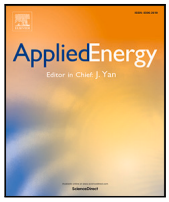
Applied Energy 305 (2022) 117871

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy



A deep generative model for probabilistic energy forecasting in power systems: normalizing flows

Jonathan Dumas [a,*], Antoine Wehenkel [a], Damien Lanaspeze [b], Bertrand Cornélusse [a], Antonio Sutera [a]

[a] Liège University, Departments of Computer Science and Electrical Engineering, Belgium
[b] Mines ParisTech, France

ARTICLE INFO

Keywords:
Deep learning
Normalizing flows
Energy forecasting
Time series
Generative adversarial networks
Variational autoencoders

ABSTRACT

Greater direct electrification of end-use sectors with a higher share of renewables is one of the pillars to power a carbon-neutral society by 2050. However, in contrast to conventional power plants, renewable energy is subject to uncertainty raising challenges for their interaction with power systems. Scenario-based probabilistic forecasting models have become a vital tool to equip decision-makers. This paper presents to the power systems forecasting practitioners a recent deep learning technique, the *normalizing flows*, to produce accurate scenario-based probabilistic forecasts that are crucial to face the new challenges in power systems applications. The strength of this technique is to directly learn the stochastic multivariate distribution of the underlying process by maximizing the likelihood. Through comprehensive empirical evaluations using the open data of the Global Energy Forecasting Competition 2014, we demonstrate that this methodology is competitive with other state-of-the-art deep learning generative models: generative adversarial networks and variational autoencoders. The models producing weather-based wind, solar power, and load scenarios are properly compared in terms of forecast value by considering the case study of an energy retailer and quality using several complementary metrics. The numerical experiments are simple and easily reproducible. Thus, we hope it will encourage other forecasting practitioners to test and use normalizing flows in power system applications such as bidding on electricity markets, scheduling power systems with high renewable energy sources penetration, energy management of virtual power plan or microgrids, and unit commitment.

1. Introduction

To limit climate change and achieve the ambitious targets prescribed by the Intergovernmental Panel on Climate Change [1], the transition towards a carbon-free society goes through an inevitable increase of the share of renewable generation in the energy mix. However, in contrast to conventional power plants, renewable energy is subject to uncertainty. Therefore, the operational predictability of modern power systems has been challenging as the total installed capacity of renewable energy sources (RES) increases and new distributed energy resources are introduced into the existing power networks. To address this challenge *point* forecasts are widely used in the industry as inputs to decision-making tools. However, they are inherently uncertain and in the context of decision-making, a point forecast plus an uncertainty interval is of genuine added value. In this context, *probabilistic* forecasts [2], which aim at modeling the distribution of all possible future realizations, have become an important tool [3] to equip decision-makers [4], hopefully leading to better decisions in energy applications [5].

The various types of probabilistic forecasts range from quantile to density forecasts, scenarios, and through prediction intervals [4]. This paper focuses on *scenario generation*, a popular probabilistic forecasting method to capture the uncertainty of load, photovoltaic (PV) generation, and wind generation. It consists of producing sequences of possible load or power generation realizations for one or more locations.

Forecasting methodologies can typically be classified into two groups: *statistical* and *machine learning* models. On the one hand, statistical approaches are more interpretable than machine learning techniques, sometimes referred to as black-box models. On the other hand, they are generally more robust, user-friendly, and successful in addressing the non-linearity in the data than statistical techniques. We provide in the following a few examples of statistical approaches. More references can be found in Khoshrou and Pauwels [6] and Mashlakov et al. [7].

* Corresponding author.
E-mail address: jdumas@uliege.be (J. Dumas).

https://doi.org/10.1016/j.apenergy.2021.117871
Received 17 June 2021; Received in revised form 10 August 2021; Accepted 10 September 2021
Available online 27 September 2021

Multiple linear regression models [8] and autoregressive integrated moving average [9] are among the most fundamental and widely-used models. For instance, an autoregressive moving average model is used by Morales et al. [10] to generate spatiotemporal scenarios with given power generation profiles at each renewables generation site. These models mostly learn a relationship between several explanatory variables and a dependent target variable. Therefore, the performance of such models is only satisfactory if the dependent variables are well formulated based on explanatory variables. However, they require some expert knowledge to formulate the relevant interaction between different variables. Another class of statistical approaches consists of using simple parametric distributions, *e.g.*, the Weibull distribution for wind speed [11], or the beta distribution for solar irradiance [12] to model the density associated with the generative process. In this line, the (Gaussian) copula method has been widely used to model the spatial and temporal characteristics of wind [13] and PV generation [14]. For instance, the problem of generating probabilistic forecasts for the aggregated power of a set of renewable power plants harvesting different energy sources is addressed by Camal et al. [15].

Overall, these approaches usually make statistical assumptions increasing the difficulty to model the underlying stochastic process. The generated scenarios approximate the future uncertainty but cannot correctly describe all the salient features in the power output from renewable energy sources. *Deep learning* is one of the newest trends in artificial intelligence and machine learning to tackle the limitations of statistical methods with promising results across various application domains.

1.1. Related work

Recurrent neural networks (RNNs) are among the most famous deep learning techniques adopted in energy forecasting applications. A novel pooling-based deep recurrent neural network is proposed by Shi et al. [16] in the field of short-term household load forecasting. It outperforms statistical approaches such as autoregressive integrated moving average and classical RNN. A tailored forecasting tool, named encoder–decoder, is implemented in Dumas et al. [17] to compute intraday multi-output PV quantiles forecasts. Guidelines and best practices are developed by Hewamalage et al. [18] for forecasting practitioners on an extensive empirical study with an open-source software framework of existing RNN architectures. In the continuity, Toubeau et al. [19] implemented a bidirectional long short-term memory (BLSTM) architecture. It is trained using quantile regression and combined with a copula-based approach to generate scenarios. This methodology is compared with other models in terms of forecast quality and value using a scenario-based stochastic optimization case study. Finally, Salinas et al. [20] trained an autoregressive recurrent neural network on several real-world datasets producing accurate probabilistic forecasts with little or no hyper-parameter tuning.

Deep generative modeling is a class of techniques that trains deep neural networks to model the distribution of the observations. In recent years, there has been a growing interest in this field made possible by the appearance of large open-access datasets and breakthroughs in both general deep learning architectures and generative models. Several approaches exist such as energy-based models, variational autoencoders, generative adversarial networks, autoregressive models, normalizing flows, and numerous hybrid strategies. They all make trade-offs in terms of computation time, diversity, and architectural restrictions. We recommend two papers to get a broader knowledge of this field. (1) The comprehensive overview of generative modeling trends conducted by Bond-Taylor et al. [21]. It presents generative models to forecasting practitioners under a single cohesive statistical framework. (2) The thorough comparison of normalizing flows, variational autoencoders, and generative adversarial networks provided by Ruthotto and Haber [22]. It describes the advantages and disadvantages of each approach using numerical experiments in the field of computer vision. In the following, we focus on the applications of generative models in power systems.

In contrast to statistical approaches, deep generative models such as *Variational AutoEncoders* (VAEs) [23] and *Generative Adversarial Networks* (GANs) [24] directly learn a generative process of the data. They have demonstrated their effectiveness in many applications to compute accurate probabilistic forecasts including power system applications. They both make probabilistic forecasts in the form of Monte Carlo samples that can be used to compute consistent quantile estimates for all sub-ranges in the prediction horizon. Thus, they cannot suffer from the issue raised by Ordiano et al. [25] on the non-differentiable quantile loss function. Note that generative models such as GANs and VAEs allow generating scenarios of the variable of interest directly. In contrast with methods that first compute weather scenarios to generate probabilistic forecasts such as implemented by Sun et al. [26] and Khoshrou and Pauwels [6]. A VAE composed of a succession of convolutional and feed-forward layers is proposed by Zhanga et al. [27] to capture the spatial–temporal complementary and fluctuant characteristics of wind and PV power with high model accuracy and low computational complexity. Both single and multi-output PV forecasts using a VAE are compared by Dairi et al. [28] to several deep learning methods such as LSTM, BLSTM, convolutional LSTM networks and stacked autoencoders, where the VAE consistently outperformed the other methods. A GAN is used by Chen et al. [29] to produce a set of wind power scenarios that represent possible future behaviors based only on historical observations and point forecasts. This method has a better performance compared to Gaussian Copula. A Bayesian GAN is introduced by Chen et al. [30] to generate wind and solar scenarios, and a progressive growing of GANs is designed by Yuan et al. [31] to propose a novel scenario forecasting method. In a different application, a GAN is implemented for building occupancy modeling without any prior assumptions [32]. Finally, a conditional version of the GAN using several labels representing some characteristics of the demand is introduced by Lan et al. [33] to output power load data considering demand response programs.

Improved versions of GANs and VAEs have also been studied in the context of energy forecasting. The Wasserstein GAN consists of enforcing the Lipschitz continuity through a gradient penalty term (WGAN-GP), as the original GANs are challenging to train and suffer from mode collapse and over-fitting. Several studies applied this improved version in power systems: (1) a method using unsupervised labeling and conditional WGAN-GP models the uncertainties and variation in wind power [34]; (2) a WGAN-GP models both the uncertainties and the variations of the load [35]; (3) Jiang et al. [36] implemented scenario generation tasks both for a single site and for multiple correlated sites without any changes to the model structure. Concerning VAEs, they suffer from inherent shortcomings, such as the difficulties of tuning the hyper-parameters or generalizing a specific generative model structure to other databases. An improved VAE is proposed by Qi et al. [37] with the implementation of a β hyper-parameter into the VAE objective function to balance the two parts of the loss. This improved VAE is used to generate PV and power scenarios from historical values.

However, most of these studies did not benefit from conditional information such as weather forecasts to generate improved PV, wind power, and load scenarios. In addition, to the best of our knowledge, only Ge et al. [38] compared NFs to these techniques for the generation of daily load profiles. Nevertheless, the comparison only considers quality metrics, and the models do not incorporate weather forecasts.

1.2. Research gaps and scientific contributions

This study investigates the implementation of *Normalizing Flows* [39, NFs] in power system applications. NFs define a new class of probabilistic generative models that has gained increasing interest from the deep learning community in recent years. A NF learns a sequence of transformations, a *flow*, from a density known analytically, *e.g.*, a *Normal*

distribution, to a complex target distribution. In contrast to other deep generative models, NFs can directly be trained by maximum likelihood estimation. They have proven to be an effective way to model complex data distributions with neural networks in many domains such as speech synthesis [40], fundamental physics to increase the speed of gravitational wave inference by several orders of magnitude [41] or for sampling Boltzmann distributions of lattice field theories [42], and have been applied in the capacity firming framework by Dumas et al. [43].

This present work goes several steps further than Ge et al. [38] that demonstrated the competitiveness of NFs regarding GANs and VAEs for generating daily load profiles. First, we study the conditional version of these models to demonstrate that they can handle additional contextual information such as weather forecasts or geographical locations. Second, we extensively compare the model's performances both in terms of forecast value and quality. The forecast quality corresponds to the ability of the forecasts to genuinely inform of future events by mimicking the characteristics of the processes involved. The forecast value relates to the benefits of using forecasts in decision-making, such as participation in the electricity market. Third, we consider PV and wind generations in addition to load profiles. Finally, in contrast to the affine NFs used in their work, we rely on monotonic transformations, which are universal density approximators [44].

Given that Normalizing Flows are rarely used in the power systems community despite their potential, our main aim is to present this recent deep learning technique and demonstrate its interest and competitiveness with state-of-the-art generative models such as GANs and VAEs on a simple and easily reproducible case study. The research gaps motivating this paper are three-fold:

1. To the best of our knowledge, only Ge et al. [38] compared NFs to GANs and VAEs for the generation of daily load profiles. Nevertheless, the comparison is only based on quality metrics, and the models do not take into account weather forecasts;
2. Most of the studies that propose or compare forecasting techniques only consider the forecast quality, such as Ge et al. [38], Sun et al. [26], and Mashlakov et al. [7];
3. The conditional versions of the models are not always addressed such as in Ge et al. [38]. However, weather forecasts are essential for computing accurate probabilistic forecasts.

