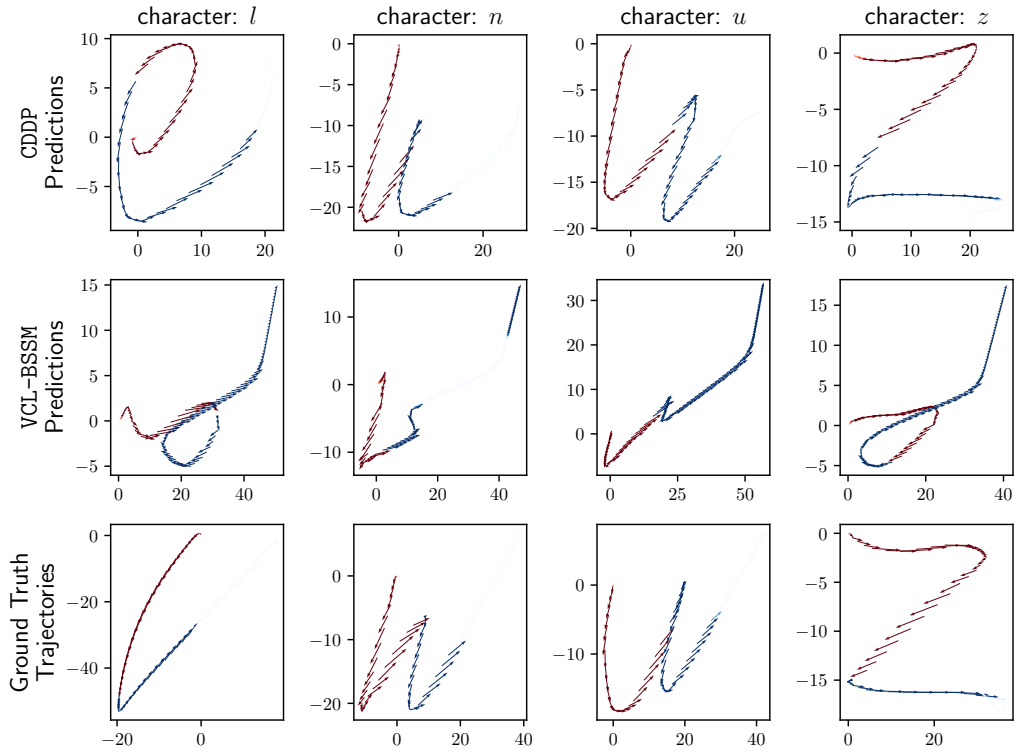


Figure 3.5 : Predictions of CDDP (top row) and VCL-BSSM (middle row) on sample sequences from four tasks of the Character Trajectories data set after CL is over. Bottom row shows the corresponding ground-truth trajectories. The first one-third of each trajectory is considered as context (plotted in red), and the rest is predicted by the models (plotted in blue). Our CDDP remembers the first task much more precisely than VCL-BSSM (first and second columns) after adapting itself accurately to the last task (third and fourth columns). This is achieved by virtue of its external episodic memory.

descriptors of the learned dynamics in an external memory, retrieving them in the subsequent tasks, and feeding them into the state transition kernel prevents catastrophic forgetting more effectively than plain parameter transfer. As seen in Figure 3.5, in the challenging character trajectories data set, the prediction accuracy of our CDDP significantly outperforms VCL-BSSM in both early and late stages of the CL period.

Computational Cost. The difference of the computational footprints of our neural episodic memory based approach and the parameter transfer approach is negligible. The average wall-clock Time Per Epoch (TPE) measured on the Sine Wave data set is 0.20 ± 0.00 seconds for VCL-BSSM and 0.21 ± 0.00 seconds for CDDP. The computational cost of CDDP increases linearly with memory size, which is balanced by its parsimonious memory usage by virtue of the DP prior.

Table 3.4 : Ablation study results given as the Area Under Curve (AUC) on the Sine Waves data set. The reported numbers are mean ± standard error over 5 repetitions.

