Journal of Machine Learning Research 25 (2024) 1-64 Submitted 1/23; Revised 12/23; Published 2/24

Probabilistic Forecasting with Generative Networks via Scoring Rule Minimization

Lorenzo Pacchiardi LORENZO.PACCHIARDI@GMAIL.COM
Department of Statistics, University of Oxford
Oxford, OX1 3LB
United Kingdom

Rilwan A. Adewoyin* RILWAN.ADEWOYIN@WARWICK.AC.UK
Department of Statistics, University of Warwick
Coventry, CV4 7AL
United Kingdom

Peter Dueben PETER.DUEBEN@ECMWF.INT
Earth System Modelling Section, European Centre for Medium-Range Weather Forecasts
Reading, RG2 9AX
United Kingdom

Ritabrata Dutta RITABRATA.DUTTA@WARWICK.AC.UK
Department of Statistics, University of Warwick
Coventry, CV4 7AL
United Kingdom

Editor: Daniel Roy

Abstract

Probabilistic forecasting relies on past observations to provide a probability distribution for a future outcome, which is often evaluated against the realization using a *scoring rule*. Here, we perform probabilistic forecasting with generative neural networks, which parametrize distributions on high-dimensional spaces by transforming draws from a latent variable. Generative networks are typically trained in an *adversarial* framework. In contrast, we propose to train generative networks to minimize a predictive-sequential (or *prequential*) scoring rule on a recorded temporal sequence of the phenomenon of interest, which is appealing as it corresponds to the way forecasting systems are routinely evaluated. Adversarial-free minimization is possible for some scoring rules; hence, our framework avoids the cumbersome hyperparameter tuning and uncertainty underestimation due to unstable adversarial training, thus unlocking reliable use of generative networks in probabilistic forecasting. Further, we prove consistency of the minimizer of our objective with dependent data, while adversarial training assumes independence. We perform simulation studies on two chaotic dynamical models and a benchmark data set of global weather observations; for this last example, we define scoring rules for spatial data by drawing from the relevant literature. Our method outperforms state-of-the-art adversarial approaches, especially in probabilistic calibration, while requiring less hyperparameter tuning.

Keywords: Generative Networks, GAN, Probabilistic Forecasting, Scoring Rules, Adversarial-free.

∗. Also affiliated with Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China

1. Introduction

In many disciplines (for instance econometrics and meteorology), practitioners want to forecast the future state of a phenomenon. Providing prediction uncertainty (ideally by stating a full probability distribution) is often essential. This task is called *probabilistic forecasting* (Gneiting and Katzfuss, 2014) and is commonplace in Numerical Weather Prediction (NWP, Palmer, 2012), where physics-based models are run multiple times to obtain an ensemble of forecasts representing the possible evolution of the weather (Leutbecher and Palmer, 2008). To assess the performance of NWP systems, people commonly use Scoring Rules (SRs, Gneiting and Raftery, 2007), functions quantifying the quality of a probabilistic forecast in relation to the observed outcome.

Here, we use *generative (neural) networks* to provide probabilistic forecasts. In a generative network, a neural network maps a latent random variable to the required output space; hence, samples on the latter are obtained by transforming latent variable draws. As the density is inaccessible, the distribution is implicitly defined and specialized techniques are necessary to train generative networks. Among those, the popular Generative Adversarial Networks (GANs, Goodfellow et al., 2014; Mirza and Osindero, 2014; Nowozin et al., 2016; Arjovsky et al., 2017) framework trains a generative network by defining a min-max game against a competitor, termed *critic*. However, adversarial training is unstable: it requires ad-hoc strategies (Gulrajani et al., 2017) and careful hyperparameter tuning (Salimans et al., 2016) but, even so, the trained generative network may not fully capture the data distribution, a phenomenon referred to as *mode collapse* (Goodfellow, 2016; Isola et al., 2017; Arora et al., 2017; Bellemare et al., 2017; Arora et al., 2018; Richardson and Weiss, 2018). This prevents practitioners from reliably applying GANs to tasks where calibrated uncertainty quantification is paramount, such as probabilistic forecasting. Additionally, it is unclear how to extend the GAN training objective to the temporal data considered in probabilistic forecasting. Indeed, the adversarial framework is derived from divergences between probability distributions and considers data as independent and identically distributed samples from one of those distributions.

Therefore, motivated by the use of scoring rules to evaluate traditional forecasting systems, we propose to train generative networks to minimize scoring rule values. Given a recorded temporal sequence of the phenomenon of interest, we use the generative network to forecast all steps of the sequence conditioned on the past. Then, our objective is the average over steps of the scoring rule between forecasts and realizations. In contrast to the adversarial framework, this so-called *prequential* (predictive-sequential, Dawid, 1984) scoring rule captures the temporal structure of the data. Additionally, the minimizer of the prequential scoring rule enjoys consistency under mild conditions on the temporal sequence. Furthermore, our proposal allows adversarial-free training through a reparametrization trick (Kingma and Welling, 2014) for SRs defined as expectations over the generative distribution. Training with our objective is therefore drastically easier than with GAN, requires less hyperparameter tuning and easily avoids mode collapse. More in detail, our contributions are:

- We introduce a novel training objective for probabilistic forecasting based on a prequential scoring rule.
- Under stationarity and mixing conditions of the time series, we prove that the minimizer of the prequential scoring rule coincides asymptotically with that of the expected

prequential scoring rule. Importantly, the latter corresponds to the true parameter value if the distribution induced by the generative network is well-specified.

- We leverage previous works in meteorology (Gneiting and Raftery, 2007; Scheuerer and Hamill, 2015) and design training objectives for high-dimensional spatio-temporal data, enabling good performance with no need for a learnable data transformation.
- We test our method and state-of-the-art adversarial approaches on two chaotic models and a spatio-temporal weather data set. We find our method to be more stable and perform better, particularly in terms of uncertainty quantification of the forecast.

The rest of the paper is organized as follows. In Sec. 2, we discuss how the adversarial framework is obtained from a divergence minimization setup and overview the scoring rules training formulation for independent data, which was considered in previous works. In Sec. 3, which contains the main contributions of our work, we give our training objective for probabilistic forecasting, show its consistency and discuss SRs for spatial data. We discuss some related works in Sec. 4 and show simulation results in Sec. 5. We conclude in Sec. 6.

Notation: We use upper case X, Y and Z to denote random variables, and their lower-case counterpart to denote observed values. Bold symbols denote vectors, and subscripts to bold symbols denote sample index (for instance, $\mathbf{y}_t$). Instead, subscripts to normal symbols denote component indices (for instance, y_i is the i-th component of $\mathbf{y}$, and $y_{t,j}$ is the j-th component of $\mathbf{y}_t$). Finally, we use notation $\mathbf{y}_{j:k} = (\mathbf{y}_j, \mathbf{y}_{j+1}, \ldots, \mathbf{y}_{k-1}, \mathbf{y}_k)$, for $j \leq k$.

2. Background

2.1 Generative networks via divergence minimization

A generative network represents a distribution P^ϕ on a space $\mathcal{Y}$ via a map $h_\phi : \mathcal{Z} \to \mathcal{Y}$ transforming samples from a probability distribution Q over the space $\mathcal{Z}$; the map is parametrized by a Neural Network (NN) with weights ϕ. Samples from P^ϕ are obtained by generating $\mathbf{z} \sim Q$ and computing $h_\phi(\mathbf{z}) \in \mathcal{Y}$; therefore, for any function g on $\mathcal{Y}$, the expectation $\mathbb{E}_{\mathbf{Y}\sim P^\phi}[g(\mathbf{Y})]$ can be computed by $\mathbb{E}_{\mathbf{Z}\sim Q}[g(h_\phi(\mathbf{Z}))]$. However, in general, the probability density of P^ϕ cannot be evaluated.

Assume now we observe data from a distribution $P^\star$ on $\mathcal{Y}$ and want to tune ϕ so that P^ϕ approximates $P^\star$. A divergence $D(P^\star||P^\phi)$ is a function of two distributions such that $D(P^\star||P^\phi) \geq 0$ and $D(P^\star||P^\phi) = 0 \iff P^\star = P^\phi$. Therefore, for a given D, we can attempt solving

$$\arg\min_\phi D(P^\star||P^\phi). \tag{1}$$

Various proposed approaches differ according to (i) their choice of divergence D and (ii) how they estimate the optimal solution in Eq. (1) using samples from $P^\star$ and P^ϕ. A popular strategy is choosing D to be an f-divergence (termed f-GAN, Nowozin et al., 2016), in which case a variational lower bound can be obtained

$$D_f(P^\star||P^\phi) \geq \sup_{c\in\mathcal{C}} \left(\mathbb{E}_{\mathbf{Y}\sim P^\star} c(\mathbf{Y}) - \mathbb{E}_{\mathbf{X}\sim P^\phi} f^*(c(\mathbf{X}))\right),$$

where f^* is the Fenchel conjugate of the function f (see Appendix B.1.1) and $\mathcal{C}$ is any set of functions from $\mathcal{Y}$ to the domain of f^*. By representing the set $\mathcal{C}$ by a neural network c_ψ

(termed *critic* or *discriminator*) with parameters $\psi \in \Psi$, an equivalent problem to Eq. (1) when D is an f-divergence is

$$\arg\min_{\phi}\max_{\psi}\left(\mathbb{E}_{\mathbf{Y}\sim P^\star}c_\psi(\mathbf{Y}) - \mathbb{E}_{\mathbf{X}\sim P^\phi}f^*(c_\psi(\mathbf{X}))\right). \tag{2}$$

The WGAN of Arjovsky et al. (2017), which uses the 1-Wasserstein distance as D, has a similar objective to Eq. (2), differing mainly in taking $\mathcal{C}$ to be the set of 1-Lipschitz functions. Details are given in Appendix B.1.2.

Typically, the problem in Eq. (2) is tackled by alternating gradient optimization steps over ψ and ϕ; the expectations are estimated via samples from both $P^\star$ (i.e., a minibatch of observations) and from P^ϕ (draws from the generative network). This approach is termed *adversarial* as P^ϕ and c_ψ respectively aim to minimize and maximize the same objective.

Adversarial training of generative networks is however unstable and difficult. A well-known consequence of unstable adversarial training is mode collapse (Goodfellow, 2016; Isola et al., 2017; Arora et al., 2017; Bellemare et al., 2017; Arora et al., 2018; Richardson and Weiss, 2018), in which the generative distribution underestimates uncertainty and, in extreme cases, can collapse to a single point. Mode collapse has been related to the approximations involved in adversarial training: Arora et al. (2017) showed that mode collapse can arise due to finite capacity of the critic c_ψ, while Bellemare et al. (2017) and Bińkowski et al. (2018) respectively linked it to using finite data and a finite number of steps in optimizing the c_ψ network and subsequently using it to obtain gradient estimates for ϕ, which are thus biased.

To avoid adversarial training altogether and bypass the above issues, Moment Matching Networks (Li et al., 2015; Dziugaite et al., 2015) are trained by considering D to be the squared Maximum Mean Discrepancy (MMD) induced by a positive definite kernel k

$$D_k\left(P^\star||P^\phi\right) := \mathbb{E}\left[k\left(\mathbf{X},\mathbf{X}'\right) - 2k(\mathbf{X},\mathbf{Y}) + k\left(\mathbf{Y},\mathbf{Y}'\right)\right], \quad \mathbf{X},\mathbf{X}'\sim P^\phi, \quad \mathbf{Y},\mathbf{Y}'\sim P^\star \tag{3}$$

From Eq. (3), we can obtain an empirical unbiased estimate of D_k and its gradients without introducing a critic network. However, using a fixed kernel on raw data can yield small discriminative power (as in the case of images, where numerical values have little meaning), leading to a poor fit of P^ϕ to $P^\star$. Hence, Li et al. (2017) suggested applying a learnable transformation before computing the kernel, with parameters trained to maximize the MMD. This approach, termed MMD-GAN, again leads to an adversarial setting and to the issues mentioned above. Details in Appendix B.1.3.

2.1.1 Conditional setting

To represent a conditional distribution $P^\phi(\cdot|\boldsymbol{\theta})$, for $\boldsymbol{\theta}\in\Theta$, a map $h_\phi : \mathcal{Z}\times\Theta\to\mathcal{Y}$ can be used; similarly to above, samples from $P^\phi(\cdot|\boldsymbol{\theta})$ for fixed $\boldsymbol{\theta}$ can be obtained via $h_\phi(\mathbf{z};\boldsymbol{\theta})$, $\mathbf{z}\sim Q$. In this way, f-GAN, WGAN and MMD-GAN can all be easily extended to the setting in which we have data

$$(\boldsymbol{\theta}_i,\mathbf{y}_i)_{i=1}^n, \text{ where } \boldsymbol{\theta}_i\sim\Pi \text{ and } \mathbf{y}_i\sim P^\star(\cdot|\boldsymbol{\theta}_i), \tag{4}$$

and want $P^\phi(\cdot|\boldsymbol{\theta}) = P^\star(\cdot|\boldsymbol{\theta})$ Π-almost everywhere. For instance, the f-GAN objective in Eq. (2) becomes

$$\min_{\phi}\max_{\psi}\mathbb{E}_{\boldsymbol{\theta}\sim\Pi}\big(\mathbb{E}_{\mathbf{Y}\sim P^\star(\cdot|\boldsymbol{\theta})}c_\psi(\mathbf{Y};\boldsymbol{\theta}) - \mathbb{E}_{\mathbf{Y}\sim P^\phi(\cdot|\boldsymbol{\theta})}f^*(c_\psi(\mathbf{Y};\boldsymbol{\theta}))\big),$$

where now $c_\psi : \mathcal{Y} \times \Theta \to \text{dom}_{f^*}$. More details can be found in Appendix B.1.

2.2 Generative networks via scoring rules minimization

Here, we review scoring rules and a formulation for training generative networks based on them which, for some choices, is intrinsically adversarial-free.

2.2.1 Scoring rules

A Scoring Rule (SR) S is a function of a distribution and an observation; see Gneiting and Raftery (2007); Dawid and Musio (2014) for an overview of their properties and usage. Generally, $S(P^\phi, \mathbf{y})$ represents a *penalty* assigned to the distribution P^ϕ when $\mathbf{y}$ is observed. If $\mathbf{y}$ is the realization of a random variable $\mathbf{Y} \sim P^\star$, the expected SR is $S(P^\phi, P^\star) := \mathbb{E}_{\mathbf{Y}\sim P^\star} S(P^\phi, \mathbf{Y})$. S is said to be *proper* relative to a set of distributions $\mathcal{P}$ if the expected Scoring Rule is minimized in P^ϕ when $P^\phi = P^\star$

$$S(P^\star, P^\star) \leq S(P^\phi, P^\star) \;\forall\; P^\phi, P^\star \in \mathcal{P}.$$

Moreover, S is *strictly proper* relative to $\mathcal{P}$ if $P^\phi = P^\star$ is the unique minimum. In practice, assuming that $\exists\, \phi^\star : P^{\phi^\star} = P^\star$, $P^\phi \neq P^\star$ can still minimize an expected proper SR $S(P^\phi, P^\star)$, which in turn implies there may be multiple minima (still, the different minima can be thought of as more "similar" to $P^\star$ than other distributions, in some way); instead, if S is strictly proper, $P^\star$ and P^ϕ coincide if and only if P^ϕ is the (unique) minimum of the expected SR. In case where $\nexists\, \phi : P^\phi = P^\star$, then the expected proper SR decreases as P^ϕ becomes more similar to the data distribution $P^\star$; however, nothing can be said on the number of minima without more information on $\mathcal{P}$, even if S is strictly proper.

For a strictly proper SR S, the quantity $D(P^\star||P^\phi) := S(P^\phi, P^\star) - S(P^\star, P^\star)$ is a statistical divergence, as in fact $D(P^\star||P^\phi) \geq 0$ and $D(P^\star||P^\phi) = 0 \iff P^\phi = P^\star$.

A strictly proper SR which we will employ in the following is the *Kernel Score* (Gneiting and Raftery, 2007)

$$S_k(P^\phi, \mathbf{y}) := \mathbb{E}[k(\mathbf{X}, \mathbf{X}')] - 2 \cdot \mathbb{E}[k(\mathbf{X}, \mathbf{y})], \quad \mathbf{X}, \mathbf{X}' \sim P^\phi, \tag{5}$$

where k is a positive-definite kernel. This choice is due to the expectation form of the kernel score, which, as explained in Sec. 2.2.2, is required by our method. The kernel score is associated with the MMD in Eq. (3); see more details in Appendix B.2.

2.2.2 Adversarial-free training of generative networks

SRs have been previously used to train conditional generative networks in Bouchacourt et al. (2016) and Gritsenko et al. (2020), where the authors considered

$$\min_\phi \mathbb{E}_{\boldsymbol{\theta}\sim\Pi} \mathbb{E}_{\mathbf{Y}\sim P^\star(\cdot|\boldsymbol{\theta})} S(P^\phi(\cdot|\boldsymbol{\theta}), \mathbf{Y}); \tag{6}$$

for strictly proper S, the solution is $P^\phi(\cdot|\boldsymbol{\theta}) = P^\star(\cdot|\boldsymbol{\theta})$ Π-almost everywhere. With $(\boldsymbol{\theta}_i, \mathbf{y}_i)_{i=1}^n$ as in Eq. (4), an unbiased estimate of the argument of $\min_\phi$ in Eq. (6) is

$$\frac{1}{n} \sum_{i=1}^n S(P^\phi(\cdot|\boldsymbol{\theta}_i), \mathbf{y}_i). \tag{7}$$

Thus, to optimize Eq. (6) via Stochastic Gradient Descent (SGD), it is enough to obtain unbiased estimates of $\nabla_\phi S(P^\phi(\cdot|\boldsymbol{\theta}_i), \mathbf{y}_i)$. That is possible whenever S is defined via a (possibly repeated) expectation over P^ϕ (as for the kernel score), which can be estimated unbiasedly by generating samples $\mathbf{x}_j \sim P_\phi, j = 1, \dots, m$, $m > 1$ at each SGD step. Additionally, by recalling that samples $\mathbf{x}_j \sim P_\phi$ are obtained as $\mathbf{x}_j = h_\phi(\mathbf{z}), \mathbf{z} \sim Q$, automatic-differentiation libraries (Paszke et al., 2019) can be exploited to compute gradients. Hence, considering the kernel score as an example, at each SGD step, ϕ will be updated by

$$\phi \leftarrow \phi - \gamma \cdot \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nabla_\phi \left[\frac{1}{m(m-1)} \sum_{j \neq k} k(\mathbf{x}_j, \mathbf{x}_k) - \frac{2}{m} \sum_j k(\mathbf{x}_j, \mathbf{y}_i) \right], \tag{8}$$

where γ is the learning rate. More details are given in Appendix C. This algorithm is equivalent to Moment Matching Networks (which use the objective in Eq. 3).

The Energy Score used in Bouchacourt et al. (2016) andGritsenko et al. (2020) can be obtained from S_k by choosing $k(\mathbf{x}, \mathbf{y}) = -||\mathbf{y} - \mathbf{x}||^\beta$ for $\beta \in (0, 2)$ (Gneiting and Raftery, 2007). As such, the Energy Score also takes the expectation form necessary for our method and leads to a gradient descent update similar to Eq. (8). See more details in Appendix B.2.

3. Generative networks for spatio-temporal models via SR minimization

We will now extend the SR formulation to a training objective for probabilistic forecasting (Sec. 3.1) which is intuitive for temporal data and enjoys some consistency (Sec. 3.1.1). Later (Sec. 3.2), we will discuss how to exploit the SR formulation to tackle high dimensional spatial data, by relying on a previously studied score from the probabilistic forecasting and meteorology literature (Scheuerer and Hamill, 2015) and by introducing *patched* scores. The resulting objectives can be minimized without resorting to adversarial training.

3.1 Time-series probabilistic forecasting via the prequential SR

Consider a discrete-time stochastic process $(\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_t, \dots) = (\mathbf{Y}_t)_t \sim P^\star$, where $\mathbf{Y}_t \in \mathcal{Y}$; in general, $\mathbf{Y}_t$'s are *not* independent. For a generic distribution P for $(\mathbf{Y}_t)_t$, we denote by P_t the marginal distribution for $\mathbf{Y}_t$, and by $P_{r:s}$ the marginal distribution for $\mathbf{Y}_{r:s}$; the conditional distribution for $\mathbf{Y}_t|\mathbf{y}_{u:v}$ will be denoted by $P_t(\cdot|\mathbf{y}_{u:v})$ and similar for $\mathbf{Y}_{r:s}$.

Having observed $\mathbf{y}_{1:t}$, we produce a *probabilistic forecast* for $\mathbf{Y}_{t+l}$ for a given lead time l via a generative network conditioned on the last k observations, $P^\phi_{t+l}(\cdot|\mathbf{y}_{t-k+1:t})$. We then repeat this procedure for all t's in a recorded window of length T and evaluate the forecast performance via $S(P^\phi_{t+l}(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l})$ for a SR S (Fig. 1); we then propose setting ϕ to

$$\hat{\phi}_T(\mathbf{y}_{1:T}) := \arg\min_\phi \sum_{t=k}^{T-l} S(P^\phi_{t+l}(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l}), \tag{9}$$

which selects the value of ϕ for which the average l-steps ahead forecast in the training data is optimal according to S. Operationally, Eq. (9) can be tackled in the same way as Eq. (7), i.e., by simulating from P^ϕ for each observation window $\mathbf{y}_{t-k+1:t}$ in a training batch, unbiasedly estimating the SR S and descending the gradient.

The objective in Eq. (9) evaluates sequential predictions obtained from the generative network; as such, we term it the *prequential* (or *predictive-sequential*) score (Dawid, 1984; Dawid and Musio, 2015). This reflects what is usually done in evaluating traditional (physics-based) probabilistic forecasting systems (Leutbecher and Palmer, 2008; Gneiting and Katzfuss, 2014).

3.1.1 Consistency of prequential SR minimization

Contrary to the independent-data setting of Eq. (7), Eq. (9) cannot be seen as the empirical estimate of an expected SR. Still, under some stationarity and mixing conditions of $(\mathbf{Y}_t)_t$, we prove below that the empirical minimizer $\hat{\phi}_T(\mathbf{Y}_{1:T})$ converges to the minimizer of the expected prequential SR. The reader uninterested in theoretical guarantees may skip this section, as it does not contain necessary information for understanding the remained of the paper.

First, the objective in Eq. (9) involves $P^{\phi}_{t+l}(\cdot|\mathbf{y}_{t-k+1:t})$ for $t \in \{k, k+1, \ldots, T-l-1, T-l\}$ and evaluates them against $\mathbf{y}_{k+l:T}$. In contrast, the initial part of the recorded sequence $\mathbf{y}_{1:k+l-1}$ only enters as conditioning values (indeed, the generative network cannot provide a forecast for the first $k+l-1$ elements of the sequence). Formally, we can define the joint distribution on $\mathbf{Y}_{k+l:T}$ induced by the generative network as $P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})$ and interpret the objective in Eq. (9) as a SR evaluating $P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})$ against $\mathbf{y}_{k+l:T}$

$$S_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), \mathbf{y}_{k+l:T}) := \sum_{t=k}^{T-l} S(P^{\phi}_{t+l}(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l}). \tag{10}$$

The above only makes sense as $P^{\phi}_{t+l}(\cdot|\mathbf{y}_{t-k+1:t})$ can be obtained from $P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})$, thanks to the marginal distribution for $\mathbf{Y}_{t+l}$ in $P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})$ being independent on $\mathbf{y}_{1:k+l-1}$ conditionally on $\mathbf{y}_{t-k+1:t}$. If that was not the case, $\mathbf{y}_{1:k+l-1}$ would also appear explicitly in the conditioning of P^{ϕ}_{t+l}. Indeed, $P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})$ satisfies the following property (which generalizes the standard k-Markov property):

Definition 1 *A probability distribution $P_{1:T}$ is k-Markovian with lag l if, assuming it has density $p_{1:T}$ with respect to some base measure, it can be decomposed as: $p_{1:T}(\mathbf{y}_{1:T}) = p_{1:k+l-1}(\mathbf{y}_{1:k+l-1}) \prod_{t=k}^{T-l} p_{t+l}(\mathbf{y}_{t+l}|\mathbf{y}_{t-k+1:t})$.*

Therefore, S_T defined in Eq. (10) is a SR for distributions over $\mathbf{Y}_{k+l:T}|\mathbf{y}_{1:k+l-1}$ which are k-Markovian with lag l. The following result (proved in Appendix A.2.2) establishes that S_T meaningfully evaluates $P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})$ although it only employs $P^{\phi}_{t+l}(\cdot|\mathbf{y}_{t-k+1:t})$ explicitly:

Theorem 2 *If S is (strictly) proper, then S_T is (strictly) proper for distributions over $\mathbf{Y}_{k+l:T}|\mathbf{y}_{1:k+l-1}$ which are k-Markovian with lag l.*

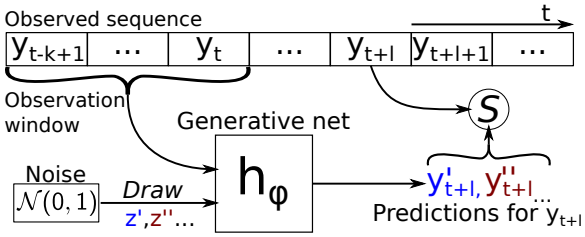


Figure 1: Estimation of the SR evaluating the forecast $P^{\phi}_{t+l}(\cdot|\mathbf{y}_{t-k+1:t})$ for the realization $\mathbf{y}_{t+l}$. The prequential SR is obtained by repeating this procedure for all t's and summing the scores.