With these research gaps in mind, the main contributions of this paper are three-fold:

1. We provide a fair comparison both in terms of quality and value with the state-of-the-art deep learning generative models, GANs and VAEs, using the open data of the Global Energy Forecasting Competition 2014 (GEFcom 2014) [45]. To the best of our knowledge, it is the first study that extensively compares the NFs, GANs, and VAEs on several datasets, PV generation, wind generation, and load with a proper assessment of the quality and value based on complementary metrics, and an easily reproducible case study;
2. We implement conditional generative models to compute improved weather-based PV, wind power, and load scenarios. In contrast to most of the previous studies that focused mainly on past observations;
3. Overall, we demonstrate that NFs are more accurate in quality and value, providing further evidence for deep learning practitioners to implement this approach in more advanced power system applications.

In addition to these contributions, this study also provides open-access to the Python code[1] to help the community to reproduce the experiments. Fig. 1 provides the framework of the proposed method and Table 1 presents a comparison of the present study to three state-of-the-art papers using deep learning generative models to generate scenarios.

[1] https://github.com/jonathandumas/generative-models

Table 1
Comparison of the paper's contributions to three state-of-the-art studies using deep generative models.

Criteria	[35]	[37]	[38]	study
GAN	✓	×	✓	✓
VAE	×	✓	✓	✓
NF	×	×	✓	✓
Number of models	4	1	3	3
PV	×	✓	×	✓
Wind power	×	✓	×	✓
Load	✓	~	✓	✓
Weather-based	✓	×	×	✓
Quality assessment	✓	✓	✓	✓
Quality metrics	5	3	5	8
Value assessment	×	✓	×	✓
Open dataset	~	×	✓	✓
Value replicability	–	~	–	✓
Open-access code	×	×	×	✓

✓: criteria fully satisfied, ~: criteria partially satisfied, ×: criteria not satisfied, ?: no information, -: not applicable. GAN: a GAN model is implemented; VAE: a VAE model is implemented; NF: a NF model is implemented; PV: PV scenarios are generated; Wind power: wind power scenarios are generated; Load: load scenarios are generated; Weather-based: the model generates weather-based scenarios; Quality assessment: a quality evaluation is conducted: Quality metrics: number of quality metrics considered; Value assessment: a value evaluation is considered with a case study; Open dataset: the data used for the quality and value evaluations are in open-access; Value replicability: the case study considered for the value evaluation is easily reproducible; Open-access code: the code used to conduct the experiments is in open-access. Note: the justifications are provided in Appendix A.1.

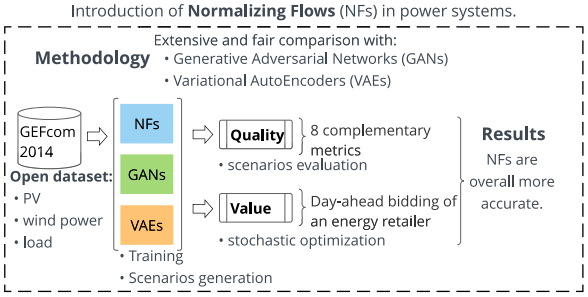


Fig. 1. The framework of the paper.
The paper's primary purpose is to present and demonstrate the potential of NFs in power systems. A fair comparison is conducted both in terms of quality and value with the state-of-the-art deep learning generative models, GANs and VAEs, using the open data of the Global Energy Forecasting Competition 2014 [45]. The PV, wind power, and load datasets are used to assess the models. The quality evaluation is conducted by using eight complementary metrics, and the value assessment by considering the day-ahead bidding of an energy retailer using stochastic optimization. Overall, NFs tend to be more accurate both in terms of quality and value and are competitive with GANs and VAEs.

1.3. Applicability of the generative models

Probabilistic forecasting of PV, wind generation, electrical consumption, and electricity prices plays a vital role in renewable integration and power system operations. The deep learning generative models presented in this paper can be integrated into practical engineering applications. We present a non-exhaustive list of five applications in the following. (1) The forecasting module of an energy management system (EMS) [46]. Indeed, EMSs are used by several energy market players to operate various power systems such as a single renewable plant, a grid-connected or off-grid microgrid composed of several generations, consumption, and storage devices. An EMS is composed of several key modules: monitoring, forecasting, planning, control, *etc*. The forecasting module aims to provide the most accurate forecast of the variable

of interest to be used as inputs of the planning and control modules. (2) Stochastic unit commitment models that employ scenarios to model the uncertainty of weather-dependent renewables. For instance, the optimal day-ahead scheduling and dispatch of a system composed of renewable plants, generators, and electrical demand are addressed by Camal et al. [15]. (3) Ancillary services market participation. A virtual power plant aggregating wind, PV, and small hydropower plants is studied by Camal et al. [15] to optimally bid on a day-ahead basis the energy and automatic frequency restoration reserve. (4) More generally, generative models can be used to compute scenarios for any variable of interest, *e.g.*, energy prices, renewable generation, loads, water inflow of hydro reservoirs, as long as data are available. (5) Finally, quantiles can be derived from scenarios and used in robust optimization models such as in the capacity firming framework [43].

1.4. Organization

The remainder of this paper is organized as follows. Section 2 presents the generative models implemented: NFs, GANs, and VAEs. Section 3 provides the quality and value assessment methodologies. Section 4 details empirical results on the GEFcom 2014 dataset, and Section 5 summarizes the main findings and highlights ideas for further work. Appendix A presents the justifications of Tables 1 and 5, Appendix B provides additional information on the generative models, Appendices C and D detail the quality metrics and the retailer energy case study formulation, and Appendix E presents additional quality results.

2. Background

This section formally introduces the conditional version of NFs, GANs, and VAEs implemented in this study. We assume the reader is familiar with the neural network's basics. However, for further information Goodfellow et al. [47], Zhang et al. [48] provide a comprehensive introduction to modern deep learning approaches.

2.1. Multi-output forecast

Let us consider some dataset $\mathcal{D} = \{\mathbf{x}^i, \mathbf{c}^i\}_{i=1}^N$ of N independent and identically distributed samples from the joint distribution $p(\mathbf{x}, \mathbf{c})$ of two continuous variables X and C. X being the wind generation, PV generation, or load, and C the weather forecasts. They are both composed of T periods per day, with $\mathbf{x}^i := [x_1^i, \dots, x_T^i]^\intercal \in \mathbb{R}^T$ and $\mathbf{c}^i := [c_1^i, \dots, c_T^i]^\intercal \in \mathbb{R}^T$. The goal of this work is to generate multi-output weather-based scenarios $\hat{\mathbf{x}} \in \mathbb{R}^T$ that are distributed under $p(\mathbf{x}|\mathbf{c})$.

A generative model is a probabilistic model $p_\theta(\cdot)$, with parameters θ, that can be used as a generator of the data. Its purpose is to generate synthetic but realistic data $\hat{\mathbf{x}} \sim p_\theta(\mathbf{x}|\mathbf{c})$ whose distribution is as close as possible to the unknown data distribution $p(\mathbf{x}|\mathbf{c})$. In our application, it computes on a day-ahead basis a set of M scenarios at day $d-1$ for day d

$$\hat{\mathbf{x}}_d^i := \left[\hat{x}_{d,1}^i, \dots, \hat{x}_{d,T}^i\right]^\intercal \in \mathbb{R}^T \quad i = 1, \dots, M. \tag{1}$$

For the sake of clarity, we omit the indexes d and i when referring to a scenario $\hat{\mathbf{x}}$ in the following.

2.2. Deep generative models

Fig. 2 provides a high-level comparison of three categories of generative models considered in this paper: Normalizing Flows, Generative Adversarial Networks, and Variational AutoEncoders.

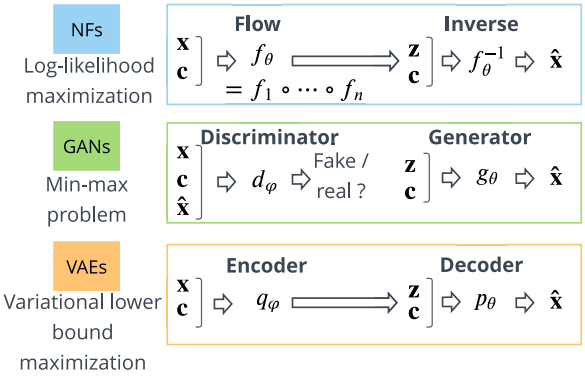


Fig. 2. High-level comparison of three categories of generative models considered in this paper: normalizing flows, generative adversarial networks, and variational autoencoders.
All models are conditional as they use the weather forecasts $\mathbf{c}$ to generate scenarios $\hat{\mathbf{x}}$ of the distribution of interest $\mathbf{x}$: PV generation, wind power, load. Normalizing flows allow exact likelihood calculation. In contrast to generative adversarial networks and variational autoencoders, they explicitly learn the data distribution and directly access the exact likelihood of the model's parameters. The inverse of the flow is used to generate scenarios. The training of generative adversarial networks relies on a min–max problem where the generator and the discriminator parameters are jointly optimized. The generator is used to compute the scenarios. Variational autoencoders indirectly optimize the log-likelihood of the data by maximizing the variational lower bound. The decoder computes the scenarios. Note: Section 2.3 provides a theoretical comparison of these models.

2.2.1. Normalizing flows

A normalizing flow is defined as a sequence of invertible transformations $f_k : \mathbb{R}^T \to \mathbb{R}^T$, $k = 1, \dots, K$, composed together to create an expressive invertible mapping $f_\theta := f_1 \circ \dots \circ f_K : \mathbb{R}^T \to \mathbb{R}^T$. This composed function can be used to perform density estimation, using f_θ to map a sample $\mathbf{x} \in \mathbb{R}^T$ onto a latent vector $\mathbf{z} \in \mathbb{R}^T$ equipped with a known and tractable probability density function p_z, *e.g.*, a Normal distribution. The transformation f_θ implicitly defines a density $p_\theta(\mathbf{x})$ that is given by the change of variables

$$p_\theta(\mathbf{x}) = p_z(f_\theta(\mathbf{x}))|\det J_{f_\theta}(\mathbf{x})|, \tag{2}$$

where J_{f_θ} is the Jacobian of f_θ regarding $\mathbf{x}$. The model is trained by maximizing the log-likelihood $\sum_{i=1}^N \log p_\theta(\mathbf{x}^i, \mathbf{c}^i)$ of the model's parameters θ given the dataset $\mathcal{D}$. For simplicity let us assume a single-step flow f_θ to drop the index k for the rest of the discussion.

In general, f_θ can take any form as long as it defines a bijection. However, a common solution to make the Jacobian computation tractable in (2) consists of implementing an *autoregressive* transformation [49], *i.e.*, such that f_θ can be rewritten as a vector of scalar bijections f^i

$$f_\theta(\mathbf{x}) := [f^1(x_1; h^1), \dots, f^T(x_T; h^T)]^\intercal, \tag{3a}$$

$$h^i := h^i(\mathbf{x}_{<i}; \varphi^i) \quad 2 \le i \le T, \tag{3b}$$

$$\mathbf{x}_{<i} := [x_1, \dots, x_{i-1}]^\intercal \quad 2 \le i \le T, \tag{3c}$$

$$h^1 \in \mathbb{R}, \tag{3d}$$

where $f^i(\cdot; h^i) : \mathbb{R} \to \mathbb{R}$ is partially parameterized by an autoregressive conditioner $h^i(\cdot; \varphi^i) : \mathbb{R}^{i-1} \to \mathbb{R}^{|h^i|}$ with parameters φ^i, and θ the union of all parameters φ^i.

There is a large choice of transformers f^i: affine, non-affine, integration-based, *etc.* In this work, an integration-based transformer is implemented by using the class of Unconstrained Monotonic Neural Networks (UMNN) proposed by Wehenkel and Louppe [50], which have been demonstrated to be a universal density approximator of continuous random variables when combined with autoregressive functions. The UMNN consists of a neural network architecture that enables

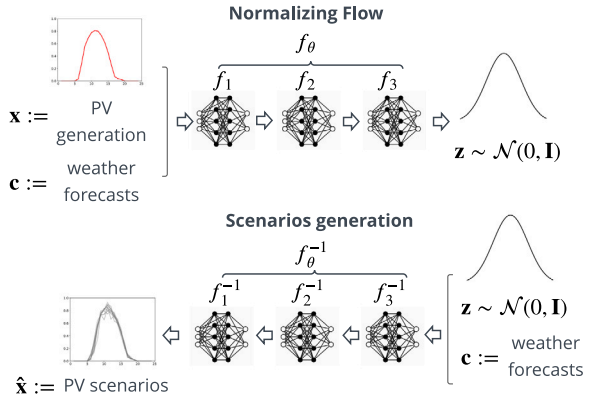


Fig. 3. The process of conditional normalizing flows is illustrated with a three-step NF for PV generation.
The model f_θ is trained by maximizing the log-likelihood of the model's parameters θ given a dataset composed of PV observations and weather forecasts. Recall f_θ defines a bijection between the variable of interest $\mathbf{x}$, PV generation, and a Normal distribution $\mathbf{z}$. Then, the PV scenarios $\hat{\mathbf{x}}$ are generated by using the inverse of f_θ that takes as inputs samples from the Normal distribution $\mathbf{z}$ and the weather forecasts $\mathbf{c}$.

