Abstract—Hidden Markov Chains (HMC) and Recurrent Neural Networks (RNN) are two well known tools for predicting time series. Even though these solutions were developed independently in distinct communities, they share some similarities when considered as probabilistic structures. So in this paper we first consider HMC and RNN as generative models, and we embed both structures in a common generative unified model (GUM). We next address a comparative study of the expressivity (or modeling power) of these models, which here refers to the range of the joint probability distribution of an observations sequence, induced by the underlying latent variables. To that end we assume that the models are furthermore linear and Gaussian. The probability distributions produced by these models are characterized by structured covariance series, and as a consequence expressivity reduces to comparing sets of structured covariance series, which enables us to call for stochastic realization theory (SRT). We finally provide conditions under which a given covariance series can be realized by a GUM, an HMC or an RNN.

Index Terms—Hidden Markov Chains, Recurrent Neural Networks, generative models, expressivity, modeling power, stochastic realization theory.

I. INTRODUCTION

Let \( x_{0:t} = (x_0, \cdots, x_t) \) be a sequence of random variables (r.v.). We focus on the general problem of predicting a future observation \( x_{t+1} \) from a realisation of \( x_{0:t} = (x_0, \cdots, x_t) \). This problem has many applications such as speech recognition, finance or geology [1], [2] and can be addressed through Bayesian estimation in two ways. The first way consists in estimating a generative model \( p_\theta(x_{0:t}) \), for all \( t \in \mathbb{N} \), and next computing the posterior distribution \( p_\theta(x_{t+1}|x_{0:t}) \). The second approach aims at building directly a function \( f_\theta \) such that \( f_\theta(x_{0:t}) \) is close to \( x_{t+1} \) is a given sense. The objective of this paper is to propose a comparison between two key tools associated with each approach. The first one is the popular hidden Markov Chain (HMC) model, based on a generative distribution deduced from latent variables, while the second one is the recurrent neural network (RNN), which aims at building a parameterized function. However, we will transform this last architecture into a generative model in order to compare it with the HMC. Our study is not of an experimental nature (see e.g. [3], [4] for such comparisons), but rather aims at quantifying the modeling power of each model. Before further comparing these two models, let us briefly review the rationale of the two approaches by recalling the prediction problem in the static case.

A. Bayesian Problem

Let us consider a sample \( (x, y) \) from a joint probability density function (pdf) \( p(x, y) \). The objective is to predict \( y \) from \( x \), so we look for an estimator \( \hat{y} = f(x) \) such that \( \hat{y} \) is “close” to \( y \). In a Bayesian context, building estimator \( f(.) \) is induced by the choice of a loss function \( L(.,.) \), which depends on the problem at hand, and quantifies the error between the prediction \( f(x) = \hat{y} \) and the true variable \( y \). Building the associate estimator amounts to minimizing the Bayesian risk

\[
R(f) = E[L(f(x), y)],
\]

i.e. build \( f^*(x) = \hat{y} \) in which \( f^* = \arg\min_f R(f) \). One can show that \( f^* \) depends on the posterior density \( p(y|x) = \frac{p(x,y)}{p(x)} \) (also called predictive distribution in the context of prediction). For instance, if the loss is quadratic, the Bayesian estimator is well known to be the conditional expectation, \( f^*(x) = \hat{y} = E[y|x] \) [5], [6, §2.5]. However, \( p(x,y) \) is not known in practice. To cope with this problem one can estimate the Bayesian risk by two different ways: by introducing a parameterized distribution \( p_\theta(x,y) \), or by estimating integral (1) from Monte Carlo samples.

1) Parameterizing the Joint Distribution \( p(x,y) \): The first approach consists in proposing a model of the unknown pdf \( p \). We thus restrict ourselves to a parameterized set of pdfs \( \{p_\theta(.,.)\}_{\theta \in \Theta} \) in which \( \theta \) can be multidimensional. If we have a set of labelled independent samples

\[
\mathcal{E} = \{(x_i, y_i) \sim p(x,y)\}_{1 \leq i \leq n},
\]

IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 71, 2023
4178

1053-587X © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.
the relevance of $p_0$ can be quantified via the likelihood function [7], [8]

$$L(.; \mathcal{E}) : \theta \mapsto \prod_{i=1}^{n} p_0(x_i, y_i),$$

so approximating $p$ amounts to computing a parameter $\theta$ which maximizes the likelihood. Note however that the choice of the parametric family is critical: $p_0$ should model the data at hand, and in the same time function (3) should be computed and optimized efficiently. In general, maximizing the likelihood can only be done approximately. For instance in the case of latent variables models (i.e. the model is defined through the introduction of an unobserved random variable $h$ such that $p_0(x, y) = \int p_0(h, x, y)dh$), maximizing the likelihood requires approximating schemes such as the Expectation Maximization (EM) algorithm [9].

2) Parameterizing the Estimator $f(x)$: The second approach does not make any assumption on $p(x, y)$, but rather estimates (1) from the same dataset (2) (see e.g. [10], [11]). The problem of building an estimator becomes that of minimizing the empirical risk

$$f^*_n = \arg\min_{f} \frac{1}{n} \sum_{i=1}^{n} L(f(x_i), y_i).$$

However, since the dataset is finite, in the absence of further constraints, any function interpolating the points $(x_i, y_i)$ satisfies the optimisation problem (4). In such a case, the model overfits and proves unable to generalize to new observations. This problem is often overcome by choosing a family of functions $(f_\theta)_{\theta \in \Theta}$, and finally (4) turns into the parameter estimation problem:

$$\theta^*_n = \arg\min_{\theta} R_n(f_\theta),$$

$$R_n(f_\theta) = \frac{1}{n} \sum_{i=1}^{n} L(f_\theta(x_i), y_i)$$

which eventually produces the estimator $\hat{y} = f_{\theta^*_n}(x)$ (notation $\theta^*_n$ underlines the fact that the estimator depends on the training set $\mathcal{E}$, which is of dimension $n$).

As above, the choice of the family $(f_\theta)_{\theta \in \Theta}$ should be balanced: a poor set of functions will lead to unrealistic predictions, while a rich set of functions can lead to overfitting. Moreover $(f_\theta)_{\theta \in \Theta}$ should lead to tractable learning, i.e. it should be possible to solve (5) efficiently. Classical solutions include the functions belonging to a reproducing kernel Hilbert space (RKHS) [12], [13] and the functions defined by a neural network (NN) [14], [15]. Optimizing (5) for these families of functions leads to well known algorithms such as (linear or kernel based) least squares [10], Support Vector Machines (SVM) [16], [17], [18], or deep learning algorithms [10], [19] for regression or classification.

3) Discussion: As we have just seen, for minimizing (5) it is not necessary to mimic the distribution $p(x, y)$ by $p_0(x, y)$. In addition, under some assumptions about the family $(f_\theta)_{\theta \in \Theta}$ it is possible to derive concentration inequalities such as

$$\mathbb{P}( |R_n(f_{\theta^*_n}) - R(f_{\theta^*_n}) | > \varepsilon ) \leq \delta_{\varepsilon,n},$$

where $\delta_{\varepsilon,n} \to 0$ when $n \to \infty$ [20]; in other words, such a bound ensures that $f_{\theta^*_n}$ generalizes well and also provides a rate of convergence. However, note that in some contexts, and in particular for times series analysis, we may be interested in predicting $\phi(x)$ from $y$ for a large class of functions $\phi$; once the joint distribution has been estimated, it is possible to comply with such a constraint without running a new estimation algorithm for each function $\phi$. In addition, the knowledge of the posterior distribution enables to quantify (even approximately) the uncertainty of the prediction.

B. Goal of This Paper

The two previous approaches can be adapted to the sequential constraints induced by time series analysis. Let $t$ be the current time parameter. In order to represent the joint distribution of $x_{0:t}$, we need to choose a parametric generative model $p_0(\theta;x_{0:t})$ such that $\theta$ does not depend on $t \in \mathbb{N}$ (otherwise, the model cannot be used with new observations). $p_0$ should model the time series $x_{0:t}$ in a realistic way, and so take into account the dependencies between the observations; in the same time, we should be able to compute an estimator of $\theta$ and to approximate the posterior distribution $p_0(x_{t+1}|x_{0:t})$ for any realization $x_{0:t}$.

A popular model satisfying these requirements is the HMC, particularly developed in the signal processing community, see e.g. [21], [22]. In the same way, it is possible to parameterize a function $f_0(\theta_0;x_{0:t})$ in such way that $\theta$ does not depend on $t$. This is the principle of RNN particularly developed in the machine learning community [23]. Even if such models were basically proposed for point estimation, they can be easily used for building generative models. So from now on, in order to compare the two approaches in a common framework, we will consider that we have at our disposal two generative models $p_0(\theta_0;x_{0:t})$ for all $t$, the HMC and the RNN. Starting from the observation that both models actually rely on a set of latent variables and that they share some common features in the construction of these variables, our objective in this paper is to quantify thoroughly (under some assumptions) how the structural differences of these models impact on their expressivity.

The rest of this paper is organized as follows. In Section II we start by formalizing both models under a common framework. Next in Section III, we see that comparing both models under the linear and Gaussian stationary assumptions reduces to comparing the covariance series of the stochastic process $x_{0:t}$, and consequently the study calls on Stochastic Realization Theory (SRT) (a branch of systems theory). In particular, this theory enables us to extend the preliminary results of [24], obtained for monodimensional states only, to the more general case of states of arbitrary dimension. Section IV provides a brief summary of SRT. Finally the expressivity of HMC and RNN is compared in Section V.