Next, we introduce two quantities:

$$\tilde{\phi}_T(\mathbf{y}_{1:k+l-1}) := \arg\min_{\phi} \overbrace{\mathbb{E}_{\mathbf{Y}_{k+l:T}|\mathbf{y}_{1:k+l-1}} S_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), \mathbf{Y}_{k+l:T})}^{:=\tilde{S}_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}))},$$
$$\phi^{\star}_T := \arg\min_{\phi} \underbrace{\mathbb{E} S_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{Y}_{1:k+l-1}), \mathbf{Y}_{k+l:T})}_{:=S^{\star}_T(P^{\phi}_{k+l:T})}.$$

$\tilde{\phi}_T(\mathbf{y}_{1:k+l-1})$ minimizes the expected prequential SR with respect to $\mathbf{Y}_{k+l:T}|\mathbf{y}_{1:k+l-1}$, for which we introduced the short-hand notation $\tilde{S}_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}))$; by Theorem 2, if S is strictly proper and the distribution of $\mathbf{Y}_{k+l:T}|\mathbf{y}_{1:k+l-1}$ is k-Markovian with lag l, $\tilde{\phi}_T(\mathbf{y}_{1:k+l-1})$ parametrizes the true distribution. $\phi^{\star}_T$ instead minimizes the expectation of S_T with respect to the full sequence $\mathbf{Y}_{1:T}$, which we shorten to $S^{\star}_T(P^{\phi}_{k+l:T})$.

Each term in the sum defining S_T depends on a finite number of observations; therefore, if $(\mathbf{Y}_t)_t$ satisfies some mixing and stationarity properties, we expect $\tilde{\phi}_T(\mathbf{y}_{1:k+l-1})$ to not depend on $\mathbf{y}_{1:k+l-1}$ for large T; similarly, we expect the empirical estimator $\hat{\phi}_T(\mathbf{y}_{1:T})$ to converge to a fixed quantity. The following Theorem proves such consistency of $\hat{\phi}_T(\mathbf{y}_{1:T})$ and $\tilde{\phi}_T(\mathbf{y}_{1:k+l-1})$ to $\phi^{\star}_T$.

Theorem 3 *Let the following assumptions hold almost surely for $\mathbf{Y}_{1:k+l-1} \sim P^{\star}_{1:k+l-1}$:*

1. *Φ is compact.*
2. *$\phi^{\star}_T$ and $\tilde{\phi}_T(\mathbf{Y}_{1:k+l-1})$ are unique; additionally, there exist a metric d on Φ such that, for all $\epsilon > 0$,*

$$\liminf_{T\to+\infty} \left\{ \min_{\phi: d(\phi,\phi^{\star}_T)\geq\epsilon} S^{\star}_T(P^{\phi}_{k+l:T}) - S^{\star}_T(P^{\phi^{\star}_T}_{k+l:T}) \right\} > 0 \qquad \textit{and}$$

$$\liminf_{T\to+\infty} \left\{ \min_{\phi: d(\phi,\tilde{\phi}_T(\mathbf{Y}_{1:k+l-1}))\geq\epsilon} \tilde{S}_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})) - \tilde{S}_T(P^{\tilde{\phi}_T(\mathbf{Y}_{1:k+l-1})}_{k+l:T}(\cdot|\mathbf{Y}_{1:k+l-1})) \right\} > 0.$$

3. *(Asymptotic stationarity) Let G_t be the marginal distribution of $\mathbf{Y}_{t-k+1:t+l}$ and $\tilde{G}_t$ be the marginal distribution of $\mathbf{Y}_{t-k+1:t+l}|\mathbf{Y}_{1:k+l-1}$ for $t \geq k$. Then, $(T-l-k+1)^{-1}\sum_{t=k}^{T-l} G_t$ and $(T-l-k+1)^{-1}\sum_{t=k}^{T-l} \tilde{G}_t$ both converge weakly to some probability measures on $\mathcal{Y}^{k+l}$ as $T \to \infty$.*

4. *Both conditions below are satisfied:*

   (a) *(Mixing)*[1] *Both $(\mathbf{Y}_t)_t \sim P^\star$ and $(\mathbf{X}_t)_t \sim P^\star(\cdot|\mathbf{Y}_{1:k+l-1})$ satisfy either one of these mixing properties (defined in Appendix A.3.5; $(\mathbf{X}_t)_t$ and $(\mathbf{Y}_t)_t$ can satisfy different ones):*

      i. *α-mixing with mixing coefficient of size $r/(2r-1)$, with $r \geq 1$, or*
      ii. *φ-mixing with mixing coefficient of size $r/(r-1)$ with $r > 1$.*

   (b) *(Moment boundedness) Define $H(\mathbf{y}_{t-k+1:t+l}) = \sup_{\phi\in\Phi} |S(P^\phi(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l})|$; then,*

$$\sup_{t\geq k} \mathbb{E}\left[H(\mathbf{Y}_{t-k+1:t+l})^{r+\delta}\right] \text{ and } \sup_{t\geq k} \mathbb{E}_{\mathbf{Y}_{t-k+1:t+l}|\mathbf{y}_{1:k+l-1}}\left[H(\mathbf{Y}_{t-k+1:t+l})^{r+\delta}\right]$$

   are finite for some $\delta > 0$, for the value of r corresponding to the condition above which is satisfied.

Then, $d(\phi_T^\star, \hat{\phi}_T(\mathbf{Y}_{1:T})) \to 0$ and $d(\tilde{\phi}_T(\mathbf{Y}_{1:k+l-1}), \hat{\phi}_T(\mathbf{Y}_{1:T})) \to 0$ when $T \to \infty$ almost surely with respect to $(\mathbf{Y}_t)_t \sim P^\star$. It also follows that $d(\tilde{\phi}_T(\mathbf{Y}_{1:k+l-1}), \phi_T^\star) \to 0$.

Under the assumptions of Theorem 3, with large enough T, $\hat{\phi}_T(\mathbf{y}_{1:T})$ and $\tilde{\phi}_T(\mathbf{y}_{1:k+l-1})$ will be independent of the observed sequence $\mathbf{y}_{1:T}$ and will converge to $\phi_T^\star$. Therefore, minimizing the prequential SR in Eq. (9) asymptotically recovers the minimizer of an expected proper SR, which does not depend on the initial conditions of the sequence $\mathbf{y}_{1:k+l-1}$.

Proof of Theorem 3 is given in Appendix A.3. The proof holds when P_{t+l}^ϕ depends on t only through the value of the past observations, which is our case of interest as we use the same generative network for all t's. The proof relies on the following steps: first, Assumptions 1, 2 and 4 are used to obtain a uniform law of large numbers using Theorem 2 in Pötscher and Prucha (1989) (Appendix A.3.7); then, this is combined with Assumption 2 to obtain the results thanks to Theorem 5.1 in Skouras (1998) (Appendix A.3.6). As such, Theorem 3 is a consequence of classical results in empirical process theory, adapted to our specific objective function in Eq. (10). To make the intermediary results more easily usable and the proof easier to follow, Appendix A.3 separately states and proves convergence of $\hat{\phi}_T$ to $\phi_T^\star$ (Appendix A.3.1) and that of $\hat{\phi}_T$ to $\tilde{\phi}_T$ (Appendix A.3.2), by splitting the assumptions in two sets.

The assumptions in Theorem 3 may be hard to verify. To make this easier, in Appendix A.3.4, we show that Assumption 2 is satisfied if S is strictly proper and P^ϕ is a well-specified model with ϕ being identifiable (Lemma 12). Moreover, we also provide simple sufficient conditions under which the moment boundedness condition in Assumption 4 holds for the Energy and Kernel score (Lemmas 14 and 13); the simplest of these conditions require the kernel or the space to be bounded for the Kernel score and the Energy score respectively.

1. Roughly speaking, both mixing properties imply that $\mathbf{Y}_{t-m}$ and $\mathbf{Y}_t$ become independent as $m \to \infty$.

3.2 Scoring rules for spatial data

In contrast to multivariate data, spatial data is structured: the relation between different entries depends on their spatial distance. Computing, say, the Kernel SR in Eq. (5) would discard this structure; we discuss here SRs which instead capture it, and which we will use for a spatio-temporal data set (Sec. 5.2).

3.2.1 Variogram Score

Say now $\mathcal{Y} \subseteq \mathbb{R}^d$. For any $p > 0$, the Variogram Score Scheuerer and Hamill (2015) is defined as

$$S_{\mathrm{v}}^{(p)}(P^\phi, \mathbf{y}) := \sum_{i,j=1}^{d} w_{ij} \left(|y_i - y_j|^p - \mathbb{E}_{\mathbf{X}\sim P^\phi} |X_i - X_j|^p \right)^2, \tag{11}$$

where $w_{ij} > 0$ are fixed scalars. If $\mathcal{Y}$ has a spatial structure w_{ij} can be set to be inversely proportional to the spatial distance of locations i and j (Scheuerer and Hamill, 2015). However, $S_{\mathrm{v}}^{(p)}$ is proper but not strictly so: it is invariant to change of sign and shift of all entries of $\mathbf{X}$ by a constant, and only depends on the moments of P^ϕ up to order $2p$ (Scheuerer and Hamill, 2015). We will fix $p = 1$ in the rest of our work.

3.2.2 Patched SR

To convey the spatial structure of the data, we can compute a SR on a localized *patch* of the data. In this way, the resulting score only considers the correlation between nearby components. We can then shift the patch across the map and cumulate the resulting score (see Fig. 4 in Appendix). However, this SR is non-strictly proper as it does not evaluate long-range dependencies. A similar approach was suggested for an adversarial setting in Isola et al. (2017), where the critic outputs separate numerical values for different patches of an input image.

3.2.3 Sum of SRs

Both SRs introduced above are non-strictly proper; we can however obtain a strictly proper SR by adding a strictly proper SR to a proper one, as stated by the lemma below (proof in Appendix A.1).

Lemma 4 *Consider two proper SRs S_1 and S_2, and let $\alpha_1, \alpha_2 > 0$; the quantity*

$$S_+(P, \mathbf{y}) = \alpha_1 \cdot S_1(P, \mathbf{y}) + \alpha_2 \cdot S_2(P, \mathbf{y})$$

is a proper SR. If at least one of S_1 and S_2 is also strictly proper, then S_+ is strictly proper.

3.2.4 Probabilistic forecasting for spatial data

Inserting the spatial SRs discussed above in the prequential score in Eq. (9) enables probabilistic forecasting for spatial data using generative networks. For the Variogram Score, unbiased gradient estimates can be computed by simulating from P^ϕ; same holds for the patched SR if the underlying SR admits unbiased gradient estimates (Appendix C).

4. Related works

Scoring rules have long been used in statistics: early characterisations are given in McCarthy (1956) and Savage (1971). Their usage for parameter estimation is also commonplace, see Gneiting and Raftery (2007) for an overview and Dawid et al. (2016) for theoretical properties. Closer to our method, Dawid and Musio (2013) used SRs to infer parameters for spatial models, considering the conditional distribution in each location given all the others to be available; instead, Dawid and Musio (2015) considered model selection based on SRs and studied a prequential application.

Prior works proved theoretical results related to our consistency result in Sec. 3.1.1: Theorem 3 combines Theorem 5.1 in Skouras (1998), which proves parameter consistency under uniform law of large numbers, with Theorem 2 in Pötscher and Prucha (1989), which is a classical result in empirical process theory obtaining a uniform law of large numbers for dependent data. Skouras (1998) also discusses other properties of prequential losses for forecasting systems, such as our Eq. (9). Analogous results to our Theorem 3 in similar settings were also shown: for instance, Dziugaite et al. (2015) showed consistency of the minimizer of an unbiased MMD empirical estimate to minimizer of the population MMD; they also rely on uniform convergence arguments, but, in contrast to our Theorem 3, their result applied to i.i.d. data.

As mentioned before, SR minimization for generative networks had been previously sparsely employed; however, a rigorous formulation such as the one we provide here was missing; moreover, no work specifically applied SR minimization to forecasting. Specifically, Bouchacourt et al. (2016) used a formulation corresponding to SR minimization with the Energy Score, but obtained it using different arguments. Similarly to the latter, Gritsenko et al. (2020) trained a generative network via a generalized Energy Distance for a speech synthesis task, again considering independent samples. More recent works use SR minimization for simulation-based Bayesian inference (Pacchiardi and Dutta, 2022), Neural SDEs (Issa et al., 2023) and self-supervised representation learning (Vahidi et al., 2024).

A research niche focuses on generating full time series with GANs (Brophy et al., 2023) often by using Recurrent NNs (RNN) as both discriminator and generator. In contrast, we focus on forecasting a single time step by conditioning on previous elements of the time-series. Some work aiming at generating full time-series can however be adapted for forecasting: for instance, the trained generator of Yoon et al. (2019) can be conditioned on past data; still, our training method is more convenient if forecasting is the task at hand, as we do not require a temporal discriminator nor multiple independent time-series as training data.

Some works instead directly used GANs for probabilistic forecasting, such as Kwon and Park (2019); Koochali et al. (2021); Bihlo (2021); Ravuri et al. (2021). However, they considered the training samples as independent and did not study theoretically the consequence of using dependent data. Bihlo (2021) tested their method on a similar data set to ours (which we privileged as it is a standardized benchmark) and found the GAN to underestimate uncertainty, so they considered a GANs ensemble to mitigate uncertainty underestimation. Instead, Ravuri et al. (2021) exploited GANs for a precipitation nowcasting task (i.e., predicting for small lead time), achieving good deterministic and probabilistic performance. Rasul et al. (2021) instead performed probabilistic forecasting with a normalizing flow (Papamakarios et al., 2021), by conditioning it on the output of a RNN or a Transformed network (Vaswani

et al., 2017) to which the past elements of the time series were input. While their method is adversarial-free, the use of a normalizing flow reduces its flexibility, possibly inhibiting its capacity to efficiently represent spatial data, which is instead straightforward with generative networks (Sec. 5.2).

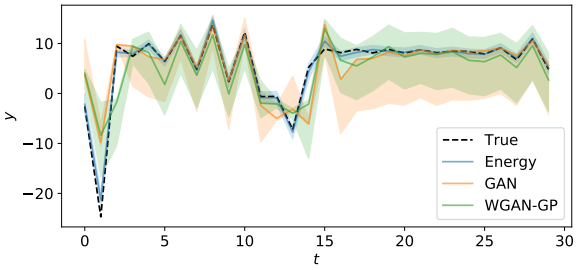
Deterministic forecasting with NNs for the WeatherBench data set (Sec. 5.2) was studied extensively Dueben and Bauer (2018); Scher (2018); Scher and Messori (2019); Weyn et al. (2019). Fewer studies tackled probabilistic forecasting: Scher and Messori (2021) combined deterministic NNs with ad-hoc strategies, not guaranteed to lead to the correct distribution. Clare et al. (2021) binned instead the data, thus mapping the problem to that of estimating a categorical distribution.

5. Simulation study

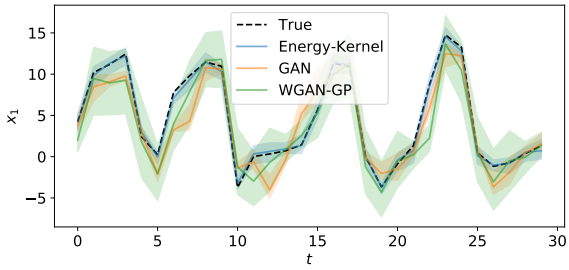
We first study two low-dimensional time-series models which allow exhaustive hyperparameter tuning and architecture comparison but still present challenging dynamics due to their chaotic nature. We then move to a high-dimensional spatio-temporal meteorology data set. For all examples, we train generative models with the Energy and the Kernel Scores (Appendix B.2) and their sum, termed *Energy-Kernel* Score (a strictly proper SR due to Lemma 4). As discussed in Sec. 2.2.2, we choose these scores as they can be written via an expectation, which makes our method applicable. Other scores (such as, for instance, the log score, Gneiting and Raftery, 2007), do not enjoy this property and are therefore unsuitable to our method. Additional SRs, discussed later in Sec. 5.2, are used for the meteorology example. For the Kernel Score, we use the Gaussian kernel (Appendix B.2) with bandwidth γ tuned from the validation set (Appendix E.1). For all SR methods, we use 10 forecasts from the generator for each observation window to estimate SR values during training; however, performance does not degrade when using as few as 3 simulations (Appendix F.3.2), which lowers the computational cost (Appendix F.3.3). We compare with the original GAN (Goodfellow et al., 2014) and WGAN with gradient penalties (WGAN-GP, Gulrajani et al., 2017). The latent variable $\mathbf{Z}$ has independent components with standard normal distribution. To have a reference for the deterministic performance of the probabilistic methods, we compare them with deterministic networks trained to minimize the standard regression loss.

All data sets consist of a long time series, which we split into training, validation and test set. We use the validation set for early stopping and hyperparameter tuning and report the final performance on the test set. The adversarial methods do not allow early stopping or hyperparameter selection using the training objective, as the generator loss depends on the critic state. For these methods, therefore, we use other metrics to pick the best hyperparameters (see below).

On the test set, we assess the calibration of the probabilistic forecasts by the *calibration error* (the discrepancy between credible intervals in the forecast distribution and the actual frequencies). We also evaluate how close the means of the forecast distributions are to the observation by the *Normalized Root Mean-Square Error* (NRMSE) and the *coefficient of determination* R^2; we detail all these metrics in Appendix D. As all these metrics are for scalar variables, we compute their values independently for each component and report their average (standard deviation in Appendix F).



(a) Lorenz63



(b) Lorenz96 (first data component).

Figure 2: Results for selected methods for Lorenz63 and Lorenz96 (first data component): median forecasts (solid line) and 99% credible area (shaded area) for a part of the test set. For each t, forecasts are obtained using the previous observation window. Credible regions for GAN and WGAN-GP are broader but contain the truth less frequently.

Our simulations show how the SR methods are easier to train and provide better uncertainty quantification. The adversarial methods require more hyperparameter tuning. We find the original GAN to be unstable and very poor at quantifying uncertainty due to mode collapse; WGAN-GP performs better but still has inferior performance than the SR approaches. Likely, ad-hoc adversarial training strategies could lead to better performance; however, the possibility of effortlessly training with off-the-shelf methods is an advantage of the SR approaches. Code for reproducing results is available here.

5.1 Time-series models

We consider the Lorenz63 (Lorenz, 1963) and Lorenz96 (Lorenz, 1996) chaotic models (Appendices E.2.1 and E.3.1). The former is defined on a 3-dimensional variable, a single component of which we assume to observe. The latter contains two sets of variables; we observe only one of them, which is 8-dimensional. In both cases, we generate an observed trajectory from a long model integration, from which we take the first 60% as training set, the following 20% as validation and the remaining 20% as test set.

We train the generative networks to forecast the next time step ($l = 1$) from an observation window of size $k = 10$. We use recurrent NNs based on Gated Recurrent Units (GRU, Cho et al., 2014; Appendices E.2.2 and E.3.2); we also tested fully connected networks but they had worse performance, so we do not report them here. For the SR methods, we select the best learning rate among 6 values according to the validation loss. For the adversarial methods, we consider instead 14 learning rates for both generator and critic; we also try two hidden dimensions for the GRU layers and four numbers of critic training steps for WGAN-GP; overall, we run 392 experiments for GAN and 1568 for WGAN-GP. As the validation loss is not a meaningful metric for adversarial approaches, we report results for 3 different configurations for GAN and WGAN-GP, maximizing either deterministic performance (1) or calibration (2), or striking the best balance between these two (3). More

	Lorenz63			Lorenz96		
	Cal. error ↓	NRMSE ↓	R^2 ↑	Cal. error ↓	NRMSE ↓	R^2 ↑
Regression	-	0.0079	0.9977	-	0.0198	0.9905
Energy	0.0380	0.0105	0.9960	0.0205	0.0166	0.9933
Kernel	0.0910	**0.0083**	**0.9975**	0.2196	**0.0164**	**0.9935**
Energy-Kernel	0.1000	0.0114	0.9953	**0.0104**	0.0173	0.9928
GAN (1)	0.4830	0.0274	0.9729	0.4644	0.0354	0.9696
GAN (2)	0.0860	0.2425	-1.1166	0.2671	0.1500	0.4537
GAN (3)	0.3590	0.0698	0.8245	0.3700	0.0763	0.8590
WGAN-GP (1)	0.4710	0.0398	0.9429	0.4134	0.0330	0.9736
WGAN-GP (2)	**0.0270**	0.1243	0.4440	0.0565	0.1081	0.7165
WGAN-GP (3)	0.2100	0.0914	0.6996	0.1648	0.0786	0.8502

Table 1: Performance on test set for the different methods, on the Lorenz63 and Lorenz96 models. Results with three hyperparameter configurations are reported for GAN and WGAN-GP, see text. Overall, SR methods perform well on both calibration and deterministic forecast metrics (NMRSE and R^2), while adversarial approaches are incapable of doing so.

details are in Appendix E.2.3 and E.3.3). These experiments are run on CPU machines and take at most a few minutes to complete.

In Table 1, we report performance metrics on the test set. The Kernel Score excels in deterministic forecasts, getting close to or outperforming the regression loss; however, all SR methods lead to combined great deterministic and probabilistic performance. On the other hand, adversarial methods are capable of good deterministic performance (1) or calibration (2) independently; but either of these two is at the expense of the other; the configuration with the best trade-off (3) is much worse than the SR methods (with WGAN-GP better than GAN). In Fig. 2, we show observation and forecast for a part of the test set, for GAN and WGAN-GP in configuration (3), the Energy Score for Lorenz63 and the Energy-Kernel Score for Lorenz96. For the two SR methods, the median forecast is close to the observation and the credible region contains the true observation for most time steps. For GAN and WGAN-GP, the match with the observation is worse and credible regions generally contain the truth less frequently albeit being wider. Additional results are given in Appendices F.1 and F.2.

5.2 Meteorological data set

The WeatherBench data set[2] for data-driven weather forecasting (Rasp et al., 2020) contains hourly values of several atmospheric fields from 1979 to 2018 at different resolutions; we choose here a resolution of 5.625° over both longitude and latitude, corresponding to a 32×64

2. Released under MIT license, see here.

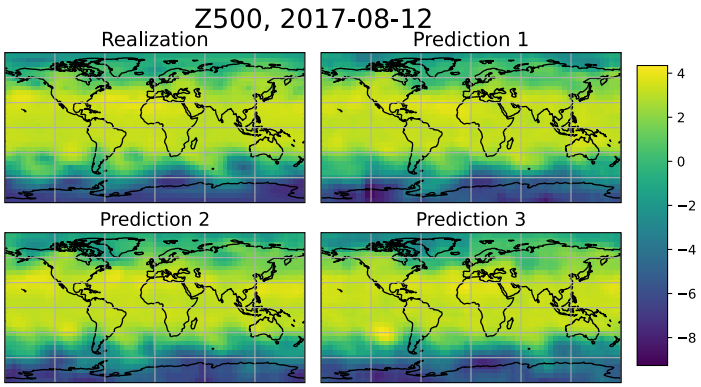


Figure 3: Realization and example of predictions obtained with the patched Energy Score (patch size 16) for a specific date in the test set for the WeatherBench data set. The predictions capture the main features but are slightly different from each other.

grid. We consider a single observation per day (12:00 UTC) and the 500 hPa geopotential (Z500) variable. We forecast with a lead of 3 days ($l = 3$) from a single observation ($k = 1$). We use the years from 1979 to 2006 as training set, 2007 to 2016 as validation test and 2017 to 2018 as test set.

In addition to the Energy, Kernel and Energy-Kernel Scores, we test the spatial SRs introduced in Sec 3.2. Specifically, we consider the Variogram Score with weights w inversely proportional to the distance on the globe (Appendix E.4.1) and sum it to the Energy (*Energy-Variogram*) or the Kernel (*Kernel-Variogram*) Scores. We also consider the Patched Energy Score with patch sizes 8 and 16; to ensure the score is strictly proper, we add the overall Energy Score (summation weights in Appendix E.4.2). We also consider patched regression loss.

We employ a U-NET architecture (Olaf et al., 2015) for the generative network and a PatchGAN discriminator (Isola et al., 2017) for the critic (Appendix E.4.3). For the SR methods, we select the best learning rate among 6 values according to the validation loss; for the adversarial ones, we consider instead 7 values for both generator and critic, resulting in 49 experiments. We then pick the setups optimizing deterministic or calibration performance. For WGAN-GP, a single configuration optimizes both; for GAN, that did not happen. As for the time-series models, we report therefore results for setups maximizing either deterministic performance (1) or calibration (2), or striking the best balance between these two (3). All training is run on a single Tesla V100 GPU; computing times are reported in Appendix F.3.3.

Table 2 reports performance on the test set. According to the calibration error, NRMSE and R^2, the Patched Energy Scores perform best, with deterministic skill only slightly worse than the regression loss. In Fig. 3 we show observation and three different predictions obtained with the Patched Energy Score for a date in the test set. More results in Appendix F.3.