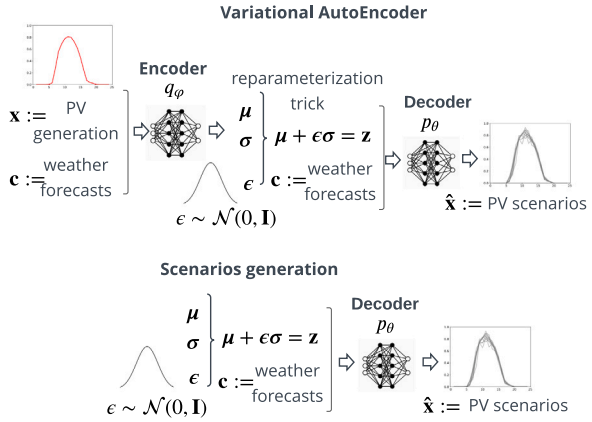


Fig. 4. The process of conditional variational autoencoder is illustrated for PV generation.
The VAE is trained by maximizing the variational lower bound given a dataset composed of PV observations and weather forecasts. The encoder q_φ maps the variable of interest $\mathbf{x}$ to a latent space $\mathbf{z}$. The decoder p_θ generates the PV scenarios $\hat{\mathbf{x}}$ by taking as inputs samples $\mathbf{z}$ from the latent space and the weather forecasts $\mathbf{c}$.

learning arbitrary monotonic functions. It is achieved by parameterizing the bijection f^i as follows

$$f^i(x_i; h^i) = \int_0^{x_i} \tau^i(x_i, h^i)dt + \beta^i(h^i), \tag{4}$$

where $\tau^i(\cdot; h^i) : \mathbb{R}^{|h^i|+1} \to \mathbb{R}^+$ is the integrand neural network with a strictly positive scalar output, $h^i \in \mathbb{R}^{|h^i|}$ an embedding made by the conditioner, and $\beta^i(\cdot) : \mathbb{R}^{|h^i|} \to \mathbb{R}$ a neural network with a scalar output. The forward evaluation of f^i requires solving the integral (4) and is efficiently approximated numerically by using the Clenshaw–Curtis quadrature. The pseudo-code of the forward and backward passes is provided by Wehenkel and Louppe [50].

Papamakarios et al. [51]'s Masked Autoregressive Network (MAF) is implemented to simultaneously parameterize the T autoregressive embeddings h^i of the flow (3). Then, the change of variables formula applied to the UMMN-MAF transformation results in the following log-density when considering weather forecasts

$$\log p_\theta(\mathbf{x}, \mathbf{c}) = \log p_z(f_\theta(\mathbf{x}, \mathbf{c}))| \det J_{f_\theta}(\mathbf{x}, \mathbf{c})|, \tag{5a}$$

$$= \log p_z(f_\theta(\mathbf{x}, \mathbf{c})) + \sum_{i=1}^{T} \log \tau^i\left(x_i, h^i(\mathbf{x}_{<i}), \mathbf{c}\right), \tag{5b}$$

that can be computed exactly and efficiently with a single forward pass. The UMNN-MAF approach implemented is referred to as NF in the rest of the paper. Fig. 3 depicts the process of conditional normalizing flows with a three-step NF for PV generation. Note: Appendix B.1 provides additional information on NFs.

2.2.2. Variational autoencoders

A VAE is a deep latent variable model composed of an *encoder* and a *decoder* which are jointly trained to maximize a lower bound on the likelihood. The encoder $q_\varphi(\cdot) : \mathbb{R}^T \times \mathbb{R}^{|\mathbf{c}|} \to \mathbb{R}^d$ approximates the intractable posterior $p(\mathbf{z}|\mathbf{x}, \mathbf{c})$, and the decoder $p_\theta(\cdot) : \mathbb{R}^d \times \mathbb{R}^{|\mathbf{c}|} \to \mathbb{R}^T$ the likelihood $p(\mathbf{x}|\mathbf{z}, \mathbf{c})$ with $\mathbf{z} \in \mathbb{R}^d$. Maximum likelihood is intractable as it would require marginalizing with respect to all possible realizations of the latent variables $\mathbf{z}$. Kingma and Welling [23] addressed this issue by maximizing the *variational lower bound* $\mathcal{L}_{\theta,\varphi}(\mathbf{x}, \mathbf{c})$ as follows

$$\log p_\theta(\mathbf{x}|\mathbf{c}) = KL[q_\varphi(\mathbf{z}|\mathbf{x}, \mathbf{c}) \parallel p(\mathbf{z}|\mathbf{x}, \mathbf{c})] + \mathcal{L}_{\theta,\varphi}(\mathbf{x}, \mathbf{c}), \tag{6a}$$

$$\geq \mathcal{L}_{\theta,\varphi}(\mathbf{x}, \mathbf{c}), \tag{6b}$$

$$\mathcal{L}_{\theta,\varphi}(\mathbf{x}, \mathbf{c}) := \mathbb{E}_{q_\varphi(\mathbf{z}|\mathbf{x},\mathbf{c})}\left[\log \frac{p(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{c})}{q_\varphi(\mathbf{z}|\mathbf{x}, \mathbf{c})}\right], \tag{6c}$$

as the Kullback–Leibler (KL) divergence [52] is non-negative. Appendix B.2 details how to compute the gradients of $\mathcal{L}_{\theta,\varphi}(\mathbf{x}, \mathbf{c})$, and its exact expression for the implemented VAE composed of fully connected neural networks for both the encoder and decoder. Fig. 4 depicts the process of a conditional variational autoencoder for PV generation.

2.2.3. Generative adversarial networks

GANs are a class of deep generative models proposed by Goodfellow et al. [24] where the key idea is the adversarial training of two neural networks, the *generator* and the *discriminator*, during which the generator learns iteratively to produce realistic scenarios until they cannot be distinguished anymore by the discriminator from real data. The generator $g_\theta(\cdot) : \mathbb{R}^d \times \mathbb{R}^{|\mathbf{c}|} \to \mathbb{R}^T$ maps a latent vector $\mathbf{z} \in \mathbb{R}^d$ equipped with a known and tractable prior probability density function $p(\mathbf{z})$, *e.g.*, a Normal distribution, onto a sample $\mathbf{x} \in \mathbb{R}^T$, and is trained to fool the discriminator. The discriminator $d_\phi(\cdot) : \mathbb{R}^T \times \mathbb{R}^{|\mathbf{c}|} \to [0, 1]$ is a classifier trained to distinguish between true samples $\mathbf{x}$ and generated samples $\hat{\mathbf{x}}$. Goodfellow et al. [24] demonstrated that solving the following min–max problem

$$\theta^\star = \arg\min_\theta \max_\phi V(\phi, \theta), \tag{7}$$

where $V(\phi, \theta)$ is the value function, recovers the data generating distribution if $g_\theta(\cdot)$ and $d_\phi(\cdot)$ are given enough capacity. The state-of-the-art conditional Wasserstein GAN with gradient penalty (WGAN-GP) proposed by Gulrajani et al. [53] is implemented with $V(\phi, \theta)$ defined as

$$V(\phi, \theta) = -\left(\mathbb{E}_{\hat{\mathbf{x}}}[d_\phi(\hat{\mathbf{x}}|\mathbf{c})] - \mathbb{E}_{\mathbf{x}}[d_\phi(\mathbf{x}|\mathbf{c})] + \lambda \text{GP}\right), \tag{8a}$$

$$\text{GP} = \mathbb{E}_{\tilde{\mathbf{x}}}\left[\left(\|\nabla_{\tilde{\mathbf{x}}} d_\phi(\tilde{\mathbf{x}}|\mathbf{c})\|_2 - 1\right)^2\right], \tag{8b}$$

where $\tilde{\mathbf{x}}$ is implicitly defined by sampling convex combinations between the data and the generator distributions $\tilde{\mathbf{x}} = \rho\hat{\mathbf{x}} + (1-\rho)\mathbf{x}$ with $\rho \sim \mathbb{U}(0, 1)$. The WGAN-GP constraints the gradient norm of the discriminator's output with respect to its input, to enforce the 1-Lipschitz conditions, in contrast to the weight clipping of WGAN that sometimes generates only poor samples or fails to converge. Appendix B.3 details the successive improvements from the original GAN to the WGAN, and the final WGAN-GP implemented, referred to as GAN in the rest of the paper. Fig. 5 depicts the process of a conditional generative adversarial network for PV generation.

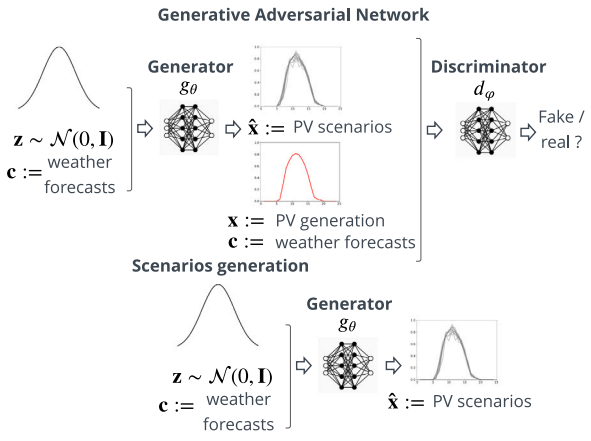


Fig. 5. The process of the conditional generative adversarial network is illustrated for PV generation.
The GAN is trained by solving a min–max problem given a dataset composed of PV observations $\mathbf{x}$ and weather forecasts. The generator g_θ computes PV scenarios $\hat{\mathbf{x}}$ by taking as inputs samples from the Normal distribution $\mathbf{z}$ and the weather forecasts $\mathbf{c}$, and the decoder d_ϕ tries to distinguishes true data from scenarios.

2.3. Theoretical comparison

Normalizing flows are a generative model that allows exact likelihood calculation. They are efficiently parallelizable and offer a valuable latent space for downstream tasks. In contrast to GANs and VAEs, NFs explicitly learn the data distribution and provide direct access to the exact likelihood of the model's parameters, hence offering a sound and direct way to optimize the network parameters [54]. However, NFs suffer from some drawbacks [21]: (1) one disadvantage of requiring transformations to be invertible is that the input dimension must be equal to the output dimension, which may make the model difficult to train or inefficient; (2) each transformation must be sufficiently expressive while being easily invertible to efficiently compute the Jacobian determinant. The first issue is also raised by Ruthotto and Haber [22] where it is said that ensuring sufficient similarity of the distribution of interest and the latent distribution is of high importance to obtain meaningful and relevant samples. However, in our numerical simulations, we did not encounter this problem. Concerning the second issue, the UMNN-MAF transformation provides an expressive and effective way of computing the Jacobian.

VAEs indirectly optimize the log-likelihood of the data by maximizing the variational lower bound. The advantage of VAEs over NFs is their ability to handle non-invertible generators and the arbitrary dimension of the latent space. However, it has been observed that when applied to complex datasets such as natural images, VAEs samples tend to be unrealistic. There is evidence that the limited approximation of the true posterior, with a common choice being a normally distributed prior with diagonal covariance, is the root cause [55]. This statement comes from the field of computer vision. However, it may explain the shape of the scenarios observed in our numerical experiments in Section 4.

The training of GANs relies on a min–max problem where the generator and the discriminator are jointly optimized. Therefore, it does not rely on estimates of the likelihood or latent variable. The adversarial nature of GANs makes them notoriously difficult to train due to the saddle point problem [56]. Another drawback is the mode collapsing, where one network stays in bad local minima, and only a small subset of the data distribution is learned. Several improvements have been designed to address these issues, such as the Wasserstein GAN with gradient penalty. Thus, GANs models are widely used in computer vision and power systems. However, most GAN approaches require cumbersome hyperparameter tuning to achieve similar results to VAEs or NFs. In our numerical simulations, the GAN is highly sensitive to hyperparameter variations, which is consistent with [22].