II. Latent Data Models for Time Series Analysis

In this section we introduce our two generative models based on a sequence of latent r.v. $h_{0:t}$, and we next embed them as two particular instances of a more general model.
A. Markovian Models

When dealing with a time series $x_{0:t}$ one major issue consists in modeling the dependency between the observations. For example, a simple Markov Chain (MC)

$$p_\theta(x_{0:t})^{\text{MC}} = p_\theta(x_0) \prod_{t=1}^{t-1} p_\theta(x_{s+1}|x_s)$$

(7)

is often unlikely to represent the distribution of $x_{0:t}$ in a realistic way, since $x_t, t' < t - 1$, becomes independent of $x_t$ when $x_{t-1}$ is observed. One way of enhancing expressivity is to introduce a latent process $h_{0:t}$, one where each $h_s$ can be discrete or continuous. The model is now described by the full joint density $p(h_{0:t}, x_{0:t})$, from which $p(x_{0:t})$ is obtained by marginalizing out the latent variables. This marginalization definitely makes pdf $p(x_{0:t})$ more complex and so increases the modeling power. A constraint is that the introduction of a latent process should preserve some computational properties in order to be used in practice. In this sense, the HMC generalizes (7) by adding a latent process in a rather parsimonious way, since its joint pdf reads:

$$p_\theta(h_{0:t}, x_{0:t})^{\text{HMC}} = p_\theta(h_0) \prod_{s=1}^{t} p_\theta(h_s|h_{s-1}) \prod_{s=0}^{t} p_\theta(x_s|h_s).$$

(8)

So an HMC benefits of three (conditional) independence properties: the latent process is an MC; given the latent variables $h_{0:t}$, observations $x_{0:t}$ are independent; and given all latent variables $h_{0:t}$, an observation only depend on the latent variable at the same time, $p_\theta(x_s|h_s) = p_\theta(x_s|h_s)$ for all $s$, $0 \leq s \leq t$. The hidden process $h_{0:t}$ can have a physical meaning (in which case estimating $h_{0:t}$ from $x_{0:t}$ is relevant). If not, the role of the latent variables $h_{0:t}$ is just to make the observed process $x_{0:t}$ more complex, and the HMC can be seen as a generative model. Associated inference algorithms for approximating the Maximum Likelihood estimator and posterior distributions have been extensively studied for these models [2], [25] and are recalled in Appendix A.

B. RNN Architectures

RNN are an adaptation of neural architectures to time series. So let us start by briefly recalling the rationale of neural networks.

Neural network architectures are versatile classes of functions [19], which have found many applications for classification or prediction, including language [26] or image [27] processing. A neural network (NN) is a succession of parameterized functions called neurons. A neuron typically computes $x \mapsto \sigma(we + b)$, where $we$ is the dot product of $w$ (a vector of weights) and $x$ (a vector of variables), $b$ is the bias, and $\sigma(.)$ is a so-called (nonlinear) activation function, such as the sigmoid, hyperbolic tangent or ReLu functions. Neurons can be gathered into layers which themselves can be cascaded, yielding increasingly complex functions. Some universal approximation theorems have been proposed [28], [29], [30], [31]; for instance, given any (possibly multi-dimensional) continuous function $f$, there exists a single-layer NN $f_\theta$ arbitrarily close to $f$, provided the activation function is not polynomial [30]. Similar results have been proposed for multiple layers NNs. So any Lebesgue-integrable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ can be approximated by an NN with ReLu activation function and layers made of at least $n + 4$ neurons, provided the net is deep enough [31]. The number of layers and of neurons per layer, as well as the activation functions, are hyperparameters which characterize the NN architecture, and the weights and biases are the model parameters learnt from a training set. However, the input of an NN as described above is of fixed size, which is not well suited to the modeling of time series in which observations accumulate - unless we use a sliding window, but in that case the prediction would not be based on the full set of observations.

In order to introduce dependencies between all the observations, RNN introduce a latent variable $h_t$ which is a function of all observations $x_{0:t}$ and serves as a memory of the past. After receiving the new information $x_t$, the new state is computed as

$$h_t = f_\theta(h_{t-1}, x_t), \quad h_{-1} = 0$$

(9)

where $\theta$ is an NN layer. In other words, $h_t$ is a summary of all the past observations until time $t$. Finally a prediction of $x_{t+1}$ is computed as $\hat{x}_{t+1} = g_\theta(h_t)$ where $g_\theta$ is an NN architecture.

As we claimed before, RNN can be transformed into generative models by replacing function $g_\theta$ by a parametric distribution $p_\theta(x_{t+1}|h_t)$. In this case, we obtain a family of models defined by

$$p_\theta(x_{0:t})^{\text{RNN}} = p_\theta(x_0) \prod_{s=0}^{t-1} p_\theta(x_{s+1}|h_s),$$

(10)

where $h_s$ depends on $\theta$ from (9).

By construction, the posterior distribution $p_\theta(x_{t+1}|x_{0:t})$ coincides with $p_\theta(x_{t+1}|h_t)$ and is directly available. The estimation of $\theta$ can be computed by a gradient backpropagation algorithm [32], [33], [34], [35] which aims at maximizing $\log(p_\theta(x_{0:t}))$ for a given observation. Due to the time component (and the parameter sharing) there can be as many computed gradients as observations for a given parameter, see e.g. [34], [36], [37], [38]. However, in practice, the gradients computed for a given parameter geometrically tend to infinity or to zero when we get back into the past. These phenomena are called exploding gradient and vanishing gradient. The exploding gradient phenomenon is often due to the repeated multiplication of high weights, a situation where learning the RNN becomes particularly unstable. An efficient way to limit this behavior is to bound the values taken by the gradient [19], [39]. One can also include a regularization term to the cost function in order to penalize weights that are too large [40]. By contrast, the vanishing gradient phenomenon results from the repeated multiplication or small size weights, as well as the iterated use of activation functions which have derivatives bounded by 1 in magnitude (e.g. the sigmoid). In that case, the oldest observations are not taken into account in the learning phase, so it is difficult to learn long term dependencies. In order to mitigate the vanishing gradient phenomenon, more sophisticated architectures have been proposed, such as the Long Short Term Memories (LSTM).
and the Gated Recurrent Units (GRU) [41] (the only difference is that the corresponding parameterization of \( f_\theta \) becomes more complex).

C. Generative Unified Model

As we have just seen, HMC and RNN models result from a different paradigm but both aim at proposing a parameterized distribution \( p_\theta(x_{0:t}) \) via the introduction of latent variables \( h_{0:t} \).

In the RNN model, \( h_{0:t} \) is deterministic given the observations \( x_{0:t} \), and \( h_s \) summarizes all the observations up to time \( s \) into a unique variable; in the HMC model, \( h_{0:t} \) is stochastic given the observations, and indeed the Bayesian estimation of \( h_{0:t} \) is of interest in cases where \( h_{0:t} \) is a physical process of interest.

When we put aside the computational aspects, the natural question that arises is to compare the set of distributions \( p_\theta(x_{0:t}) \) induced by each model. Actually, both representations can be reconciled as particular instances of the following Generative Unified Model (GUM),

\[
p_\theta(h_{0:t}, x_{0:t}) = p_\theta(h_0) \prod_{s=1}^{t} p_\theta(h_s | h_{s-1}, x_{s-1}) \prod_{s=0}^{t} p_\theta(x_s | h_s).
\]

Indeed, the HMC model (8) is a GUM where

\[
p_\theta(h_t | h_{t-1}, x_{t-1})_{\text{HMC}} = p_\theta(h_t | h_{t-1}),
\]

while the RNN (9)-(10) satisfies (up to the transformation \( h_t \leftarrow h_{t-1} \))

\[
p_\theta(h_t | h_{t-1}, x_{t-1})_{\text{RNN}} = \delta_{f_t(h_{t-1}, x_{t-1})}(h_t),
\]

\[
p_\theta(x_0 | h_0) = p_\theta(x_0),
\]

\[
h_0 \text{ RNN} \equiv 0,
\]

where \( \delta \) denotes the Dirac mass.

In the rest of this paper we will also consider deterministic GUM (D-GUM), which are defined by the first equation of (13) only (the interest of D-GUM over RNN will be clear in Section V-B).

Let us now discuss the three models, beginning with their similarities. First, in all three models the pair \((h_s, x_s)\) is an MC:

\[
p(h_s, x_s | h_{0:s-1}, x_{0:s-1}) = p(h_s | x_s | h_{0:s-1}, x_{0:s-1})
= p(h_s | h_{0:s-1}, x_{s-1}) p(x_s | h_s),
\]

which induces \( p(x_s | h_{0:s-1}) = p(x_s | h_s) \). In addition, in all three models the marginal process \( h_{0:t} \) is also an MC, since

\[
p(h_s | h_{0:s-1}) = \int p(h_s | h_{s-1}, x_{s-1}) p(x_{s-1} | h_{s-1}) dx_{s-1}
= p(h_s | h_{s-1}).
\]

As a result, the GUM, HMC and RNN models only differ via the distribution \( p_\theta(x_{0:t} | h_{0:t}) \). In an HMC, \( h_s \) only depends on \( h_{s-1} \) given the past \((h_{0:s-1}, x_{0:s-1})\), so \( p_\theta(x_{0:t} | h_{0:t}) = \prod_{s=0}^{t} p(x_s | h_s) \); on the other hand, \( h_t \) is stochastic so \( p_\theta(x_{0:t}) \) is not available in closed form. By contrast, in a D-GUM or an RNN, \( h_s \) is deterministic given the past \((h_{0:s-1}, x_{0:s-1})\), so \( p_\theta(x_{0:t}) \) is available in closed form; but \( h_s \) also depends on \( h_{s-1} \) so \( p_\theta(x_{0:t} | h_{0:t}) \) is difficult to interpret. The graphical representation of the three models is displayed in Fig. 1.