	Cal. error ↓	NRMSE ↓	R^2 ↑
Regression	-	0.1162	0.5300
Patched Regression, 8	-	0.1147	0.5459
Patched Regression, 16	-	0.1144	0.5509
Energy	0.0863	0.1208	0.4968
Kernel	0.0797	0.1200	0.5097
Energy-Kernel	0.0794	0.1194	0.5150
Energy-Variogram	0.0899	0.1192	0.5177
Kernel-Variogram	0.1704	0.1203	0.5050
Patched Energy, 8	**0.0550**	0.1189	0.5217
Patched Energy, 16	0.0690	**0.1186**	**0.5248**
GAN (1)	0.4845	0.1573	0.1418
GAN (2)	0.3130	0.2487	-2.7970
GAN (3)	0.3625	0.1693	-0.0117
WGAN-GP	0.1009	0.1302	0.4340

Table 2: Performance on WeatherBench test set for different methods. Results with three hyperparameter configurations are reported for GAN, see text. SR methods perform well on both calibration and deterministic forecast metrics (NMRSE and R^2). WGAN-GP is worse and GAN is drastically worse.

6. Conclusions

We proposed a method to train generative networks for probabilistic forecasting by minimizing a prequential scoring rule. Compared to the standard adversarial framework, the advantages of the Scoring Rule formulation are: (i) it provides a principled objective for probabilistic forecasting; (ii) it yields adversarial-free training, with which better uncertainty quantification is possible, as we show empirically; (iii) it enables leveraging the literature on SRs to define objectives for spatio-temporal data sets. The resulting training method is easier to use and requires less hyperparameter tuning than adversarial methods.

We highlight the following limitations of our work: first, our Theorem 3 relies on assumptions which are hard to verify, although, for some assumptions, we provide sufficient conditions applicable to the Kernel and Energy Scores in Appendix A.3.4. However, we believe similar consistency properties hold provided the temporal process satisfies some generic stationarity and memory-less properties. Secondly, we do not experiment with forecasting multiple time-steps at once as we preferred focusing on single time-step forecast tasks for analytical simplicity while developing our framework. Doing so would be a useful extension of our work; in practice, SRs assessing temporal coherence analogous to what is done with temporal discriminators in Ravuri et al. (2021) in the adversarial setting could be developed. Finally, we presented adversarial training and SR minimization as alternative approaches, but it is plausible that combining them would be beneficial. We leave this for future work.

Acknowledgments

LP was supported by the EPSRC and MRC through the OxWaSP CDT programme (EP/L016710/1), which also funded the computational resources used to perform this work. RD was funded by EPSRC (grant nos. EP/V025899/1, EP/T017112/1) and NERC (grant no. NE/T00973X/1). PD gratefully acknowledges funding from the Royal Society for his University Research Fellowship, as well as from the ESiWACE Horizon 2020 project (#823988) and the MAELSTROM EuroHPC Joint Undertaking project (#955513). We thank Geoff Nicholls, Christian Robert, Peter Watson, Matthew Chantry, Mihai Alexe and Eugenio Clerico for valuable feedback and suggestions.

Appendix A. Proofs of theoretical results

A.1 Proof of Lemma 4

Proof By the definition of proper SR, we have that

$$\alpha_1 \cdot S_1(Q,Q) \leq \alpha_1 \cdot S_1(P,Q) \; \forall \; P,Q \in \mathcal{P},$$

and similar for S_2. By adding the two inequalities, we have therefore that

$$\alpha_1 \cdot S_1(Q,Q) + \alpha_2 \cdot S_2(Q,Q) \leq \alpha_1 \cdot S_1(P,Q) + \alpha_2 \cdot S_2(P,Q) \; \forall \; P,Q \in \mathcal{P},$$

which implies that S_+ is a proper SR.

Assume now additionally that S_1, without loss of generality, is strictly proper, i.e.

$$\alpha_1 \cdot S_1(Q,Q) < \alpha_1 \cdot S_1(P,Q) \; \forall \; P,Q \in \mathcal{P} : P \neq Q;$$

then, summing the above with the corresponding inequality for S_2 gives that

$$\alpha_1 \cdot S_1(Q,Q) + \alpha_2 \cdot S_2(Q,Q) < \alpha_1 \cdot S_1(P,Q) + \alpha_2 \cdot S_2(P,Q) \; \forall \; P,Q \in \mathcal{P} : P \neq Q,$$

which implies that S_+ is a strictly proper SR. ■

A.2 Propriety of the prequential SR

In this Section, let $P^\star$ denote the data generating distribution for $(\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_t, \ldots) = (\mathbf{Y}_t)_t$, and let P denote a generic distribution assigned to $(\mathbf{Y}_t)_t$. From the distribution on the full sequence P, conditional and marginals can be obtained, and denoted as follows: $P_{t+1}(\cdot|\mathbf{y}_{1:t})$ denotes the conditional distribution for $\mathbf{Y}_{t+1}$ given $\mathbf{y}_{1:t}$, and $P_{1:t}$ the (marginal) distribution for $(\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_t)$. Similar notation will be used for the conditional and marginals induced by $P^\star$.

A.2.1 Generic 1-step ahead prequential SR

We first consider a simplified case in which we can access the marginal for $\mathbf{Y}_1$ and all subsequent conditionals from P. Given $\mathbf{y}_{1:t}$, we use the distribution P to construct a forecast

distribution for $\mathbf{Y}_{t+1}$, namely $P_{t+1}(\cdot|\mathbf{y}_{1:t})$; we penalize the forecast, against the verifying observation $\mathbf{y}_{t+1}$, via a SR S

$$S(P_{t+1}(\cdot|\mathbf{y}_{1:t}), \mathbf{y}_{t+1}).$$

From the above, we construct the *prequential* SR for the forecast $P_{1:T}$ as follows

$$S_T(P_{1:T}, \mathbf{y}_{1:T}) = \frac{1}{T}\left[\sum_{t=1}^{T-1} S(P_{t+1}(\cdot|\mathbf{y}_{1:t}), \mathbf{y}_{t+1}) + S(P_1, \mathbf{y}_1)\right]; \tag{12}$$

the above assumes that at each time instant we obtain a probabilistic forecast $P_{t+1}(\cdot|\mathbf{y}_{1:t})$ from the distribution P and we verify it against the next observed element of the sequence $\mathbf{y}_{t+1}$. Additionally, at the first time step, we have not yet received any observation, so our forecast P_1 is unconditional. Also, let us define the expected prequential score as

$$S_T(P_{1:T}, P^\star_{1:T}) := \mathbb{E}_{\mathbf{Y}_{1:T}\sim P^\star_{1:T}} S_T(P_{1:T}, \mathbf{Y}_{1:T}),$$

Theorem 5 *If the scoring rule S is proper, then the prequential score S_T in Eq. (12) is proper for distributions over $\mathcal{Y}^T$, i.e.*

$$S_T(P^\star_{1:T}, P^\star_{1:T}) \leq S_T(P_{1:T}, P^\star_{1:T}).$$

Similarly, if S is strictly proper, the prequential score S_T is strictly proper, i.e. the equality only holds if $P_{1:T} = P^\star_{1:T}$.

Proof By definition of proper SR, we have that

$$\mathbb{E}_{\mathbf{Y}_{t+1}\sim P^\star_{t+1}(\cdot|\mathbf{y}_{1:t})} S(P^\star_{t+1}(\cdot|\mathbf{y}_{1:t}), \mathbf{Y}_{t+1}) \leq \mathbb{E}_{\mathbf{Y}_{t+1}\sim P^\star_{t+1}(\cdot|\mathbf{y}_{1:t})} S(P_{t+1}(\cdot|\mathbf{y}_{1:t}), \mathbf{Y}_{t+1})$$

for any conditional distribution $P_{t+1}(\cdot|\mathbf{y}_{1:t})$ and for any values $\mathbf{y}_{1:t}$.

Similarly, it holds

$$\mathbb{E}_{\mathbf{Y}_1\sim P^\star_1} S(P^\star_1, \mathbf{Y}_1) \leq \mathbb{E}_{\mathbf{Y}_1\sim P^\star_1} S(P_1, \mathbf{Y}_1), \tag{13}$$

for any distribution P_1.

For the expected prequential SR, it holds that:

$$\begin{aligned} S_T(P_{1:T}, P^\star_{1:T}) &= \mathbb{E}_{\mathbf{Y}_{1:T}\sim P^\star_{1:T}} S_T(P_{1:T}, \mathbf{Y}_{1:T}) \\ &= \frac{1}{T}\left[\sum_{t=1}^{T-1} \mathbb{E}_{\mathbf{Y}_{1:T}\sim P^\star_{1:T}} S(P_{t+1}(\cdot|\mathbf{Y}_{1:t}), \mathbf{Y}_{t+1}) + \mathbb{E}_{\mathbf{Y}_{1:T}\sim P^\star_{1:T}} S(P_1, \mathbf{Y}_1)\right] \\ &= \frac{1}{T}\left[\sum_{t=1}^{T-1} \mathbb{E}_{\mathbf{Y}_{1:t+1}\sim P^\star_{1:t+1}} S(P_{t+1}(\cdot|\mathbf{Y}_{1:t}), \mathbf{Y}_{t+1}) + \mathbb{E}_{\mathbf{Y}_1\sim P^\star_1} S(P_1, \mathbf{Y}_1)\right]; \end{aligned}$$

but now

$$\begin{aligned} \mathbb{E}_{\mathbf{Y}_{1:t+1}\sim P^\star_{1:t+1}} S(P_{t+1}(\cdot|\mathbf{Y}_{1:t}), \mathbf{Y}_{t+1}) &= \mathbb{E}_{\mathbf{Y}_{1:t}\sim P^\star_{1:t}}\left[\mathbb{E}_{\mathbf{Y}_{t+1}\sim P^\star_{t+1}(\cdot|\mathbf{Y}_{1:t})} S(P_{t+1}(\cdot|\mathbf{Y}_{1:t}), \mathbf{Y}_{t+1})\right] \\ &\geq \mathbb{E}_{\mathbf{Y}_{1:t}\sim P^\star_{1:t}}\left[\mathbb{E}_{\mathbf{Y}_{t+1}\sim P^\star_{t+1}(\cdot|\mathbf{Y}_{1:t})} S(P^\star_{t+1}(\cdot|\mathbf{Y}_{1:t}), \mathbf{Y}_{t+1})\right], \end{aligned} \tag{14}$$

so that

$$\begin{aligned} S_T(P_{1:T}, P^\star_{1:T}) &\geq \frac{1}{T}\left[\sum_{t=1}^{T-1} \mathbb{E}_{\mathbf{Y}_{1:t+1}\sim P^\star_{1:t+1}} S(P^\star_{t+1}(\cdot|\mathbf{Y}_{1:t}), \mathbf{Y}_{t+1}) + \mathbb{E}_{\mathbf{Y}_1\sim P^\star_1} S(P^\star_1, \mathbf{Y}_1)\right] \\ &= \frac{1}{T}\left[\sum_{t=1}^{T-1} \mathbb{E}_{\mathbf{Y}_{1:T}\sim P^\star_{1:T}} S(P^\star_{t+1}(\cdot|\mathbf{Y}_{1:t}), \mathbf{Y}_{t+1}) + \mathbb{E}_{\mathbf{Y}_{1:T}\sim P^\star_{1:T}} S(P^\star_1, \mathbf{Y}_1)\right] \\ &= S_T(P^\star_{1:T}, P^\star_{1:T}), \end{aligned} \tag{15}$$

which proves that S_T is proper.

To show that S_T is strictly proper if S is, we first notice that $P_{1:T}$ is fully determined by the marginal P_1 and by the conditionals $P_{t+1}(\cdot|\mathbf{y}_{1:t})$ for all possible values of $\mathbf{y}_{1:t}$, $1 \leq t \leq T-1$. In fact, if $P_{1:T}$ and its conditional marginals have densities, you can write

$$p_{1:T}(\mathbf{y}_{1:T}) = p_1(\mathbf{y}_1)p_2(\mathbf{y}_2|\mathbf{y}_1)p_3(\mathbf{y}_3|\mathbf{y}_{1:2})\dots p_{T-1}(\mathbf{y}_{T-1}|\mathbf{y}_{1:T-2})p_T(\mathbf{y}_T|\mathbf{y}_{1:T-1}).$$

Next, notice that the $\geq$ sign in Eq. (15) is an equality if and only if the $\leq$ sign in Eq. (13) is an equality and the $\geq$ sign in (14) is an equality for all $1 \leq t \leq T$. As S is proper, the latter being true requires

$$\mathbb{E}_{\mathbf{Y}_{t+1}\sim P^\star_{t+1}(\cdot|\mathbf{y}_{1:t})} S(P_{t+1}(\cdot|\mathbf{y}_{1:t}), \mathbf{Y}_{t+1}) = \mathbb{E}_{\mathbf{Y}_{t+1}\sim P^\star_{t+1}(\cdot|\mathbf{y}_{1:t})} S(P^\star_{t+1}(\cdot|\mathbf{y}_{1:t}), \mathbf{Y}_{t+1})$$

for all values of $\mathbf{y}_{1:t}$ in the support of $P^\star_{1:t}$. If S is strictly proper, however, the above conditions require that $P_1 = P^\star_1$ and $P_{t+1}(\cdot|\mathbf{y}_{1:t}) = P^\star_{t+1}(\cdot|\mathbf{y}_{1:t})\ \forall\ \mathbf{y}_{1:t}$ in the support of $P^\star_{1:t}$ and for $1 \leq t \leq T-1$, which implies that $P_{1:T} = P^\star_{1:T}$ due to distributions on $\mathbf{Y}_{1:T}$ being determined by the marginal for $\mathbf{Y}_1$ and the conditional on $\mathbf{Y}_{t+1}|\mathbf{y}_{1:t}$ for all values of $\mathbf{y}_{1:t}$ in the support of $P^\star_{1:t}$. ■

A.2.2 l-steps ahead prequential SR (Theorem 2)

We now go back to the specific setting considered in the main body of the paper. By discarding the model parameter ϕ in the notation for simplicity, the generative network induces conditional distributions $P_{t+l}(\cdot|\mathbf{y}_{1:t})$ for $\mathbf{Y}_{t+l}$ which only depend on the last k observations, i.e. $P_{t+l}(\cdot|\mathbf{y}_{1:t}) = P_{t+l}(\cdot|\mathbf{y}_{t-k+1:t})$. Therefore, the joint distribution for $\mathbf{Y}_{k+l:T}$ induced by the generative network satisfies the following property:

Definition 6 *A probability distribution $P_{1:T}$ is k-Markovian with lag l if it can be decomposed as follows, assuming it has density $p_{1:T}$ with respect to some base measure:*

$$p_{1:T}(\mathbf{y}_{1:T}) = p_{1:k+l-1}(\mathbf{y}_{1:k+l-1}) \prod_{t=k}^{T-l} p_{t+l}(\mathbf{y}_{t+l}|\mathbf{y}_{t-k+1:t}).$$

Setting $l = 1$ recovers the standard definition of k-Markovian models.

Notice also that the set of distributions which are k-Markovian with lag l is a subset of $(k+l-1)$-Markovian distributions, for which in fact

$$\begin{aligned} p_{1:T}(\mathbf{y}_{1:T}) &= p_{1:k+l-1}(\mathbf{y}_{1:k+l-1}) \prod_{t=k+l}^{T} p_t(\mathbf{y}_t|\mathbf{y}_{t-k-l+1:t-1}) \\ &= p_{1:k+l-1}(\mathbf{y}_{1:k+l-1}) \prod_{t=k}^{T-l} p_{t+l}(\mathbf{y}_{t+l}|\mathbf{y}_{t-k+1:t+l-1}); \end{aligned}$$

the additional assumption in Definition 6 with respect to $(k+l-1)$-Markovian is that the conditional distribution for $\mathbf{Y}_t$ is *not* influenced by the last $l-1$ elements.

In our setting, we can only access $P_{t+l}(\cdot|\mathbf{y}_{t-k+1:t})$ for $k \leq t \leq T-l$; the marginals $P_{1:k+l-1}$ are not available. Therefore, we consider the following quantity

$$S_T^{k,l}(P_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), \mathbf{y}_{k+l:T}) := \frac{1}{T-l-k+1} \sum_{t=k}^{T-l} S(P_{t+l}(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l}); \tag{16}$$

in contrast to Eq. (9) in the main text, we make explicit the dependence on k and l in the notation for $S_T^{k,l}$ and introduce a scaling constant for simplicity, which however does not impact the following arguments. The notation in Eq. (16) only makes sense if P is a $(k+l-1)$-Markovian distribution, as otherwise $\mathbf{y}_{1:k+l-1}$ would also appear explicitly in the conditioning of P_{t+l} on the right hand-side. The notation therefore makes sense for P obtained from the generative network, as that is k-Markovian with lag l which, as mentioned above, is a specific case of $(k+l-1)$-Markovian.

As mentioned in the main text, $S_T^{k,l}$ is the prequential score and is a SR for distributions over $\mathbf{Y}_{k+l:T}|\mathbf{y}_{1:k+l-1}$ which are $(k+l-1)$-Markovian.

From Eq. (16), we can define the expected SR as

$$\begin{aligned} &S_T^{k,l}(P_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})) := \\ &\qquad \mathbb{E}_{\mathbf{Y}_{k+l:T} \sim P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})} S_T^{k,l}(P_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), \mathbf{Y}_{k+l:T}). \end{aligned}$$

For the scoring rule defined in Eq. (16), the following Theorem holds, which we state in more generality with respect to Theorem 2 in the main text:

Theorem 7 *If the scoring rule S is proper, then, for all choices of $\mathbf{y}_{1:k+l-1}$, the prequential score $S_T^{k,l}$ in Eq. (16) is proper for distributions on $\mathbf{Y}_{k+l:T}|\mathbf{y}_{1:k+l-1}$ which are $(k+l-1)-$Markovian; namely, the following inequality holds*

$$S_T^{k,l}(P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})) \leq S_T^{k,l}(P_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})), \tag{17}$$

where $P_{1:T}$ and $P^\star_{1:T}$ are $(k+l-1)$-Markovian.

If additionally S is strictly proper, then, for all choices of $\mathbf{y}_{1:k+l-1}$, $S_T^{k,l}$ is proper for distributions on $\mathbf{Y}_{k+l:T}|\mathbf{y}_{1:k+l-1}$ which are k-Markovian with lag l, i.e. the equality in Eq. (17) only holds if $P_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}) = P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})$, where $P_{1:T}$ and $P^\star_{1:T}$ are k-Markovian with lag l.

The prequential score S_T^{k+l} is non-strictly proper for distributions that are $(k+l-1)$-Markovian but not k-Markovian with lag l. In fact, it builds forecasts from $P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})$ with lead of l timesteps, meaning that the information included in observations $\mathbf{y}_{t+1:t+l-1}$ is not used in formulating the forecast for $\mathbf{Y}_{t+l}$. It is therefore unable to distinguish between different distributions for $\mathbf{Y}_{k+l:T}|\mathbf{y}_{1:k+l-1}$ which have the same conditionals at lead l, but for which the conditionals change if one takes into account $\mathbf{y}_{t+1:t+l-1}$ in forecasting $\mathbf{Y}_{t+l}$. Therefore, you need to restrict the class of distributions to those in which the value $\mathbf{y}_{t+1:t+l-1}$ does not impact the distribution for $\mathbf{Y}_{t+l}$ in order to get strict propriety.

We now prove the Theorem.

Proof The proof steps follow those of Theorem 5.

By definition of proper SR, we have that, for all $t \geq k$

$$\mathbb{E}_{\mathbf{Y}_{t+l}\sim P^\star_{t+l}(\cdot|\mathbf{y}_{t-k+1:t})} S(P^\star_{t+l}(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{Y}_{t+l}) \leq \mathbb{E}_{\mathbf{Y}_{t+l}\sim P^\star_{t+l}(\cdot|\mathbf{y}_{t-k+1:t})} S(P_{t+l}(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{Y}_{t+l}) \tag{18}$$

for any conditional distribution $P_{t+l}(\cdot|\mathbf{y}_{t-k+1:t})$ and for any values $\mathbf{y}_{t-k+1:t}$.

For the expected prequential SR, it holds that

$$\begin{aligned}
&S_T^{k,l}(P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})) \\
&\qquad = \mathbb{E}_{\mathbf{Y}_{k+l:T}\sim P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})} S_T^{k,l}(P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), \mathbf{Y}_{k+l:T}) \\
&\qquad = \mathbb{E}_{\mathbf{Y}_{1:T}\sim P^\star_{1:T}(\cdot|\mathbf{y}_{1:k+l-1})} S_T^{k,l}(P^\star_{k+l:T}(\cdot|\mathbf{Y}_{1:k+l-1}), \mathbf{Y}_{k+l:T}) \\
&\qquad = \frac{1}{T-l-k+1}\sum_{t=k}^{T-l} \mathbb{E}_{\mathbf{Y}_{1:T}\sim P^\star_{1:T}(\cdot|\mathbf{y}_{1:k+l-1})} S(P^\star_{t+l}(\cdot|\mathbf{Y}_{t-k+1:t}), \mathbf{Y}_{t+l}) \\
&\qquad = \frac{1}{T-l-k+1}\sum_{t=k}^{T-l} \mathbb{E}_{\mathbf{Y}_{1:t+l}\sim P^\star_{1:t+l}(\cdot|\mathbf{y}_{1:k+l-1})} S(P^\star_{t+l}(\cdot|\mathbf{Y}_{t-k+1:t}), \mathbf{Y}_{t+l});
\end{aligned}$$

the second equality in the Equation above is trivial but we use it to simplify notation in the following. Now

$$\begin{aligned}
&\mathbb{E}_{\mathbf{Y}_{1:t+l}\sim P^\star_{1:t+l}(\cdot|\mathbf{y}_{1:k+l-1})} S(P^\star_{t+l}(\cdot|\mathbf{Y}_{t-k+1:t}), \mathbf{Y}_{t+l}) \\
&= \mathbb{E}_{\mathbf{Y}_{t-k+1:t}\sim P^\star_{t-k+1:t}(\cdot|\mathbf{y}_{1:k+l-1})} \left[\mathbb{E}_{\mathbf{Y}_{t+l}\sim P^\star_{t+l}(\cdot|\mathbf{Y}_{t-k+1:t},\mathbf{y}_{1:k+l-1})} S(P^\star_{t+l}(\cdot|\mathbf{Y}_{t-k+1:t}), \mathbf{Y}_{t+l})\right] \\
&= \mathbb{E}_{\mathbf{Y}_{t-k+1:t}\sim P^\star_{t-k+1:t}(\cdot|\mathbf{y}_{1:k+l-1})} \left[\mathbb{E}_{\mathbf{Y}_{t+l}\sim P^\star_{t+l}(\cdot|\mathbf{Y}_{t-k+1:t})} S(P^\star_{t+l}(\cdot|\mathbf{Y}_{t-k+1:t}), \mathbf{Y}_{t+l})\right] \\
&\leq \mathbb{E}_{\mathbf{Y}_{t-k+1:t}\sim P^\star_{t-k+1:t}(\cdot|\mathbf{y}_{1:k+l-1})} \left[\mathbb{E}_{\mathbf{Y}_{t+l}\sim P^\star_{t+l}(\cdot|\mathbf{Y}_{t-k+1:t})} S(P_{t+l}(\cdot|\mathbf{Y}_{t-k+1:t}), \mathbf{Y}_{t+l})\right] \\
&= \mathbb{E}_{\mathbf{Y}_{1:t+l}\sim P^\star_{1:t+l}(\cdot|\mathbf{y}_{1:k+l-1})} S(P_{t+l}(\cdot|\mathbf{Y}_{t-k+1:t}), \mathbf{Y}_{t+l});
\end{aligned} \tag{19}$$

in the first equality above, we have marginalized over all components of $\mathbf{Y}_{1:t+l}$ which do not appear in the expected quantity and we have used the definition of conditional probability together with the tower property of expectations. In the second equality, we have exploited the $(k+l-1)-$Markov property[3] of $P^\star$ which ensures that the distribution for $\mathbf{Y}_{t+l}$ does not depend on $\mathbf{Y}_{1:t-k}$. The inequality holds for any conditional distribution $P_{t+l}(\cdot|\mathbf{y}_{t-k+1:t})$

3. Technically, you can relax the $(k+l-1)-$Markov assumption for the full sequence to assuming $(k+l-1)-$Markovianity for $\mathbf{Y}_{1:2k+l-1}$ and independence of $\mathbf{Y}_{2k+l:T}$ on $\mathbf{Y}_{1:k+l-1}$; this is however quite artificial.

and for any values $\mathbf{y}_{t-k+1:t}$ thanks to Eq. (18). Finally, the last equality is obtained via the reverse of the argument used for the first one.

Now, we can write

$$\begin{aligned}
&S_T^{k,l}(P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})) \\
&\qquad \leq \frac{1}{T-l-k+1}\sum_{t=k}^{T-l} \mathbb{E}_{\mathbf{Y}_{1:t+l}\sim P^\star_{1:t+l}(\cdot|\mathbf{y}_{1:k+l-1})} S(P_{t+l}(\cdot|\mathbf{Y}_{t-k+1:t}), \mathbf{Y}_{t+l}) \\
&\qquad = \frac{1}{T-l-k+1}\sum_{t=k}^{T-l} \mathbb{E}_{\mathbf{Y}_{1:T}\sim P^\star_{1:T}(\cdot|\mathbf{y}_{1:k+l-1})} S(P_{t+l}(\cdot|\mathbf{Y}_{t-k+1:t}), \mathbf{Y}_{t+l}) \qquad (20) \\
&\qquad = \frac{1}{T-l-k+1}\sum_{t=k}^{T-l} \mathbb{E}_{\mathbf{Y}_{k+l:T}\sim P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})} S(P_{t+l}(\cdot|\mathbf{Y}_{t-k+1:t}), \mathbf{Y}_{t+l}) \\
&\qquad = S_T^{k,l}(P_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})),
\end{aligned}$$

which proves that $S_T^{k,l}$ is proper for distributions over $\mathbf{Y}_{k+l:T}|\mathbf{y}_{1:k+l-1}$ which are $(k+l)-$Markov.