	Parameter Transfer	**Probabilistic Parameters**	**Memory Content**	**NMSE**	**NLL**
RNN	✓	×	N/A	1.03 ± 0.05	3.50 ± 0.10
VCL-BSSM	✓	✓	N/A	0.93 ± 0.03	3.59 ± 0.16
CDDP Variants	×	✓	Zeros	0.94 ± 0.04	3.56 ± 0.12
	×	✓	Ones	1.12 ± 0.13	3.75 ± 0.14
	×	✓	Twos	1.35 ± 0.23	3.80 ± 0.18
	✓	✓	Learned	0.91 ± 0.04	3.56 ± 0.12
CDDP Target	×	✓	Learned	$\mathbf{0.87 \pm 0.03}$	$\mathbf{3.35 \pm 0.15}$

Ablation Study. We investigate the contribution of individual design choices to the total performance of our target model. We study the effect of three design choices: i) knowledge transfer via parameters θ of the learned transition dynamics, ii) quantifying the uncertainty of the parameters of transition dynamics by a distribution $q_\psi(\theta)$, and iii) maintaining an external memory with learned or unlearned content. Table 3.4 shows a map of model variants corresponding to the activation status of these three design choices, as well as the corresponding numerical results on the Sine Waves data set over five repetitions. We observe a performance increase when knowledge transfer is done via the external memory *instead of —but not together with —* parameter transfer, supporting the central assumptions of our target model. Setting the memory content to unlearned values causes rapid performance deterioration as values diverge from the learned values. This outcome demonstrates the essential role of the external memory in the CL performance of CDDP.

3.7 Broad Impact, Limitations and Ethical Concerns

Broad Impact. Our work can be used for numerous applications such as: i) weather forecasting, where features can be transferred from one climate to another, ii) autonomous driving, where driving patterns can be adapted across different countries, and iii) model-based reinforcement learning algorithms, when the environment changes due either to the actions of the ego agent or to external factors. The memory architecture of CDDP may be improved by alternative embedding, update, and attention mechanisms. Our formulation of the transition rule is agnostic to the

architecture that governs the transition kernel. Transformer Networks [31], Neural Stochastic Differential Equations [76], or a blend of recent approaches can be used as a transition rule instead of the plain RNN.

Limitations and Ethical Concerns. CDDP demonstrates high sensitivity to the performance of its base model on individual tasks. The interpretability of the memory descriptors of the CDDP deserves further investigation. We present CDDP as a general-purpose method, hence its use in fairness-sensitive applications may require post-hoc allocation of existing fairness-inducing algorithms. The uncertainty calibration footprint of CDDP calls for another empirical study prior to its deployment into safety-critical setups.

4. CONCLUSION

In ETP, we initially develop the first formal definition of total calibration. Then we analyze how the design of two mainstream Bayesian modeling approaches affect their uncertainty characteristics. Next, we introduce CBMs as a unifying framework that inherits the complementary strengths of existing ones. We develop the ETP as an optimal realization of CBMs. We derive an experiment setup from our formal definition of total calibration and a systematic ablation of our target model into the strongest representatives of the state of the art. We observe in five real-world tasks that the ETP is the only model that can excel in all three aspects of total calibration simultaneously.

In CDDP, we report the first study on CL of multi-modal dynamical systems. We curate a competitive baseline for this new problem setup from an adaptation of VCL to BSSMs. We introduce a novel alternative to the parameter transfer approach of VCL for within-task knowledge acquisition and cross-task knowledge transfer using an original combination of neural episodic memory, DPs, and BSSMs. We observe in CL of challenging multi-modal dynamics modeling that our alternative approach compares favorably to the established parameter transfer approach.

In this thesis, we show that external memory is highly beneficial for problems of probabilistic modeling, based on the experiments. According to the experimental results, external memory helps neural networks to fit in-domain data, provide calibrated predictions, and detect out-of-domain queries simultaneously. Furthermore, external memory allows neural networks to avoid the catastrophic forgetting problem by transferring mode descriptors of the multi-modal dynamics.

4.1 Future Directions

We summarize future directions may lead by this thesis as follows:

For ETP: We evaluate our model ETP in some important vision tasks; however, the application and evaluation of ETP in safety-critical areas such as health care could be a future direction. Furthermore, the performance of the ETP with transformer nets and deeper neural networks could be another future direction because ETP is not tested with whose architectures.

For CDDP: One of the future directions is evaluating the performance of our model CDDP with model-based reinforcement learning. Furthermore, the interpretation of mode descriptors and the explainability of external memory are other leading research topics.