Each method has its advantages and drawbacks and makes trade-offs in terms of computing time, hyper-parameter tuning, architecture complexity, *etc*. Therefore, the choice of a particular method is dependent on the user criteria and the dataset considered. In addition, the challenges of power systems are different from computer vision. Therefore, the limitations established in the computer vision literature such as Bond-Taylor et al. [21] and Ruthotto and Haber [22] must be addressed with caution. Therefore, we encourage the energy forecasting practitioners to test and compare these methods in power systems applications.

3. Value and quality assessment

For predictions in any form, one must differentiate between their quality and their value [4]. Forecast quality corresponds to the ability of the forecasts to genuinely inform of future events by mimicking the characteristics of the processes involved. Forecast value relates, instead, to the benefits from using forecasts in a decision-making process, such as participation in the electricity market.

3.1. Forecast quality

Evaluating and comparing generative models remains a challenging task. Several measures have been introduced with the emergence of new models, particularly in the field of computer vision. However, there is no consensus or guidelines as to which metric best captures the strengths and limitations of models. Generative models need to be evaluated directly to the application they are intended for [57]. Indeed, good performance to one criterion does not imply good performance to the other criteria. Several studies propose metrics and make attempts to determine the pros and cons. We selected two that provide helpful information: (1) 24 quantitative and five qualitative measures for evaluating generative models are reviewed and compared by Borji [58] with a particular emphasis on GAN-derived models; (2) several representative sample-based evaluation metrics for GANs are investigated by Xu et al. [59] where the kernel Maximum Mean Discrepancy (MMD) and the 1-Nearest-Neighbor (1-NN) two-sample test seem to satisfy most of the desirable properties. The key message is to combine several complementary metrics to assess the generative models. Some of the metrics proposed are related to image generation and cannot directly be transposed to energy forecasting.

Therefore, we used eight complementary quality metrics to conduct a relevant quality analysis inspired by the energy forecasting and computer vision fields. They can be divided into four groups: (1) the *univariate* metrics composed of the continuous ranked probability score, the quantile score, and the reliability diagram. They can only assess the quality of the scenarios to their marginals; (2) the *multivariate* metrics are composed of the energy and the variogram scores. They can directly assess multivariate scenarios; (3) the *specific* metrics composed of a classifier-based metric and the correlation matrix between scenarios for a given context; (4) the Diebold and Mariano test statistical test. The basics of these metrics are provided in the following, and Appendix C presents the mathematical definitions and the details of implementation.

Univariate metrics

The continuous ranked probability score (CRPS) [60] is a univariate scoring rule that penalizes the lack of resolution of the predictive distributions as well as biased forecasts. It is negatively oriented, *i.e.*, the lower, the better, and for deterministic forecasts, it turns out to be the mean absolute error (MAE). The CRPS is used to compare the skill of predictive marginals for each component of the random variable of interest. In our case, for the twenty-four time periods of the day.

It allows to quantitatively assess the performance of the generative methods similar to the MAE when considering point forecasts.

The quantile score (QS), also known as the pinball loss score, is complementary to the CRPS as it permits obtaining detailed information about the forecast quality at specific probability levels, *i.e.*, over-forecasting or under-forecasting, and particularly those related to the tails of the predictive distribution [61]. It is negatively oriented and assigns asymmetric weights to negative and positive errors for each quantile.

Finally, the reliability diagram is a visual verification used to evaluate the reliability of the quantiles derived from the scenarios. Quantile forecasts are reliable if their nominal proportions are equal to the proportions of the observed value.

Multivariate metrics

The energy score (ES) is the most commonly used scoring rule when a finite number of trajectories represents distributions. It is a multivariate generalization of the CRPS and has been formulated and introduced by Gneiting and Raftery [60]. The ES is proper and negatively oriented, *i.e.*, a lower score represents a better forecast. The ES is used as a multivariate scoring rule by Golestaneh et al. [62] to investigate and analyze the spatio-temporal dependency of PV generations. They emphasize the ES pros and cons. It is capable of evaluating forecasts relying on marginals with correct variances but biased means. Unfortunately, its ability to detect incorrectly specified correlations between the components of the multivariate quantity is somewhat limited. The ES is selected as a multivariate scoring rule in this study to quantitatively assess the performance of the generative methods similar to the mean absolute error when considering point forecasts.

An alternative class of proper scoring rules based on the geostatistical concept of variograms is proposed by Scheuerer and Hamill [63]. The sensitivity of these variogram-based scoring rules to incorrectly predicted means, variances, and correlations is studied. The results indicate that these scores are shown to be distinctly more discriminative to the correlation structure. Thus, in contrast to the Energy score, the Variogram score captures correlations between multivariate components.

Specific metrics

A conditional classifier-based scoring rule is designed by implementing an Extra-Trees classifier [64] to discriminate true from generated samples. The receiver operating characteristic (ROC) curves are computed for each generative model on the testing set. The best generative model should achieve an area under the ROC curve (AUC) of 0.5, *i.e.*, each sample is equally likely to be predicted as true or false, meaning the classifier is unable to discriminate generated scenarios from the actual observations. Note: ROC curve is the relationship between True Positive Rate and False Positive Rate given by different thresholds. AUC ROC is the area under the ROC curve, and it is the metric used to measure how well the model can distinguish two classes. We recommend the article of Fawcett [65] that is designed both as a tutorial introduction to ROC graphs and as a practical guide for using them in research.

The second specific metric consists of computing the correlation matrix between the scenarios generated for given weather forecasts. Formally, let $\{\hat{\mathbf{x}}^i\}_{i=1}^{M}$ be the set of M scenarios generated for a given day of the testing set. It is a matrix ($M \times 24$) where each row is a scenario. Then, the Pearson's correlation coefficients are computed into a correlation matrix (24×24). This metric indicates the variety of scenario shapes.

Statistical testing

Using relevant metrics to assess the forecast quality is essential. However, it is also necessary to analyze whether any difference in accuracy is statistically significant. Indeed, when different models have almost identical values in the selected error measures, it is difficult to draw statistically significant conclusions on the outperformance of the forecasts of one model by those of another. The Diebold–Mariano (DM) test [66] is probably the most commonly used statistical testing tool to evaluate the significance of differences in forecasting accuracy. It is model-free, *i.e.*, it compares the forecasts of models, and not models themselves. The DM test is used in this study to assess the CRPS, QS, ES, and VS metrics. The CRPS and QS are univariate scores, and a value of CRPS and QS is computed per marginal (time period of the day). Therefore, the multivariate variant of the DM test is implemented following Ziel and Weron [67], where only one statistic for each pair of models is computed based on the 24-dimensional vector of errors for each day.

3.2. Forecast value

A model that yields lower errors in terms of forecast quality may not always point to a more effective model for forecast practitioners [5]. To this end, similarly to Toubeau et al. [19], the forecast value is assessed by considering the day-ahead market scheduling of electricity aggregators, such as energy retailers or generation companies. The energy retailer aims to balance its portfolio on an hourly basis to avoid financial penalties in case of imbalance by exchanging the surplus or deficit of energy in the day-ahead electricity market. The energy retailer may have a battery energy storage system (BESS) to manage its portfolio and minimize imports from the main grid when day-ahead prices are prohibitive.

Let e_t [MWh] be the net energy retailer position on the day-ahead market during the tth hour of the day, modeled as a first stage variable. Let y_t [MWh] be the realized net energy retailer position during the tth hour of the day, which is modeled as a second stage variable due to the stochastic processes of the PV generation, wind generation, and load. Let π_t [€/MWh] the clearing price in the spot day-ahead market for the tth hour of the day, q_t ex-post settlement price for negative imbalance $y_t < e_t$, and λ_t ex-post settlement price for positive imbalance $y_t > e_t$. The energy retailer is assumed to be a price taker in the day-ahead market. It is motivated by the individual energy retailer capacity being negligible relative to the whole market. The forward settlement price π_t is assumed to be fixed and known. As imbalance prices tend to exhibit volatility and are difficult to forecast, they are modeled as random variables, with expectations denoted by $\bar{q}_t = \mathbb{E}[q_t]$ and $\bar{\lambda}_t = \mathbb{E}[\lambda_t]$. They are assumed to be independent random variables from the energy retailer portfolio.

A stochastic planner with a linear programming formulation and linear constraints is implemented using a scenario-based approach. The planner computes the day-ahead bids e_t that cannot be modified in the future when the uncertainty is resolved. The second stage corresponds to the dispatch decisions $y_{t,\omega}$ in scenario ω that aims at avoiding portfolio imbalances modeled by a cost function f^c. The second-stage decisions are therefore scenario-dependent and can be adjusted according to the realization of the stochastic parameters. The stochastic planner objective to maximize is

$$J_S = \mathbb{E}\left[\sum_{t\in\mathcal{T}} \pi_t e_t + f^c(e_t, y_{t,\omega})\right], \tag{9}$$

where the expectation is taken to the random variables, the PV generation, wind generation, and load. Using a scenario-based approach, (9) is approximated by

$$J_S \approx \sum_{\omega\in\Omega} \alpha_\omega \sum_{t\in\mathcal{T}} \left[\pi_t e_t + f^c(e_t, y_{t,\omega})\right], \tag{10}$$

with α_ω the probability of scenario $\omega \in \Omega$, and $\sum_{\omega\in\Omega} \alpha_\omega = 1$. The optimization problem is detailed in Appendix D.

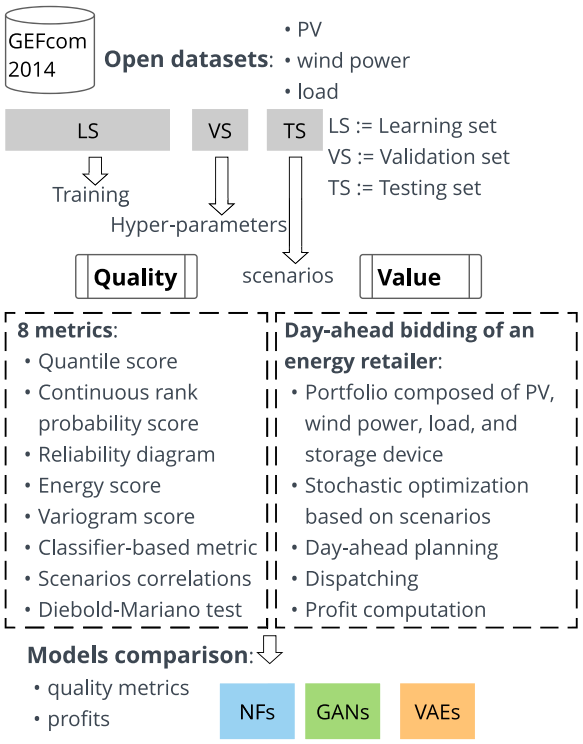


Fig. 6. Methodology to assess both the quality and value of the GAN, VAE, and NF models implemented in this study.
The PV, wind power, and load datasets of the open-access Global Energy Forecasting Competition 2014 are divided into three parts: learning, validation, and testing sets. The learning set is used to train the models, the validation set to select the optimal hyper-parameters, and the testing set to conduct the numerical experiments. The quality and value of the models are assessed by using the scenarios generated on the testing set. The quality evaluation consists of eight complementary metrics, and the value assessment is performed by using the simple and easily reproducible case study of the day-ahead bidding of an energy retailer. The energy retailer portfolio is composed of PV, wind power generation, load, and a storage system device. The retailer bids on the day-ahead market by computing a planning based on stochastic optimization. The dispatch is computed by using the observations of the PV generation, wind power, and load. Then, the profits are evaluated and compared.

4. Numerical results

The quality and value evaluations of the models are conducted on the load, wind, and PV tracks of the open-access GEFCom 2014 dataset [45], composed of one, ten, and three zones, respectively. Fig. 6 depicts the methodology to assess both the quality and value of the GAN, VAE, and NF models implemented in this study.