Now, consider $P_{1:T}$ and $P^\star_{1:T}$ to be k-Markovian with lag l. The $\leq$ sign in Eq. (20) is an equality if and only if the $\leq$ sign in Eq. (19) is an equality for all $k \leq t \leq T-l$. As S is proper, the latter requires

$$\mathbb{E}_{\mathbf{Y}_{t+l}\sim P^\star_{t+l}(\cdot|\mathbf{y}_{t-k+1:t})} S(P^\star_{t+l}(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{Y}_{t+l}) = \mathbb{E}_{\mathbf{Y}_{t+l}\sim P^\star_{t+l}(\cdot|\mathbf{y}_{t-k+1:t})} S(P_{t+l}(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{Y}_{t+l})$$

for all values of $\mathbf{y}_{t-k+1:t}$, If S is strictly proper, however, the latter is satisfied if and only if $P_{t+l}(\cdot|\mathbf{y}_{t-k+1:t}) = P^\star_{t+l}(\cdot|\mathbf{y}_{t-k+1:t})\ \forall\ \mathbf{y}_{t-k+1:t}$ and for $k \leq t \leq T-l$, which implies that $P_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}) = P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})$ due to the k-Markov with lag l property. This implies that S_T is strictly proper for distributions which are k-Markov with lag l. ■

A.3 Proof and precise statement of the consistency result (Theorem 3)

We follow here the notation introduced at the start of Appendix A.2. Specifically, $P^\star$ denotes the data generating distribution for $(\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_t, \ldots) = (\mathbf{Y}_t)_t$.

We consider a model class parametrized by a set of parameters ϕ. For such models, we assume the conditional distributions $P^\phi_{t+l}(\cdot|\mathbf{y}_{1:t})$ for $\mathbf{Y}_{t+l}$ only depends on the last k observations, i.e. $P^\phi_{t+l}(\cdot|\mathbf{y}_{1:t}) = P^\phi_{t+l}(\cdot|\mathbf{y}_{t-k+1:t})$. Additionally, we assume that the conditional distribution does not depend explicitly on t, such that $P^\phi_{t+l}(\cdot|\mathbf{y}_{t-k+1:t}) = P^\phi_{(l)}(\cdot|\mathbf{y}_{t-k+1:t})$, where the bracketed subscript denotes that the forecast is for l steps ahead. This is the setting considered in the main manuscript.

In this specific case, therefore, the scoring rule used to penalize the forecast $P^\phi_{(l)}(\cdot|\mathbf{y}_{t-k+1:t})$ against the verification $\mathbf{y}_{t+l}$ (Eq. 16) becomes

$$S(P^\phi_{(l)}(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l}).$$

Therefore, the prequential score defined in Eq. (16) becomes

$$S_T^{k,l}(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), \mathbf{y}_{k+l:T}) = \frac{1}{T-l-k+1}\sum_{t=k}^{T-l} S(P^{\phi}_{(l)}(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l}); \tag{21}$$

notice that we introduce here a scaling constant for simplicity; that however does not impact any of the following arguments. Recall also the definition of the expected prequential score

$$\begin{aligned} &S_T^{k,l}(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), P^{\star}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})) \\ &\qquad := \mathbb{E}_{\mathbf{Y}_{k+l:T}\sim P^{\star}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})} S_T^{k,l}(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), \mathbf{Y}_{k+l:T}), \end{aligned} \tag{22}$$

for which we will use the following notation for brevity

$$\tilde{S}_T^{k,l}(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})) := S_T^{k,l}(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), P^{\star}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}))$$

As discussed in Appendix A.2.2 and shown in Theorem 7, provided that S is strictly proper, $S_T^{k,l}$ is a strictly proper SR for k-Markovian with lag l distributions over $\mathbf{Y}_{k+l:T}|\mathbf{y}_{1:k+l-1}$, for all values of $\mathbf{y}_{1:k+l-1}$.

We will also consider the minimizer of the expectation of the expected prequential SR in Eq. (22) with respect to the initial data $\mathbf{y}_{1:k+l-1}$, i.e.

$$\begin{aligned} S_T^{k,l\star}(P^{\phi}_{k+l:T}) :&= \mathbb{E}_{\mathbf{Y}_{1:k+l-1}\sim P^{\star}_{1:k+l-1}} S_T^{k,l}(P^{\phi}_{k+l:T}(\cdot|\mathbf{Y}_{1:k+l-1}), P^{\star}_{k+l:T}(\cdot|\mathbf{Y}_{1:k+l-1})) \\ &= \mathbb{E}_{\mathbf{Y}_{1:T}\sim P^{\star}_{1:T}} S_T^{k,l}(P^{\phi}_{k+l:T}(\cdot|\mathbf{Y}_{1:k+l-1}), \mathbf{Y}_{k+l:T}). \end{aligned} \tag{23}$$

Theorem 3 in the main text states that the value of ϕ minimizing the empirical prequential SR (Eq. (21)) converges to both the minimizer of the expected (with respect to $\mathbf{Y}_{k+l:T}|\mathbf{y}_{1:k+l-1}$ for fixed $\mathbf{y}_{1:k+l-1}$) SR in Eq. (22) and to the minimizer of the expected (with respect to $\mathbf{Y}_{k+l:T}$) SR in Eq. (23). We will split the original result in two separate statements, which hold under similar Assumptions.

We now set notation and introduce the relevant quantities. From now onwards, we will drop k and l for brevity in the definition of S_T; all following results hold for each fixed value of k and l. We write therefore $S_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), \mathbf{y}_{k+l:T}) = S_T^{k,l}(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), \mathbf{y}_{k+l:T})$, $\tilde{S}_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})) = \tilde{S}_T^{k,l}(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}))$ and $S_T^{\star}(P^{\phi}_{k+l:T}) = S_T^{k,l\star}(P^{\phi}_{k+l:T})$. Next, we define the minimizers of the empirical and expected prequential scores

$$\begin{aligned} \hat{\phi}_T(\mathbf{y}_{1:T}) &: S_T(P^{\hat{\phi}_T(\mathbf{y}_{1:T})}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), \mathbf{y}_{k+l:T}) = \min_{\phi\in\Phi} S_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), \mathbf{y}_{k+l:T}) \\ \tilde{\phi}_T(\mathbf{y}_{1:k+l-1}) &: \tilde{S}_T(P^{\tilde{\phi}_T(\mathbf{y}_{1:k+l-1})}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})) = \min_{\phi\in\Phi} \tilde{S}_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})). \\ \phi_T^{\star} &: S_T^{\star}(P^{\phi_T^{\star}}_{k+l:T}) = \min_{\phi\in\Phi} S_T^{\star}(P^{\phi}_{k+l:T}). \end{aligned}$$

A.3.1 Convergence of $\hat{\phi}_T$ to $\phi_T^{\star}$

We first introduce Assumptions and give the statement linking $\hat{\phi}_T(\mathbf{y}_{1:T})$ to $\phi_T^{\star}$ (Theorem 8). We require the sequence $(\mathbf{Y}_t)_t$ to be stationary and to satisfy some mixing properties. Specifically, the following Assumptions are required. The precise definition of the mixing properties is postponed to later in Appendix A.3.5.

A1 Φ is compact.

A2 $\phi_T^\star$ is unique; additionally, there exist a metric d on Φ such that, for all $\epsilon > 0$

$$\liminf_{T\to+\infty} \left\{ \min_{\phi: d(\phi,\phi_T^\star)\geq\epsilon} S_T^\star(P^\phi_{k+l:T}) - S_T^\star(P^{\phi_T^\star}_{k+l:T}) \right\} > 0$$

A3 (Asymptotic stationarity) Let G_t be the marginal distribution of $\mathbf{Y}_{t-k+1:t+l}$ for $t \geq k$; then, $(T-l-k+1)^{-1}\sum_{t=k}^{T-l} G_t$ converges weakly to some probability measure on $\mathcal{Y}^{k+l}$ as $T \to \infty$.

A4 Both conditions below are satisfied:

(a) (Mixing) Either one of the following holds:

i. $(\mathbf{Y}_t)_t$ is α-mixing with mixing coefficient of size $r/(2r-1)$, with $r \geq 1$, or

ii. $(\mathbf{Y}_t)_t$ is φ-mixing with mixing coefficient of size $r/(r-1)$ with $r > 1$.

(b) (Moment boundedness) Define $H(\mathbf{y}_{t-k+1:t+l}) = \sup_{\phi\in\Phi} |S(P^\phi(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l})|$; then,

$$\sup_{t\geq k} \mathbb{E}\left[H(\mathbf{Y}_{t-k+1:t+l})^{r+\delta} \right] < \infty$$

for some $\delta > 0$, for the value of r corresponding to the condition above which is satisfied.

S being strictly proper and $P^\phi_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})$ being a well specified model for $\mathbf{Y}_{k+l:T}|\mathbf{y}_{1:k+l-1}$ is a sufficient (but not necessary) condition for the uniqueness of $\phi_T^\star$ in Assumption **A2** (see Lemma 12 in Appendix A.3.4), provided that the parameters ϕ are identifiable. Notice that neural networks do not have identifiable parameters; we require however this assumption to prove the Theorem. In case the parameters are not identifiable, we believe it is possible to show asymptotic convergence of the distributions minimizing the empirical and expected prequential SR, instead of convergence of the parameters. Extending the proof to this setting is technically challenging, as the distance in Assumption **A1** needs to be replaced by a divergence between probability distributions. We leave this extension for future work.

The rest of Assumption **A2** is a standard condition ensuring that the function which we are minimizing does not get flatter and flatter around the optimal value as $T \to \infty$. The asymptotic stationarity condition in Assumption **A3** is implied by the stronger condition of the marginals G_t being the same for each t. Assumption **A4**(a) is a mixing condition, ensuring that the dependence between two different $\mathbf{Y}_t, \mathbf{Y}'_t$ decreases as $t - t' \to \infty$ (defined precisely in Appendix A.3.5). Finally, Assumption **A4**(b) is a boundedness condition; for the specific case of the Kernel and Energy SR, that can be verified by simpler conditions as discussed in Lemmas 13 and 14 in Appendix A.3.4.

We will now state our first result.

Theorem 8 *If $(\mathbf{y}_{t-k+1:t+l}, \phi) \to S(P^\phi(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l})$ is continuous on $\mathcal{Y}^{k+l} \times \Phi$, and if Assumptions **A1**, **A2**, **A3** and **A4** hold, then $d(\hat{\phi}_T(\mathbf{Y}_{1:T}), \phi_T^\star) \to 0$ with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^\star$.*

The Theorem above relies on a generic consistency result (discussed in Appendix A.3.6) for which a uniform law of large numbers is required. Such a uniform law of large numbers can be obtained under stationarity and mixing conditions; we report in Appendix A.3.7 a result ensuring this. We prove Theorem 8 by combining the above two elements in Appendix A.3.8.

A.3.2 Convergence of $\hat{\phi}_T$ to $\tilde{\phi}_T$

We now give the statement linking $\hat{\phi}_T(\mathbf{y}_{1:T})$ to $\tilde{\phi}_T(\mathbf{y}_{1:t})$ (Theorem 10). We will require similar Assumptions to what considered above, but holding for fixed values of $\mathbf{y}_{1:k+l-1}$:

B1 $\tilde{\phi}_T(\mathbf{y}_{1:k+l-1})$ is unique; additionally, there exist a metric d on Φ such that, for all $\epsilon > 0$

$$\liminf_{T\to+\infty} \left\{ \min_{\phi: d(\phi, \tilde{\phi}_T(\mathbf{y}_{1:k+l-1}))\geq\epsilon} \tilde{S}_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})) - \tilde{S}_T(P^{\tilde{\phi}_T(\mathbf{y}_{1:k+l-1})}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})) \right\} > 0$$

B2 (Asymptotic stationarity) Let $\tilde{G}_t$ be the marginal distribution of $\mathbf{Y}_{t-k+1:t+l}|\mathbf{y}_{1:k+l-1}$ for $t \geq k$; then,

$$(T-l-k+1)^{-1} \sum_{t=k}^{T-l} \tilde{G}_t$$

converges weakly to some probability measure on $\mathcal{Y}^{k+l}$ as $T \to \infty$.

B3 Both conditions below are satisfied:

(a) (Mixing) Let $(\mathbf{X}_t)_t \sim P^{\star}(\cdot|\mathbf{y}_{1:k+l-1})$; then, either one of the following holds:

i. $(\mathbf{X}_t)_t$ is α-mixing with mixing coefficient of size $r/(2r-1)$, with $r \geq 1$, or

ii. $(\mathbf{X}_t)_t$ is φ-mixing with mixing coefficient of size $r/(r-1)$ with $r > 1$.

(b) (Moment boundedness) Define $H(\mathbf{y}_{t-k+1:t+l}) = \sup_{\phi\in\Phi} |S(P^{\phi}(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l})|$; then,

$$\sup_{t\geq k} \mathbb{E}_{\mathbf{Y}_{t-k+1:t+l}|\mathbf{y}_{1:k+l-1}} \left[H(\mathbf{Y}_{t-k+1:t+l})^{r+\delta} \right] < \infty$$

for some $\delta > 0$, for the value of r corresponding to the condition above which is satisfied.

We can therefore state the following:

Theorem 9 *If $(\mathbf{y}_{t-k+1:t+l}, \phi) \to S(P^{\phi}(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l})$ is continuous on $\mathcal{Y}^{k+l} \times \Phi$, and if Assumptions **A1**, **B1**, **B2** and **B3** hold, then $d(\hat{\phi}_T(\mathbf{y}_{1:k+l-1}, \mathbf{Y}_{k+l:T}), \tilde{\phi}_T(\mathbf{y}_{1:k+l-1})) \to 0$ with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^{\star}(\cdot|\mathbf{y}_{1:k+l-1})$.*

Notice how now in $\hat{\phi}_T$ we split the dependence with respect to the fixed $\mathbf{y}_{1:k+l-1}$ and the random $\mathbf{Y}_{k+l:T}$.

Proof Theorem 9 is proven following the same steps as Theorem 8 (given in Appendix A.3.8). Specifically, Corollary 21 can be used to obtain a uniform Law of Large Numbers such as in Assumption **A5**. Then, an equivalent to Theorem 18 can be shown following the exact same steps. That implies the result of Theorem 9. ■

The above result is saying that, for the sequence $(\mathbf{Y}_t)_t$ conditioned on $\mathbf{y}_{1:k+l-1}$, if stationarity and mixing conditions hold for a fixed $\mathbf{y}_{1:k+l-1}$, then the empirical minimizer $\hat{\phi}_T$ converges to the minimizer $\tilde{\phi}$, both with fixed $\mathbf{y}_{1:k+l-1}$.

Clearly, if the above Assumptions hold for all values of $\mathbf{y}_{1:k+l-1}$, the statement also does. This is made precise by the following Corollary:

Corollary 10 *If Assumptions **A1**, **B1**, **B2** and **B3** hold almost surely for $\mathbf{Y}_{1:k+l-1} \sim P^\star_{1:k+l-1}$, and if $(\mathbf{y}_{t-k+1:t+l}, \phi) \to S(P^\phi(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l})$ is continuous on $\mathcal{Y}^{k+l} \times \Phi$, then*

$$d(\hat{\phi}_T(\mathbf{Y}_{1:k+l-1}, \mathbf{Y}_{k+l:T}), \tilde{\phi}_T(\mathbf{Y}_{1:k+l-1})) \to 0$$

with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^\star$.

Proof If Assumptions **A1**, **B1**, **B2** and **B3** hold almost surely for $\mathbf{Y}_{1:k+l-1} \sim P^\star_{1:k+l-1}$, and under the continuity condition, the following statement holds with probability 1 with respect to $\mathbf{Y}_{1:k+l-1} \sim P^\star_{1:k+l-1}$: "$d(\hat{\phi}_T(\mathbf{Y}_{1:k+l-1}, \mathbf{Y}_{k+l:T}), \tilde{\phi}_T(\mathbf{Y}_{1:k+l-1})) \to 0$ with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^\star(\cdot|\mathbf{Y}_{1:k+l-1})$," from which the result follows by considering that a statement holding with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^\star(\cdot|\mathbf{Y}_{1:k+l-1})$, for each value $\mathbf{Y}_{1:k+l-1}$ takes, and with probability 1 with respect to $\mathbf{Y}_{1:k+l-1} \sim P^\star_{1:k+l-1}$ holds almost surely with respect to $(\mathbf{Y}_t)_t \sim P^\star$. ■

A.3.3 Putting the two results together

Finally, we also have the following, which correspond to Theorem 3 in the main text with the two sets of assumptions for the conditional and unconditional case kept separate:

Corollary 11 *If Assumptions **A1**, **A2**, **A3** and **A4** hold, and if Assumptions **B1**, **B2** and **B3** hold almost surely for $\mathbf{Y}_{1:k+l-1} \sim P^\star_{1:k+l-1}$, and if $(\mathbf{y}_{t-k+1:t+l}, \phi) \to S(P^\phi(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l})$ is continuous on $\mathcal{Y}^{k+l} \times \Phi$, then*

1. *$d(\hat{\phi}_T(\mathbf{Y}_{1:T}), \phi^\star_T) \to 0$ with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^\star$;*
2. *$d(\hat{\phi}_T(\mathbf{Y}_{1:T}), \tilde{\phi}_T(\mathbf{Y}_{1:k+l-1})) \to 0$ with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^\star$;*
3. *$d(\phi^\star_T, \tilde{\phi}_T(\mathbf{Y}_{1:k+l-1})) \to 0$ with probability 1 with respect to $\mathbf{Y}_{1:k+l-1} \sim P^\star_{1:k+l-1}$.*

Proof Under the Assumptions, both Theorem 8 and Corollary 10 hold, from which the first two statements follow. For the last statement, applying the triangle inequality yields

$$d(\phi^\star_T, \tilde{\phi}_T(\mathbf{Y}_{1:k+l-1})) \leq d(\hat{\phi}_T(\mathbf{Y}_{1:T}), \tilde{\phi}_T(\mathbf{Y}_{1:k+l-1})) + d(\hat{\phi}_T(\mathbf{Y}_{1:T}), \phi^\star_T) \to 0.$$

As the left-hand side above depends only on $\mathbf{Y}_{1:k+l-1}$, the result holds almost surely with respect to $\mathbf{Y}_{1:k+l-1} \sim P^\star_{1:k+l-1}$. ■

In case in which all the Assumption hold, therefore, the minimizer of the expected prequential SR over $\mathbf{Y}_{k+l:T}|\mathbf{Y}_{1:k+l-1}$ converges to the minimizer of the expected prequential SR over $\mathbf{Y}_{1:T}$, which is a deterministic quantity. Therefore, this result is saying that for

large T, $\tilde{\phi}_T$ does not depend on the initial conditions, as it is intuitive under mixing and stationarity of $(\mathbf{Y}_t)_t$. Indeed, the same holds for the empirical minimizer $\hat{\phi}_T$, in which no expectation at all is computed.

In the next Subsections, we will discuss how to verify the Assumptions in some specific cases, and then move to introducing preliminary results for proving Theorem 8, which we do in Appendix A.3.8. As mentioned above, the proof of Theorem 9 follows the same steps as the one for Theorem 8, but with the corresponding set of Assumptions. For this reason, we do not give that in details.

A.3.4 Verifying the Assumptions in specific cases

Before delving into proving Theorem 8, we here show sufficient conditions under which $\phi_T^\star$ and $\tilde{\phi}_T(\mathbf{y}_{1:k+l-1})$ are unique and under which Assumption **A4**(b) holds. Specifically, for the former (Lemma 12), we consider the model $P^\phi_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})$ to be a well specified model and the scoring rule S to be strictly proper; for the latter, we consider instead the Kernel and Energy SR and obtain more precise conditions, which are easily satisfied.

First, consider uniqueness of $\phi_T^\star$:

Lemma 12 *If both*

- *S is strictly proper, and*
- *for all values of T, $P^\phi_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})$ is a well specified model for $\mathbf{Y}_{k+l:T}|\mathbf{y}_{1:k+l-1}$ and the mapping $\phi \to P^\phi_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})$ is unique,*

then $\phi_T^\star$ and $\tilde{\phi}_T(\mathbf{y}_{1:k+l-1})$ are unique for all values of T and $\mathbf{y}_{1:k+l-1}$.

Proof If P^ϕ is well specified, there exists a $\phi^\star$ such that

$$P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}) = P^{\phi^\star}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}) \ \forall\ T,\ \forall\ \mathbf{y}_{1:k+l-1}.$$

Notice that this implies that $P^\star$ is k-Markovian with lag l. If S is strictly proper, we have by Theorem 7 that

$$\phi^\star = \arg\min_{\phi\in\Phi} S_T(P^\phi_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), P^\star_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1})$$

is unique, for all $\mathbf{y}_{1:k+l-1}$. Therefore, $\tilde{\phi}_T(\mathbf{y}_{1:k+l-1}) = \phi^\star$ for all values of $\mathbf{y}_{1:k+l-1}$. Recalling now the definition of $S^\star_T(P^\phi_{k+l:T})$ in Eq. (23), notice that the quantity inside the expectation $\mathbb{E}_{\mathbf{Y}_{1:k+l-1}\sim P^\star_{1:k+l-1}}$ is minimized uniquely by $\phi = \phi^\star$, so that $S^\star_T(P^\phi_{k+l:T})$ is also uniquely minimized by $\phi_T^\star = \phi^\star$. ■

The following two Lemmas show conditions under which Assumption **A4**(b) holds.

Lemma 13 *When $S = S_k$, Assumption **A4**(b) is verified for a kernel k which satisfies either of the following:*

1. *with probability 1 with respect to* $(\mathbf{Y}_t)_t \sim P^\star$,[4] *for all* $t \geq k$ *and* ϕ, $\mathbb{E}_{\mathbf{X},\mathbf{X}'\sim P^\phi_{(l)}(\cdot|\mathbf{Y}_{t-k+1:t})}|k(\mathbf{X},\mathbf{X}')| < \infty$ *and* $\mathbb{E}_{\mathbf{X}\sim P^\phi_{(l)}(\cdot|\mathbf{Y}_{t-k+1:t})}|k(\mathbf{X},\mathbf{Y}_{t+l})| < \infty$*;*

2. k *is bounded, i.e.* $|k(\mathbf{y},\mathbf{x})| < \kappa < +\infty \ \forall\ \mathbf{y},\mathbf{x} \in \mathcal{Y}$ *(this implies the above condition).*

Proof First, notice that $\sup_{t\geq k}\mathbb{E}\left[H(\mathbf{Y}_{t-k+1:t+l})^{r+\delta}\right] < \infty \iff \mathbb{E}\left[H(\mathbf{Y}_{t-k+1:t+l})^{r+\delta}\right] < \infty\ \forall\ t \geq k$.

Consider the kernel SR $S = S_k$

$$\begin{aligned}|S_k(P^\phi_{(l)}(\cdot|\mathbf{y}_{t-k+1:t}),\mathbf{y}_{t+l})| &= |\mathbb{E}_{\mathbf{X},\mathbf{X}'\sim P^\phi_{(l)}(\cdot|\mathbf{y}_{t-k+1:t})}[k(\mathbf{X},\mathbf{X}') - 2k(\mathbf{X},\mathbf{y}_{t+l})]| \\ &\leq \mathbb{E}_{\mathbf{X},\mathbf{X}'\sim P^\phi_{(l)}(\cdot|\mathbf{y}_{t-k+1:t})}|k(\mathbf{X},\mathbf{X}') - 2k(\mathbf{X},\mathbf{y}_{t+l})| \\ &\leq \mathbb{E}_{\mathbf{X},\mathbf{X}'\sim P^\phi_{(l)}(\cdot|\mathbf{y}_{t-k+1:t})}[|k(\mathbf{X},\mathbf{X}')| + 2|k(\mathbf{X},\mathbf{y}_{t+l})|].\end{aligned} \tag{24}$$

We first show why condition 1 yields the result. If, with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^\star$, for all $t \geq k$ and ϕ

$$\mathbb{E}_{\mathbf{X},\mathbf{X}'\sim P^\phi_{(l)}(\cdot|\mathbf{Y}_{t-k+1:t})}|k(\mathbf{X},\mathbf{X}')| \leq \kappa_1 < \infty \text{ and } \mathbb{E}_{\mathbf{X}\sim P^\phi_{(l)}(\cdot|\mathbf{Y}_{t-k+1:t})}|k(\mathbf{X},\mathbf{Y}_{t+l})| \leq \kappa_2 < \infty,$$

we have that

$$|S_k(P^\phi_{(l)}(\cdot|\mathbf{Y}_{t-k+1:t}),\mathbf{Y}_{t+l})| \leq \kappa_1 + 2\kappa_2 < \infty,$$

from which

$$\begin{aligned}\mathbb{E}\left[H(\mathbf{Y}_{t-k+1:t+l})^{r+\delta}\right] &= \mathbb{E}\left[\left(\sup_{\phi\in\Phi}|S_k(P^\phi_{(l)}(\cdot|\mathbf{Y}_{t-k+1:t}),\mathbf{Y}_{t+l})|\right)^{r+\delta}\right] \\ &\leq \mathbb{E}\left[\left(\sup_{\phi\in\Phi}\kappa_1 + 2\kappa_2\right)^{r+\delta}\right] = (\kappa_1+2\kappa_2)^{r+\delta} < \infty.\end{aligned}$$

Now, condition 2 implies condition 1. Therefore, condition 2 yields the result. ■

Lemma 14 *When* $S = S_E^{(\beta)}$*, Assumption* ***A4****(b) is verified when either of the following holds:*

1. *with probability 1 with respect to* $(\mathbf{Y}_t)_t \sim P^\star$*, for all* $t \geq k$ *and* ϕ*,* $\mathbb{E}_{\mathbf{X},\mathbf{X}'\sim P^\phi_{(l)}(\cdot|\mathbf{Y}_{t-k+1:t})}||\mathbf{X}-\mathbf{X}'|| < \infty$ *and* $\mathbb{E}_{\mathbf{X}\sim P^\phi_{(l)}(\cdot|\mathbf{Y}_{t-k+1:t})}||\mathbf{X}-\mathbf{Y}_{t+l}|| < \infty$*;*

2. *the space* $\mathcal{Y}$ *is bounded, such that* $||\mathbf{y}|| \leq B < \infty\ \forall\ \mathbf{y} \in \mathcal{Y}$ *(this implies the first condition);*

3. $\beta \geq 1$*,* $\mathbb{E}||\mathbf{Y}_{t+l}||^{\beta(r+\delta)} < \infty$ *for all* t *and, with probability 1 with respect to* $(\mathbf{Y}_t)_t \sim P^\star$*, for all* t *and* ϕ*,* $\mathbb{E}_{\mathbf{X}\sim P^\phi_{(l)}(\cdot|\mathbf{y}_{t-k+1:t})}||\mathbf{X}||^\beta \leq B < \infty$*.*

4. Put simply, this condition means that the following has to be true for all observed sequences $(\mathbf{y}_t)_t$ which can be generated by the distribution $P^\star$.