Now that we have cast the HMC and the RNN in a common framework, we can address the comparison of their expressivity from the GUM perspective. Actually, this perspective is very general, and recent Stochastic RNN architectures used in machine learning can also be cast into that GUM framework [42]. It is why understanding such a representation is of interest. More precisely, our objective is to set some assumptions which enable us to discuss on the distribution of the observation \( p_\theta(x_{0:t}) \) associated to the GUM, and next to discuss on the restrictions on this distribution induced by (12) and (13).

III. STRUCTURE OF THE MAPPING \( \theta \rightarrow p_\theta(x_{0:t}) \):

THE LINEAR AND GAUSSIAN CASE

A. Linear and Gaussian GUM

We now address the expressivity of the HMC and RNN models. In order to compare the observations pdf \( p(x_{0:t}) \) induced by the HMC and RNN models, we set ourselves in the general framework of GUM, in which \( p(x_{0:t}) \) is a marginal of (11). Of course, \( p(x_{0:t}) = \int_{h_{0:t}} p(h_{0:t}, x_{0:t}) dh_{0:t} \) cannot, in most cases, be computed in closed form. In order to be able to provide a compared analysis of the expressivity of those models (and thus to understand, when we reduce to the particular cases of HMC and of RNN, the role of stochastic vs. deterministic transitions: see (12) and (13)), we consider the simplified linear and Gaussian framework, i.e. a GUM model in which the elementary factors in (11) read

\[
p(h_0) = \mathcal{N}(h_0; 0, \eta),
\]

\[
p(h_t | h_{t-1}, x_{t-1}) = \mathcal{N}(h_t; ah_{t-1} + cx_{t-1}; \alpha),
\]

\[
p(x_t | h_t) = \mathcal{N}(x_t; bh_t; \beta),
\]

Fig. 1. Conditional dependencies in HMC, RNN, and GUM. The dashed (resp. solid) lines stand for deterministic (resp. probabilistic) dependencies, i.e., one r.v. is a function of the other(s), or one r.v. is given by a conditional pdf, given the other(s).
in which $h_t$ is an $n$-dimensional vector and $x_t$ is a scalar; so $a$, $b$ and $c$ are respectively $n \times n$, $1 \times n$, $n \times 1$, $\eta$ and $\alpha$ are $n \times n$ covariance matrices, and $\beta \geq 0$. $\theta = (a, b, c, \alpha, \beta, \eta)$ is the parameter of the model.

It is easy to see that in model (11) (15)-(17), the joint pdf $p(x_{0:t})$ is a zero-mean multivariate Gaussian density, which is fully characterized by its covariance matrix. Let $h_t$ be the covariance matrix of $h_t$ (we will later see how $h_t$ depends on $\eta_t$ and on time $t$). Using the linear structure of the model (16)-(17) and the tower property of conditional expectations, we have $\text{Var}(x_t) = \beta + bT \eta T$, and, for all $t \in \mathbb{N}$, $\eta_t$ is a wide sense stationary process. Let us finally remark that in which $h_t$ is an $n$-dimensional vector and $x_t$ is a scalar; so $a$, $b$ and $c$ are respectively $n \times n$, $1 \times n$, $n \times 1$, $\eta$ and $\alpha$ are $n \times n$ covariance matrices, and $\beta \geq 0$. $\theta = (a, b, c, \alpha, \beta, \eta)$ is the parameter of the model.

Due to $\eta_t$ and factor $N_t$, $\text{Var}(x_t)$ and $\text{Cov}(x_t, x_{t+k})$ are at least asymptotically stationary.

### B. Stationarity

First, it is easy to see that the matrix series $(\eta_t)_{t \in \mathbb{N}}$ is defined by

$$\eta_{t+1} = (\alpha + c\beta c^T) + (a + cb)\eta_t(a + cb)^T.$$  \hspace{1cm} (19)

Using (19) recursively we get $\eta_{t+1} - \eta_t = (a + cb)^T[\eta_t - \eta_0](a + cb)^T$ for all $t, t \in \mathbb{N}$. The series $(\eta_t)$ is thus constant if

$$\eta_0 = \eta_1 = \eta.$$  \hspace{1cm} (20)

Assumption (20) implies in turn that $\text{Var}(x_t)$ and $\text{Cov}(x_t, x_{t+k})$ no longer depend on time, so that $(x_t)_{t \in \mathbb{N}}$ is a wide sense stationary process. Let us finally remark that under assumption (20), equation (19) becomes:

$$\eta = (\alpha + c\beta c^T) + (a + cb)^T \eta(a + cb)^T.$$  \hspace{1cm} (21)

This equation in variable $\eta$ is meaningful (recall that $\eta$ is a covariance matrix) only if it admits a semi-definite positive $(\geq 0)$ solution, which implies [43], [44] that

$$(a + cb)$$ has all its eigenvalues in $\{z \in \mathbb{C}; |z| < 1\}$.  \hspace{1cm} (22)

We will now assume that (22) and (20) hold, which implies that $x_t$ is stationary. Let us note that this stationarity assumption is reasonable, because under assumption (22), the series $(\eta_t)$ converges when $t$ tends to infinity. So $(h_t)_{t \in \mathbb{N}}$ as well as $(x_t)_{t \in \mathbb{N}}$ are at least asymptotically stationary.

**Remark 1:** In the HMC case, we have the additional constraint $c = 0$ since $h_t$ does not depend on $x_{t-1}$ given $(h_{t-1}, x_{t-1})$. In the case of D-GUM, $\alpha = 0$; if we also consider the full RNN case with its particular initial conditions (see (13)), the constraint $\text{Var}(h_1) = \eta = e\text{Var}(x_0)c^T = c(\beta + b\eta b^T)c^T$ has also to be satisfied.

### C. The Mapping $\theta \rightarrow p_\theta(x_{0:t})$

Let $r_k = \text{Cov}(x_t, x_{t+k})$. We observe that this covariance series has a very specific structure:

$$r_0 = \beta + b\eta b^T;$$

for all $k \in \mathbb{N}^*$,  \hspace{1cm} (23)

$$r_k = b\eta b^T.$$  \hspace{1cm} (24)

Since this covariance series $(r_k)_{k \in \mathbb{N}}$ characterizes the distribution of $p_\theta(x_{0:t})$ for all $t$, we now consider function

$$\phi : (a, b, c, \alpha, \beta, \eta) \rightarrow \phi(\theta) = (r_k)_{k \in \mathbb{N}}.$$  \hspace{1cm} (25)

in which $(r_k)_{k \in \mathbb{N}}$ is given by (23) (24). Since a study of the direct range of $\phi$ under the HMC or RNN constraints seems a difficult task, we rather consider the inverse mapping.

Let us first observe that the factored structure of the covariance series (i.e., there exists $(H, F, N)$ s.t. $r_k = H F^{k-1} N$ for all $k \geq 1$) is remarkable, and is directly related to system theory. More precisely, the output of any linear time invariant (LTI) state space system has a stationary factorized covariance series, and conversely, any such series can be realized by an LTI system. This second point is the topic of Stochastic realization theory (SRT), which indeed is of interest here since we shall look for parameters $\theta = (a, b, c, \alpha, \beta, \eta)$ s.t. $\phi(\theta) = (r_k)_{k \in \mathbb{N}}$ for a given $(r_k)_{k \in \mathbb{N}}$. Before we proceed to the analysis we thus briefly provide a brief reminder of SRT (the reader familiar with SRT can skip Section IV and directly jump to Section V).

### IV. A SHORT REVIEW OF SRT

Let us briefly review some points from SRT [43], [45], [46], [47], [48] which we will need in Section V. SRT is a part of systems theory, which deals with modeling, controlling and estimating dynamic systems (see e.g. [49], [50], [51]). Before we proceed (see Section IV-B) we need to recall some algebraic facts from deterministic realization theory (DR).

#### A. DRT

Let us consider a linear discrete time system with state $h_t$:

$$\begin{aligned}
h_{t+1} &= F h_t + N u_t, \\
x_t &= H h_t,
\end{aligned}$$  \hspace{1cm} (26)

where $F$, $N$, $H$ are $n \times n$ (resp. $n \times 1$, $1 \times n$) matrices (we only deal here with the case where observation $x_t$ and input $u_t$ are one-dimensional). The mapping between input $u_t$ and output $x_t$ is given by the convolution equation $x_t = \sum_{k=1}^{+\infty} H_k u_{t-k}$, where the lags $H_k$ of the impulse response (the so-called Markov parameters of the system) satisfy

$$H_k = H F^{k-1} N$$  \hspace{1cm} (27)

for all $k \geq 1$. Equivalently, the strictly causal transfer function $H(z) = \sum_{k=1}^{+\infty} H_k z^{-k}$ can be written as $H(z) = H(z I - F)^{-1} N$.  