4.1. Implementation details

By appropriate normalization, we standardize the weather forecasts to have a zero mean and unit variance. Table 2 provides a summary of the implementation details described in what follows. For the sake of proper model training and evaluation, the dataset is divided into three parts per track considered: learning, validation, and testing sets. The learning set (LS) is used to fit the models, the validation set (VS) to select the optimal hyper-parameters, and the testing set (TS) to assess the forecast quality and value. The number of samples (#), expressed in days, of the VS and TS is $50 \cdot n_z$, with n_z the number of zones of the track considered. The 50 days are selected randomly from the dataset, and the learning set is composed of the remaining part with $D \cdot n_z$ samples, where D is provided for each track in Table 2. The NF, VAE, and GAN use the weather forecasts as inputs to generate on a day-ahead basis M scenarios $\hat{\mathbf{x}} \in \mathbb{R}^T$. The hyper-parameters values used for the experiments are provided in Appendix B.4.

Table 2
Dataset and implementation details.

	Wind	PV	Load
T periods	24	16	24
n_z zones	10	3	–
n_f features	10	5	25
$\mathbf{c}_d$ dimension	$n_f \cdot T + n_z$	$n_f \cdot T + n_z$	$n_f \cdot T$
# LS (days)	$631 \cdot n_z$	$720 \cdot n_z$	1999
# VS/TS (days)	$50 \cdot n_z$	$50 \cdot n_z$	50

Each dataset is divided into three parts: learning, validation, and testing sets. The number of samples (#) is expressed in days and is set to 50 days for the validation and testing sets. T is the number of periods per day considered, n_z the number of zones of the dataset, n_f the number of weather variables used, and $\mathbf{c}_d$ is the dimension of the conditional vector for a given day that includes the weather forecasts and the one hot-encoding variables when there are several zones. Note: the days of the learning, validation, and testing sets are selected randomly.

Wind track

The zonal $\mathbf{u}^{10}$, $\mathbf{u}^{100}$ and meridional $\mathbf{v}^{10}$, $\mathbf{v}^{100}$ wind components at 10 and 100 meters are selected, and six features are derived following the formulas provided by Landry et al. [68] to compute the wind speed $\mathbf{ws}^{10}$, $\mathbf{ws}^{100}$, energy $\mathbf{we}^{10}$, $\mathbf{we}^{100}$ and direction $\mathbf{wd}^{10}$, $\mathbf{wd}^{100}$ at 10 and 100 meters

$$\mathbf{ws} = \sqrt{\mathbf{u} + \mathbf{v}}, \tag{11a}$$

$$\mathbf{we} = \frac{1}{2}\mathbf{ws}^3, \tag{11b}$$

$$\mathbf{wd} = \frac{180}{\pi} \arctan(\mathbf{u}, \mathbf{v}). \tag{11c}$$

For each generative model, the wind zone is taken into account with one hot-encoding variable $Z_1, \dots, Z_{10}$, and the wind feature input vector for a given day d is

$$\begin{aligned}\mathbf{c}_d^{\text{wind}} = [&\mathbf{u}_d^{10}, \mathbf{u}_d^{100}, \mathbf{v}_d^{10}, \mathbf{v}_d^{100}, \mathbf{ws}_d^{10}, \mathbf{ws}_d^{100},\\ &\mathbf{we}_d^{10}, \mathbf{we}_d^{100}, \mathbf{wd}_d^{10}, \mathbf{wd}_d^{100}, Z_1, \dots, Z_{10}],\end{aligned} \tag{12}$$

of dimension $n_f \cdot T + n_z = 10 \cdot 24 + 10$.

PV track

The solar irradiation $\mathbf{I}$, the air temperature $\mathbf{T}$, and the relative humidity $\mathbf{rh}$ are selected, and two features are derived by computing $\mathbf{I}^2$ and $\mathbf{IT}$. For each generative model, the PV zone is taken into account with one hot-encoding variable Z_1, Z_2, Z_3, and the PV feature input vector for a given day d is

$$\mathbf{c}_d^{\text{PV}} = [\mathbf{I}_d, \mathbf{T}_d, \mathbf{rh}_d, \mathbf{I}_d^2, \mathbf{IT}_d, Z_1, Z_2, Z_3], \tag{13}$$

of dimension $n_f \cdot T + n_z$. For practical reasons, the periods where the PV generation is always 0, across all zones and days, are removed, and the final dimension of the input feature vector is $n_f \cdot T + n_z = 5 \cdot 16 + 3$.

Load track

The 25 weather station temperature $\mathbf{w}_1, \dots, \mathbf{w}_{25}$ forecasts are used. There is only one zone, and the load feature input vector for a given day d is

$$\mathbf{c}_d^{\text{load}} = [\mathbf{w}_1, \dots, \mathbf{w}_{25}], \tag{14}$$

of dimension $n_f \cdot T = 25 \cdot 24$.

The number of samples (#), expressed in days, of the VS and TS is $50 \cdot n_z$, with n_z the number of zones of the track considered. The 50 days are selected randomly from the dataset, and the learning set is composed of the remaining part with $D \cdot n_z$ samples, where D is provided for each track.

4.2. Quality results

A thorough comparison of the models is conducted on the wind track, and Appendix E provides the Figures of the other tracks for the sake of clarity. Note: the model ranking slightly differs depending on the track.

Wind track

In addition to the generative models, a naive approach is designed (RAND), where the scenarios of the learning, validation, and testing sets are sampled randomly from the learning, validation, and testing sets, respectively. Intuitively, it assumes that past observations are repeated, and these scenarios are realistic but may not be compatible with the context. Each model generates a set of 100 scenarios for each day of the testing set, and the scores are computed following the mathematical definitions provided in Appendix C. Fig. 7 compares the QS, reliability diagram, and CRPS of the wind (markers), PV (plain), and load (dashed) tracks. Overall, for the wind track in terms of CRPS, QS, and reliability diagrams, the VAE achieves slightly better scores, followed by the NF and the GAN. The ES and VS multivariate scores confirm this trend with 54.82 and 18.87 for the VAE *vs* 56.71 and 18.54 for the NF, respectively.

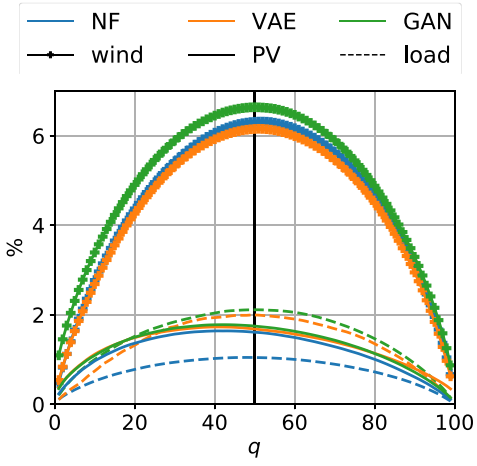
Fig. 8 provides the results of the DM tests for these metrics. The heat map indicates the range of the p-values. The closer they are to zero, dark green, the more significant the difference between the scores of two models for a given metric. The statistical threshold is set to 5%, but the scale color is capped at 10% for a better exposition of the relevant results. For instance, when considering the DM test for the RAND CRPS, all the columns of the RAND row are in dark green, indicating that the RAND scenarios are always significantly outperformed by the other models. These DM tests confirm that the VAE outperforms the NF for the wind track considering these metrics. Then, the NF is only outperformed by the VAE and the GAN by both the VAE and NF. These results are consistent with the classifier-based metric depicted in Fig. 9, where the VAE is the best to mislead the classifier, followed by the NF and GAN.

The left part of Fig. 10 provides 50 scenarios, (a) NF, (c) GAN, and (e) VAE, generated for a given day selected randomly from the testing set. Notice how the shape of the NF's scenarios differs significantly from the GAN and VAE as they tend to be more variable with no identifiable trend. In contrast, the VAE and GAN scenarios seem to differ mainly in nominal power but have similar shapes. This behavior is even more pronounced for the GAN, where the scenarios rarely crossed over time periods. For instance, there is a gap in generation around periods 17 and 18 where all the GAN's scenarios follow this trend. These observations are confirmed by computing the corresponding time correlation matrices, depicted by the right part of Fig. 10 demonstrating there is no correlation between NF's scenarios. On the contrary, the VAE and GAN correlation matrices tend to be similar with a time correlation of the scenarios over a few periods, with more correlated periods when considering the GAN. This difference in the scenario's shape is striking and not necessarily captured by metrics such as the CRPS, QS, or even the classifier-based metric and is also observed on the PV and load tracks, as explained in the next paragraph.

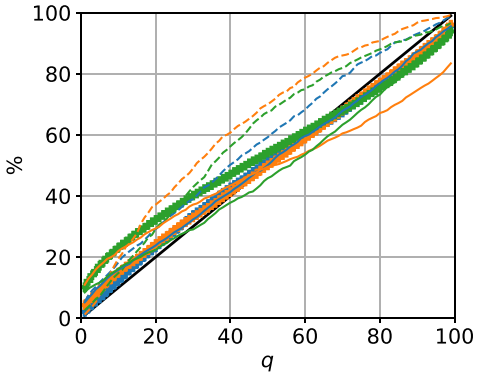
All tracks

Table 3 provides the averaged quality scores for all the datasets considered: wind, PV, and load. The CRPS is averaged over the 24 time periods $\overline{\text{CRPS}}$. The QS over the 99 percentiles $\overline{\text{QS}}$. The MAE-r is the mean absolute error between the reliability curve and the diagonal, and $\overline{\text{AUC}}$ is the mean of the 50 AUC. Overall, for the PV and load tracks in CRPS, QS, reliability diagrams, AUC, ES, and VS, the NF outperforms the VAE and GAN and is slightly outperformed by the VAE on the wind track. On the load track, the VAE outperforms the GAN. However, the VAE and GAN achieved similar results on the PV track, and the GAN performed better in terms of ES and VS. These results are confirmed by the DM tests depicted in Fig. E.16. The classifier-based metric results for both the load and PV tracks, provided by Fig. E.17, confirm this trend where the NF is the best to trick the classifier followed by the VAE and GAN.

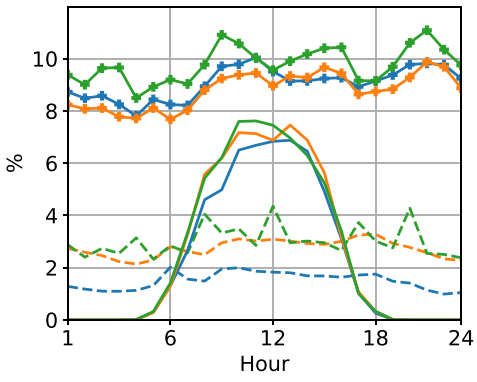
Similar to the wind track, the shape of the scenarios differs significantly between the NF and the other models for both the load and PV tracks as indicated by the left part of Figs. E.18 and E.19, and the corresponding correlation matrices provided by the right part of Figs. E.18 and E.19. Note: the load track scenarios are highly correlated for both the VAE and GAN. Finally, Fig. E.20 provides the average of the correlation matrices over all days of the testing set for each dataset. The trend depicted above is confirmed. This difference between



(a) Quantile score.



(b) Reliability diagram.



(c) CRPS per marginal.

Fig. 7. Quality common metrics comparison on the wind (markers), PV (plain), and load (dashed) tracks.
Quantile score (a): the lower and the more symmetrical the better. Note: the quantile score has been averaged over the marginals (the 24 time periods of the day). Reliability diagram (b): the closer to the diagonal, the better. Continuous ranked probability score per marginal (c): the lower, the better.
NF outperforms the VAE and GAN for both the PV and load tracks and is slightly outperformed by the VAE on the wind track. Note: all models tend to have more difficulties forecasting the wind power that seems less predictable than the PV generation or the load.