Proof First, notice that $\sup_{t\geq k}\mathbb{E}\left[H(\mathbf{Y}_{t-k+1:t+l})^{r+\delta}\right] < \infty \iff \mathbb{E}\left[H(\mathbf{Y}_{t-k+1:t+l})^{r+\delta}\right] < \infty\ \forall\ t \geq k$.

Notice how the kernel SR recovers the Energy SR when $k(\mathbf{y},\mathbf{x}) = -||\mathbf{y}-\mathbf{x}||^{\beta}$; condition 1 for the kernel SR corresponds therefore to condition 1 for the Energy SR; therefore, the result holds under condition 1.

For condition 2 for the Energy SR, notice that

$$|k(\mathbf{y},\mathbf{x})| = ||\mathbf{y}-\mathbf{x}||^{\beta} \leq (||\mathbf{y}|| + ||\mathbf{x}||)^{\beta} \leq (2B)^{\beta},$$

where the first inequality comes from applying the triangle inequality and the second comes from condition 2 for the Energy SR. Therefore, condition 2 for the Energy SR implies condition 2 for the corresponding Kernel SR, from which the result follows.

Finally, an alternative route leads to condition 3. Specifically, for the Energy SR, Equation (24) becomes

$$\begin{aligned}
&|S_E^{(\beta)}(P_{(l)}^{\phi}(\cdot|\mathbf{y}_{t-k+1:t}),\mathbf{y}_{t+l})| \\
&\quad\leq \mathbb{E}_{\mathbf{X},\mathbf{X}'\sim P_{(l)}^{\phi}(\cdot|\mathbf{y}_{t-k+1:t})}[||\mathbf{X}-\mathbf{X}'||^{\beta} + 2||\mathbf{X}-\mathbf{y}_{t+l}||^{\beta}] \\
&\quad\leq \mathbb{E}_{\mathbf{X},\mathbf{X}'\sim P_{(l)}^{\phi}(\cdot|\mathbf{y}_{t-k+1:t})}[(||\mathbf{X}|| + ||\mathbf{X}'||)^{\beta} + 2(||\mathbf{X}|| + ||\mathbf{y}_{t+l}||)^{\beta}]
\end{aligned}$$

by triangle inequality. Now, for any $\beta > 1$, $a, b > 0$, $(a+b)^{\beta} \leq 2^{\beta-1}(a^{\beta}+b^{\beta})$;[5] therefore,

$$\begin{aligned}
&|S_E^{(\beta)}(P_{(l)}^{\phi}(\cdot|\mathbf{y}_{t-k+1:t}),\mathbf{y}_{t+l})| \\
&\quad\leq \mathbb{E}_{\mathbf{X},\mathbf{X}'\sim P_{(l)}^{\phi}(\cdot|\mathbf{y}_{t-k+1:t})}[2^{\beta-1}(||\mathbf{X}||^{\beta} + ||\mathbf{X}'||^{\beta}) + 2^{\beta}(||\mathbf{X}||^{\beta} + ||\mathbf{y}_{t+l}||^{\beta})].
\end{aligned}$$

From the above, we have that

$$\begin{aligned}
\mathbb{E}\left[H(\mathbf{Y}_{t-k+1:t+l})^{r+\delta}\right] &= \mathbb{E}\left[\left(\sup_{\phi\in\Phi}|S_E^{(\beta)}(P_{(l)}^{\phi}(\cdot|\mathbf{Y}_{t-k+1:t}),\mathbf{Y}_{t+l})|\right)^{r+\delta}\right] \\
\leq \mathbb{E}&\left[\left(\sup_{\phi\in\Phi}\mathbb{E}_{\mathbf{X},\mathbf{X}'\sim P_{(l)}^{\phi}(\cdot|\mathbf{Y}_{t-k+1:t})}[2^{\beta-1}(||\mathbf{X}||^{\beta} + ||\mathbf{X}'||^{\beta}) + 2^{\beta}(||\mathbf{X}||^{\beta} + ||\mathbf{Y}_{t+l}||^{\beta})]\right)^{r+\delta}\right].
\end{aligned}$$

If, with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^{\star}$, for all $t \geq k$ and ϕ, $\mathbb{E}_{\mathbf{X}\sim P_{(l)}^{\phi}(\cdot|\mathbf{Y}_{t-k+1:t})}||\mathbf{X}||^{\beta} \leq B < \infty$, we have therefore

$$\begin{aligned}
\mathbb{E}\left[H(\mathbf{Y}_{t-k+1:t+l})^{r+\delta}\right] &\leq \mathbb{E}\left[\left(2^{\beta-1}(B+B) + 2^{\beta}(B + ||\mathbf{Y}_{t+l}||^{\beta})\right)^{r+\delta}\right] \\
&= \mathbb{E}\left[\left(2^{\beta+1}B + 2^{\beta}||\mathbf{Y}_{t+l}||^{\beta}\right)^{r+\delta}\right].
\end{aligned}$$

Now, denote $\delta' = r+\delta$; $\delta' > 1$ by assumption. It holds therefore, as above, $(a+b)^{\delta'} \leq 2^{\delta'-1}(a^{\delta'}+b^{\delta'})$ for $a, b > 0$; we have therefore that

$$\mathbb{E}\left[H(\mathbf{Y}_{t-k+1:t+l})^{\delta'}\right] \leq 2^{-1}\left(2^{\beta+2}B\right)^{\delta'} + 2^{\delta'(\beta+1)-1}\mathbb{E}||\mathbf{Y}_{t+l}||^{\beta\delta'};$$

5. This inequality is well-known and can be shown by convexity.

the above expression is therefore bounded whenever $\mathbb{E}||\mathbf{Y}_{t+l}||^{\beta(r+\delta)} < \infty$. ■

A.3.5 Defining the mixing conditions

Here, we give the precise definitions for the mixing conditions stated in Assumption **A4**(a). More background on the following definitions can be found, for instance, in Bradley (2005).

Definition 15 (Measures of dependence) *Consider a probability space $(\Omega, \mathcal{F}, P)$; for any two sigma algebras $\mathcal{A} \subseteq \mathcal{F}$ and $\mathcal{B} \subseteq \mathcal{F}$, define*

$$\alpha_P(\mathcal{A}, \mathcal{B}) := \sup_{A \in \mathcal{A}, B \in \mathcal{B}} |P(A \cap B) - P(A)P(B)|,$$
$$\varphi_P(\mathcal{A}, \mathcal{B}) := \sup_{A \in \mathcal{A}, B \in \mathcal{B}: P(B) > 0} |P(B|A) - P(B)|.$$

For $1 \leq r \leq s \leq \infty$, define the Borel σ-algebra of events generated from $(\mathbf{Y}_r, \mathbf{Y}_{r+1}, \ldots, \mathbf{Y}_{s-1}, \mathbf{Y}_s)$ as $\mathcal{G}_r^s$. Then, we define

$$\alpha^{\mathbf{Y}}(m) = \sup_{r \geq 1} \alpha_{P^\star}(\mathcal{G}_1^r, \mathcal{G}_{r+m}^{+\infty}), \quad \varphi^{\mathbf{Y}}(m) = \sup_{r \geq 1} \varphi_{P^\star}(\mathcal{G}_1^r, \mathcal{G}_{r+m}^{+\infty}).$$

Definition 16 *The random sequence $(\mathbf{Y}_t)_t$ is said α-mixing if $\alpha^{\mathbf{Y}}(m) \to 0$ as $m \to \infty$ and φ-mixing if $\varphi^{\mathbf{Y}}(m) \to 0$ as $m \to \infty$. It can be seen that φ-mixing implies α-mixing (Domowitz and White, 1982).*

Definition 17 *We say that the mixing coefficients $\varphi^{\mathbf{Y}}(m)$ are of size s (Domowitz and White, 1982) if $\varphi^{\mathbf{Y}}(m) = \mathcal{O}(m^{-\lambda})$ for $\lambda > s$; similar definition can be given for the coefficients $\alpha^{\mathbf{Y}}(m)$.*

In Bradley (2005), the definitions for the quantities above consider a sequence $(\mathbf{X}_t)_{t \in \mathbb{Z}}$, and defined

$$\alpha^{\mathbf{X}}(m) = \sup_{r \in \mathbb{Z}} \alpha_P(\mathcal{G}_{-\infty}^r, \mathcal{G}_{r+m}^{+\infty}),$$

for some distribution P, and similar for $\phi^{\mathbf{X}}(m)$. Our definition can be cast in this way by defining $\mathbf{X}_t = \mathbf{Y}_t \; \forall \; t \geq 1$ and $\mathbf{X}_t = 0 \; \forall \; t \leq 0$.

A.3.6 Generic consistency result

We consider here the following Assumption:

A5 (Uniform Law of Large Numbers.) The following holds with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^\star$

$$\sup_{\phi \in \Phi} \left| S_T(P^\phi_{k+l:T}(\cdot|\mathbf{Y}_{1:k+l-1}), \mathbf{Y}_{k+l:T}) - S^\star_T(P^\phi_{k+l:T}) \right| \to 0.$$

We give here a consistency result more general than Theorem 8, as in fact Assumption **A5** is more general than the stationarity and mixing conditions in Assumption **A3** and **A4**.

Theorem 18 (Theorem 5.1 in Skouras, 1998) *If Assumptions **A2** and **A5** hold, then* $d(\hat{\phi}_T(\mathbf{Y}_{1:T}), \phi_T^\star) \to 0$ *with probability 1 with respect to* $(\mathbf{Y}_t)_t \sim P^\star$.

We report here a proof for ease of reference.

Proof By the definition of lim inf, for a fixed $\epsilon > 0$, Assumption **A2** implies that there exists $T_1(\epsilon)$ such that

$$\delta(\epsilon) := \left\{ \inf_{T > T_1(\epsilon)} \min_{\phi : d(\phi, \phi_T^\star) \geq \epsilon} S_T^\star(P_{k+l:T}^{\phi}) - S_T^\star(P_{k+l:T}^{\phi_T^\star}) \right\} > 0. \tag{25}$$

Due to Assumption **A5**, with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^\star$, there exists $T_2((\mathbf{Y}_t)_t, \delta(\epsilon))$ such that, for all $T > T_2((\mathbf{Y}_t)_t, \delta(\epsilon))$

$$\left| S_T(P_{k+l:T}^{\phi_T^\star}(\cdot|\mathbf{Y}_{1:k+l-1}), \mathbf{Y}_{k+l:T}) - S_T^\star(P_{k+l:T}^{\phi_T^\star}) \right| < \delta(\epsilon)/2,$$

which implies

$$\begin{aligned} S_T^\star(P_{k:T}^{\phi_T^\star}) &> S_T(P_{k:T}^{\phi_T^\star}(\cdot|\mathbf{Y}_{1:k+l-1}), \mathbf{Y}_{k+l:T}) - \delta(\epsilon)/2 \\ &\geq S_T(P_{k+l:T}^{\hat{\phi}_T(\mathbf{Y}_{1:T})}(\cdot|\mathbf{Y}_{1:k+l-1}), \mathbf{Y}_{k+l:T}) - \delta(\epsilon)/2, \end{aligned} \tag{26}$$

where the second inequality is valid thanks to the definition of $\hat{\phi}_T(\mathbf{Y}_{1:T})$.

Similarly, by exploiting Assumption **A5** again, with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^\star$, there exists $T_3((\mathbf{Y}_t)_t, \delta(\epsilon))$ such that, for all $T > T_3((\mathbf{Y}_t)_t, \delta(\epsilon))$

$$\left| S_T^\star(P_{k+l:T}^{\hat{\phi}_T(\mathbf{Y}_{1:T})}) - S_T(P_{k+l:T}^{\hat{\phi}_T(\mathbf{Y}_{1:T})}(\cdot|\mathbf{Y}_{1:k+l-1}), \mathbf{Y}_{k+l:T}) \right| < \delta(\epsilon)/2. \tag{27}$$

Then, with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^\star$, for all
$T > \max\{T_2((\mathbf{Y}_t)_t, \delta(\epsilon)), T_3((\mathbf{Y}_t)_t, \delta(\epsilon))\}$

$$\begin{aligned} S_T^\star(P_{k+l:T}^{\hat{\phi}_T(\mathbf{Y}_{1:T})}) - S_T^\star(P_{k+l:T}^{\phi_T^\star}) &\leq S_T^\star(P_{k+l:T}^{\hat{\phi}_T(\mathbf{Y}_{1:T})}) - S_T(P_{k+l:T}^{\hat{\phi}_T(\mathbf{Y}_{1:T})}(\cdot|\mathbf{Y}_{1:k+l-1}), \mathbf{Y}_{k+l:T}) + \delta(\epsilon)/2 \\ &< \delta(\epsilon)/2 + \delta(\epsilon)/2 = \delta(\epsilon), \end{aligned} \tag{28}$$

where the first inequality is thanks to Eq. (26) and the second is thanks to Eq (27).

Now, Eq. (25) and Eq. (28) both hold with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^\star$ for all $T > \max\{T_1(\delta(\epsilon)), T_2((\mathbf{Y}_t)_t, \delta(\epsilon)), T_3((\mathbf{Y}_t)_t, \delta(\epsilon))\}$. Notice that Eq. (28) ensures that the difference considered in Eq. (25) is smaller than $\delta(\epsilon)$ for $\phi = \hat{\phi}_T(\mathbf{Y}_{1:T})$; However, Eq. (25) states that the same difference is larger or equal than $\delta(\epsilon)$ for all $\phi : d(\phi, \phi_T^\star) \geq \epsilon$, from which it follows that $d(\hat{\phi}_T(\mathbf{Y}_{1:T}), \phi_T^\star) < \epsilon$ with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^\star$. As ϵ is however arbitrary, it follows that, with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^\star$

$$d(\hat{\phi}_T(\mathbf{Y}_{1:T}), \phi_T^\star) \to 0.$$

■

A.3.7 Uniform law of large numbers

We will here show how the Uniform Law of Large Numbers in Assumption **A5** can be obtained from the stationarity and mixing conditions in **A3** and **A4**. To this aim, we exploit a result in Pötscher and Prucha (1989).

We consider now a generic sequence of random variables $\mathbf{Z}_t \in \mathcal{Z}$, and a function $q : \mathcal{Z} \times \Phi \to \mathbb{R}$. Let us denote now by $\mathcal{F}$ the Borel σ-algebra generated by the sequence $(\mathbf{Z}_t)_t$, $\Omega_{\mathbf{Z}}$ the space of realizations of $(\mathbf{Z}_t)_t$ and $Q^\star$ the probability distribution for it.

Consider the following Assumptions:

C1 (Dominance condition) For $D(\mathbf{z}) = \sup_{\phi\in\Phi} |q(\mathbf{z}, \phi)|$, there is some $\delta > 0$ such that

$$\sup_t \frac{1}{N} \sum_{t=1}^{N} \mathbb{E}\left[D(\mathbf{Z}_t)^{1+\delta}\right] < \infty.$$

C2 (Asymptotic stationarity) Let $Q_t^\star$ be the marginal distribution of $\mathbf{Z}_t$; then, $N^{-1}\sum_{t=1}^{N} Q_t^\star$ converges weakly to some probability measure F on $\mathcal{Z}$.

C3 (Pointwise law of large numbers) For some metric ρ on Φ, let

$$\bar{q}(\mathbf{z}, \phi, \tau) := \sup_{\phi':\rho(\phi,\phi')<\tau} q(\mathbf{z}, \phi'), \quad \underline{q}(\mathbf{z}, \phi, \tau) := \inf_{\phi':\rho(\phi,\phi')<\tau} q(\mathbf{z}, \phi').$$

For all $\phi \in \Phi$, there exists a sequence of positive numbers $\tau_i(\phi)$ such that $\tau_i(\phi) \to 0$ as $i \to \infty$, and such that for each τ_i the random variables $\bar{q}(\mathbf{Z}_t, \phi, \tau_i)$ and $\underline{q}(\mathbf{Z}_t, \phi, \tau_i)$ satisfy a strong law of large numbers, i.e., as $N \to \infty$:

$$\frac{1}{N}\sum_{t=1}^{N} \{\bar{q}(\mathbf{Z}_t, \phi, \tau_i) - \mathbb{E}\left[\bar{q}(\mathbf{Z}_t, \phi, \tau_i)\right]\} \to 0$$
$$\frac{1}{N}\sum_{t=1}^{N} \left\{\underline{q}(\mathbf{Z}_t, \phi, \tau_i) - \mathbb{E}\left[\underline{q}(\mathbf{Z}_t, \phi, \tau_i)\right]\right\} \to 0,$$

where the two above equations hold with probability 1 with respect to $(\mathbf{Z}_t)_t \sim Q^\star$.

Theorem 19 (Theorem 2 in Pötscher and Prucha, 1989) *If Assumptions **A1**, **C1**, **C2** and **C3** hold and if $q(\mathbf{z}, \phi)$ is continuous on $\mathcal{Z} \times \Phi$, then:*

(i) with probability 1 with respect to $(\mathbf{Z}_t)_t \sim Q^\star$,

$$\lim_{t\to\infty} \sup_{\phi\in\Phi} \left| \frac{1}{N}\sum_{t=1}^{N} \{q(\mathbf{Z}_t, \phi) - \mathbb{E}\left[q(\mathbf{Z}_t, \phi)\right]\} \right| = 0;$$

(ii) $\int q(\mathbf{z}, \phi) dF(\mathbf{z})$ exists and is finite, continuous on Φ and, with probability 1 with respect to $(\mathbf{Z}_t)_t \sim Q^\star$,

$$\lim_{t\to\infty} \sup_{\phi\in\Phi} \left| \frac{1}{N}\sum_{t=1}^{N} q(\mathbf{Z}_t, \phi) - \int q(\mathbf{z}, \phi) dF(\mathbf{z}) \right| = 0;$$

We now give sufficient conditions for Assumption **C3** to hold. In fact, sequences for which the dependence of $\mathbf{Z}_t$ on a past observation $\mathbf{Z}_{t-m}$ decreases to 0 quickly enough as $m \to \infty$ satisfy Assumption **C3**. This can be made more rigorous considering the definitions of α- and φ-mixing sequences given in Appendix A.3.5.

Given the sequence $(\mathbf{Z}_t)_t$, for $1 \leq r \leq s \leq \infty$, define the Borel σ-algebra of events generated from $(\mathbf{Z}_r, \mathbf{Z}_{r+1}, \ldots, \mathbf{Z}_{s-1}, \mathbf{Z}_s)$ as $\mathcal{F}_r^s$. Then, we define the mixing coefficients for $(\mathbf{Z}_t)_t$ as

$$\alpha^{\mathbf{Z}}(m) = \sup_{r \geq 1} \alpha_{Q^\star}(\mathcal{F}_1^r, \mathcal{F}_{r+m}^{+\infty}), \quad \varphi^{\mathbf{Z}}(m) = \sup_{r \geq 1} \varphi_{Q^\star}(\mathcal{F}_1^r, \mathcal{F}_{r+m}^{+\infty}).$$

Similarly to before, the random sequence $(\mathbf{Z}_t)_{t \in \mathbb{Z}}$ is said α-mixing if $\alpha^{\mathbf{Z}}(m) \to 0$ as $m \to \infty$ and φ-mixing if $\varphi^{\mathbf{Z}}(m) \to 0$ as $m \to \infty$. Additionally, we say that the mixing coefficients $\varphi^{\mathbf{Z}}(m)$ are of size s (Domowitz and White, 1982) if $\varphi^{\mathbf{Z}}(m) = \mathcal{O}(m^{-\lambda})$ for $\lambda > s$; similar definition can be given for the coefficients $\alpha^{\mathbf{Z}}(m)$.

Let us define now the following additional assumption:

C4 Both conditions below hold:

(a) (Mixing) Either one of the following holds:

i. $(\mathbf{Z}_t)_t$ is α-mixing with mixing coefficient of size $r/(2r-1)$, with $r \geq 1$, or

ii. $(\mathbf{Z}_t)_t$ is φ-mixing with mixing coefficient of size $r/(r-1)$ with $r > 1$.

(b) (Moment boundedness) $\sup_t \mathbb{E}\left[D(\mathbf{Z}_t)^{r+\delta}\right] < \infty$ for some $\delta > 0$, for the value of r corresponding to the condition above which is satisfied.

We give the following Lemma, which is contained in Corollary 1 in Pötscher and Prucha (1989).

Lemma 20 (Corollary 1 in Pötscher and Prucha, 1989) *Assumption **C4** implies Assumptions **C1** and **C3**.*

We can therefore state the following.

Corollary 21 *If Assumptions **A1**, **C2** and **C4** hold and if $q(\mathbf{z}, \phi)$ is continuous on $\mathcal{Z} \times \Phi$, then the conclusions of Theorem 19 are satisfied.*

A.3.8 Proving Theorem 8

Here, we finally prove Theorem 8 by combining the generic consistency result in Appendix A.3.6 with the uniform law of large number result reported in Appendix A.3.7.

Notice that, in stating Theorem 19 and Corollary 21, we have considered a generic sequence $(\mathbf{Z}_t)_t$. In the setting of our interest, however, we want to study the prequential scoring rule defined in Eq. (21), and use Corollary 21 to state conditions under which Assumption **A5**, and therefore Theorem 18, hold.

To this aim, we identify now $N = T - k - l + 1$, $\mathbf{Z}_t = \mathbf{Y}_{t:t+k+l-1}$ and $q(\mathbf{Z}_t, \phi) = S(P^{\phi}_{(l)}(\cdot|\mathbf{Y}_{t:t+k-1}), \mathbf{Y}_{t+k+l-1})$; which leads to

$$\begin{aligned}\frac{1}{N}\sum_{t=1}^{N} q(\mathbf{Z}_t, \phi) &= \frac{1}{T-k-l+1}\sum_{t=1}^{T-k-l+1} S(P^{\phi}_{(l)}(\cdot|\mathbf{Y}_{t:t+k-1}), \mathbf{Y}_{t+k+l-1}) \\ &= \frac{1}{T-k-l+1}\sum_{t=k}^{T-l} S(P^{\phi}_{(l)}(\cdot|\mathbf{Y}_{t-k+1:t}), \mathbf{Y}_{t+l}) \\ &= S_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{Y}_{1:k+l-1}), \mathbf{Y}_{k+l:T}).\end{aligned}$$

The distribution $Q^\star$ on $(\mathbf{Z}_t)_t$ considered in the previous section is induced therefore by $P^\star$ over $(\mathbf{Y}_t)_t$.

We want now to relate $\alpha^{\mathbf{Y}}(m)$ and $\varphi^{\mathbf{Y}}(m)$ to $\alpha^{\mathbf{Z}}(m)$ and $\varphi^{\mathbf{Z}}(m)$; in order to do so, notice that, as $\mathbf{Z}_t = \mathbf{Y}_{t:t+k+l-1}$, $\mathcal{F}^s_r = \mathcal{G}^{s+k+l-1}_r$. Therefore,

$$\begin{aligned}\alpha^{\mathbf{Z}}(m) &= \sup_{r\geq 1} \alpha^{\mathbf{Z}}(\mathcal{F}^r_1, \mathcal{F}^{+\infty}_{r+m}) = \sup_{r\geq 1} \alpha^{\mathbf{Y}}(\mathcal{G}^{r+k+l-1}_1, \mathcal{G}^{+\infty}_{r+m}) \\ &= \sup_{r\geq k+l} \alpha^{\mathbf{Y}}(\mathcal{G}^{r}_1, \mathcal{G}^{+\infty}_{r+m-k-l+1}) \leq \sup_{r\geq 1} \alpha^{\mathbf{Y}}(\mathcal{G}^{r}_1, \mathcal{G}^{+\infty}_{r+m-k-l+1}) = \alpha^{\mathbf{Y}}(m-k-l+1),\end{aligned}$$

and, similarly, $\varphi^{\mathbf{Z}}(m) \leq \varphi^{\mathbf{Y}}(m-k-l+1)$. As k is fixed, $\varphi^{\mathbf{Y}}(m) \to 0 \implies \varphi^{\mathbf{Z}}(m) \to 0$ as $m \to \infty$, which is to say, $(\mathbf{Y}_t)_t$ being φ-mixing implies $(\mathbf{Z}_t)_t$ is φ-mixing as well, and similar for α-mixing. Additionally, if the mixing coefficients for $(\mathbf{Z}_t)_t$ have a given size s, then the mixing coefficients for $(\mathbf{Y}_t)_t$ will have the same size, and viceversa. In fact, $\varphi^{\mathbf{Z}}(m) \leq \varphi^{\mathbf{Y}}(m-k-l+1) = \mathcal{O}(m^{-\lambda})$ implies either $\varphi^{\mathbf{Y}}(m) = \mathcal{O}(m^{-\lambda})$ or $\varphi^{\mathbf{Y}}(m) = o(m^{-\lambda})$, and similar for α-mixing.