REFERENCES

[1] **Kim, H.**, **Mnih, A.**, **Schwarz, J.**, **Garnelo, M.**, **Eslami, A.**, **Rosenbaum, D.**, **Vinyals, O. and Teh, Y.** (2019). Attentive Neural Processes, *ICLR*.

[2] **Krizhevsky, A.**, **Sutskever, I. and Hinton, G.** (2012). ImageNet Classification with Deep Convolutional Neural Networks, *NeurIPS*.

[3] **He, K.**, **Zhang, X.**, **Ren, S. and Sun, J.** (2016). Deep residual learning for image recognition, *CVPR*.

[4] **Ren, S.**, **He, K.**, **Girshick, R. and Sun, J.** (2015). Faster r-cnn: Towards real-time object detection with region proposal networks, *NeurIPS*.

[5] **Redmon, J.**, **Divvala, S.**, **Girshick, R. and Farhadi, A.** (2016). You only look once: Unified, real-time object detection, *CVPR*.

[6] **Hochreiter, S. and Schmidhuber, J.** (1997). Long Short-Term Memory, *Neural Computation*, *9*(8), 1735–1780.

[7] **Cho, K.**, **van Merrienboer, B.**, **Bahdanau, D. and Bengio, Y.** (2014). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches, *arXiv preprint arXiv:1409.1259*.

[8] **Schwalbe, G. and Schels, M.** (2020). A Survey on Methods for the Safety Assurance of Machine Learning Based Systems, *European Congress on Embedded Real Time Software and Systems*.

[9] **Naeini, M.**, **Cooper, G. and Hauskrecht, M.** (2015). Obtaining Well Calibrated Probabilities using Bayesian Binning, *AAAI*.

[10] **Guo, C.**, **Pleiss, G.**, **Sun, Y. and Weinberger, K.** (2017). On Calibration of Modern Neural Networks, *ICML*.

[11] **Kuleshov, V.**, **Fenner, N. and Ermon, S.** (2018). Accurate Uncertainties for Deep Learning using Calibrated Regression, *ICML*.

[12] **Sensoy, M.**, **Kaplan, L. and Kandemir, M.** (2018). Evidential Deep Learning to Quantify Classification Uncertainty, *NeurIPS*.

[13] **Malinin, A. and Gales, M.** (2018). Predictive Uncertainty Estimation via Prior Networks, *NeurIPS*.

[14] **Kirkpatrick, J.**, **Pascanu, R.**, **Rabinowitz, N.**, **Veness, J.**, **Desjardins, G.**, **Rusu, A.**, **Milan, K.**, **Quan, J.**, **Ramalho, T.**, **Grabska-Barwinska, A.**, **Hassabis, D.**, **Clopath, C.**, **Kumaran, D. and Hadsell, R.** (2017). Overcoming catastrophic forgetting in neural networks, *Proceedings of the National Academy of Sciences*, *114*(13).

[15] **Nguyen, C.**, **Li, Y.**, **Bui, T. and Turner, R.** (2018). Variational Continual Learning, *ICLR*.

[16] **Singh, G.**, **Yoon, J.**, **Son, Y. and Ahn, S.** (2019). Sequential Neural Processes, *NeurIPS*.

[17] **Zenke, F.**, **Poole, B. and Ganguli, S.** (2017). Continual Learning Through Synaptic Intelligence, *ICML*.

[18] **Garnelo, M.**, **Schwarz, J.**, **Rosenbaum, D.**, **Viola, F.**, **Rezende, D.**, **Eslami, S. and Teh, Y.** (2018). Neural Processes, *ICML WS on Theoretical Foundations and Applications of Deep Generative Models*.

[19] **Fraccaro, M.**, **Rezende, D.**, **Zwols, Z.**, **Pritzel, A.**, **Eslami, S. and Viola, F.** (2018). Generative Temporal Models with Spatial Memory for Partially Observed Environments, *ICML*.

[20] **Cossu, A.**, **Carta, A.**, **Lomonaco, V. and Bacciu, D.** (2021). Continual learning for recurrent neural networks: An empirical evaluation, *Neural Networks*, *143*, 607–627.

[21] **Deisenroth, M. and Rasmussen, C.** (2011). PILCO: A Model-Based and Data-Efficient Approach to Policy Search, *ICML*.