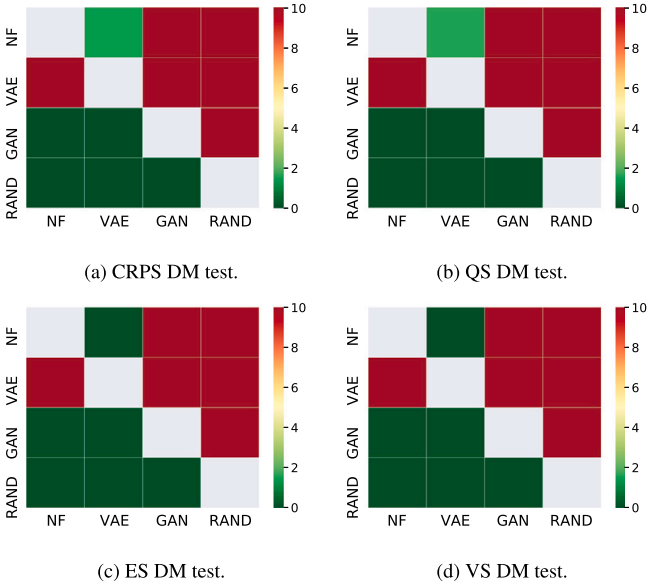


Fig. 8. Wind track Diebold–Mariano tests of the CRPS, QS, ES, and VS metrics.
The Diebold–Mariano tests of the continuous ranked probability, quantile, energy, and variogram scores confirm that the VAE outperforms the NF on the wind track for these metrics. The NF is only outperformed by the VAE and the GAN by both the VAE and NF.
The heat map indicates the range of the p-values. The closer they are to zero, dark green, the more significant the difference between the scores of two models for a given metric. The statistical threshold is set to 5% but the scale color is capped at 10% for a better exposition of the relevant results.

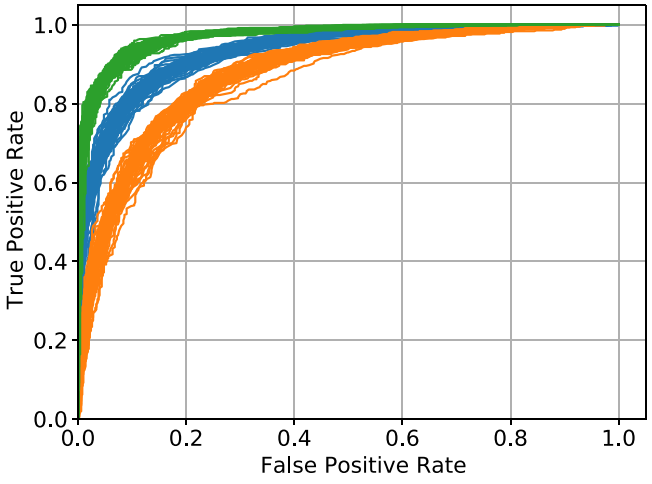


Fig. 9. Wind track classifier-based metric.
The VAE (orange) is the best to mislead the classifier, followed by the NF (blue) and GAN (green). Note: there are 50 ROC curves depicted for each model, each corresponding to a scenario generated used as input of the classifier. It allows taking into account the variability of the scenarios to avoid having results dependent on a particular scenario.

the NF and the other generative model may be explicated by the design of the methods. The NF explicitly learns the probability density function (PDF) of the multi-dimensional random variable considered. Thus, the NF scenarios are generated according to the learned PDF producing multiple shapes of scenarios. In contrast, the generator of the GAN is trained to fool the discriminator, and it may find a shape particularly efficient leading to a set of similar scenarios. Concerning the VAE, it is less obvious. However, by design, the decoder is trained to generate scenarios from the latent space assumed to follow a Gaussian distribution that may lead to less variability.

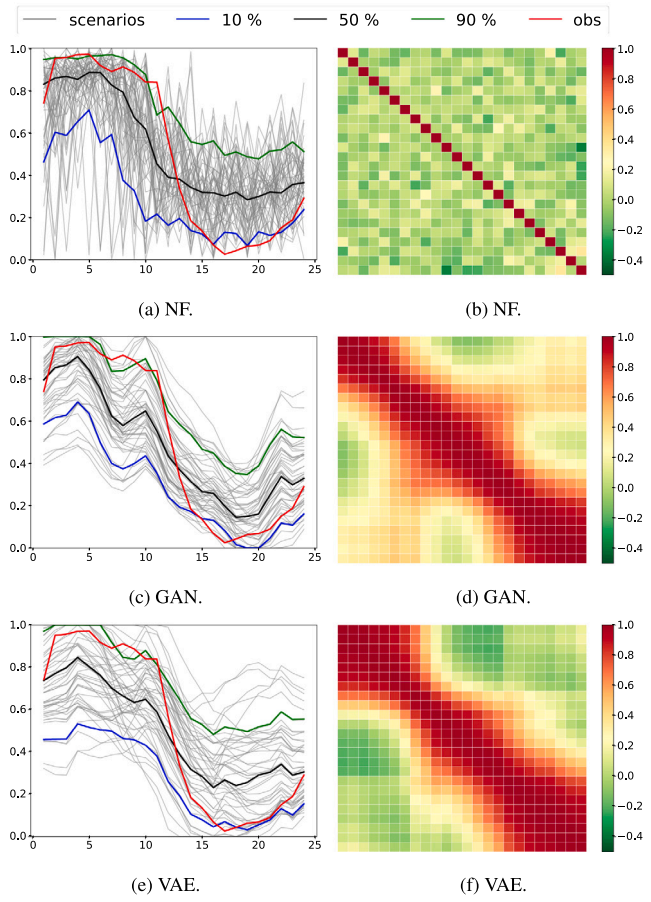


Fig. 10. Wind power scenarios shape comparison and analysis.
Left part (a) NF, (c) GAN, and (e) VAE: 50 wind power scenarios (gray) of a randomly selected day of the testing set along with the 10% (blue), 50% (black), and 90% (green) quantiles, and the observations (red). Right part (b) NF, (d) GAN, and (f) VAE: the corresponding Pearson time correlation matrices of these scenarios with the time periods as rows and columns. The NF tends to exhibit no time correlation between scenarios. In contrast, the VAE and GAN tend to be partially time-correlated over a few periods.

4.3. Value results

The energy retailer portfolio comprises wind power, PV generation, load, and a battery energy storage device. The 50 days of the testing set are used and combined with the 30 possible PV and wind generation zones (three PV zones and ten wind farms), resulting in 1500 independent simulated days. A two-step approach is employed to evaluate the forecast value:

- First, for each generative model and the 1500 days simulated, the two-stage stochastic planner computes the day-ahead bids of the energy retailer portfolio using the PV, wind power, and load scenarios. After solving the optimization, the day-ahead decisions are recorded.
- Then, a real-time dispatch is carried out using the PV, wind power, and load observations, with the day-ahead decisions as parameters.

This two-step methodology is applied to evaluate the three generative models, namely the NF, GAN, and VAE. Fig. 11 illustrates an arbitrary random day of the testing set with the first zone for both the PV and wind. π_t [€/MWh] is the day-ahead prices on *February* 6, 2020 of the Belgian day-ahead market used for the 1500 days simulated. The negative $\bar{q}_t$ and positive $\bar{\lambda}_t$ imbalance prices are set to $2 \times \pi_t$, $\forall t \in \mathcal{T}$. The retailer aims to balance the net power, red curve in

Table 3
Averaged quality scores per dataset.

		NF	VAE	GAN	RAND
Wind	$\overline{\text{CRPS}}$	9.07	**8.80**	9.79	16.92
	$\overline{\text{QS}}$	4.58	**4.45**	4.95	8.55
	MAE-r	2.83	**2.67**	6.82	1.01
	$\overline{\text{AUC}}$	0.935	**0.877**	0.972	0.918
	ES	56.71	**54.82**	60.52	96.15
	VS	18.54	**17.87**	19.87	23.21
PV	$\overline{\text{CRPS}}$	**2.35**	2.60	2.61	4.92
	$\overline{\text{QS}}$	**1.19**	1.31	1.32	2.48
	MAE-r	**2.66**	9.04	4.94	3.94
	$\overline{\text{AUC}}$	**0.950**	0.969	0.997	0.947
	ES	**23.08**	24.65	24.15	41.53
	VS	**4.68**	5.02	4.88	13.40
Load	$\overline{\text{CRPS}}$	**1.51**	2.74	3.01	6.74
	$\overline{\text{QS}}$	**0.76**	1.39	1.52	3.40
	MAE-r	**7.70**	13.97	9.99	0.88
	$\overline{\text{AUC}}$	**0.823**	0.847	0.999	0.944
	ES	**9.17**	15.11	17.96	38.08
	VS	**1.63**	1.66	3.81	7.28

The best performing deep learning generative model for each track is written in bold. The CRPS, QS, MAE-r, and ES are expressed in %. Overall, for both the PV and load tracks, the NF outperforms the VAE and GAN and is slightly outperformed by the VAE on the wind track.

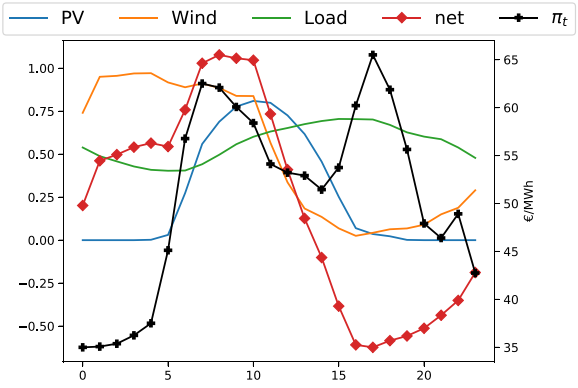


Fig. 11. Energy retailer case study: illustration of the observations on a random day of the testing set.
The energy retailer portfolio comprises PV generation, wind power, load, and a storage device. The PV, wind power, and load scenarios from the testing set are used as inputs of the stochastic day-ahead planner to compute the optimal bids. The net is the power balance of the energy retailer portfolio. The day-ahead prices π_t are obtained from the Belgian day-ahead market on *February* 6, 2020.

Fig. 11, by importing/exporting from/to the main grid. Usually, the net is positive (negative) at noon (evening) when the PV generation is maximal (minimal), and the load is minimal (maximal). As the day-ahead spot price is often maximal during the evening load peak, the retailer seeks to save power during the day by charging the battery to decrease the import during the evening. Therefore, the more accurate the PV, wind generation, and load scenarios are, the better is the day-ahead planning.

The battery minimum $s^{\min}$ and maximum $s^{\max}$ capacities are 0 and 1, respectively. It is assumed to be capable of fully (dis)charging in two hours with $y^{\text{dis}}_{\max} = y^{\text{cha}}_{\max} = s^{\max}/2$, and the (dis)charging efficiencies are $\eta^{\text{dis}} = \eta^{\text{cha}} = 95$ %. Each simulation day is independent with a fully discharged battery at the first and last period of each day $s^{\text{ini}} = s^{\text{end}} = 0$. The 1500 stochastic optimization problems are solved with 50 PV, wind generation, and load scenarios. The python Gurobi library is used to implement the algorithms in Python 3.7, and Gurobi[2] 9.0.2 is used to solve the optimization problems. Numerical experiments are performed on an Intel Core i7-8700 3.20 GHz based computer with 12 threads and 32 GB of RAM running on Ubuntu 18.04 LTS.

[2] https://www.gurobi.com/

Table 4
Total net profit (k€) and cumulative ranking (%).

	NF	VAE	GAN
Net profit (k€)	**107**	97	93
1 (%)	**39.0**	31.8	29.2
1 & 2 (%)	**69.6**	68.3	62.1
1 & 2 & 3 (%)	100	100	100

The stochastic planner using the NF PV, wind power, and load scenarios achieved the highest net profit with 107 k€, ranked first 39.0%, second 30.6%, and third 30.4% over 1500 days of simulation. In comparison, the second-best model, the VAE, achieved a net profit of 97 k€, ranked first 31.8%, second 36,5%, and third 31.7%.

The net profit, that is, the profit minus penalty, is computed for the 1500 days of the simulation and aggregated in the first row of Table 4. The ranking of each model is computed for the 1500 days, and the cumulative ranking is expressed in terms of percentage in Table 4. NF outperformed both the GAN and VAE with a total net profit of 107 k€. There is still room for improvement as the oracle, which has perfect future knowledge, achieved 300 k€. NF ranked first 39.0% during the 1500 simulation days and achieved the first and second ranks 69.6%. Overall, in terms of forecast value, the NF outperforms the VAE and GAN. However, this case study is "simple," and stochastic optimization relies mainly on the quality of the average of the scenarios. Therefore, one may consider taking advantage of the particularities of a specific method by considering more advanced case studies. In particular, the specificity of the NFs to provide direct access to the probability density function may be of great interest in specific applications. It is left for future investigations as more advanced case studies would prevent a fair comparison between models.