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
The DR problem consists in building three matrices $H, F, N$, with $F_{n \times N}$ of minimal dimension, from the impulse response of the system, i.e., move from the infinite representation $(H_k)_{k \in \mathbb{N}^+}$ to the finite representation $(H, F, N)$, with $F$ of minimal dimension. The key tool for this problem is the infinite Hankel matrix
\[
\mathcal{H}_\infty = \begin{bmatrix}
H_1 & H_2 & H_3 & \ldots \\
H_2 & H_3 & & \\
& H_3 & & \\
& & & \\
& & & \\
\end{bmatrix}.
\] (28)

From (27), $\mathcal{H}_\infty$ factorizes as
\[
\mathcal{H}_\infty = \begin{bmatrix}
H & H F & H F^2 & \ldots \\
H F & H F^2 & & \\
& H F^2 & & \\
& & & \\
& & & \\
\end{bmatrix} [N, FN, F^2N, \ldots],
\] (29)

and so has finite rank, which moreover is equal to $n$ (the dimension of $F$) if and only if (iff.) each factor is itself full rank $n$. Conversely, if $\mathcal{H}_\infty$ has finite rank $n$, then it can be factorized as a product of two factors of dimensions $(\infty \times n)$ and $(n \times \infty)$, both of them being of full rank $n$, and due to the Hankel structure, there exists $F_{n \times N}, N_{n \times 1}, H_{1 \times n}$ so that (29) (and thus (27)) is satisfied. Moreover, from the proposition below, all minimal realizations of $H(z)$ are isomorphic:

**Proposition 1:** ([52], proposition 3) $(H_1, F_1, N_1)$ and $(H_2, F_2, N_2)$ are two minimal realizations of $H(z)$ if and only if there exists $T$ invertible such that $F_2 = TF_1T^{-1}$, $N_2 = TN_1$ and $H_2 = H_1T^{-1}$.

Finally numerically efficient DR algorithms have been proposed in [53], [54].

**B. SRT**

Let us now consider the state space system
\[
\begin{cases}
\dot{h}_{t+1} = Fh_t + u_t \\
x_t = Hh_t + v_t,
\end{cases}
\] (30)

where $h_0$ is zero-mean and uncorrelated with $(u_t, v_t)$, and where $(u_t, v_t)$ is a zero-mean, uncorrelated, stationary random process with
\[
\mathbb{E} \begin{bmatrix}
u_t^T \\
v_t^T
\end{bmatrix} = \begin{bmatrix} Q & S \\
S^T & R \end{bmatrix}_0
\] (31)

and $\delta_{t,t'} = 1$ iff. $t = t'$. Let us assume that $\{x_t\}_{t \geq 0}$ is (wide sense) stationary and purely non-deterministic. This together with an observability condition on $(F, H)$ implies that $\{h_t\}_{t \geq 0}$ is (wide-sense) stationary and purely non-deterministic as well. Let $P = \mathbb{E}[h_th_t^T]$; $P$ satisfies
\[
P = FP F^T + Q,
\] (32)

which in turn implies that $F$ has all its eigenvalues in the open unit disc. Finally the covariance function of $x_t$ is given by
\[
r_0 = \mathbb{E}[x_t^2] = R + HPHT;
\] (33)

for all $k \in \mathbb{N}^*$, $r_k = \mathbb{E}[x_{t+k}] = HF^{k-1}(FPHT^T + S)$. (34)

Starting from a covariance $(r_k)_{k \in \mathbb{N}}$, the SR problem consists in building a minimal “Markovian representation” of $(x_t)_{t \in \mathbb{N}}$, i.e., a state-space system (30)-(31), with $F$ of minimal dimension.

**Step 1:** Thanks to the structure of function $(r_k)_{k \in \mathbb{N}^+}$, we can as in Section IV-A build a Hankel matrix
\[
\mathcal{R}_\infty = \begin{bmatrix}
r_1 & r_2 & r_3 & \ldots \\
r_2 & r_3 & & \\
r_3 & & & \\
& & & \\
& & & \\
\end{bmatrix} [H, HF, HF^2, N, FN, F^2N, \ldots]
\] (35)

which should be compared to factorization (29). The first (and, in fact, “deterministic”) step of an SR algorithm consists in building a minimal realization $(H, F, N)$ of $(r_k)_{k \in \mathbb{N}^+}$ (unique up to an invertible matrix):

**Step 2:** At this point, we dispose of $(H, F, N)$ but $N$ remains a function of $P$ and $S$ (see (34)), and it remains to identify $Q$ and $R$. This second step is more delicate for the problem must be solved under positivity constraints: $P$ and $Q\begin{bmatrix} S & S^T \\ S^T & R \end{bmatrix}$ are covariance matrices and so must be semi-definite positive ($\geq 0$). If these constraints were not satisfied, the solution would be meaningless. Finally, the problem is as follows: knowing $(H, F, N, r_0)$, we look for $(P, Q, R, S)$ such that
\[
\begin{bmatrix}
P \\
N^T \\
r_0
\end{bmatrix} = \begin{bmatrix} F & H \\
F^T & HT \\
H^T & \end{bmatrix} \begin{bmatrix} Q & S \\
S^T & R \end{bmatrix},
\] (36)

$P > 0$, (37)

\[
\begin{bmatrix} Q \\
S^T \\
S \\
R
\end{bmatrix} \geq 0,
\] (38)

in which $> 0$ stands for definite positive (the constraint on $P$ should be, a priori, that $P$ is semi-definite positive, but indeed it happens that any solution $P$ must be definite positive [46] (see theorem 1 below), whence (37)). Let us notice that equation (36) gives the covariance of $(h_{t+1}, x_t)$. Since $(u_t, v_t)$ is a white noise, this covariance satisfies a (Ricatti) equation of the same kind as that satisfied by $P$ (equation (32), which in fact is a submatrix of (36)).

System (36) can be seen as a system with three equations and four unknowns $(P, Q, R$ and $S$), or rather as a system with three equations and three unknowns $(Q, R$ and $S$), parameterized by $P$. Finally, $P$ parameterizes solutions of the constrained system (36)-(38). Let $\mathcal{P}$ be the set of parameters
\[
\mathcal{P} = \{P \ s.t. \ (36) \ - (38) \ are \ satisfied\}.
\] (39)

Positive Real Lemma, Positivity of $(r_k)_{k \in \mathbb{N}}$, Structure of $\mathcal{P}$

A result known as the positive real lemma (initially proved in the spectral domain) connects the positivity of the series $(r_k)_{k \in \mathbb{N}}$ (in other words, whether $(r_k)_{k \in \mathbb{N}}$ is a covariance series) to the existence of at least one solution to the constrained system (36)-(38). Let us recall that the infinite series $(r_k)_{k \in \mathbb{N}}$ is
a covariance series iff. the Toeplitz form $\sum_{i,j=0}^{m} u_i u_j r_{j-i}$ is positive or null for all $m$, i.e. iff. the associated Toeplitz matrix is semi-definite positive for all $m$.

**Lemma 1:** (Positive real lemma [46]) The series $(r_k)_{k \in \mathbb{N}}$ is a covariance series iff. $\mathcal{P}$ is non void.

We now consider the structure of $\mathcal{P}$.

**Theorem 1:** (46) The set $\mathcal{P}$ is closed, convex and bounded; it admits (for the usual order relation between symmetric matrices) a maximum $P^*$ and a minimum $P_*$.

Let us finally notice that there exist efficient algorithms for building elements of $\mathcal{P}$ (see [46], [48]).

V. EXPRESSIVITY OF GUM, HMC, AND RNN

We are now ready to come back to mapping (25) (23) (24), and first need to study the range of $\phi$.

A. Algebraic Properties Induced by the Factorizability and Positivity Constraints

The range of $\phi$ is strictly included into $\mathbb{R}^{\mathbb{N}}$, since $(r_k)_{k \in \mathbb{N}}$ in (25) is indeed a factorized and covariance series. As we now see, it is possible to characterize this range, via algebraic tests which determine whether a given real series satisfies these two constraints.

1) Factorizability: First, factorization (24) implies that the doubly infinite Hankel matrix built on $(r_k)_{k \in \mathbb{N}}$ factorizes as (35). So the rank of $\mathcal{R}_\infty$ is finite and lower than or equal to $n$ (the dimension of $F$), and is equal to $n$ if and only if each factor has itself full rank $n$. In this case, $(H,F,N)$ is a so-called minimal (deterministic) realization of $(r_k)_{k \in \mathbb{N}}$ (see Section IV for more details). One can show (see [52], proposition 3 or Section IV) that all minimal realizations are isomorphic: $(H_1,F_1,N_1)$ and $(H_2,F_2,N_2)$ are two minimal realizations of $(r_k)_{k \in \mathbb{N}}$ if and only if there exists $T_{1,2}$ invertible such that

$$ (H_2,F_2,N_2) = (H_1T_{12}^{-1},T_{12}F_1T_{12}^{-1},T_{12}N_1). \quad (40) $$

2) Positivity: Apart from being factorizable, any sequence $(r_k)_{k \in \mathbb{N}} \equiv (r_k)_{k \in \mathbb{N}}$ is also a covariance series, which can be characterized either by the constraint that for all $k \in \mathbb{N}$, the Toeplitz matrix with first row $[r_0, \ldots, r_k]$: is positive semi definite or, equivalently, that $C(z)^{def} = r_0 + 2 \sum_{k=1}^{\infty} r_k z^k$ is a Carathéodory function, i.e. has positive real part in the open unit disk $\{z \in \mathbb{C}; |z| < 1\}$ (Carathéodory-Toeplitz theorem, see e.g. [55]).