We are now ready to prove Theorem 8.

Proof [Proof of Theorem 8.]

Notice that, by identifying $\mathbf{Z}_t = \mathbf{Y}_{t:t+k+l-1}$ and $q(\mathbf{Z}_t, \phi) = S(P^{\phi}_{(l)}(\cdot|\mathbf{y}_{t:t+k-1}), \mathbf{y}_{t+k+l-1})$, Assumption **A3** corresponds to Assumption **C2**, and Assumption **A4** implies Assumption **C4**, due to the conservation of size of the mixing coefficients discussed above.

Together with Assumption **A1** and the continuity condition, therefore, Corollary 21 holds, from which you have that, with probability 1 with respect to $(\mathbf{Y}_t)_t \sim P^\star$,

$$\lim_{T\to\infty} \sup_{\phi\in\Phi} \left| \frac{1}{T-k-l+1}\sum_{t=k}^{T-l} \left\{ S(P^{\phi}_{(l)}(\cdot|\mathbf{Y}_{t-k+1:t}), \mathbf{Y}_{t+l}) - \mathbb{E}\left[S(P^{\phi}_{(l)}(\cdot|\mathbf{Y}_{t-k+1:t}), \mathbf{Y}_{t+l}) \right] \right\} \right| = 0;$$

which, recalling the definition of $S_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{Y}_{1:k+l-1}), \mathbf{Y}_{k+l:T})$ and $S^\star_T(P^{\phi}_{k+l:T})$ in Eqs. (21) and (22), is the same as Assumption **A5**. Thanks to this and Assumption **A2**, therefore, Theorem 18 holds, from which the result follows. ■

Appendix B. More details on the different methods

B.1 Training generative networks via divergence minimization

B.1.1 f-GAN

The f-GAN approach is defined by considering an f-divergence in place of D in Eq. (1) in the main text

$$D_f(P^\star||P^\phi) = \int_{\mathcal{Y}} p^\phi(\mathbf{y}) f\left(\frac{p^\star(\mathbf{y})}{p^\phi(\mathbf{y})}\right) d\mu(\mathbf{y}),$$

where $f : \mathbb{R}_+ \to \mathbb{R}$ is a convex, lower-semicontinuous function for which $f(1) = 0$, and where p^ϕ and $p^\star$ are densities of P^ϕ and $P^\star$ with respect to a base measure μ. Let now dom_f denote the domain of f. By exploiting the Fenchel conjugate $f^*(t) = \sup_{u\in\mathrm{dom}_f}\{ut - f(u)\}$, Nowozin et al. (2016) obtain the following variational lower bound

$$D_f(P^\star||P^\phi) \geq \sup_{c\in\mathcal{C}} \left(\mathbb{E}_{\mathbf{Y}\sim P^\star} c(\mathbf{Y}) - \mathbb{E}_{\mathbf{X}\sim P^\phi} f^*(c(\mathbf{X}))\right),$$

which holds for any set of functions $\mathcal{C}$ from $\mathcal{Y}$ to dom_{f^*}. By considering a parametric set of functions $\mathcal{C} = \{c_\psi : \mathcal{Y} \to \mathrm{dom}_{f^*}, \psi \in \Psi\}$, a surrogate to the problem in Eq. (1) in the main text becomes:

$$\min_\phi \max_\psi \left(\mathbb{E}_{\mathbf{Y}\sim P^\star} c_\psi(\mathbf{Y}) - \mathbb{E}_{\mathbf{X}\sim P^\phi} f^*(c_\psi(\mathbf{X}))\right).$$

In the conditional setting discussed in Section 2.1 in the main text, the above generalizes to

$$\min_\phi \max_\psi \mathbb{E}_{\boldsymbol{\theta}\sim\Pi}\left(\mathbb{E}_{\mathbf{Y}\sim P^\star(\cdot|\boldsymbol{\theta})} c_\psi(\mathbf{Y};\boldsymbol{\theta}) - \mathbb{E}_{\mathbf{Y}\sim P^\phi(\cdot|\boldsymbol{\theta})} f^*(c_\psi(\mathbf{Y};\boldsymbol{\theta}))\right), \tag{30}$$

By denoting as $P^\star_{\boldsymbol{\theta},\mathbf{Y}}$ and $P^\phi_{\boldsymbol{\theta},\mathbf{Y}}$ the joint distributions over $\Theta \times \mathcal{Y}$, Eq. (30) corresponds to the relaxation of $D_f(P^\star_{\boldsymbol{\theta},\mathbf{Y}}||P^\phi_{\boldsymbol{\theta},\mathbf{Y}})$ under the constraint that the marginal of $P^\phi_{\boldsymbol{\theta},\mathbf{Y}}$ for $\boldsymbol{\theta}$ is equal to Π.

In order to solve the problem in Eq. (30), alternating optimization over ϕ and ψ can be performed; in Algorithm 1, we show a single epoch (i.e., a loop on the full training data set) of conditional f-GAN training; for simplicity, we consider here using a single pair $(\boldsymbol{\theta}_i, \mathbf{y}_i)$ to estimate the expectations in Eq. (30) (i.e., the batch size is 1), but using a larger number of samples is indeed possible. Notice how in Algorithm 1 we update the critic once every generator update; however, multiple critic updates can be done.

Algorithm 1 Single epoch conditional f-GAN training.

Require: Parametric map h_ϕ, critic network c_ψ, learning rates ϵ, γ.
  for each training pair $(\boldsymbol{\theta}_i, \mathbf{y}_i)$ **do**
    Sample $\mathbf{z} \sim Q$
    Obtain $\hat{\mathbf{x}}_i^\phi = h_\phi(\mathbf{z}, \boldsymbol{\theta}_i)$
    Set $\psi \leftarrow \psi + \gamma \cdot \nabla_\psi \Big[c_\psi(\mathbf{y}_i, \boldsymbol{\theta}_i) - f^*(c_\psi(\hat{\mathbf{x}}_i^\phi, \boldsymbol{\theta}_i))\Big]$
    Set $\phi \leftarrow \phi - \epsilon \cdot \nabla_\phi \Big[- f^*(c_\psi(\hat{\mathbf{x}}_i^\phi, \boldsymbol{\theta}_i))\Big]$
  end for

B.1.2 Wasserstein-GAN (WGAN)

Arjovsky et al. (2017) exploited the following expression for the 1-Wasserstein distance

$$W\left(P^\star, P^\phi\right) = \sup_{c:\|c\|_L \leq 1} \mathbb{E}_{\mathbf{Y}\sim\mathbb{P}^\star}[c(\mathbf{Y})] - \mathbb{E}_{\mathbf{X}\sim\mathbb{P}^\phi}[c(\mathbf{X})], \tag{31}$$

where $||c||_L$ denotes the Lipschitz constant of the function c. The different notation here highlights how W is a symmetric function. Plugging Eq. (31) into Eq. (1) in the main text leads again to an adversarial setting; here, the Lipschitz constraint can be enforced by clipping the weights of the neural network to a given range (Arjovsky et al., 2017). Alternatively, this hard constraint can be relaxed to a soft one via gradient penalization (Gulrajani et al., 2017).

B.1.3 MMD-GAN

A specific case of the MMD (Eq. 3 in the main text) is the Energy Distance

$$\mathcal{E}\left(P^\star, P^\phi\right) = \mathbb{E}\left[2||\mathbf{X}-\mathbf{Y}||_2^\beta - ||\mathbf{X}-\mathbf{X}'||_2^\beta - ||\mathbf{Y}-\mathbf{Y}'||_2^\beta\right], \tag{32}$$

where $\beta \in (0,2)$ and $||\cdot||_2$ denotes the ℓ_2 norm. In Bellemare et al. (2017), the above is used to define an algorithm to train generative networks, termed Cramer-GAN.

In Li et al. (2017), the authors proposed to compute the kernel k in Eq. (3) in the main text on a learnable transformation c_ψ, whose weights are trained to maximize the discrepancy. Specifically, that leads to a new discrepancy measure

$$\max_\psi \mathbb{E}\big[k\left(c_\psi(\mathbf{X}), c_\psi(\mathbf{X}')\right) - 2k(c_\psi(\mathbf{X}), c_\psi(\mathbf{Y})) + k\left(c_\psi(\mathbf{Y}), c_\psi(\mathbf{Y}')\right)\big],$$

which is a meaningful divergence between probability distributions (Li et al., 2017). In this setting, again people resort to alternating maximization steps over ψ with minimization over ϕ. This, as mentioned in the main text, leads to biased estimates of gradients. However, for MMD-GANs, training is made easier by applying the gradient regularization techniques described in Gulrajani et al. (2017), as shown in Bińkowski et al. (2018).

Notice that, in minimizing Equations (3) in the main text with respect to ϕ, one could ignore the term involving $\mathbf{Y}, \mathbf{Y}'$; however, when introducing c_ψ, this cannot be done as that term depends on ψ as well.

In the conditional setting, a natural approach for MMD-GAN is minimizing $\mathbb{E}_{\boldsymbol{\theta}\sim\Pi}[\mathrm{MMD}^2\left(P^\star(\cdot|\boldsymbol{\theta}), P^\phi(\cdot|\boldsymbol{\theta})\right)]$, as $\mathrm{MMD}^2(P^\star_{\boldsymbol{\theta},\mathbf{Y}}, P^\phi_{\boldsymbol{\theta},\mathbf{Y}})$ would require computing kernel over $\Theta \times \mathcal{Y}$.

Notice however how, in estimating $\mathrm{MMD}^2\left(P^\star(\cdot|\boldsymbol{\theta}), P^\phi(\cdot|\boldsymbol{\theta})\right)$, multiple samples $\mathbf{Y}, \mathbf{Y}' \sim P^\star(\cdot|\boldsymbol{\theta})$ are used (see Eq. 3 in the main text), but those are unavailable (empirical samples are of the form in Eq. 4 in the main text); as discussed before, however, $k(\mathbf{Y}, \mathbf{Y}')$ does not depend on ϕ, so that it can be discarded in the minimization process. However, if the data is transformed via c_ψ, $k(c_\psi(\mathbf{Y}), c_\psi(\mathbf{Y}'))$ cannot be dropped anymore, which makes the problem intractable. In Bellemare et al. (2017), this problem is solved by replacing $k(c_\psi(\mathbf{Y}), c_\psi(\mathbf{Y}'))$ with some other tractable terms; however, that approach leads to an ill-defined statistical divergence, as it can be minimized by two distributions which are not the same (Bińkowski et al., 2018).

B.2 Scoring Rules

We now introduce some common SRs; let $\mathbf{X}, \mathbf{X}' \sim P^\phi$ be independent samples for the forecast distribution P^ϕ.

B.2.1 Energy Score

For $\beta \in (0, 2)$, the energy score is

$$S_{\mathrm{E}}^{(\beta)}(P^\phi, \mathbf{y}) = 2 \cdot \mathbb{E}\|\mathbf{X} - \mathbf{y}\|_2^\beta - \mathbb{E}\|\mathbf{X} - \mathbf{X}'\|_2^\beta. \tag{33}$$

The probabilistic forecasting literature (Gneiting and Raftery, 2007) use a different convention of the energy score and the subsequent kernel score, which amounts to multiplying our definitions by 1/2. We follow here the convention used in the statistical inference literature (Rizzo and Székely, 2016; Chérief-Abdellatif and Alquier, 2020; Nguyen et al., 2020)

The Energy Score is strictly proper for the class of probability measures P^ϕ such that $\mathbb{E}_{\mathbf{X}\sim P^\phi}\|\mathbf{X}\|^\beta < \infty$ (Gneiting and Raftery, 2007). The Energy Score is related to the Energy distance (Eq. (32)), which is a metric between probability distributions (Rizzo and Székely, 2016). We will fix $\beta = 1$ in the rest of this work. Additionally, for a univariate distribution and $\beta = 1$, the Energy Score recovers the Continuous Ranked Probability Score (CRPS), widely used in meteorology (e.g, see Hersbach, 2000).

B.2.2 Kernel Score

For a positive definite kernel $k(\cdot, \cdot)$, the kernel Scoring Rule can be defined as (Gneiting and Raftery, 2007)

$$S_k(P^\phi, \mathbf{y}) = \mathbb{E}[k(\mathbf{X}, \mathbf{X}')] - 2 \cdot \mathbb{E}[k(\mathbf{X}, \mathbf{y})].$$

The Kernel Score is connected to the squared Maximum Mean Discrepancy (MMD, Gretton et al., 2012) relative to the kernel k, see Eq. (3) in the main text. S_k is proper for the class of probability distributions for which $\mathbb{E}[k(\mathbf{X}, \mathbf{X}')]$ is finite (by Theorem 4 in Gneiting and Raftery, 2007). Additionally, it is strictly proper under conditions on k ensuring that the MMD is a metric for probability distributions on $\mathcal{Y}$ Gretton et al. (2012). These conditions are satisfied, among others, by the Gaussian kernel (which we will use in this work)

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|_2^2}{2\gamma^2}\right), \tag{34}$$

in which γ is a scalar bandwidth.

B.2.3 Patched Score

For the Patched Score, we consider different overlapping patches of the input data; denote as $\mathcal{P}$ the set of patches and as $p \in \mathcal{P}$ an individual patch; the patches are of a given size and spaced by a given spacing.

Then, we compute a SR S for multivariate distributions on each patch separately, and then add the results

$$S_p(P^\phi, \mathbf{y}) = \sum_{p\in\mathcal{P}} S(P^\phi|_p, \mathbf{y}|_p), \tag{35}$$

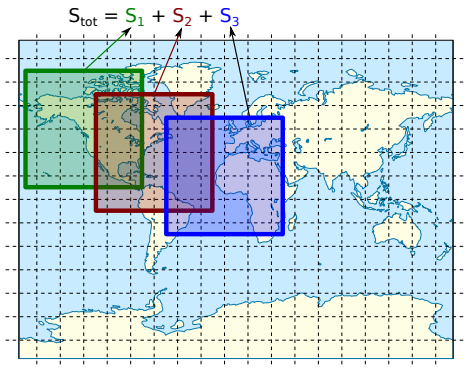


Figure 4: Patched SR: a SR for multivariate data is computed on localized patches, and the resulting values are summed.

where $\mathbf{y}|_p$ denotes the components of $\mathbf{y}$ in the patch p and $P^\phi|_p$ denotes the marginal distribution induced by P^ϕ for components in the patch p. See Figure 4 for a representation. As mentioned in the main body (Sec. 3.2 in the main text), the resulting SR is not strictly proper, as far away correlations are discarded. Notice how the topology of data for our global weather data set is periodic along the longitudinal direction (i.e., horizontally in Figure 4). The patches we define follow this.

Appendix C. Stochastic Gradient Descent for generative-SR networks

We discuss here how we can get unbiased gradient estimates for the prequential SR in Eq. (9) in the main text with respect to the parameters of the generative network ϕ.

In order to do that, we first discuss how to obtain unbiased estimates of the SRs we use across this work. Then, we show how those allow to obtain unbiased gradient estimates.

C.1 Unbiased scoring rule estimates

Consider we have draws $\mathbf{x}_j \sim P, j = 1, \ldots, m$.

C.1.1 Energy Score

An unbiased estimate of the energy score can be obtained by unbiasedly estimating the expectations in $S_{\mathrm{E}}^{(\beta)}(P, \mathbf{y})$ in Eq. (33)

$$\hat{S}_{\mathrm{E}}^{(\beta)}(\{\mathbf{x}_j\}_{j=1}^m, \mathbf{y}) = \frac{2}{m}\sum_{j=1}^{m} \|\mathbf{x}_j - \mathbf{y}\|_2^\beta - \frac{1}{m(m-1)}\sum_{\substack{j,k=1 \\ k\neq j}}^{m} \|\mathbf{x}_j - \mathbf{x}_k\|_2^\beta .$$

C.1.2 Kernel Score

Similarly to the energy score, we obtain an unbiased estimate of $S_k(P, y)$ by

$$\hat{S}_k(\{\mathbf{x}_j\}_{j=1}^m, \mathbf{y}) = \frac{1}{m(m-1)} \sum_{\substack{j,k=1 \\ k\neq j}}^{m} k(\mathbf{x}_j, \mathbf{x}_k) - \frac{2}{m} \sum_{j=1}^{m} k(\mathbf{x}_j, \mathbf{y}).$$

C.1.3 Variogram Score

It is immediate to obtain an unbiased estimate of $S_{\mathrm{v}}^{(p)}(P, \mathbf{y})$ in Eq. (11) in the main text by

$$\hat{S_{\mathrm{v}}}^{(p)}(\{\mathbf{x}_j\}_{j=1}^m, \mathbf{y}) = \sum_{i,j=1}^{d} w_{ij} \left(|y_i - y_j|^p - \frac{1}{m} \sum_{k=1}^{m} |x_{k,i} - x_{k,j}|^p \right)^2.$$

C.1.4 Patched SR

Assume the patched SR in Eq. (35) is built from a SR S which admits an unbiased empirical estimate $\hat{S}(\{\mathbf{x}_j\}_{j=1}^m, \mathbf{y})$. Therefore, an unbiased estimate of the patched SR can be obtained as

$$\hat{S}_p(\{\mathbf{x}_j\}_{j=1}^m, \mathbf{y}) = \sum_{p\in\mathcal{P}} S(\{\mathbf{x}_j|_p\}_{j=1}^m, \mathbf{y}|_p),$$

as in fact the components of samples $\mathbf{x}_j$ in the patch p are samples from the marginal distribution over the patch $P|_p$.

C.1.5 Sum of SRs

When adding multiple SRs, an unbiased estimate of the sum can be obtained by adding unbiased estimates of the two addends.

C.2 Unbiased estimate for gradient of S_T

Recall now we want to solve:

$$\hat{\phi}_T(\mathbf{y}_{1:T}) := \arg\min_{\phi} S_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), \mathbf{y}_{k+l:T}),$$

where, for simplicity, we re-define S_T in Eq. (9) in the main text with an additional scaling constant:

$$S_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), \mathbf{y}_{k+l:T}) := \frac{1}{T-l-k+1} \sum_{t=k}^{T-l} S(P^{\phi}_{t+l}(\cdot|\mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l}). \tag{36}$$

In order to do this, we exploit Stochastic Gradient Descent (SGD), which requires unbiased estimates of $S_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), \mathbf{y}_{k+l:T})$ (notice we are not talking here of unbiased estimates with respect to the observed sequence $\mathbf{y}_{1:T}$).

Notice how, for all the Scoring Rules used across this work, as well as any weighted sum of those, we can write: $S(P, \mathbf{y}) = \mathbb{E}_{\mathbf{Y},\mathbf{Y}'\sim P}\left[g(\mathbf{Y}, \mathbf{Y}', \mathbf{y})\right]$ for some function g; namely, the SR

is defined through an expectation over (possibly multiple) samples from P. That is the form exploited in Appendix C.1 to obtain unbiased SR estimates.

Now, we will use this fact to obtain unbiased estimates for the objective in Eq. (36). For brevity, let us now denote $J(\phi) = S_T(P^{\phi}_{k+l:T}(\cdot|\mathbf{y}_{1:k+l-1}), \mathbf{y}_{k+l:T})$, which we can rewrite as (letting $N = T - l - k + 1$ for brevity)

$$\begin{aligned} J(\phi) &= \frac{1}{N}\sum_{t=k}^{T-l} \mathbb{E}_{\mathbf{Y},\mathbf{Y}'\sim P^{\phi}(\cdot|\mathbf{y}_{t-k+1:t})}\left[g(\mathbf{Y}, \mathbf{Y}', \mathbf{y}_{t+l})\right] \\ &= \frac{1}{N}\sum_{t=k}^{T-l} \mathbb{E}_{\mathbf{Z},\mathbf{Z}'\sim Q}\left[g(h_{\phi}(\mathbf{Z}; \mathbf{y}_{t-k+1:t}), h_{\phi}(\mathbf{Z}'; \mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l})\right], \end{aligned}$$

where we used the fact that P^{ϕ} is the distribution induced by a generative network with transformation h_{ϕ}; this is called the reparametrization trick Kingma and Welling (2014). Now

$$\begin{aligned} \nabla_{\phi} J(\phi) &= \nabla_{\phi} \frac{1}{N}\sum_{t=k}^{T-l} \mathbb{E}_{\mathbf{Z},\mathbf{Z}'\sim Q}\left[g(h_{\phi}(\mathbf{Z}; \mathbf{y}_{t-k+1:t}), h_{\phi}(\mathbf{Z}'; \mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l})\right] \\ &= \frac{1}{N}\sum_{t=k}^{T-l} \mathbb{E}_{\mathbf{Z},\mathbf{Z}'\sim Q}\left[\nabla_{\phi} g\left(h_{\phi}(\mathbf{Z}; \mathbf{y}_{t-k+1:t}), h_{\phi}(\mathbf{Z}'; \mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l}\right)\right]. \end{aligned}$$

In the latter equality, the exchange between expectation and gradient is not a trivial step, due to the non-differentiability of functions (such as ReLU) used in h_{ϕ}. Luckily, Theorem 5 in Bińkowski et al. (2018) proved the above step to be valid almost surely with respect to a measure on Φ, under mild conditions on the NN architecture.

We can now easily obtain an unbiased estimate of the above. Additionally, Stochastic Gradient Descent usually consider a small batch of training samples, obtained by considering a random subset $\mathcal{T} \subseteq \{k, k+1 \ldots, n-l-1, n-l\}$. Therefore, the following unbiased estimator of $\nabla_{\phi} J(\phi)$ can be obtained, with samples $\mathbf{z}_{t,j} \sim Q, j = 1, \ldots, m$

$$\widehat{\nabla_{\phi} J(\phi)} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \frac{1}{m(m-1)} \sum_{\substack{i,j=1 \\ i\neq j}}^{m} \nabla_{\phi} g(h_{\phi}(\mathbf{z}_{t,i}; \mathbf{y}_{t-k+1:t}), h_{\phi}(\mathbf{z}_{t,j}; \mathbf{y}_{t-k+1:t}), \mathbf{y}_{t+l}).$$

In practice, we then use autodifferentiation libraries (see for instance Paszke et al., 2019) to compute the gradients in the above quantity.

In Algorithm 2, we train a generative network for a single epoch using a scoring rule S for which unbiased estimators can be obtained by using more than one sample from P^{ϕ}. As in Algorithm 1, we use a single pair $(\boldsymbol{\theta}_i, \mathbf{y}_i)$ to estimate the gradient.

Appendix D. Performance measures for probabilistic forecast

D.1 Deterministic performance measures

We discuss two measures of performance of a deterministic forecast $\hat{y}_{t+l}$ for a realization y_{t+l}; across our work, we take $\hat{y}_{t+l}$ to be the mean of the probability distribution $P^{\phi}(\cdot|\mathbf{y}_{t-k+1:t})$.

Algorithm 2 Single epoch generative-SR training.

Require: Parametric map h_ϕ, SR S, learning rate ϵ.
  for each training pair $(\boldsymbol{\theta}_i, \mathbf{y}_i)$ **do**
    Sample **multiple** $\mathbf{z}_1, \dots, \mathbf{z}_m$
    Obtain $\hat{\mathbf{x}}^\phi_{i,j} = h_\phi(\mathbf{z}_j, \boldsymbol{\theta}_i)$
    Obtain unbiased estimate $\hat{S}(P^\phi(\cdot|\boldsymbol{\theta}_i), \mathbf{y}_i)$ from $\hat{\mathbf{x}}^\phi_{i,j}$
    Set $\phi \leftarrow \phi - \epsilon \cdot \nabla_\phi \hat{S}(P^\phi(\cdot|\boldsymbol{\theta}_i), \mathbf{y}_i)$
  end for

D.1.1 Normalized RMSE

We first introduce the Root Mean-Square Error (RMSE) as

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N} (\hat{y}_{t+l} - y_{t+l})^2},$$

where we consider here for simplicity $t = 1, \dots, N$. From the above, we obtain the Normalized RMSE (NRMSE) as

$$\text{NRMSE} = \frac{RMSE}{\max_t\{y_{t+l}\} - \min_t\{y_{t+l}\}}.$$

NRMSE $= 0$ means that $\hat{y}_{t+l} = y_{t+l}$ for all t's.

D.1.2 Coefficient of determination

The coefficient of determination R^2 measures how much of the variance in $\{y_{t+l}\}_{t=1}^N$ is explained by $\{\hat{y}_{t+l}\}_{t=1}^N$. Specifically, it is given by

$$\text{R}^2 = 1 - \frac{\sum_{t=1}^N (y_{t+l} - \hat{y}_{t+l})^2}{\sum_{t=1}^N (y_{t+l} - \bar{y})^2},$$

where $\bar{y} = \frac{1}{N}\sum_{t=1}^{N} y_{t+l}$. $R^2 \leq 1$ and, when $\text{R}^2 = 1$, $\hat{y}_{t+l} = y_{t+l}$ for all t's. Notice how R^2 is unbounded from below, and can thus be negative.