4.4. Results summary

Table 5 summarizes the main results of this study by comparing the VAE, GAN, and NF implemented through easily comparable star ratings. The rating for each criterion is determined using the following rules - 1 star: third rank, 2 stars: second rank, and 3 stars: first rank. Specifically, training speed is assessed based on reported total training times for each dataset: PV generation, wind power, and load; sample speed is based on reported total generating times for each dataset; quality is evaluated with the metrics considered; value is based on the case study of the day-ahead bidding of the energy retailer; the hyper-parameters search is assessed by the number of configurations tested before reaching satisfactory and stable results over the validation set; the hyper-parameters sensitivity is evaluated by the impact on the quality metric of deviations from the optimal the hyper-parameter values found during the hyper-parameter search; the implementation-friendly criterion is appraised regarding the complexity of the technique and the amount of knowledge required to implement it.

5. Conclusion

This paper proposes a fair and thorough comparison, both in terms of quality and value, of normalizing flows with the state-of-the-art deep learning generative models: generative adversarial networks and variational autoencoders. The experiments adopt the open data of the Global Energy Forecasting Competition 2014, where the generative models use the conditional information to compute improved weather-based PV power, wind power, and load scenarios. The results demonstrate that normalizing flows can challenge generative adversarial networks and variational autoencoders. Overall, they are more accurate in quality and value and can be used effectively by non-expert deep learning practitioners. In addition, normalizing flows have several advantages over more traditional deep learning approaches that should motivate their introduction into power system applications:

Table 5
Comparison between the deep generative models.

Criteria	VAE	GAN	NF
Train speed	★★★	★★☆	★☆☆
Sample speed	★★★	★★☆	★☆☆
Quality	★★☆	★☆☆	★★★
Value	★★☆	★☆☆	★★★
Hp search	★★☆	★☆☆	★★★
Hp sensibility	★★☆	★☆☆	★★★
Implementation	★★★	★★☆	★☆☆

The rating for each criterion is determined using the following rules - 1 star: third rank, 2 stars: second rank, and 3 stars: first rank. Train speed: training computation time; Sample speed: scenario generation computation time; Quality: forecast quality based on the eight complementary metrics considered; Value: forecast value based on the day-ahead energy retailer case study; Hp search: assess the difficulty to identify relevant hyper-parameters; Hp sensibility: assess the sensitivity of the model to a given set of hyper-parameters (the more stars, the more robust to hyper-parameter modifications); Implementation: assess the difficulty to implement the model (the more stars, the more implementation-friendly). Note: the justifications are provided in Appendix A.2.

(i) Normalizing flows directly learn the stochastic multivariate distribution of the underlying process by maximizing the likelihood. Therefore, in contrast to variational autoencoders and generative adversarial networks, they provide access to the exact likelihood of the model's parameters, hence offering a sound and direct way to optimize the network parameters. It may open a new range of advanced applications benefiting from this advantage. For instance, to transfer scenarios from one location to another based on the knowledge of the probability density function. A second application is the importance sampling for stochastic optimization based on a scenario approach. Indeed, normalizing flows provide for each generated scenario its likelihood making it possible to filter relevant scenarios used in stochastic optimization.

(ii) In our opinion, normalizing flows are easier to use by non-expert deep learning practitioners once the libraries are available, as they are more reliable and robust in terms of hyper-parameters selection. Generative adversarial networks and variational autoencoders are particularly sensitive to the latent space dimension, the structure of the neural networks, the learning rate, etc. Generative adversarial networks convergence, by design, is unstable, and for a given set of hyper-parameters, the scenario's quality may differ completely. In contrast, it was easier to retrieve relevant normalizing flows hyper-parameters by manually testing a few sets of values that led to satisfying training convergence and quality results.

Nevertheless, their usage as a base component of the machine learning toolbox is still limited compared to generative adversarial networks or variational autoencoders.

CRediT authorship contribution statement

Jonathan Dumas: Conceptualization, Methodology, Software, Validation, Investigation, Formal analysis, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration. **Antoine Wehenkel:** Conceptualization, Methodology, Writing – review & editing, Supervision, Project administration. **Damien Lanaspeze:** Conceptualization, Methodology, Software, Investigation. **Bertrand Cornélusse:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Antonio Sutera:** Conceptualization, Methodology, Software, Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to acknowledge the authors and contributors of the Scikit-Learn [69], Pytorch [70], and Weights & Biases [71] Python libraries. In addition, the authors would like to thank the editor and the reviewers for the comments that helped improve the paper. Antoine Wehenkel is a recipient of an F.R.S.- FNRS fellowship and acknowledges the financial support of the FNRS (Belgium). Antonio Sutera is supported via the Energy Transition Funds project EPOC 2030–2050 organized by the FPS economy, S.M.E.s, Self-employed and Energy.

Appendix A. Additional arguments

A.1. *Table 1 justifications*

Wang et al. [35] use a Wasserstein GAN with gradient penalty to model both the uncertainties and the variations of the load. Specifically, point forecasting is first conducted, and the corresponding residuals are calculated. Then, the GAN generates residual scenarios conditional on the day type, temperatures, and historical loads. The GAN model is compared with the same version without gradient penalty and two quantile regression models: random forest and gradient boosting regression tree. The quality evaluation is conducted on open load datasets from the Independent System Operator-New England[3] with five metrics: (1) the continuous ranked probability score; (2) the quantile score; (3) the Winkler score; (4) reliability diagrams; (5) Q-Q plots. Note: the forecast value is not assessed.

Qi et al. [37] propose a concentrating solar power (CSP) configuration method to determine the CSP capacity in multi-energy power systems. The configuration model considers the uncertainty by scenario analysis. The scenarios are produced by a β VAE that is an improved version of the original VAE. The weather forecasts are not considered, and the VAE is trained only by using historical observations. The quality evaluation is conducted on two wind farms and six PV plants using three metrics: (1) the leave-one-out accuracy of the 1-nearest neighbor classifier; (2) the comparison of the frequency distributions of the real data and the generated scenarios; (3) the comparison of the spatial and temporal correlations of the real data and the scenarios by computing Pearson correlation coefficients. The value is assessed by considering the case study of the CSP configuration model, where the β VAE is used to generate PV, wind power, and load scenarios. However, the VAE is not compared to another generative model for both the quality and value evaluations. Note: the dataset does not seem to be in open-access. Finally, the value evaluation case study is not trivial due to the mathematical formulation that requires a certain level of knowledge of the system. Thus, the replicability criterion is partially satisfied.

Ge et al. [38] compared NFs to VAEs and GANs for the generation of daily load profiles. The models do not take into account weather forecasts but only historical observations. However, an example is given to illustrate the principle of generating conditional daily load profiles by using three groups: light load, medium load, and heavy load. The quality evaluation uses five indicators. Four to assess the temporal correlation: (1) probability density function; (2) autocorrelation function; (3) load duration curve; (4) a wave rate is defined to evaluate the volatility of the daily load profile. Furthermore, one additional for the spatial correlation: (5) Pearson correlation coefficient is used to measure the spatial correlation among multiple daily load profiles. The simulations use the open-access London smart meter and Spanish transmission service operator datasets of Kaggle. The forecast value is not assessed.

[3] https://www.iso-ne.com/

A.2. Table 5 justifications

The VAE is the fastest to train, with a recorded computation time of 7 s on average per dataset. The training time of the GAN is approximately three times longer, with an average computation time of 20 s per dataset. Finally, the NF is the slowest, with an average training time of 4 min. This ranking is preserved with the VAE the fastest concerning the sample speed, followed by the GAN and NF models. The VAE and the GAN generate the samples over the testing sets, 5000 in total, in less than a second. However, the NF considered takes a few minutes. In contrast, the affine autoregressive version of the NF is much faster to train and generate samples. Note: even a training time of a few hours is compatible with day-ahead planning applications. In addition, once the model is trained, it is not necessarily required to retrain it every day.

The quality and value assessments have already been discussed in Section 4. Overall, the NF outperforms both the VAE and GAN models.

Concerning the hyper-parameters search and sensibility, the NF tends to be the most straightforward model to calibrate. Compared with the VAE and GAN, we found relevant hyper-parameter values by testing only a few combinations. In addition, the NF is robust to hyper-parameter modifications. In contrast, the GAN is the most sensitive. Variations of the hyper-parameters may result in very poor scenarios both in terms of quality and shape. Even for a fixed set of hyper-parameters values, two separate training may not converge towards the same results illustrating the GAN training instabilities. The VAE is more accessible to train than the GAN but is also sensitive to hyper-parameters values. However, it is less evident than the GAN.

Finally, we discuss the implementation-friendly criterion of the models. Note: this discussion is only valid for the models implemented in this study. There exist various architectures of GANs, VAEs, and NFs with simple and complex versions. In our opinion, the VAE is the effortless model to implement as the encoder and decoder are both simple feed-forward neural networks. The only difficulty lies in the reparameterization trick that should be carefully addressed. The GAN is a bit more difficult to deploy due to the gradient penalty to handle but is similar to the VAE with both the discriminator and the generator that are feed-forward neural networks. The NF is the most challenging model to implement from scratch because the UMNN-MAF approach requires an additional integrand network. An affine autoregressive NF is easier to implement. Nevertheless, it may be less capable of modeling the stochasticity of the variable of interest. However, forecasting practitioners do not necessarily have to implement generative models from scratch and can use numerous existing Python libraries.

Appendix B. Background

B.1. Normalizing flows

Normalizing flow computation

Evaluating the likelihood of a distribution modeled by a normalizing flow requires computing (2), *i.e.*, the normalizing direction, as well as its log-determinant. Increasing the number of sub-flows by K of the transformation results in only $\mathcal{O}(K)$ growth in the computational complexity as the log-determinant of J_{f_θ} can be expressed as

$$\log|\det J_{f_\theta}(\mathbf{x})| = \log\left|\prod_{k=1}^{K}\det J_{f_{k,\theta}}(\mathbf{x})\right|, \tag{B.1a}$$

$$= \sum_{k=1}^{K}\log|\det J_{f_{k,\theta}}(\mathbf{x})|. \tag{B.1b}$$

However, with no further assumption on f_θ, the computational complexity of the log-determinant is $\mathcal{O}(K\cdot T^3)$, which can be intractable for large T. Therefore, the efficiency of these operations is critical during training, where the likelihood is repeatedly computed. There are many possible implementations of NFs detailed by Papamakarios et al. [72], Kobyzev et al. [73] to address this issue.

Autoregressive flow

The Jacobian of the autoregressive transformation f_θ defined by (3) is lower triangular, and its log-absolute-determinant is

$$\log|\det J_{f_\theta}(\mathbf{x})| = \log\prod_{i=1}^{T}\left|\frac{\partial f^i}{\partial x_i}(x_i;h^i)\right|, \tag{B.2a}$$

$$= \sum_{i=1}^{T}\log\left|\frac{\partial f^i}{\partial x_i}(x_i;h^i)\right|, \tag{B.2b}$$

that is calculated in $\mathcal{O}(T)$ instead of $\mathcal{O}(T^3)$.

Affine autoregressive flow

A simple choice of transformer is the class of affine functions

$$f^i(x_i;h^i) = \alpha_i x_i + \beta_i, \tag{B.3}$$

where $f^i(\cdot;h^i):\mathbb{R}\to\mathbb{R}$ is parameterized by $h^i=\{\alpha_i,\beta_i\}$, α_i controls the scale, and β_i controls the location of the transformation. Invertibility is guaranteed if $\alpha_i\neq 0$, and this can be easily achieved by *e.g.* taking $\alpha_i=\exp(\tilde{\alpha}_i)$, where $\tilde{\alpha}_i$ is an unconstrained parameter in which case $h^i=\{\tilde{\alpha}_i,\beta_i\}$. The derivative of the transformer with respect to x_i is equal to α_i. Hence the log-absolute-determinant of the Jacobian becomes

$$\log|\det J_{f_\theta}(\mathbf{x})| = \sum_{i=1}^{T}\log|\alpha_i| = \sum_{i=1}^{T}\tilde{\alpha}_i. \tag{B.4}$$

Affine autoregressive flows are simple and computation efficient but are limited in expressiveness requiring many stacked flows to represent complex distributions. It is unknown whether affine autoregressive flows with multiple layers are universal approximators or not [72], in contrast to the UMNN autoregressive transformation implemented in this paper.