In the context of this paper, it is however more interesting to recall the positive real lemma, which relies on the factorizability constraint we just evoked. So assume that rank($\mathcal{R}_\infty$) is finite, which enables to build a minimal set $(F,H,N)$ satisfying $r_k = HF^{-1} N$ for all $k \geq 1$. As we recalled in Section IV, positivity of the series $(r_k)_{k \in \mathbb{N}}$ is related to whether there exists at least one matrix $P > 0$ satisfying

$$ \begin{bmatrix} P & N^T \\ N & r_0 \end{bmatrix} - \begin{bmatrix} F & F^T \\ H & H^T \end{bmatrix} P \begin{bmatrix} F^T \\ H^T \end{bmatrix} \succeq 0, \quad (41) $$

three unknowns $(Q,R$ and $S)$ parameterized by $P$, i.e. whether the set $\mathcal{P}$ defined in (39) is non void.

B. Compared Expressivity of the Three Models

Let us summarize Section V-A.

- Starting from any real valued series $(r_k)_{k \in \mathbb{N}}$, this series is factorizable (i.e., there exists a triplet $(H,F,N)$ such that $r_k = HF^{-1} N$ for all $k \in \mathbb{N}$) iff. the Hankel matrix $\mathcal{R}_\infty$ is finite rank; the rank $n$ of $\mathcal{R}_\infty$ is also the minimal dimension of any realization of $(r_k)_{k \in \mathbb{N}}$.
- Starting from a factorizable series $(r_k)_{k \in \mathbb{N}}$, this series is a covariance series if and only if there exists at least one matrix $P > 0$ satisfying (41).

This discussion is summarized in Fig. 2 below, the South-West part of which is the range of function $\phi$ in (25).

We now study any point of the South-West corner (i.e., any factorizable covariance function) can be realized by a GUM, an HMC and/or an RNN.

1) Expressivity of GUM: This question does not raise any particular difficulty. Since $(r_k)_{k \in \mathbb{N}}$ is a factorized covariance function, it can be realized (see Section IV) by the state space system (30)-(31) for some $(F,H,Q,R,S)$. System (30) can be rewritten (if $R \neq 0$) as

$$ \begin{cases} h_{t+1} = ah_t + cx_t + u'_t \\ x_t = bh_t + v' - t \end{cases}, \quad (42) $$

$$ \mathbb{E}\left[ \begin{bmatrix} u'_t \\ v'_t \end{bmatrix} \cdot \begin{bmatrix} u'_t^T \\ v'_t^T \end{bmatrix} \right] = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix}, \quad (43) $$

in which

$$ a = F - SR^{-1} H, b = H, c = SR^{-1}, \quad \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} = \begin{bmatrix} I & -SR^{-1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} Q & S \\ T^* R \end{bmatrix} \begin{bmatrix} I & 0 \\ -R^{-1} S^T & 1 \end{bmatrix}; \quad (45) \quad \geq 0 $$

Eq. (45) ensures that $\begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix} \geq 0$ (and thus $\alpha \geq 0$). Equations (42)-(43) are a state space representation of (11) (15)-(17). In other words, any point of the South-West corner can be realized by some linear and Gaussian GUM model.
2) Expressivity of HMC: We know that any factorizable covariance function \( (r_k)_{k \in \mathbb{N}} \) such that \( \dim(R_\infty) = n \) can be realized by a GUM of dimension \( n \). Starting from such a series, under which conditions does there exist an HMC of the same degree \( n \) which produces that same covariance series? We have the following result (see Appendix B for a proof).

Proposition 2: Let \((r_k)_{k \in \mathbb{N}}\) a factorizable covariance function and let \((H, F, N)\) a triplet (with \( F \) of minimal dimension \( n \)) produced by DR. The series \((r_k)_{k \in \mathbb{N}}\) can be realized by an HMC of dimension \( n \) if and only if there exists \( P \) (and thus \( \tilde{Q}(P) \) et \( \tilde{R}(P) \)) such that

\[
\begin{bmatrix}
\tilde{P} \\
N^T \\
r_0
\end{bmatrix} = \begin{bmatrix}
F \\
H
\end{bmatrix} \tilde{P} \begin{bmatrix}
F^T \\
H^T
\end{bmatrix} = \begin{bmatrix}
\tilde{Q} \\
0 \\
0 \\
\tilde{R}
\end{bmatrix},
\]

(46)

\[
\tilde{P} > 0,
\]

(47)

\[
\begin{bmatrix}
\tilde{Q} \\
0 \\
0 \\
\tilde{R}
\end{bmatrix} \geq 0.
\]

(48)

Remark 2: Finally, let \( \tilde{P} \) the set of solutions \( \tilde{P} \) of the constrained problem (46)-(48). One can note that \( \tilde{P} \) is a convex subset of \( P \). On the other hand, as compared to \( P \), (46) yields the supplementary constraint (74). This equation can be satisfied only if \( N \in \text{Span}(F) \). Moreover, if \( F \) is invertible, (74) also implies

\[
H^TF^{-1}N > 0.
\]

(49)

So if (74) and/or (49) is not satisfied, then the series \((r_k)_{k \in \mathbb{N}}\) cannot be realized by an HMC of dimension \( n \).

3) Expressivity of D-GUM and RNN: Similarly as the HMC, the study can be done from any triplet \((H, F, N)\) provided by the DR step. We have to take into account the D-GUM constraint \( \alpha = 0 \) (or \( Q - SR^{-1}ST = 0 \), see (44)-(45)) and the RNN constraint \( \eta = e\text{Var}(x_0) \) (or \( P = SR^{-1}RST \)).

Proposition 3: Let \((r_k)_{k \in \mathbb{N}}\) a factorizable covariance function and let \((H, F, N)\) a triplet (with \( F \) of minimal dimension \( n \)) produced by DR. Let us note \( P \) the set of solutions of system (36)-(38). Then \((r_k)_{k \in \mathbb{N}}\) can be realized by a D-GUM if and only if there exists \( P \in P \) such that

\[
\tilde{P} = \begin{bmatrix}
F \\
H
\end{bmatrix} \tilde{P} \begin{bmatrix}
F^T \\
H^T
\end{bmatrix} - (N - FPH^T)(r_0 - H\tilde{P}H^T)^{-1}(N - FPH^T)^T = 0.
\]

(50)

If in addition \( \tilde{P} \) satisfies

\[
\tilde{P} = r_0(r_0 - H\tilde{P}H^T)^{-2}(N - F\tilde{P}H^T)(N - F\tilde{P}H^T)^T,
\]

(51)

the covariance series can be produced by a traditional RNN initialized to \( h_0 = 0 \), with a linear activation function.

Remark 3: Note that by construction \( \tilde{P} \) in (51) is a rank 1 \( n \times n \) semi-definite positive matrix, and is positive definite only if \( n = 1 \). In other words, a factorizable covariance series can be realized by an RNN if the latent vector is monodimensional and (51) holds (as we will check in the worked example below), but can never be realized by an RNN if \( n > 1 \).

C. A Worked Example (Unidimensional Case)

We now illustrate the preceding section in the scalar case. We first look for conditions on the triplet \((H, F, N)\) such that a factorizable series \((r_k)_{k \in \mathbb{N}}\) is a covariance function. We next give conditions on this triplet to determine if the covariance series can be produced by one of the generative models of this paper.

1) SR Step: Let \((r_k)_{k \in \mathbb{N}}\) be a covariance series, factorizable as \( HF^{k-1}N \) for all \( k \in \mathbb{N} \) with \( H, F \) and \( N \) scalar. We assume that such a triplet \((H, F, N)\) has been produced by DR of \((r_k)_{k \in \mathbb{N}}\), and we search for the scalar parameters \( P, Q, R, S \) satisfying (36)-(38). Each of these equations becomes respectively

\[
\begin{align*}
Q & = P(1 - F^2) \\
R & = r_0 - PH^2 \\
S & = N - HFP \\
P & \geq 0,
\end{align*}
\]

(52)

(53)

\[
\begin{align*}
QR - S^2 & \geq 0 \\
Q & \geq 0 \\
R & \geq 0.
\end{align*}
\]

(54)

In particular, (54) corresponds to the semi-definite positive constraint (38) when \( Q, R \) and \( S \) are scalar.

Let us build the set \( P \) of positive numbers \( P \) which satisfy this system. The second inequality of (54) is satisfied when \( F^2 \leq 1 \); by using (52), the first inequality of (54) reads

\[
\Xi_{HF^2}(P) \overset{\text{def}}{=} -H^2P^2 + |r_0(1 - F^2) + 2HN|P - N^2 \geq 0.
\]

(55)

Since polynomial \( \Xi_{HF^2} \) is concave, one can show easily that (55) admits a solution provided

\[
\frac{r_0(F - 1)}{2} \leq HN \leq \frac{r_0(F + 1)}{2},
\]

(56)

and that \( P \) is included in \([P_{HF^1}, P_{HF^2}] \) with

\[
P_{HF^1} = \frac{(2HF + r_0(1 - F^2)) + (-1)^i\sqrt{\delta}}{2H^2},
\]

(57)

\[
\delta = (1 - F^2)(r_0(1 + F) - 2HN)(r_0(1 - F) + 2HN).
\]

(58)

Moreover, constraints (53) and \( R \geq 0 \) in (54) imply that \( P \) belongs to \([0, \frac{1}{H^2}] \); but it can be checked that \([P_{HF^2}, P_{HF^2}^2] \subseteq [0, \frac{1}{H^2}] \) so constraints \( P \geq 0 \) and \( R \geq 0 \) do not yield further interval restrictions. Finally, factorizable series \((r_k)_{k \in \mathbb{N}}\) is a covariance function if \( F^2 \leq 1 \) and \( HN \) satisfies (56); \( P \) then coincides with

\[
P = [P_{HF^1}, P_{HF^2}^2]
\]

(59)

and is non void. It can also be produced by a GUM whose parameters are deduced from (44)-(45).