[22] **Hafner, D.**, **Lillicrap, T.**, **Fischer, I.**, **Villegas, R.**, **Ha, D.**, **Lee, H. and Davidson, J.** (2019). Learning Latent Dynamics for Planning from Pixels, *ICML*.

[23] **Hafner, D.**, **Lillicrap, T.**, **Ba, J. and Norouzi, M.** (2020). Dream to Control: Learning Behaviors by Latent Imagination, *ICLR*.

[24] **Åström, K.** (2012). *Introduction to stochastic control theory*, Courier Corporation.

[25] **Kandemir, M.**, **Akgül, A.**, **Haussmann, M. and Unal, G.** (2022). Evidential Turing Processes, *ICLR*.

[26] **Akgül, A.**, **Unal, G. and Kandemir, M.** (2022). *Continual Learning of Multi-modal Dynamics with External Memory*.

[27] **Graves, A.**, **Wayne, G. and Danihelka, I.** (2014). Neural Turing Machines, *arXiv preprint arXiv:1410.5401*.

[28] **Higgins, I.**, **Matthey, L.**, **Pal, A.**, **Burgess, C.**, **Glorot, X.**, **Botvinick, M.**, **Mohamed, S. and Lerchner, A.** (2017). β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework, *ICLR*.

[29] **Chen, W.**, **Shen, Y.**, **Jin, H. and Wang, W.** (2019). A Variational Dirichlet Framework for Out-of-Distribution Detection, *abs/1811.07308*.

[30] **Øksendal, B.** (1992). *Stochastic Differential Equations: An Introduction with Applications*, Springer-Verlag.

[31] **Vaswani, A.**, **Shazeer, N.**, **Parmar, N.**, **Uszkoreit, J.**, **Jones, L.**, **Gomez, A.**, **Kaiser, L. and Polosukhin, I.** (2017). Attention is All You Need, *NeurIPS*.

[32] **Neal, R.** (1995). *Bayesian Learning for Neural Networks*.

[33] **MacKay, D.** (1992). A Practical Bayesian Framework for Backpropagation Networks, *Neural Computation*.

[34] **MacKay, D.** (1995). Probable Networks and Plausible Predictions – A Review of Practical Bayesian Methods for Supervised Neural Networks, *Network: Computation in Neural Systems*.

[35] **Kingma, D.**, **Salimans, T. and Welling, M.** (2015). Variational Dropout and the Local Reparameterization Trick, *NeurIPS*.

[36] **Molchanov, D.**, **Ashukha, A. and Vetrov, D.** (2017). Variational Dropout Sparsifies Deep Neural Networks, *ICML*.

[37] **Garnelo, M.**, **Rosenbaum, D.**, **Maddison, C.**, **Ramalho, T.**, **Saxton, D.**, **Shanahan, M.**, **Teh, Y.**, **Rezende, D. and Eslami, S.** (2018). Conditional Neural Processes, *ICML*.

[38] **Gordon, J.**, **Bruinsma, W.**, **Foong, A.**, **Requeima, J.**, **Dubois, Y. and Turner, R.** (2020). Convolutional Conditional Neural Processes, *ICLR*.

[39] **Foong, A.**, **Bruinsma, W.**, **Gordon, J.**, **Dubois, Y.**, **Requeima, J. and Turner, R.** (2020). Meta-Learning Stationary Stochastic Process Prediction with Convolutional Neural Processes, *NeurIPS*.

[40] **Singh, G.**, **Yoon, J.**, **Son, Y. and Ahn, S.** (2019). Sequential Neural Processes, *NeurIPS*.

[41] **Yoon, J.**, **Singh, G. and Ahn, S.** (2020). Robustifying Sequential Neural Processes, *ICML*.

[42] **Hendrycks, D. and Dietterich, T.** (2019). Benchmarking Neural Network Robustness to Common Corruptions and Perturbations, *ICML*.

[43] **Lorenz, K.** (1978). *Die Rückseite des Spiegels: Versuch einer Naturgeschichte menschlichen Erkennens*, Piper.

[44] **Abramowitz, M. and Stegun, I.** (1965). *Handbook of Mathematical Functions*, Dover.

[45] **Kingma, D. and Ba, J.** (2014). Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980*.