D.2 Calibration error

We review here a measure of calibration of a probabilistic forecast; this measure considers the univariate marginals of the probabilistic forecast distribution $P^\phi(\cdot|\mathbf{y}_{t-k+1:t})$; for component i, let us denote that by $P_{\phi,i}(\cdot|\mathbf{y}_{t-k+1:t})$.

The calibration error (Radev et al., 2020) quantifies how well the credible intervals of the probabilistic forecast $P_{\phi,i}(\cdot|\mathbf{y}_{t-k+1:t})$ match the distribution of the verification $Y_{t+l,i}$. Specifically, let $\alpha^\star(i)$ be the proportion of times the verification $y_{t+l,i}$ falls into an α-credible interval of $P_{\phi,i}(\cdot|\mathbf{y}_{t-k+1:t})$, computed over all values of t. If the marginal forecast distribution is perfectly calibrated for component i, $\alpha^\star(i) = \alpha$ for all values of $\alpha \in (0, 1)$.

We define therefore the calibration error as the median of $|\alpha^\star(i) - \alpha|$ over 100 equally spaced values of $\alpha \in (0, 1)$. Therefore, the calibration error is a value between 0 and 1, where 0 denotes perfect calibration.

In practice, the credible intervals of the predictive are estimated using a set of samples from $P^\phi(\cdot|\mathbf{y}_{t-k+1:t})$.

Appendix E. Additional experimental details

E.1 Tuning γ in the Gaussian kernel

Similar to what was suggested for instance in Park et al. (2016), we set γ in the Gaussian kernel in Eq. (34) to be the median of the pairwise distances $||\mathbf{y}_i - \mathbf{y}_j||$ over all pairs of observations $\mathbf{y}_i, \mathbf{y}_j, i \neq j$ in the validation window.

E.2 Lorenz63 model

E.2.1 Model definition

The Lorenz63 model (Lorenz, 1963) is defined by the following differential equations

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x),$$
$$\frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho - z) - y,$$
$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z.$$

To generate our data set, we consider $\sigma = 10$, $\rho = 28$, $\beta = 2.667$ and integrate the model using Euler scheme with $dt = 0.01$ starting from $x = 0, y = 1, z = 1.05$. We discard the first 10 time units and integrate the model for additional 9000 time units, during which we record the value of y every $\Delta t = 0.3$ and discard the values of x and z.

E.2.2 Neural Networks architecture

We experiment with Recurrent Neural Networks (RNNs), which capture the temporal structure in the data.

For the generative network, the observation window is passed through a Gated Recurrent Units (GRU, Cho et al., 2014) layer with depth 1 and hidden size 8 or 16 (that is a tuning hyperparameter, the choice of which we discuss below). The output of the GRU layer is then concatenated to the latent variable $\mathbf{Z}$ with size 1 and passed through 3 fully connected layers, which output a forecast for the next timestep. For the deterministic setting trained with the regression loss, the architecture is analogous, the only difference being that no latent variable $\mathbf{Z}$ is concatenated to the output of the GRU layer.

In the adversarial settings, the critic has a GRU layer with depth 1 that, analogously to the generative net, processes the information in the past observation window. As above, we try hidden sizes 8 and 16. Then, the output of the GRU layer and the observation/forecast are concatenated and transformed by 3 fully connected layers. In the GAN case, the critic

Energy	Kernel	Energy-Kernel	Regression
0.01	0.001	0.01	0.001

Table 3: Optimal learning rate values for SR and regression (deterministic) approaches for Lorenz63.

	GAN (1)	GAN (2)	GAN (3)	WGAN-GP (1)	WGAN-GP (2)	WGAN-GP (3)
Generator l.r.	0.0003	0.001	0.0001	0.003	0.0003	0.0003
Critic l.r.	0.03	0.01	0.001	0.001	0.1	0.03
GRU hidden size	16	8	8	8	8	8
Critic training steps	1	1	1	5	5	5

Table 4: Optimal hyperparameter values for adversarial approaches for Lorenz63 model.

outputs a value between 0 and 1 indicating how confident the critic believes that is a fake sample. In the WGAN-GP case, the critic output is a real number.

E.2.3 Training hyperparameters

For the experiments on Lorenz63, we considered the batch size to be 1000. For the SR and deterministic approaches, we used Adam optimizer and tested the following learning rate values: 10^{-i} for $i = 1, \dots, 6$ for the SR methods and 10^{-i-1} and $3 \cdot 10^{-i-1}$ for $i = 1, \dots, 3$ for regression. We fix the GRU hidden size to 8. We report then the performance achieved with the learning rate yielding lower loss on the validation set, which is indicated in Table 3.

For the GAN and WGAN-GP approach, we used Adam optimizer and we tested the following learning rate values for both critic and generative network: 10^{-i} and $3 \cdot 10^{-i}$ for $i = 1, \dots, 7$. In total, those are 14 learning rate values. We tested GRU hidden size to 8 and 16; further, we experiment with 4 number of critic training steps for WGAN-GP (1, 3, 5, 10), in order to have the best possible results to compare with our SR methods, while we left the number of critic training steps to 1 for GAN. Overall, therefore, we had $2 \cdot 14^2 = 392$ experiments for GAN and $2 \cdot 4 \cdot 14^2 = 1568$ for WGAN-GP; notice the extremely larger number number of experiments for the adversarial approaches with respect to SR ones, which highlights an advantage of our approach. We stress that such a number of trials could be possible only for the low-dimensional setting of the Lorenz63 and Lorenz96 models, in which training is cheap, but not in real-life applications.

Additionally, the adversarial approaches do not allow to select hyperparameters according to loss on a validation set, as the generator loss depends on the current state of the discriminator (i.e., there is no absolute loss scale). Therefore, we report results for 3 different configurations for GAN and WGAN-GP, maximizing either deterministic performance (1) or calibration (2), or striking the best balance between these two (3). The resulting learning rates are in Table 4.

E.3 Lorenz96 model

E.3.1 Model definition

The Lorenz96 model (Lorenz, 1996) is a toy representation of atmospheric behavior containing slow ($\mathbf{x}$) and fast ($\mathbf{y}$) evolving variables.

Specifically, the evolution of the variables is determined by the following differential equations

$$\frac{\mathrm{d}x_k}{\mathrm{d}t} = -x_{k-1}(x_{k-2} - x_{k+1}) - x_k + F - \frac{hc}{b}\sum_{j=J(k-1)+1}^{kJ} y_j;$$
$$\frac{\mathrm{d}y_j}{\mathrm{d}t} = -cby_{j+1}(y_{j+2} - y_{j-1}) - cy_j + \frac{hc}{b}X_{\mathrm{int}[(j-1)/J]+1},$$

where $k = 1, \ldots, K$, and $j = 1, \ldots, JK$, and cyclic boundary conditions are assumed, so that index $k = K + 1$ corresponds to $k = 1$ and similarly for j. The above equations connect the fast and slow variables in a cyclic way. Additionally, x_k reciprocally depends on J fast variables.

Following Gagne et al. (2020), we take $K = 8$, $J = 32$, $h = 1$, $b = 10$, $c = 10$ and $F = 20$. We then integrate the above equations with RK4 scheme with $dt = 0.001$, starting from $x_k = y_j = 0$ for $k = 2, \ldots, K$ and $j = 2, \ldots JK$ and $x_1 = y_1 = 1$. We discard the first 2 time units and record the values of $\mathbf{x}$ every $\Delta t = 0.2$ (which corresponding to roughly one atmospheric day with respect to predictability, Gagne et al., 2020). We do this for additional 4000 time units, and split the resulting data set in training, validation and test according to the proportions 60%, 20% and 20%.

E.3.2 Neural Networks architecture

We experiment with Recurrent Neural Networks (RNNs), which capture the temporal structure in the data.

For the generative network, the observation window is passed through a Gated Recurrent Units (GRU, Cho et al., 2014) layer with depth 1 and hidden size 32 or 64 (that is a tuning hyperparameter, the choice of which we discuss below). The output of the GRU layer is then concatenated to the latent variable $\mathbf{Z}$ with size 1 and passed through 3 fully connected layers, which output a forecast for the next timestep. For the deterministic setting trained with the regression loss, the architecture is analogous, the only difference being that no latent variable $\mathbf{Z}$ is concatenated to the output of the GRU layer.

In the adversarial settings, the critic has a GRU layer with depth 1 that, analogously to the generative net, processes the information in the past observation window. As above, we try hidden sizes 8 and 16. Then, the output of the GRU layer and the observation/forecast are concatenated and transformed by 3 fully connected layers. In the GAN case, the critic outputs a value between 0 and 1 indicating how confident the critic believes that is a fake sample. In the WGAN-GP case, the critic output is a real number.

E.3.3 Training hyperparameters

For the experiments on Lorenz96, we considered the batch size to be 1000. For the SR and deterministic approaches, we used Adam optimizer and tested the following learning rate

Energy	Kernel	Energy-Kernel	Regression
0.01	0.001	0.001	0.003

Table 5: Optimal learning rate values for SR and regression (deterministic) approaches for Lorenz96.

	GAN (1)	GAN (2)	GAN (3)	WGAN-GP (1)	WGAN-GP (2)	WGAN-GP (3)
Generator l.r.	0.01	0.0001	0.0001	0.001	0.00003	0.0001
Critic l.r.	0.001	0.003	0.001	0.001	0.1	0.01
GRU hidden size	64	32	64	64	64	64
Critic training steps	1	1	1	10	1	5

Table 6: Optimal hyperparameter values for adversarial approaches for Lorenz96 model.

values: 10^{-i} for $i = 1, \dots, 6$ for the SR methods and 10^{-i-1} and $3 \cdot 10^{-i-1}$ for $i = 1, \dots, 3$ for regression. We fix the GRU hidden size to 32. We report then the performance achieved with the learning rate yielding lower loss on the validation set, which is indicated in Table 5.

For the GAN and WGAN-GP approach, we used Adam optimizer and we tested the following learning rate values for both critic and generative network: 10^{-i} and $3 \cdot 10^{-i}$ for $i = 1, \dots, 7$. In total, those are 14 learning rate values. We tested hidden size 32 and 64; further, we experiment with 4 number of critic training steps for WGAN-GP (1, 3, 5, 10), in order to have the best possible results to compare with our SR methods, while we left the number of critic training steps to 1 for GAN. Overall, therefore, we had $2 \cdot 14^2 = 392$ experiments for GAN and $2 \cdot 4 \cdot 14^2 = 1568$ for WGAN-GP; notice the extremely larger number number of experiments for the adversarial approaches with respect to SR ones, which highlights an advantage of our approach. We stress that such a number of trials could be possible only for the low-dimensional setting of the Lorenz63 and Lorenz96 models, in which training is cheap, but not in real-life applications.

Additionally, the adversarial approaches do not allow to select hyperparameters according to loss on a validation set, as the generator loss depends on the current state of the discriminator (i.e., there is no absolute loss scale). Therefore, we report results for 3 different configurations for GAN and WGAN-GP, maximizing either deterministic performance (1) or calibration (2), or striking the best balance between these two (3). The resulting learning rates are in Table 6. Notice that, for GAN, there was no configuration leading to intermediate performance between (1) and (2), so that the column for (3) is left empty.

E.4 WeatherBench data set

E.4.1 Variogram Score

For the Variogram Score, we use a weight matrix which is inversely proportional to the Haversine distance, which measures the angular distance between two points on the surface of a sphere. Specifically, by denoting the longitude and latitude (in radians) of component i

	Energy-Kernel	Energy-Variogram	Kernel-Variogram
α_1	1/70	1	1
α_2	1	$6.94 \cdot 10^{-7}$	$1.3 \cdot 10^{-8}$

Table 7: Weights for summed Scores.

of $\mathbf{y}$ as $\text{lon}_i, \text{lat}_i$, the Haversine distance is defined as:

$$d_{ij} = 2 \arcsin \left[\sqrt{\sin^2((\text{lat}_i - \text{lat}_j)/2) + \cos(\text{lat}_i)\cos(\text{lat}_j)\sin^2((\text{lon}_i - \text{lon}_j)/2)} \right]$$

The physical distance along the sphere can be computed by multiplying the above by Earth's radius (approximately 6371 km). However, that is just a scaling constant, therefore we ignore it in defining the variogram, which we take to be $w_{ij} = 1/d_{ij}$.

E.4.2 Choice of weights for summed scores

In the summed Scores (Energy-Variogram, Kernel-Variogram, Energy-Kernel and Patched Energy Score), we need to select the weights for the two addends. Notice that, in the Patched Energy Score, we consider the Energy Score computed on the full data to be the first addend, and the sum of the Energy Scores computed on each patch to be the second addend.

We fix the weights such that the two addends have roughly the same magnitude. This results, for the Energy-Variogram, Kernel-Variogram, Energy-Kernel, in the choices reported the Table 7.

For the Patched Energy Score, we use the following two setups in our experiments:

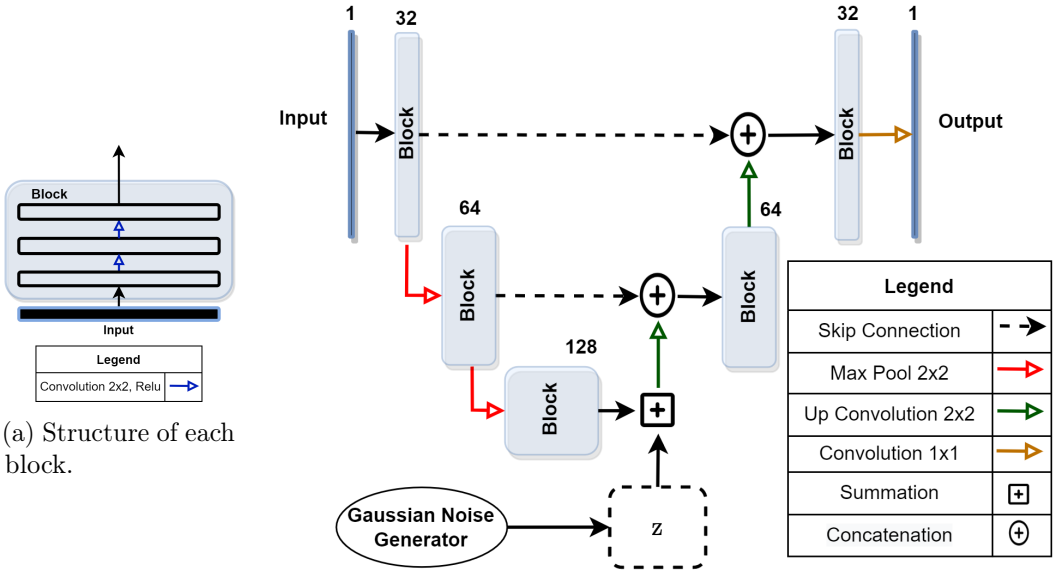
- Patches of size 16 separated by 8 grid points: this leads to 32 patches. As the Energy Score scales as the data dimensionality, each of the $16 \times 16 = 256$ patches has relative magnitude with respect to Energy Score computed on the full WeatherBench grid $256/2048 = 0.125$, where $32 \times 64 = 2048$ is the size of the WeatherBench grid. However, we sum the Score for each of the 32 patches, which leads to a quantity with magnitude 4 times the one of the overall Energy Score.

- Patches of size 8 separated by 4 grid points: this leads to 128 patches. Following the argument above, each $8 \times 8 = 64$ patch gives a Score with relative magnitude $64/2048 = 0.03125$. As there are 128 patches, again the cumulative patched score has magnitude 4 times the overall one.

In both cases, we leave therefore $\alpha_1 = \alpha_2 = 1$, as the patched and overall components are already of similar magnitude (they just differ by a factor 4).

E.4.3 Neural Networks architecture

For the generative network, we use a U-NET architecture (Olaf et al., 2015), which is an encoder-decoder structure, where each subsequent layer of the encoder outputs a downscaled latent representation of the input variables. The final output of the encoder is passed to a bottleneck layer, which performs no up/down scaling. The output of this bottleneck layer is



(a) Structure of each block.

(b) Full U-NET architecture.

Figure 5: U-NET architecture.

then passed to the decoder. Conversely to the encoder, each subsequent layer of the decoder outputs an upscaled latent representation of the bottleneck layer output. Additionally, skip connections allow information to pass directly between layers of the encoder and decoder at the same scale; in this way, both large scale structures and high-frequency information contributes to the output. The latent variable $\mathbf{Z}$ is summed to the latent representation in the bottleneck layer. Figure 5 gives a graphical representation of the UNet. For the deterministic setting trained with the regression loss, the architecture is analogous, the only difference being that no latent variable $\mathbf{Z}$ is summed to the latent representation.

In the adversarial setups, we use the PatchGAN critic suggested in Isola et al. (2017). Specifically, this is a convolutional network which considers separate patches of the input image and outputs a numerical value for each patch, corresponding, in the original GAN setting of Goodfellow et al. (2014), to the confidence with which the critic believes that patch is real, in contrast to generated from the generative network. The GAN or WGAN loss is then computed for each of the output values and averaged.

The PatchGAN critic employs some Batch Normalization layers; however, these cannot be used when the gradient penalization strategy of WGAN-GP is used (Gulrajani et al., 2017). Therefore, as suggested in Gulrajani et al. (2017), we replace the Batch Normalization layers with Layer Normalization.

As before, in the GAN case, the critic outputs a value between 0 and 1 indicating how confident the critic believes that is a fake sample. In the WGAN-GP case, the critic output is a real number.

	Regression	Energy	kernel	Energy-Kernel	Energy-Variogram
Learning rate	0.01	0.0001	0.0001	0.0001	10^{-5}

	Kernel-Variogram	Patched Energy (8)	Patched Energy (16)
Learning rate	10^{-5}	10^{-5}	10^{-5}

Table 8: Optimal learning rate values for the SR and regression (deterministic) approaches for WeatherBench.

E.4.4 Training hyperparameters

For the SR approaches for the WeatherBench data set, we considered the batch size to be 128 for all experiments, except for those on the Energy-Variogram and Kernel-Variogram score, which resulted in GPU memory overflow with that batch size (in fact, computing the Variogram Score is an operation requiring quadratic memory with respect to data size); for these two, we fixed therefore the batch size to be 48. We used Adam optimizer and tested the following learning rate values 10^{-i} for $i = 1, \dots, 6$. We report then the performance achieved with the learning rate yielding lower loss on the validation set in Table 8.

For the deterministic network trained via regression, we test learning rule values 10^{-i-1} for $i = 1, \dots, 4$; additionally, we use an exponential learning rate scheduler which reduces the learning rate by multiplying it by a factor γ every 10 training epochs. We also use a $\ell 2$ weight regularization with weight λ. We try different values of these parameters in conjunction with the learning rate values; the ones with which best validation loss is obtained are $\gamma = 0.8$ and $\lambda = 0.001$. The best learning rate value is reported in Table 8. Notice that the same learning rate value was optimal for the full (non-patched) regression loss and for the patched loss in both configurations.

For the GAN and WGAN-GP approach, we used Adam optimizer and we tested the following learning rate values for both critic and generative network: 10^{-i}, $i = 1, \dots, 7$. In total, those are 7 learning rate values, which result in $7^2 = 49$ experiments. Notice additionally that the adversarial approaches does not allow to select hyperparameters according to loss on a validation set, as the generator loss depends on the current state of the discriminator (i.e., there is no absolute loss scale). Additionally, the adversarial approaches do not allow to select hyperparameters according to loss on a validation set, as the generator loss depends on the current state of the discriminator (i.e., there is no absolute loss scale). Therefore, we report results for 3 different configurations for GAN, maximizing either deterministic performance (1) or calibration (2), or striking the best balance between these two (3). For WGAN-GP, a single configuration maximized both calibration and deterministic performance, so that we report that one. The resulting learning rates are in Table 9.

	GAN (1)	GAN (2)	GAN (3)	WGAN-GP
Generator learning rate	0.001	10^{-6}	10^{-5}	10^{-5}
Critic learning rate	0.0001	0.0001	10^{-5}	0.01

Table 9: Optimal hyperparameter values for adversarial approaches for WeatherBench.

	Cal. error ↓	NRMSE ↓	R^2 ↑
Regression	-	0.0198 ± 0.0006	0.9905 ± 0.0006
Energy	0.0205 ± 0.0176	0.0166 ± 0.0014	0.9933 ± 0.0012
Kernel	0.2196 ± 0.0123	0.0164 ± 0.0003	0.9935 ± 0.0003
Energy-Kernel	0.0104 ± 0.0060	0.0173 ± 0.0004	0.9928 ± 0.0004
GAN (1)	0.4644 ± 0.0062	0.0354 ± 0.0026	0.9696 ± 0.0044
GAN (2)	0.2671 ± 0.0559	0.1500 ± 0.0090	0.4537 ± 0.0619
GAN (3)	0.3700 ± 0.0369	0.0763 ± 0.0030	0.8590 ± 0.0099
WGAN-GP (1)	0.4134 ± 0.0051	0.0330 ± 0.0007	0.9736 ± 0.0009
WGAN-GP (2)	0.0565 ± 0.0339	0.1081 ± 0.0037	0.7165 ± 0.0200
WGAN-GP (3)	0.1648 ± 0.0444	0.0786 ± 0.0041	0.8502 ± 0.0149

Table 10: Average and standard deviation of performance measures for forecasts obtained with the different methods, on the test set for the Lorenz96 data set. Metrics are computed on each data component individually; then, the average and standard deviation is computed.

Appendix F. Additional experimental results

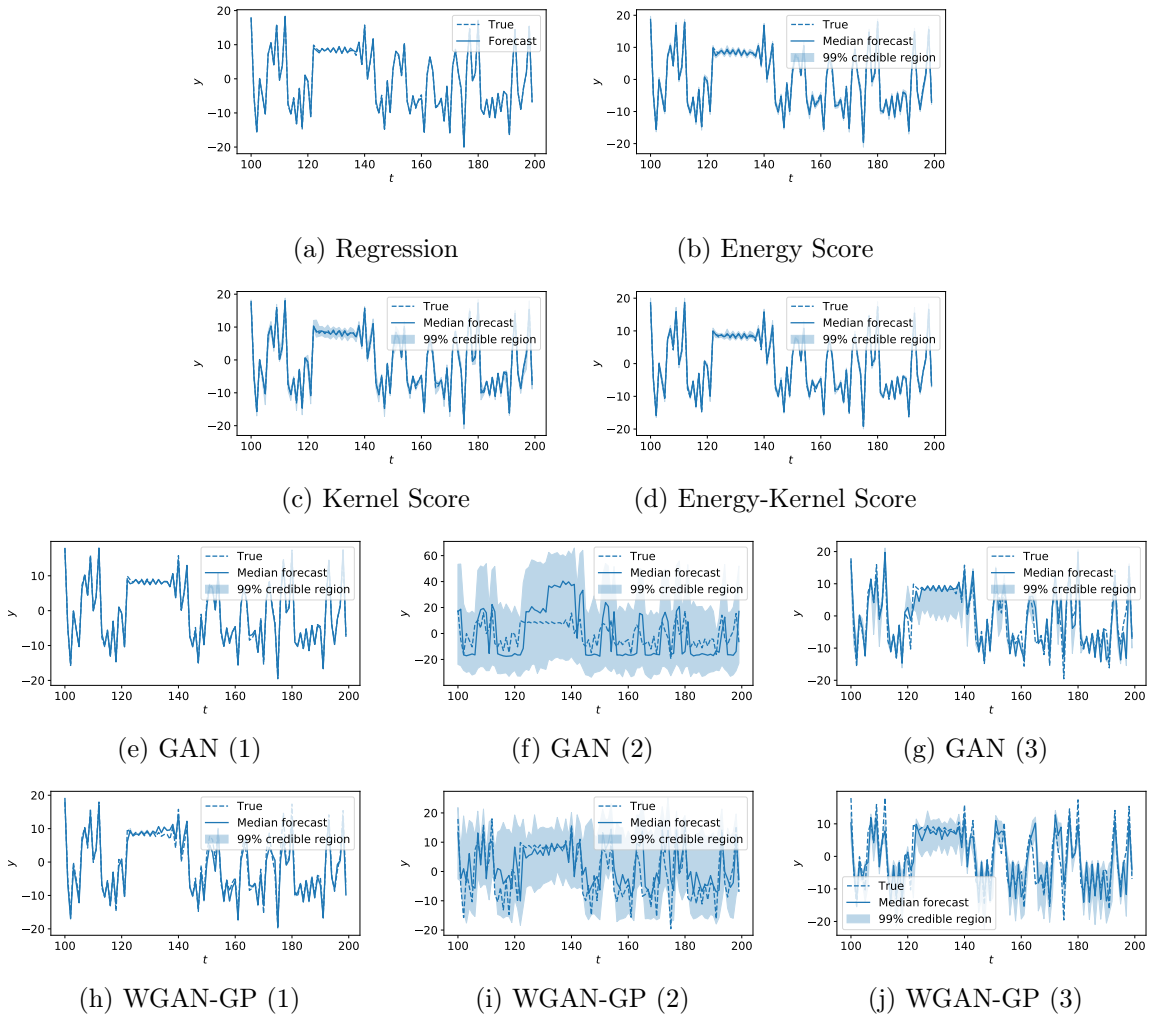
F.1 Additional results for Lorenz63 model

We report here additional results. Figure 6 contains separate plots for all methods showing forecasts and realization for a portion of the test set (the same used in Section 5.1 in the main text).

F.2 Additional results for Lorenz96 model

We report here additional results. Table 10 reports the average and standard deviation of the different performance measures computed across the different data components. It contains the same results as Table 1 in the main text, where however the standard deviation was not reported.