Remark 4: Finally one can show easily that \( P_{HF^2,1} > 0 \) if \( N > 0 \) (the case \( P_{HF^2,1} = 0 \) is possible only if \( \Xi_{HF^2}(0) = -N^2 \geq 0 \), and so \( N = 0 \), which corresponds to the degenerate case of a series \((r_k)_{k \in \mathbb{N}}\) which is null everywhere except at \( k = 0 \) where it is equal to \( r_0 \), so \( P \) is a definite positive set, which is in concordance with Faurre’s theory [46, theorem 7].
Fig. 3. Expressivity of RNN, HMC, D-GUM and GUM with regards to parameters $F$ and $HN$ in the scalar case. The parallelogram (blue and cyan) coincides with the factorizable covariance series $r_k = F^{k-1}HN$. Such series can be produced by a GUM or a D-GUM. The blue (resp. orange) area (resp. curves) coincides with the value of $F$ and $HN$ which can be taken by the HMC (resp. the RNN).

2) HMC Case: As a consequence of Proposition 2, a factorizable covariance series $(r_k)_{k \in \mathbb{N}}$ can be produced by an HMC if there exists a $\tilde{P}$ in $\mathcal{P}$ such that $\tilde{P} = N(HF)^{-1}$; equivalently, $\tilde{P}$ has to satisfy $\tilde{P} = N(HF)^{-1}$ and $\tilde{P} \in (0, r_0 H^{-2}]$. So condition (56) for $HN$ becomes

$$
\begin{align*}
0 < HN & \leq r_0 F, \quad \text{if } F \geq 0 \\
r_0 F & \leq HN < 0, \quad \text{if } F \leq 0.
\end{align*}
$$

3) D-GUM and RNN Cases: Remember that for a D-GUM, the first inequality of (54) becomes an equality. So polynomial $\Xi_{/H^2}$ in (55) is equal to zero, and system (52)-(54) admits two solutions, $P_{/H^2,1}$ and $P_{/H^2,2}$. Consequently, as the GUM, a D-GUM can produce any covariance series, but requires less parameters since $\alpha = 0$.

Finally, the additional RNN constraint becomes $P = r_0 S^2 R^{-2}$. In the same time $P$ has to satisfy $P = P_{/H^2,1}$ or $P = P_{/H^2,2}$. Using elementary calculus, these new systems have a solution if $HN = r_0 F$, or $HN = r_0 F(2F^2 - 1)$.

A graphical representation of the expressivity of each generative model in function of parameters $F$ and $HN$ is given in Fig. 3 below. Note that starting from a pair $(F,HN)$ it is possible to describe all the corresponding state-space models (42) described by a GUM, an HMC or an RNN, by identifying the possible associated values of $P$, and next applying (52) and (44)-(45). Actually, by applying this reversion step we turn back to the one-dimensional results which were obtained in [24] via the Caratheodory theorem.

VI. NUMERICAL EXPERIMENTS

A. About the Worked Example

We first illustrate the results of Section V-C by providing experiments in the scalar case, under the stationary assumption. Precisely, we generate a sequence from a particular HMC or an RNN of Fig. 3 and we next estimate the parameters of each model (under the stationary constraint) by using a gradient approach based on the auto-differentiation method [56]. We use the ADAM optimiser with a constant learning rate [57] to optimise the likelihoods $p_\theta(x_{0:T})$. Note that they can be computed exactly and sequentially from the KF (see Appendix A3). The corresponding code can be found at https://github.com/AchilleSalaun/GUM. Once the models have been trained, we compare them with regard to the negative log-likelihood associated to 100 new sequences of length $T = 1000$.

Fig. 4 describes the behaviour of each model during the training step. In the left subplot, the targeted distribution is an HMC (but not an RNN). The observed behaviour matches our theoretical expectations: the HMC outperforms the RNN, achieving after training an averaged negative log-likelihood of 1.39 (variance: 0.00045) versus 1.40 (variance: 0.00048) for the RNN. In the central figure, the targeted distribution is an RNN (not an HMC). If the HMC’s behaviour is expected, it can be noted that the RNN does not converge to the exact targeted distribution. However, it can be seen on the right subplot of Fig. 4 that the final model has the same likelihood as the targeted one and so it illustrates the classical problem of the initialisation of the parameters in such a gradient approach. Finally, in this case, the RNN outperforms the HMC with a negative log-likelihood of 1.39 (variance: 0.00034) versus 1.42 (variance: 0.00041) for the HMC.

This experiment recalls that while expressivity is crucial, other factors like model initialisation also determine model performance. That said, theoretical analysis is key to disentangling these aspects, since they cannot be isolated purely experimentally.

B. General Comparison on Synthetic Data

We now generate sequences of $T = 1000$ observations according to GUMs (15)-(17) (or equivalently the state-space model (30)-(31)) with $n = 1$, $n = 3$ or $n = 5$ from which we estimate the parameters of each model. For $n = 1$, we have extracted a GUM from Fig. 3 with $F = 0.05$ and $HN = 0.5$, while for $n = 3$ and $n = 5$ the parameters of the GUMs of reference have been chosen randomly.

We underline that even if we consider linear models, such state-space models are widely used for time series data since they can represent such well known models as the ARMA model [1], [58].

In Fig. 5, we have displayed the evolution of the negative log-likelihood on the test data sets during the estimation step. It can be observed that the RNN tends to converge faster to its limit, which is in part due to the small number of parameters to estimate (remember that the covariance matrix $\alpha$ is null in this case). However, as is shown in Table 1, the (estimated) GUM is the model that provides the best results in the sense that its averaged log-likelihood on the test data sets are the largest. Because we have considered linear models, it can be noted that these averaged likelihoods are close but that their associated variances have the same order of magnitude as the experiment of Section VI-A.

These results tend to confirm that the GUM has to be considered directly as the reference generative model, but due to its large number of parameters, the HMM and the RNN provide interesting, faster alternatives; we also note that the performances of the HMC and the RNN depend on the considered scenario, which is in accordance with Fig. 3.
DESBOUVRIES et al.: EXPRESSIVITY OF HMC VS. RNN FROM A SYSTEM THEORETIC VIEWPOINT 4187

Fig. 4. Illustration of the worked example of Section V-C. The left and central figures show the evolution of the variables \( A = F \) and \( B = HN \) during the training process of the HMC and RNN models when the data follow an HMC distribution (left) or an RNN one (center). The blue cross is the targeted model and the circle is the initial model. The right figure describes the negative log-likelihood across the parallelogram w.r.t. the RNN distribution targeted in the central figure. Unlikely and likely models are displayed in yellow and green, respectively.

Fig. 5. Evolution of the negative log-likelihood on the test data sets during the parameters estimation.

TABLE I
NEGATIVE LOG-LIKELIHOODS FOR EACH ESTIMATED MODEL. THE COLUMN TARGET DISPLAYS THE NEGATIVE LOG-LIKELIHOOD REACHED BY THE TARGETED GUM ON TEST DATA

| n    | learning rate | iteration | HMC  | RNN  | GUM  | Target |
|------|---------------|-----------|------|------|------|--------|
| 1    | \( 2e^{-4} \) | 5000      | 1.2870 | 1.3384 | **1.2847** | 1.1860 |
| 3    | \( 1e^{-4} \) | 5000      | .0679  | .0677 | **.0672** | .0631  |
| 5    | \( 1e^{-4} \) | 6000      | .5491  | .5503 | **.5486** | .5483  |

VII. CONCLUSION

In this paper we addressed a comparative study of HMC and RNN, which are familiar tools for predicting time series. Even though both tools were developed in different communities, we first showed that they indeed share close features when the RNN is turned into a generative model, and thus when HMC and RNN are considered as two latent variables probabilistic models with close enough (conditional) independence structures. Under this framework, both structures can be seen as two different particular instances of a common generative unified model. We next compared both models from the point of view of expressivity, i.e. the relative complexity of the joint probability distribution of an observations sequence, induced by the underlying latent variables. By contrast with previous studies, which were of an experimental nature, our approach consisted in thoroughly quantifying the modeling power of both models. To that end we considered the linear and Gaussian assumption, which induces that the probability distributions of an observations sequence produced by each model are characterized by structured covariance series, which enabled us to call for SRT. We provided implicit conditions under which a given covariance series (and thus a given probability distribution) can be realized by a GUM, an HMC and/or a D-GUM or an RNN; these implicit conditions turn to an explicit cartography of the models in the mono-dimensional case. From this study it happens that the GUM outperforms the RNN and HMC models, in the sense that i) by contrast to the HMC and RNN models, it can produce stationary observed processes endowed with any geometric autocovariance series, while ii) benefiting from the HMC inference tools (computation of the likelihood and of the posterior distributions, for example). Consequently, it happens retrospectively that rather than focusing on the RNN or HMC structures, one should rather focus directly on the GUM structure, which encompasses and extends both models. In practice, since we only have at our disposal observations, one should work directly with a GUM structure and estimate its parameters. This is confirmed by our
can be computed from the predictive likelihoods $p_\theta(x_t|x_{0:t-1}).$ In model (8), for all $s$, $0 \leq s \leq t$, we have

$$p_\theta(x_s|x_{0:s-1}) = \int \left( \int \left( \int \left( \int \right) \right) \right) dh_{s-1:s},$$

where $p_\theta(x_s|h_s)$ and $p_\theta(h_s|x_{0:s-1})$ are the elementary factors in (8). On the other hand, the filtering pdf $p_\theta(h_s|x_{0:s-1})$ can be computed recursively:

$$p_\theta(h_s|x_{0:s}) = \frac{p_\theta(x_s|h_s)}{p_\theta(x_s|x_{0:s-1})} \times \int p_\theta(h_s|x_{0:s-1}) p_\theta(h_{s-1}|x_{0:s-1}) \, dh_{s-1}.$$  