[46] **Paszke, A.**, **Gross, S.**, **Massa, F.**, **Lerer, A.**, **Bradbury, J.**, **Chanan, G.**, **Killeen, T.**, **Lin, Z.**, **Gimelshein, N.**, **Antiga, L.**, **Desmaison, A.**, **Kopf, A.**, **Yang, E.**, **DeVito, Z.**, **Raison, M.**, **Tejani, A.**, **Chilamkurthy, S.**, **Steiner, B.**, **Fang, L.**, **Bai, J. and Chintala, S.**, (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library, NeurIPS.

[47] **LeCun, Y.**, **Cortes, C. and Burges, C.** (2010). MNIST Handwritten Digit Database, *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, *2*.

[48] **Netzer, Y.**, **Wang, T.**, **Coates, A.**, **Bissacco, A.**, **Wu, B. and Ng, A.** (2011). Reading Digits in Natural Images with Unsupervised Feature Learning, *NeurIPS WS on Deep Learning and Unsupervised Feature Learning*.

[49] **Robbins, H. and Monro, S.** (1951). A Stochastic Approximation Method, *The Annals of Mathematical Statistics*, *22*(3), 400 – 407.

[50] **Frigola, R.**, **Chen, Y. and Rasmussen, C.E.** (2014). Variational Gaussian Process State-Space Models, *NeurIPS*.

[51] **Doerr, A.**, **Daniel, C.**, **Schiegg, M.**, **Duy, N.**, **Schaal, S.**, **Toussaint, M. and Sebastian, T.** (2018). Probabilistic Recurrent State-Space Models, *ICML*.

[52] **Ialongo, A.**, **van der Wilk, M.**, **Hensman, J. and Rasmussen, C.** (2019). Overcoming Mean-Field Approximations in Recurrent Gaussian Process Models, *ICML*.

[53] **Yildiz, C.**, **Heinonen, M. and Lähdesmäki, H.** (2019). ODE^{2}VAE: Deep generative second order ODEs with Bayesian neural networks, *ICML*.

[54] **Teh, Y.**, **Jordan, M.**, **Beal, M. and Blei, D.** (2006). Hierarchical Dirichlet Processes, *Journal of the American Statistical Association*, *101*(476), 1566–1581.

[55] **Pitman, J.** (2002). Combinatorial stochastic processes, **Technical Report**.

[56] **Sethuraman, J.** (1994). A CONSTRUCTIVE DEFINITION OF DIRICHLET PRIORS, *Statistica Sinica*, *4*(2), 639–650.

[57] **Fox, E.**, **Sudderth, E.**, **Jordan, M. and Willsky, A.** (2011). A STICKY HDP-HMM WITH APPLICATION TO SPEAKER DIARIZATION, *The Annals of Applied Statistics*, 1020–1056.

[58] **Hochreiter, S. and Schmidhuber, J.** (1997). Long Short-term Memory, *Neural computation*, *9*, 1735–80.

[59] **Graves, G.**, **Wayne, G.**, **Reynolds, M.**, **Harley, T.**, **Danihelka, I.**, **Grabska-Barwinska, A.**, **Colmenarejo, S.**, **Grefenstette, E.**, **Ramalho, T.**, **Agapiou, J.**, **Badia, A.**, **Hermann, K.**, **Zwols, Y.**, **Ostrovski, G.**, **Cain, A.**, **King, H.**, **Summerfield, C.**, **Blunsom, P.**, **Kavukcuoglu, K. and Hassabis, D.** (2016). Hybrid computing using a neural network with dynamic external memory, *Nature*, *538*, 471–476.

[60] **Ramsauer, H.**, **Schäfl, B.**, **Lehner, J.**, **Seidl, P.**, **Widrich, M.**, **Adler, T.**, **Gruber, L.**, **Holzleitner, M.**, **Pavlovic´, M.**, **Sandve, G.**, **Greiff, V.**, **Kreil, D.**, **Kopp, M.**, **Klambauer, G.**, **Brandstetter, J. and Hochreiter, S.** (2021). Hopfield Networks is All You Need, *ICLR*.

[61] **Finn, C.**, **Abbeel, P. and Levine, S.** (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, *ICML*.

[62] **Snell, J.**, **Swersky, K. and Zemel, R.** (2017). Prototypical Networks for Few-shot Learning, *NeurIPS*.