Figure 7 contains separate plots for all methods showing forecasts and realization for a portion of the test set (the same used in Section 5.1 in the main text).



(a) Regression (b) Energy Score

(c) Kernel Score (d) Energy-Kernel Score

(e) GAN (1) (f) GAN (2) (g) GAN (3)

(h) WGAN-GP (1) (i) WGAN-GP (2) (j) WGAN-GP (3)

Figure 6: Results for the Lorenz63 model with all considered methods. The figures show observations, median forecast and 99% credible interval for a portion of the test set. For each time-step, forecasts are obtained using the previous observation window.

F.3 WeatherBench data set

F.3.1 Standard deviation of performance measures

In Table 11, the average and standard deviation of the different performance measures are computed across the different data components.

F.3.2 Number of generator simulations for the SR methods

We study here the effect of using different numbers of simulations from the generative network for each input (i.e., how many forecasts the generative network provides) during training.

	Cal. error ↓	NRMSE ↓	R^2 ↑
Regression	-	0.1162 ± 0.0256	0.5300 ± 0.2559
Patched Regression, 8	-	0.1147 ± 0.0238	0.5459 ± 0.2297
Patched Regression, 16	-	0.1144 ± 0.0227	0.5509 ± 0.2188
Energy	0.0863 ± 0.0407	0.1208 ± 0.0256	0.4968 ± 0.2596
Kernel	0.0797 ± 0.0455	0.1200 ± 0.0226	0.5097 ± 0.2226
Energy-Kernel	0.0794 ± 0.0433	0.1194 ± 0.0226	0.5150 ± 0.2225
Energy-Variogram	0.0899 ± 0.0541	0.1192 ± 0.0220	0.5177 ± 0.2180
Kernel-Variogram	0.1704 ± 0.0607	0.1203 ± 0.0238	0.5050 ± 0.2399
Patched Energy, 8	0.0550 ± 0.0348	0.1189 ± 0.0209	0.5217 ± 0.2064
Patched Energy, 16	0.0690 ± 0.0478	0.1186 ± 0.0208	0.5248 ± 0.2034
GAN (1)	0.4845 ± 0.0089	0.1573 ± 0.0391	0.1418 ± 0.5267
GAN (2)	0.3130 ± 0.1143	0.2487 ± 0.2248	-2.7970 ± 17.1346
GAN (3)	0.3625 ± 0.0545	0.1693 ± 0.0494	-0.0117 ± 0.8348
WGAN-GP	0.1009 ± 0.0679	0.1302 ± 0.0214	0.4340 ± 0.2271

Table 11: Average and standard deviation of performance measures for forecasts obtained with the different methods, on the test section of the WeatherBench data set. Metrics are computed on each data component individually; then, the average and standard deviation is computed.

	Cal. error ↓	NRMSE ↓	R^2 ↑
2	0.0625 ± 0.0340	0.1211 ± 0.0258	0.4935 ± 0.2656
3	0.0701 ± 0.0342	0.1176 ± 0.0208	0.5338 ± 0.1961
5	0.0727 ± 0.0348	0.1164 ± 0.0198	0.5446 ± 0.1842
10	0.0863 ± 0.0407	0.1208 ± 0.0256	0.4968 ± 0.2596
20	0.0738 ± 0.0336	0.1179 ± 0.0206	0.5329 ± 0.1925
30	0.0738 ± 0.0350	0.1169 ± 0.0202	0.5407 ± 0.1864
50	0.0749 ± 0.0356	0.1172 ± 0.0203	0.5379 ± 0.1889

Table 12: Performance on test set of probabilistic forecasts obtained by training with the Energy Score, with different numbers of generator simulations, for the WeatherBench data set.

Recall in fact how the Energy and Kernel Score need multiple samples to be estimated (Appendix B.2).

Specifically, we consider the WeatherBench data set and the Energy Score, with learning rate 0.0001, which was found to be the optimal value when using 10 generator simulations (Appendix E.4.4). We report the measures used in the main text in Table 12. Notice how good performance is achieved when using as little as 2 or 3 simulations.

	Per-epoch Computational cost	Early stopping at epoch	Total computational cost
Regression	8.45	250	2112
Patched Regression, 8	8.65	200	1729
Patched Regression, 16	8.5	250	2122
Energy	54.2	100	5417
Kernel	53.3	100	5329
Energy-Kernel	55.4	100	5542
Energy-Variogram	97.38	250	24346
Kernel-Variogram	95.52	250	24393
Patched Energy, 8	56.71	400	22682
Patched Energy, 16	54.93	450	24717
GAN (1)	8.36	-	8357
GAN (2)	8.37	-	8373
GAN (3)	8.33	-	8326
WGAN-GP	7.00	-	7000

Table 13: Per-epoch and total computational cost, in seconds, for the different methods reported in the main text. We also report epoch at which early stopping occurred.

F.3.3 Computational cost and early stopping

In Table 13, we report the computational cost and the early stopping achieved by the methods presented in the main text. All experiments are run on a Tesla v100 GPU, and methods are run for a maximum of 1000 epochs. We use early stopping for the SR methods, but not for GAN and WGAN-GP, for which early stopping is not possible. Recall that the methods with the Variogram Score used training batch size 48, while all others used 128; this fact contributes to the larger computational time for both the Energy-Variogram and Kernel-Variogram Scores.

Additionally, recall that, in order to achieve the performance reported in the main text, we tried 49 learning rate values for GAN and WGAN-GP, but only 6 for the SR methods. Therefore, the total computing time for GAN and WGAN-GP is the one below multiplied by 49, with respect to 6 for the SR methods. Under that perspective, even the total computing time for Energy-Variogram and Kernel-Variogram Scores is smaller than the one for the adversarial methods. For instance, if we consider Energy-Variogram, do not use early stopping and run for 1000 epochs 6 times, we get a total of $97.38 \times 6000 = 584280$ seconds. For WGAN-GP, we obtain instead $7.00 \times 49 \times 1000 = 343000$ seconds, which is only slightly smaller than the grand total for Energy-Variogram. For the latter, this number does not take into account early stopping which, as can be seen from Table 13, reduces largely the total number of epochs required for training.

Additionally, we highlight how, in the results used for Table 13, the SR methods were trained using 10 simulations from the generator for each observation window (i.e., 10 forecasts). In Appendix F.3.2, we studied the effect of the number of simulations used on training, highlighted how the performance is good with as little as 2 or 3 simulations. This greatly reduces the computational cost; we report that in Table 14; for this study, the Energy Score was used.

	Per-epoch Computational cost	Early stopping at epoch	Total computational cost
2	13.7	100	1371
3	19.1	100	1913
5	29.6	100	2967
10	54.2	100	5417
20	107.0	100	10700
30	159.2	100	15916
50	258.7	100	25865

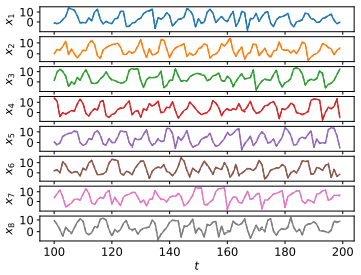
Table 14: Per-epoch and total computational cost, in seconds, for the Energy Score for different numbers of generator simulations. We also report epoch at which early stopping occurred.

F.3.4 Maps for a chosen date

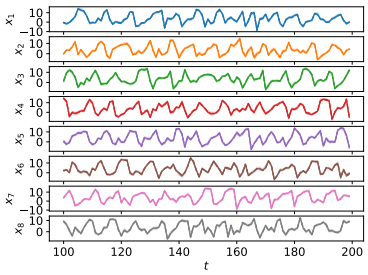
We provide figures similar to Fig. 3 in the main text in this online PDF file, due to space constraints in the present document. There, we also show deviation of draws from the forecast distribution and the realization from the forecast mean (obtained empirically from 100 draws from the forecast distribution).

F.3.5 Time-series plots for selected variables on the grid

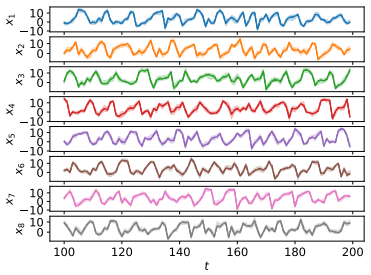
In Figures 8, 9, 10 and 11, and show the time series evolution, for a portion of the test period, for 8 randomly selected locations on the WeatherBench grid, for all considered methods (the same locations are shown for all methods). The dashed line represents the true evolution, the solid one the forecast mean, while the shaded region represents 99% credible intervals.



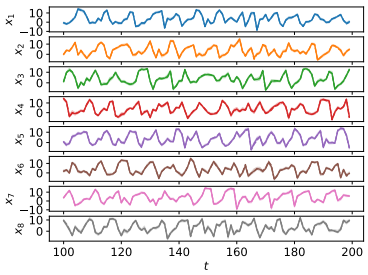
(a) Regression



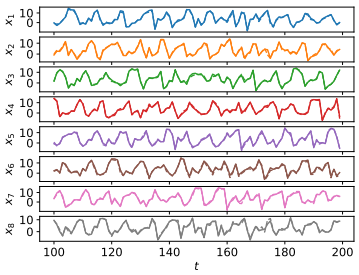
(b) Energy Score.



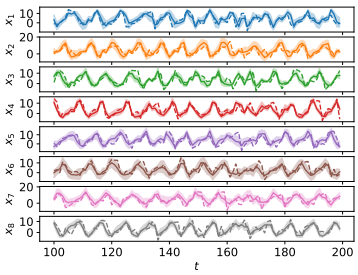
(c) Kernel Score.



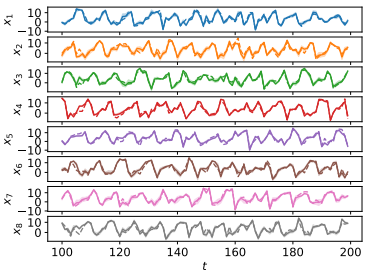
(d) Energy-Kernel Score



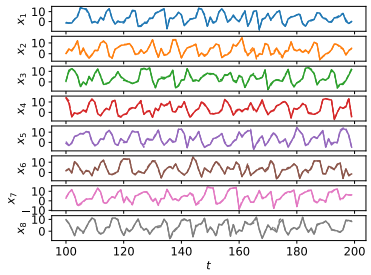
(e) GAN (1)



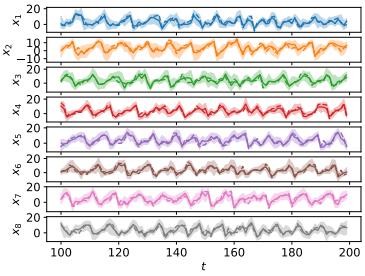
(f) GAN (2)



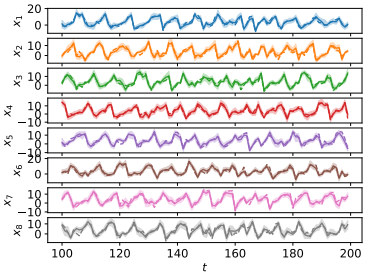
(g) GAN (3)



(h) WGAN-GP (1)

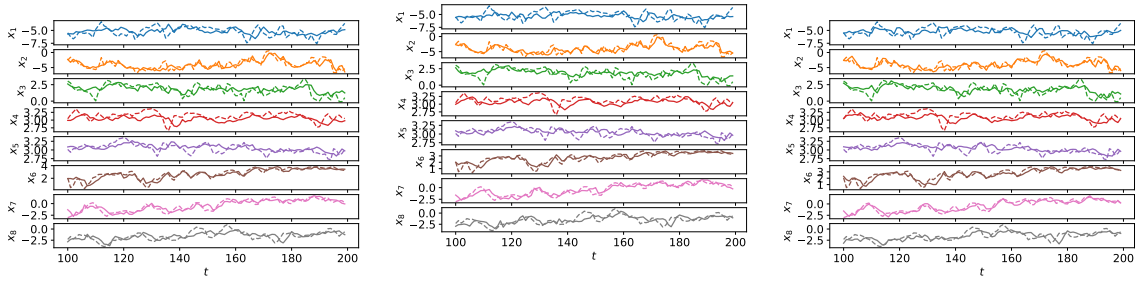


(i) WGAN-GP (2)



(j) WGAN-GP (3)

Figure 7: Results for the Lorenz96 model with all considered methods. Panels show observations (dashed line), median forecast (solid line) and 99% credible interval (shaded region) for a portion of the test set. That is done for all 8 components of $\mathbf{x}$. For each time-step, forecasts are obtained using the previous observation window.

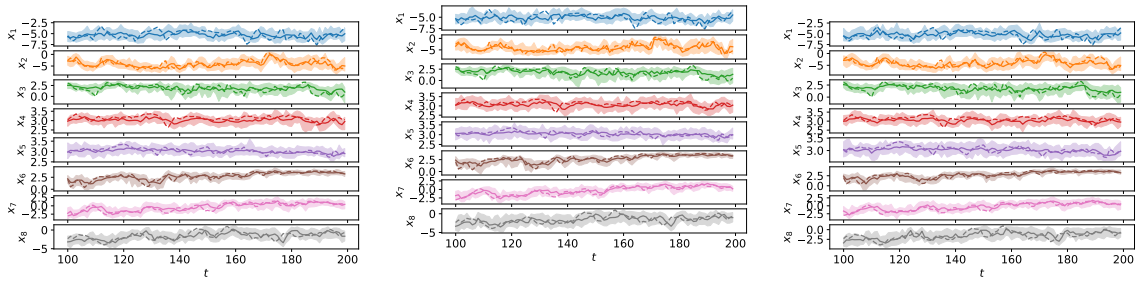


(a) Regression

(b) Patched Regression, 8

(c) Patched Regression, 16

Figure 8: Results with the Regression and patched regression losses for 8 locations on the WeatherBench grid. The panels show observations (dashed line) and median forecast (solid line)

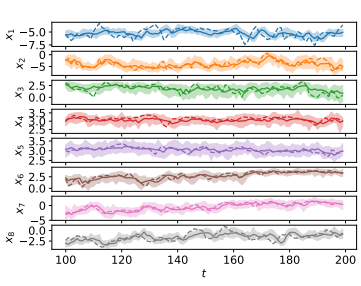


(a) Energy Score.

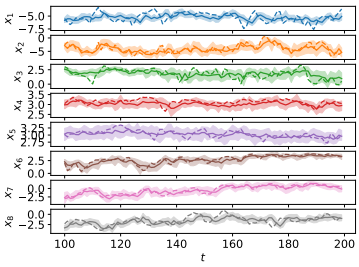
(b) Kernel Score.

(c) Energy-Kernel.

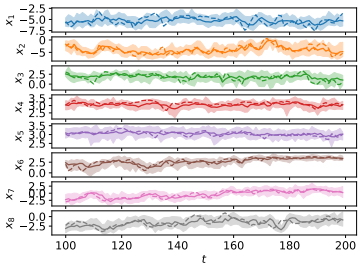
Figure 9: Results with the the Energy, Kernel and Energy-Kernel Scores for 8 locations on the WeatherBench grid. The panels show observations (dashed line), median forecast (solid line) and 99% credible interval (shaded region) for a portion of the test set.



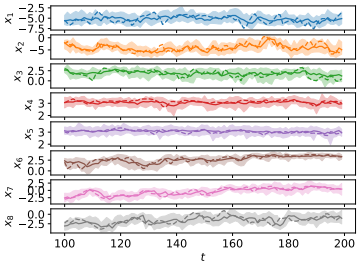
(a) Energy-Variogram Score



(b) Kernel-Variogram Score

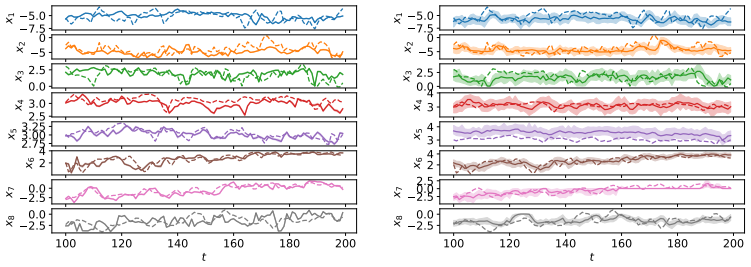


(c) Patched Energy Score (8)



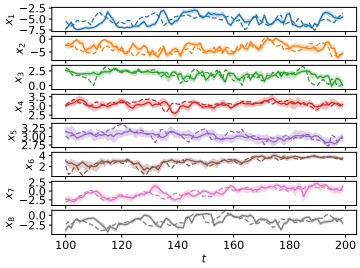
(d) Patched Energy Score (16)

Figure 10: Results with the Energy-Variogram, Kernel-Variogram and Patched Energy Score (with patch size both 8 and 16) Scores for 8 locations on the WeatherBench grid. The panels show observations (dashed line), median forecast (solid line) and 99% credible interval (shaded region) for a portion of the test set.

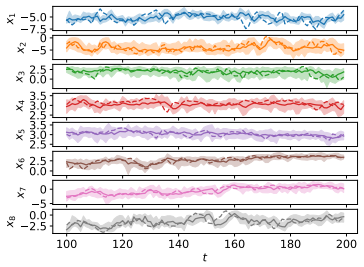


(a) GAN (1).

(b) GAN (2).



(c) GAN (3).



(d) WGAN-GP

Figure 11: Results with the three considered GAN setups and WGAN-GP Scores for 8 locations on the WeatherBench grid. The panels show observations (dashed line), median forecast (solid line) and 99% credible interval (shaded region) for a portion of the test set. Notice how the first GAN setup severely underestimates the uncertainty region, while the second one forecasts unpyhsical evolution for some time intervals.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (GANs). In *International Conference on Machine Learning*, pages 224–232. PMLR, 2017.

Sanjeev Arora, Andrej Risteski, and Yi Zhang. Do GANs learn the distribution? Some theory and empirics. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=BJehNfW0-`.

Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The Cramer distance as a solution to biased Wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

Alex Bihlo. A generative adversarial network approach to (ensemble) weather prediction. *Neural Networks*, 139:1–16, 2021.

Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018.

Diane Bouchacourt, Pawan K Mudigonda, and Sebastian Nowozin. DISCO nets: DISsimilarity COefficient networks. *Advances in Neural Information Processing Systems*, 29:352–360, 2016.

Richard C Bradley. Basic properties of strong mixing conditions. a survey and some open questions. *Probability surveys*, 2:107–144, 2005.

Eoin Brophy, Zhengwei Wang, Qi She, and Tomás Ward. Generative adversarial networks in time series: A systematic literature review. *ACM Comput. Surv.*, 55(10), feb 2023. ISSN 0360-0300. doi: 10.1145/3559540. URL `https://doi.org/10.1145/3559540`.

Badr-Eddine Chérief-Abdellatif and Pierre Alquier. MMD-Bayes: Robust Bayesian estimation via maximum mean discrepancy. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–21. PMLR, 2020.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-4012. URL `https://aclanthology.org/W14-4012`.

Mariana CA Clare, Omar Jamil, and Cyril J Morcrette. Combining distribution-based neural networks to predict weather
probabilities. *Quarterly Journal of the Royal Meteorological Society*, 147(741):4337–4357, 2021.

A Philip Dawid. Present position and potential developments: Some personal views statistical theory the prequential approach. *Journal of the Royal Statistical Society: Series A (General)*, 147(2):278–290, 1984.

A Philip Dawid and Monica Musio. Estimation of spatial processes using local scoring rules. *AStA Advances in Statistical Analysis*, 97(2):173–179, 2013.

A Philip Dawid and Monica Musio. Bayesian model selection based on proper scoring rules. *Bayesian analysis*, 10(2):479–499, 2015.

A Philip Dawid, Monica Musio, and Laura Ventura. Minimum scoring rule inference. *Scandinavian Journal of Statistics*, 43(1):123–138, 2016.

Alexander Philip Dawid and Monica Musio. Theory and applications of proper scoring rules. *Metron*, 72(2):169–183, 2014.

Ian Domowitz and Halbert White. Misspecified models with dependent observations. *Journal of Econometrics*, 20(1):35–58, 1982. ISSN 0304-4076. doi: https://doi.org/10.1016/0304-4076(82)90102-6. URL `https://www.sciencedirect.com/science/article/pii/0304407682901026`.

Peter D Dueben and Peter Bauer. Challenges and design choices for global weather and climate models based on machine learning. *Geoscientific Model Development*, 11(10): 3999–4009, 2018.

Gintare Karolina Dziugaite, Daniel M Roy, and Zoubin Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 258–267, 2015.

David John Gagne, Hannah M Christensen, Aneesh C Subramanian, and Adam H Monahan. Machine learning for stochastic parameterization: Generative adversarial networks in the Lorenz'96 model. *Journal of Advances in Modeling Earth Systems*, 12(3):e2019MS001896, 2020.

Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1:125–151, 2014.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1): 723–773, 2012.

Alexey Gritsenko, Tim Salimans, Rianne van den Berg, Jasper Snoek, and Nal Kalchbrenner. A spectral energy distance for parallel speech synthesis. *Advances in Neural Information Processing Systems*, 33:13062–13072, 2020.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of Wasserstein GANs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5769–5779, 2017.

Hans Hersbach. Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather and Forecasting*, 15(5):559–570, 2000.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

Zacharia Issa, Blanka Horvath, Maud Lemercier, and Cristopher Salvi. Non-adversarial training of neural SDEs with signature kernel scores. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=ixcsBZw5pl`.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL `http://arxiv.org/abs/1312.6114`.

Alireza Koochali, Andreas Dengel, and Sheraz Ahmed. If you like it, GAN it—probabilistic multivariate times series forecast with GAN. *Engineering Proceedings*, 5(1):40, Jul 2021. ISSN 2673-4591. doi: 10.3390/engproc2021005040. URL `http://dx.doi.org/10.3390/engproc2021005040`.

Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective cycle gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1811–1820, 2019.

Martin Leutbecher and Tim N Palmer. Ensemble forecasting. *Journal of computational physics*, 227(7):3515–3539, 2008.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. MMD GAN: Towards deeper understanding of moment matching network. In *NIPS*, 2017.

Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *International Conference on Machine Learning*, pages 1718–1727. PMLR, 2015.

Edward N Lorenz. Deterministic nonperiodic flow. *Journal of atmospheric sciences*, 20(2): 130–141, 1963.

Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on predictability*, volume 1, 1996.

John McCarthy. Measures of the value of information. *Proceedings of the National Academy of Sciences*, 42(9):654–655, 1956.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Hien Duy Nguyen, Julyan Arbel, Hongliang Lü, and Florence Forbes. Approximate Bayesian computation via the energy statistic. *IEEE Access*, 8:131683–131698, 2020.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 271–279, 2016.

Ronneberger Olaf, Fischer Philipp, and Brox Thomas. U-Net: Convolutional networks for biomedical image segmentation, 2015.

Lorenzo Pacchiardi and Ritabrata Dutta. Likelihood-free inference with generative neural networks via scoring rule minimization. *arXiv preprint arXiv:2205.15784*, 2022.

TN Palmer. Towards the probabilistic Earth-system simulator: a vision for the future of climate and weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 138(665):841–861, 2012.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. URL `http://jmlr.org/papers/v22/19-1028.html`.

Mijung Park, Wittawat Jitkrittum, and Dino Sejdinovic. K2-ABC: Approximate Bayesian computation with kernel embeddings. In *Artificial Intelligence and Statistics*, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

Benedikt M Pötscher and Ingmar R Prucha. A uniform law of large numbers for dependent and heterogeneous data processes. *Econometrica: Journal of the Econometric Society*, pages 675–683, 1989.

Stefan T Radev, Ulf K Mertens, Andreas Voss, Lynton Ardizzone, and Ullrich Köthe. BayesFlow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Stephan Rasp, Peter D Dueben, Sebastian Scher, Jonathan A Weyn, Soukayna Mouatadid, and Nils Thuerey. WeatherBench: a benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203, 2020.

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs M Bergmann, and Roland Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=WiGQBFuVRv`.

Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021.

Eitan Richardson and Yair Weiss. On GANs and GMMs. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 5852–5863, 2018.

Maria L Rizzo and Gábor J Székely. Energy distance. *Wiley interdisciplinary reviews: Computational statistics*, 8(1):27–38, 2016.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *Advances in neural information processing systems*, 29, 2016.

Leonard J Savage. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801, 1971.

Sebastian Scher. Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning. *Geophysical Research Letters*, 45(22): 12–616, 2018.

Sebastian Scher and Gabriele Messori. Weather and climate forecasting with neural networks: using general circulation models (GCMs) with different complexity as a study ground. *Geoscientific Model Development*, 12(7):2797–2809, 2019.

Sebastian Scher and Gabriele Messori. Ensemble methods for neural network-based weather forecasts. *Journal of Advances in Modeling Earth Systems*, 13(2), 2021.

Michael Scheuerer and Thomas M Hamill. Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities. *Monthly Weather Review*, 143(4):1321–1334, 2015.

Konstantinos Skouras. *On the optimal performance of forecasting systems: The prequential approach.* University of London, University College London (United Kingdom), 1998.

Amirhossein Vahidi, Simon Schosser, Lisa Wimmer, Yawei Li, Bernd Bischl, Eyke Hüllermeier, and Mina Rezaei. Probabilistic self-supervised representation learning via scoring rules minimization. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=skcTCdJz0f`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jonathan A Weyn, Dale R Durran, and Rich Caruana. Can machines learn to predict weather? Using deep learning to predict gridded 500-hPa geopotential height from historical weather data. *Journal of Advances in Modeling Earth Systems*, 11(8):2680–2693, 2019.

Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf`.