So equations (60) and (61) enable to compute the predictive pdf $p_\theta(x_s|x_{0:s-1})$ and the filtering pdf $p_\theta(h_s|x_{0:s})$ recursively. Given the initial pdf $p_\theta(h_0)$, we first compute $p_\theta(x_s|x_{0:s-1})$ from $p_\theta(h_{s-1}|x_{0:s-1})$ via (60). We next compute $p_\theta(h_s|x_{0:s})$ from $p_\theta(h_{s-1}|x_{0:s-1}), p_\theta(x_s|x_{0:s-1})$ and the HMC transition pdfs. Note that it is the HMC structure (8) that enables this likelihood calculation, at least theoretically (in practice, the one- and second-order moments of the pdfs of interest are another class of approximate solutions [72], [73], which consist in propagating a random, discrete approximation of $p_\theta(h_{0:t}|x_{0:t})$, via an importance sampling mechanism with resampling [75]. Let us start from $p_\theta(h_{0:t}|x_{0:t}) = \sum_{i=1}^{N} \delta_{h_{0:t}^{(i)}}(h_{0:t})$, where $\delta$ is the Dirac mass and $w_t^{(i)}$ is a normalized set of weights. The weighted trajectories $\{h_{0:t}^{(i)}, w_t^{(i)}\}_{i=1}^{N}$ are propagated via three steps. For all $s \in \mathbb{N}$, the $i$th particle is sampled from a conditional importance pdf $q$:

$$\tilde{h}_{s+1}^{(i)} \sim q(h_{s+1}|h_s^{(i)}).$$

Next we compute its unnormalized weight

$$w_{s+1}^{(i)} = w_s^{(i)} p_\theta(h_{s+1}^{(i)}|h_s^{(i)}) p_\theta(x_{s+1}|\tilde{h}_{s+1}^{(i)}) / q(\tilde{h}_{s+1}^{(i)}|h_s^{(i)}),$$

which is normalized as $\tilde{w}_{s+1}^{(i)} = w_{s+1}^{(i)}/\sum_{j=1}^{N} w_{s+1}^{(j)}.$ Finally the trajectories can be resampled, i.e. $h_{0:t}^{(i)} \sim \sum_{j=1}^{N} \tilde{w}_{s+1}^{(j)} \delta_{(h_{0:t}^{(i)}, \tilde{h}_{s+1}^{(i)})}(h_{0:t}),$ and given new weights $w_{s+1}^{(i)} = \frac{1}{N}$. This optional resampling step keeps a larger proportion of trajectories with strong weights, to the detriment of those of low weight. If the trajectories are not resampled,
then \( \tilde{h}^{(i)} \) (resp. \( u^{(i)} \)) reduces to \( \tilde{h}^{(i)} \) (resp. \( u^{(i)} \)). With or without resampling, the procedure is repeated from (65).

The unnormalized weights computed in (66) enable in turn to compute an approximation of the predictive likelihood: \( \tilde{p}_0(x_t|x_{0:t-1}) = 1 \sum_{i=1}^{N} \tilde{u}^{(i)} \), from which an estimate of the likelihood is computed from (7). Finally \( \tilde{p}_0(h_{0:t}|x_{0:t}) \) also provides an approximation of (62):

\[
\tilde{E}_{\theta'} [\log \tilde{p}_0(h_{0:t}, x_{0:t})|x_{0:t}] = \sum_{i=1}^{N} \tilde{u}^{(i)} \log \tilde{p}_0(h^{(i)}_{0:t}, x_{0:t}),
\]

which still remains to be maximized (notation \( u^{(i)} \) recalls that weights are built from parameter \( \theta' \), see (66)). In practice approximation \( \tilde{p} \) can be poor, in particular when \( N \) is very small w.r.t. \( t \). As a possible rescue one can use particle smoothing algorithms [76], [77], [78], [79], which aim at improving the approximation of \( p(h_{s-1}, h_s|x_{0:t}) \) in (63).

Case 3: discrete latent states.

HMC with discrete latent states were introduced in the 1960’s [80], [81], [82] and have been used in such fields as language processing [21], [22], bioinformatics [83] or digital communications [82], [84].

In the discrete case, the problem is that computing the likelihood as \( p_0(x_{0:t}) = \sum_{h_{0:t}} p_0(h_{0:t}, x_{0:t}) \), i.e. via brute force marginalization of the full joint pdf, is unfeasible due to the exponential cost. The success of HMC comes from the fact that the computation of the likelihood \( p_0(x_{0:t}) \) can be performed in linear time. Indeed the likelihood can be seen as another marginalized pdf:

\[
p_0(x_{0:t}) = \sum_{h_s} p_0(h_s, x_{0:s}) \alpha(h_s) \tag{67},
\]

in which pdfs \( \alpha(h_s) \) can be computed recursively in linear cost in the forward time direction:

\[
\alpha(h_0) = p_0(h_0)p_0(x_0|h_0) \tag{68}
\]

\[
\alpha(h_{s+1}) = \sum_{h_s} p_0(x_{t+1}|h_{s+1}) \alpha(h_s) \tag{69}.
\]

Note that \( \alpha(h_s) \) is proportional to the filtering probability mass function, and that (69) is the discrete analog of (61). As for the predictive likelihood, it reads

\[
p(x_{s+1}|x_{0:s}) = \sum_{h_s} p(x_{s+1}|h_{s+1}) \sum_{h_s} p(h_{s+1}|h_s) \alpha(h_s) \tag{60}.
\]

From (63) (where integrals become sums), running the EM algorithm requires calculating, for all \( s \), \( 0 \leq s \leq t \), pdf \( p_0(h_{s-1}, h_s|x_{0:t}) \), which is proportional to

\[
p_0(h_{s-1}, h_s, x_{0:t}) = p_0(x_{s+1}|h_s) \beta(h_{s-1})p_0(h_{s-1}|x_{0:t}) \tag{70}
\]

In particular, pdf \( p_0(h_{s-1}, h_s, x_{0:t}) \) depends on the backward pdfs \( \beta(h_s) = p_0(x_{s+1}|h_s) \) which, similarly to pdfs \( \alpha(h_s) \), can be computed recursively at linear cost, but in the reverse time direction (whence the term backward):

\[
\beta(h_{t}) = 1 \tag{71}
\]

\[
\beta(h_s) = \sum_{h_{s+1}} \beta(h_{s+1})p_0(h_{s+1}|h_s)p_0(h_{s+1}|h_{s-1}) \tag{72},
\]

The recursive calculation of functions \( \alpha(h_s) \) and \( \beta(h_s) \), for all \( s \), is the so-called forward-backward algorithm [22], [80], [81]. Finally from (70) we have

\[
p_0(h_{s-1}, h_s|x_{0:t}) = \frac{\beta(h_s)\alpha(h_{s-1})p_0(h_{s-1})p_0(x_{s}|h_s)}{\sum_{h_{s-1}} \beta(h_{s-1})\alpha(h_{s-1})p_0(h_{s-1})p_0(x_{s}|h_s)}. \tag{73}
\]

It remains to maximize w.r.t. \( p_0(h_1|h_0) \) and \( p_0(x_1|h_0) \). Computing \( p_0(h_{s-1}, h_s, x_{0:t}) \) enables to update these pmfs \( f \) pdfs in the M step of the EM algorithm; this version of the EM algorithm, applied to discrete latent states HMC, is called the Baum-Welch algorithm [22], [85]. Finally observe that the forward-backward algorithm enables to compute the smoothing pmf (for a given, fixed parameter \( \theta \)), since from (73) we get

\[
p_0(h_s|x_{0:t}) = \frac{\alpha(h_s)\beta(h_s)}{\sum_{h_{s-1}} \alpha(h_{s-1})\beta(h_{s-1})}. \tag{8642}
\]

B. Proof of Proposition 2

According to Remark 1, the problem reduces to studying whether among the set of all solutions, there exists at least one such that \( e = 0 \), and thus (see (44)) \( S = 0 \). So we need to solve (41) with the additional constraint

\[
N = FPH^T , \tag{14}
\]

whence (46). However, this raises the following question: although the set \( \mathcal{P}_1 \) in which we look for an HMC solution is built from a triplet \( (H_1, F_1, N_1) \), produced by the DR step, this triplet is not unique; if \( \mathcal{P}_1 \) had no HMC solution, could another set \( \mathcal{P}_2 \), built from another triplet \( (H_2, F_2, N_2) \), nevertheless contain an HMC solution?