[63] **Lai, N.**, **Kan, M.**, **Han, C.**, **Song, X. and Shan, S.** (2021). Learning to Learn Adaptive Classifier–Predictor for Few-Shot Learning, *IEEE Transactions on Neural Networks and Learning Systems*, *32*(8), 3458–3470.

[64] **Loo, N.**, **Swaroop, S. and Turner, R.** (2021). Generalized Variational Continual Learning, *ICLR*.

[65] **Knoblauch, J.**, **Husain, H. and Diethe, T.** (2020). Optimal continual learning has perfect memory and is np-hard, *ICML*.

[66] **Shin, H.**, **Lee, J.**, **Kim, J. and Kim, J.** (2017). Continual Learning with Deep Generative Replay, ***I. Guyon**, **U.V. Luxburg**, **S. Bengio**, **H. Wallach**, **R. Fergus**, **S. Vishwanathan and R. Garnett**, editors, NeurIPS*.

[67] **Lopez-Paz, D. and Ranzato, M.** (2017). Gradient Episodic Memory for Continual Learning, *NeurIPS*.

[68] **Rios, A. and Itti, L.** (2019). Closed-Loop Memory GAN for Continual Learning, *IJCAI*.

[69] **Su, X.**, **Guo, S.**, **Tan, T. and Chen, F.** (2020). Generative Memory for Lifelong Learning, *IEEE Transactions on Neural Networks and Learning Systems*, *31*(6), 1884–1898.

[70] **Guo, Y.**, **Liu, M.**, **Yang, T. and Rosing, T.** (2020). Improved Schemes for Episodic Memory-based Lifelong Learning, *NeurIPS*.

[71] **Chaudhry, A.**, **Ranzato, M.**, **Rohrbach, M. and Elhoseiny, M.** (2019). Efficient Lifelong Learning with A-GEM, *ICLR*.

[72] **Yoon, J.**, **Jeong, W.**, **Lee, G.**, **Yang, E. and Hwang, S.** (2021). Federated Continual Learning with Weighted Inter-client Transfer, *ICML*.

[73] **Mundt, M.**, **Majumder, S.**, **Pliushch, I. and Ramesh, R.** (2019). Unified Probabilistic Deep Continual Learning through Generative Replay and Open Set Recognition, 1905.12019.

[74] **Caccia, M.**, **Rodriguez, P.**, **Ostapenko, O.**, **Normandin, F.**, **Lin, M.**, **Page-Caccia, L.**, **Laradji, I.**, **Rish, I.**, **Lacoste, A.**, **Vázquez, D. and Charlin, L.** (2020). Online Fast Adaptation and Knowledge Accumulation (OSAKA): a New Approach to Continual Learning, *NeurIPS*.

[75] **Mundt, M.**, **Lang, S.**, **Delfosse, Q. and Kersting, Q.** (2022). CLEVA-Compass: A Continual Learning Evaluation Assessment Compass to Promote Research Transparency and Comparability, *ICLR*.

[76] **Haußmann, M.**, **Gerwinn, S.**, **Look, A.**, **Rakitsch, B. and Kandemir, M.** (2021). Learning Partially Known Stochastic Dynamics with Empirical PAC Bayes, *AISTATS*.

[77] **Satorras, V.**, **Akata, Z. and Welling, M.** (2019). Combining Generative and Discriminative Models for Hybrid Inference, *NeurIPS*.

[78] **Dua, D. and Graff, C.** (2017). *UCI Machine Learning Repository*, http://archive.ics.uci.edu/ml.

CURRICULUM VITAE

Name SURNAME: Abdullah AKGÜL

EDUCATION:

- **B.Sc.:** 2020, Istanbul Technical University, Faculty of Computer and Informatics, Department of Computer Engineering
- **M.Sc.:** 2022, Istanbul Technical University, Graduate School, Department of Computer Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2020 Machine Learning Engineer, VakıfBank
- 2021-ongoing Research Assistant at Department of Computer Engineering, Istanbul Technical University

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- Kandemir, M., **Akgül A.**, Haussman M., Unal G. (2022). Evidential Turing Processes, *International Conference on Learning Representations*,
- **Akgül A.**, Unal G., Kandemir, M. (2022). Continual Learning of Multi-modal Dynamics with External Memory, *arXiv preprint.*.