We thus need to study the relation between \( \mathcal{P}_1 \) and \( \mathcal{P}_2 \). Since \( (H_1, F_1, N_1) \) is of minimal degree, from (40) any other minimal degree solution \( (H_2, F_2, N_2) \) can be computed from \( (H_1, F_1, N_1) \) via an invertible matrix \( T_{12} \). Let \( P_2 \) be an element of \( \mathcal{P}_2 \). By pre- (respectively post-) multiplying equation (41), with parameters \( (H_1, F_1, N_1) \), by \( \begin{bmatrix} T_{12} & 0 \\ 0 & 1 \end{bmatrix} \) (respectively \( \begin{bmatrix} T_{12}^T & 0 \\ 0 & 1 \end{bmatrix} \), one can show easily that

\[
P_2 = T_{12}P_1T_{12}^T, \tag{75}
\]

which we denote simply by the set equation \( \mathcal{P}_2 = T_{12}P_1T_{12}^T \). This is summarized by Fig. 6 below.

Let now a triplet \( (H_1, F_1, N_1) \) and a solution \( P_1 \in \mathcal{P}_1 \), such that

\[
N_1 = F_1P_1H_1^T . \tag{16}
\]

This matrix \( P_1 \) is thus an HMC solution of \( (r_k)_{k \in \mathbb{N}} \). Equation (76) is equivalent to

\[
T_{12}N_1 = T_{12}F_1T_{12}^{-1}T_{12}P_1T_{12}^T T_{12}^T H_1^T ; \tag{77}
\]

so from (40) and (75), we see that \( P_2 = T_{12}P_1T_{12}^T \) is one HMC element belonging to set \( \mathcal{P}_2 \). In other words, if there exists an HMC element in the set \( \mathcal{P}_1 \) associated to a triplet \( (H_1, F_1, N_1) \) from the equivalence class produced by the DR step, then any
and it suffices to know the appropriate invertible matrix out of an infinity of solutions to this problem. These solutions are isomorphic (set $\{\cdots\}$).

**Fig. 6.** Starting from a factorizable covariance series $(r_k)_{k\in\mathbb{N}}$ (see Fig. 2), the deterministic realization step (in blue) consists in finding a triplet $(H, F, N)$ representing the series under study. This step provides one solution out of an infinity of solutions to this problem. These solutions are isomorphic and it suffices to know the appropriate invertible matrix $T_{12}$ to move from a given solution $(H_1, F_1, N_1)$ to another $(H_2, F_2, N_2)$ (dashed arrow). The SR step (in orange) amounts to finding a state-space system modeling function $P_2 = T_{12} P_1 T_{12}^T$ (with $T_{12}$ an arbitrary invertible matrix) also obtained at the previous step. A triplet $(H_1, F_1, N_1)$ leads to a set of solutions $P_1$. These sets are also isomorphic and $T_{12}$ suffices for moving from $P_1$ to $P_2$, respectively obtained from $(H_1, F_1, N_1)$ and $(H_2, F_2, N_2)$ (dashed arrow).

set $P_2 = T_{12} P_1 T_{12}^T$ (with $T_{12}$ an arbitrary invertible matrix) also contains an HMC solution. Similarly, if there is no such element in $P_1$, then no set $P_2 = T_{12} P_1 T_{12}^T$ will contain a solution either. Finally it suffices to look for an HMC solution in the set produced by the SR algorithm, without bothering any longer of the other elements in the equivalence class.

**REFERENCES**

[1] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications*, New York, NY, USA: Springer-Verlag, 2000.

[2] R. Douc, E. Moulines, and D. Stoffer, *Nonlinear Time Series: Theory, Methods and Applications With R Examples*. Boca Raton, FL, USA: CRC Press, 2014.

[3] A. M. Deshmukh, “Comparison of hidden Markov model and recurrent neural network in automatic speech recognition,” *Eur. J. Eng. Technol. Rev.*, vol. 5, pp. 958–965, Aug. 2020.

[4] B. Biskmukhamedov, A. Nadeev, G. Maione, and D. Striccoli, “Comparison of HMM and RNN models for network traffic modeling,” *Internet Technol. Lett.*, vol. 3, no. 2, Mar./Apr. 2020, Art. no. e147.

[5] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis*. New York, NY, USA: Springer-Verlag, 1985.

[6] C. Robert, *The Bayesian Choice* (Springer Texts in Statistics). New York, NY, USA: New York, 1994.

[7] A. Borovkov, *Statistique Mathématique*. Moscow, Russia: Editions Mir, 1987.

[8] L. Wasserman, *All of Statistics: A Concise Course in Statistical Inference*. New York, NY, USA: Springer Science & Business Media, 2013.

[9] A. P. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. Roy. Statistical Soc. B*, vol. 39, no. 1, pp. 1–22, 1977.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer Science & Business Media, 2009.

[12] J. H. Manton and P.-O. Amblard, “A primer on reproducing kernel Hilbert spaces,” *Found. Trends® Signal Process.*., vol. 8, nos. 1–2, pp. 1–126, 2015.

[13] V. I. Paulsen and M. Raghupathi, *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*, vol. 152. Cambridge, U.K.: Cambridge University Press, 2016.

[14] A. K. Jain, J. Mao, and K. M. Mohiuddin, “Artificial neural networks: A tutorial,” *Computer*, vol. 29, no. 3, pp. 31–44, Mar. 1996.

[15] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[16] C. I. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[17] W. Hu, Y. Liao, and V. R. Vemuri, “Robust anomaly detection using support vector machines,” in *Proc. Int. Conf. Mach. Learn.*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, pp. 282–289.

[18] V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer Science & Business Media, 2013.

[19] I. Goodfellow, Y. Bengio, and A. Courville, and Y. Bengio, *Deep Learning*, Cambridge, MA, USA: MIT Press, 2016.

[20] V. N. Vapnik, *Statistical Learning Theory*. New York, NY, USA: J. Wiley & Sons, cop., 1998.

[21] L. R. Rabine and B. H. Jiang, “An introduction to hidden Markov models,” *IEEE ASSP Mag.*, vol. 3, no. 1, pp. 4–16, Jan. 1986.

[22] L. R. Rabine, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[23] J. T. Connor, R. D. Martin, and L. E. Atlas, “Recurrent neural networks and robust time series prediction,” *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 240–254, Mar. 1994.

[24] A. Salain, Y. Petetin, and F. Desbovrières, “Comparing the modeling powers of RNN and HMM,” in *Proc. 18th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Piscataway, NJ, USA: IEEE Press, 2019, pp. 1496–1499.

[25] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models* (Springer Series in Statistics). Springer-Verlag, 2005.

[26] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur, “Extensions of recurrent neural network language model,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Piscataway, NJ, USA: IEEE Press, 2011, pp. 5528–5531.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[28] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Math. Control, Signals Syst.*, vol. 2, no. 4, pp. 303–314, 1989.

[29] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural Netw.*, vol. 4, no. 2, pp. 251–257, 1991.

[30] A. Pinkus, “Approximation theory of the MLP model in neural networks,” *Acta Numerica*, vol. 8, no. 1, pp. 143–195, 1999.

[31] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, “The expressive power of neural networks: A view from the width,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6231–6239.

[32] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” *California Univ San Diego La Jolla Inst for Cognitive Science, Fort Belvoir, VA, USA, Tech. Rep.*, 1985. [Online]. Available: https://apps.dtic.mil/sti/pdfs/ADA164453.pdf

[33] A. Robinson and F. Fallside, *The Utility Driven Dynamic Error Propagation Network*. Cambridge, MA, USA: University of Cambridge Department of Engineering, 1987.

[34] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.

[35] M. C. Mozer, “A focused backpropagation algorithm for temporal pattern recognition,” in *Backpropagation: Theory, Architectures, and Applications*, vol. 137, Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1995, pp. 137–169.

[36] M. Sangiorgio, F. Deroce, and G. Guariso, “Neural predictors’ sensitivity and robustness,” in *Deep Learning in Multi-Step Prediction of Chaotic Dynamics: From Deterministic Models to Real-World Systems*. Cham, Switzerland: Springer International Publishing, 2021, pp. 85–96.

[37] E. Tan, D. Correa, T. Stember, and M. Small, “Backpropagation on dynamical networks,” 2022, arXiv:2207.03093.

[38] Y. Goldberg, *Neural Network Methods for Natural Language Processing* (Synthesis Lectures on Human Language Technologies), vol. 10. Cham, Switzerland: Springer, 2017, pp. 1–309.
François Desbovriès (Senior Member, IEEE) received the B.E. degree in telecommunications and the Ph.D. degree in signal and image processing from Telecom Paris, Paris, France, in 1987 and 1991, respectively, and the HDR degree in signal processing from the Université de Marne la Vallée, France, in 2001. He is a Full Professor with Telecom SudParis, Palaiseau, France. He served as an Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2014 to 2018. His research interests are in computational statistics, Bayesian inference, Markovian models, and statistical machine learning.

Yohan Petetin (Member, IEEE) received the Ph.D. degree in applied mathematics from Telecom SudParis, Evry, France, in 2013, and the HDR degree in applied mathematics from the Institut Polytechnique de Paris, Palaiseau, France, in 2023. He is currently an Associate Professor with Telecom SudParis, Palaiseau, France. His research interests are in statistical machine learning, Monte Carlo methods, and sequential Bayesian inference.

Achille Salain graduated from Telecom Paris, in 2017. He received the Ph.D. degree in applied mathematics from the Institut Polytechnique de Paris, in 2021, with a Ph.D. devoted to alarm prediction in networks via space-time pattern matching and machine learning. He has been a Postdoctor with the Institute of Biomedical Engineering, Department of Engineering, University of Oxford, since 2022, where he currently develops interpretable AI tools to support clinical decisions in organ transplantation.