B.2. Variational autoencoders

Gradients computation

By using (6) $\mathcal{L}_{\theta,\varphi}$ is decomposed in two parts

$$\mathcal{L}_{\theta,\varphi}(\mathbf{x},\mathbf{c}) = \mathop{\mathbb{E}}_{q_\varphi(\mathbf{z}|\mathbf{x},\mathbf{c})}[\log p_\theta(\mathbf{x}|\mathbf{z},\mathbf{c})] - \mathrm{KL}[q_\varphi(\mathbf{z}|\mathbf{x},\mathbf{c})\parallel p(\mathbf{z})]. \tag{B.5}$$

$\nabla_\theta\mathcal{L}_{\theta,\varphi}$ is estimated with the usual Monte Carlo gradient estimator. However, the estimation of $\nabla_\varphi\mathcal{L}_{\theta,\varphi}$ requires the reparameterization trick proposed by Kingma and Welling [23], where the random variable $\mathbf{z}$ is re-expressed as a deterministic variable

$$\mathbf{z} = g_\varphi(\epsilon,\mathbf{x}), \tag{B.6}$$

with ϵ an auxiliary variable with independent marginal p_ϵ, and $g_\varphi(\cdot)$ some vector-valued function parameterized by φ. Then, the first right hand side of (B.5) becomes

$$\mathop{\mathbb{E}}_{q_\varphi(\mathbf{z}|\mathbf{x},\mathbf{c})}[\log p_\theta(\mathbf{x}|\mathbf{z},\mathbf{c})] = \mathop{\mathbb{E}}_{p(\epsilon)}[\log p_\theta(\mathbf{x}|g_\varphi(\epsilon,\mathbf{x}),\mathbf{c})]. \tag{B.7}$$

$\nabla_\varphi\mathcal{L}_{\theta,\varphi}$ is now estimated with Monte Carlo integration.

Conditional variational autoencoders implemented

Following Kingma and Welling [23], we implemented Gaussian multi-layered perceptrons (MLPs) for both the encoder NN_φ and decoder NN_θ. In this case, $p(\mathbf{z})$ is a centered isotropic multivariate Gaussian, $p_\theta(\mathbf{x}|\mathbf{z},\mathbf{c})$ and $q_\varphi(\mathbf{x}|\mathbf{z},\mathbf{c})$ are both multivariate Gaussian with a diagonal covariance and parameters $\boldsymbol{\mu}_\theta, \boldsymbol{\sigma}_\theta$ and $\boldsymbol{\mu}_\varphi, \boldsymbol{\sigma}_\varphi$, respectively. Note: there is no restriction on the encoder and decoder architectures, and they could as well be arbitrarily complex convolutional networks. Under these assumptions, the conditional VAE implemented is

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}), \tag{B.8a}$$

$$p_\theta(\mathbf{x}|\mathbf{z},\mathbf{c}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\theta, \boldsymbol{\sigma}_\theta^2\mathbf{I}), \tag{B.8b}$$

$$q_\varphi(\mathbf{z}|\mathbf{x},\mathbf{c}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\varphi, \boldsymbol{\sigma}_\varphi^2\mathbf{I}), \tag{B.8c}$$

$$\boldsymbol{\mu}_\theta, \log\boldsymbol{\sigma}_\theta^2 = \text{NN}_\theta(\mathbf{x},\mathbf{c}), \tag{B.8d}$$

$$\boldsymbol{\mu}_\varphi, \log\boldsymbol{\sigma}_\varphi^2 = \text{NN}_\varphi(\mathbf{z},\mathbf{c}). \tag{B.8e}$$

Then, by using the valid reparameterization trick proposed by Kingma and Welling [23]

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{B.9a}$$

$$\mathbf{z} := \boldsymbol{\mu}_\varphi + \boldsymbol{\sigma}_\varphi \boldsymbol{\epsilon}, \tag{B.9b}$$

$\mathcal{L}_{\theta,\varphi}$ is computed and differentiated without estimation using the expressions

$$\text{KL}[q_\varphi(\mathbf{z}|\mathbf{x},\mathbf{c}) \parallel p(\mathbf{z})] = -\frac{1}{2}\sum_{j=1}^{d}(1 + \log\boldsymbol{\sigma}_{\varphi,j}^2 - \boldsymbol{\mu}_{\varphi,j}^2 - \boldsymbol{\sigma}_{\varphi,j}^2), \tag{B.10a}$$

$$\mathop{\mathbb{E}}_{p(\boldsymbol{\epsilon})}[\log p_\theta(\mathbf{x}|\mathbf{z},\mathbf{c})] \approx -\frac{1}{2}\left\|\frac{\mathbf{x}-\boldsymbol{\mu}_\theta}{\boldsymbol{\sigma}_\theta}\right\|^2, \tag{B.10b}$$

with d the dimensionality of $\mathbf{z}$.

B.3. Generative adversarial network

Original generative adversarial network

The original GAN value function $V(\phi,\theta)$ proposed by Goodfellow et al. [24] is

$$V(\phi,\theta) = \underbrace{\mathop{\mathbb{E}}_{\mathbf{x}}[\log d_\phi(\mathbf{x}|\mathbf{c})] + \mathop{\mathbb{E}}_{\hat{\mathbf{x}}}[\log(1 - d_\phi(\hat{\mathbf{x}}|\mathbf{c}))]}_{:=-L_d}, \tag{B.11a}$$

$$L_g := -\mathop{\mathbb{E}}_{\hat{\mathbf{x}}}[\log(1 - d_\phi(\hat{\mathbf{x}}|\mathbf{c}))], \tag{B.11b}$$

where L_d is the cross-entropy, and L_g the probability the discriminator wrongly classifies the samples.

Wasserstein generative adversarial network

The divergences which GANs typically minimize are responsible for their training instabilities for reasons investigated by Arjovsky and Bottou [56] theoretically. Arjovsky et al. [74] proposed instead using the *Earth mover* distance, also known as the Wasserstein-1 distance

$$W_1(p,q) = \inf_{\gamma\in\Pi(p,q)} \mathbb{E}_{(x,y)\sim\gamma}[\|x-y\|], \tag{B.12}$$

where $\Pi(p,q)$ denotes the set of all joint distributions $\gamma(x,y)$ whose marginals are respectively p and q, $\gamma(x,y)$ indicates how much mass must be transported from x to y in order to transform the distribution p into q, $\|\cdot\|$ is the L1 norm, and $\|x-y\|$ represents the cost of moving a unit of mass from x to y. However, the infimum in (B.12) is intractable. Therefore, Arjovsky et al. [74] used the Kantorovich–Rubinstein duality [75] to propose the Wasserstein GAN (WGAN) by solving the min–max problem

$$\theta^\star = \arg\min_\theta \max_{\phi\in\mathcal{W}} \mathop{\mathbb{E}}_{\mathbf{x}}[d_\phi(\mathbf{x}|\mathbf{c})] - \mathop{\mathbb{E}}_{\hat{\mathbf{x}}}[d_\phi(\hat{\mathbf{x}}|\mathbf{c})], \tag{B.13}$$

where $\mathcal{W} = \{\phi : \|d_\phi(\cdot)\|_L \leq 1\}$ is the 1-Lipschitz space, and the classifier $d_\phi(\cdot) : \mathbb{R}^T \times \mathbb{R}^{|\mathbf{c}|} \to [0,1]$ is replaced by a critic function $d_\phi(\cdot) : \mathbb{R}^T \times \mathbb{R}^{|\mathbf{c}|} \to \mathbb{R}$. However, the weight clipping used to enforce d_ϕ 1-Lipschitzness can lead sometimes the WGAN to generate only poor samples or failure to converge [53]. Therefore, we implemented the WGAN-GP to tackle this issue.

Table B.6
(a) NF, (b) VAE, and (c) GAN hyper-parameters.

		Wind	PV	Load
(a)	Embedding Net	4 × 300	4 × 300	4 × 300
	Embedding size	40	40	40
	Integrand Net	3 × 40	3 × 40	3 × 40
	Weight decay	5.10^{-4}	5.10^{-4}	5.10^{-4}
	Learning rate	10^{-4}	5.10^{-4}	10^{-4}
(b)	Latent dimension	20	40	5
	E/D Net	1 × 200	2 × 200	1 × 500
	Weight decay	$10^{-3.4}$	$10^{-3.5}$	10^{-4}
	Learning rate	$10^{-3.4}$	$10^{-3.3}$	$10^{-3.9}$
(c)	Latent dimension	64	64	256
	G/D Net	2 × 256	3 × 256	2 × 1024
	Weight decay	10^{-4}	10^{-4}	10^{-4}
	Learning rate	2.10^{-4}	2.10^{-4}	2.10^{-4}

The hyper-parameters selection is performed on the validation set using the Python library Weights & Biases [71]. This library is an experiment tracking tool for machine learning, making it easier to track experiments. The GAN model was the most time-consuming during this process, followed by the VAE and NF. Indeed, the GAN is highly sensitive to hyper-parameter modifications making it challenging to identify a relevant set of values. In contrast, the NF achieved satisfactory results, both in terms of scenarios shapes and quality, by testing only a few sets of hyper-parameter values.

B.4. Hyper-parameters

Table B.6 provides the hyper-parameters of the NF, VAE, and GAN implemented. The Adam optimizer [76] is used to train the generative models with a batch size of 10% of the learning set. The NF implemented is a one-step monotonic normalizer using the UMNN-MAF.[4] The embedding size $|h^i|$ is set to 40, and the embedding neural network is composed of l layers of n neurons (l×n). The same integrand neural network $\tau^i(\cdot)\ \forall i = 1,\dots,T$ is used and composed of 3 layers of $|h^i|$ neurons (3 × 40). Both the encoder and decoder of the VAE are feed-forward neural networks ($l \times n$), ReLU activation functions for the hidden layers, and no activation function for the output layer. Both the generator and discriminator of the GAN are feed-forward neural networks (l×n). The activation functions of the hidden layers of the generator (discriminator) are ReLU (Leaky ReLU). The activation function of the discriminator output layer is ReLU, and there is no activation function for the generator output layer. The generator is trained once after the discriminator is trained five times to stabilize the training process, and the gradient penalty coefficient λ in (7) is set to 10 as suggested by Gulrajani et al. [53].

Figs. B.12, B.13, and B.14 illustrate the VAE, GAN, and NF structures implemented for the wind dataset where the number of weather variables selected and the number of zones is 10, and 10, respectively. Recall, $\mathbf{c}$:= weather forecasts, $\hat{\mathbf{x}}$:= scenarios $\mathbf{x}$:= wind power observations, $\mathbf{z}$:= latent space variable, ϵ := Normal variable (only for the VAE).

Appendix C. Quality metrics

Recall the PV generation, wind power, and load are assumed to be multivariate random variables of dimension T, $\mathbf{x} \in \mathbb{R}^T$, with T the number of time periods per day. Let $\cup_{d\in\text{TS}}\{\hat{\mathbf{x}}_d^i\}_{i=1}^M$ be the set of #TS×M scenarios generated with M scenarios per day of the testing set, where $\hat{\mathbf{x}}_d^i \in \mathbb{R}^T \forall i,d$. $\hat{x}_{d,k}^i$ is the component k of scenario i on day d of the testing set as specified by (1).

[4] https://github.com/AWehenkel/Normalizing-Flows

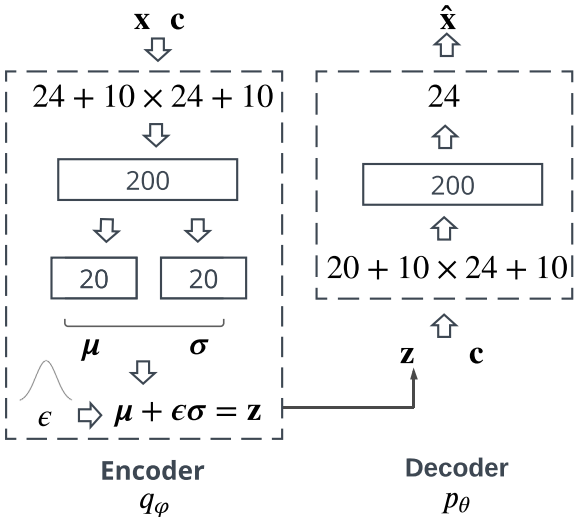


Fig. B.12. Variational autoencoder structure implemented for the wind dataset. Both the encoder and decoder are feed-forward neural networks composed of one hidden layer with 200 neurons. Increasing the number of layers did not improve the results for this dataset. The latent space dimension is 20.

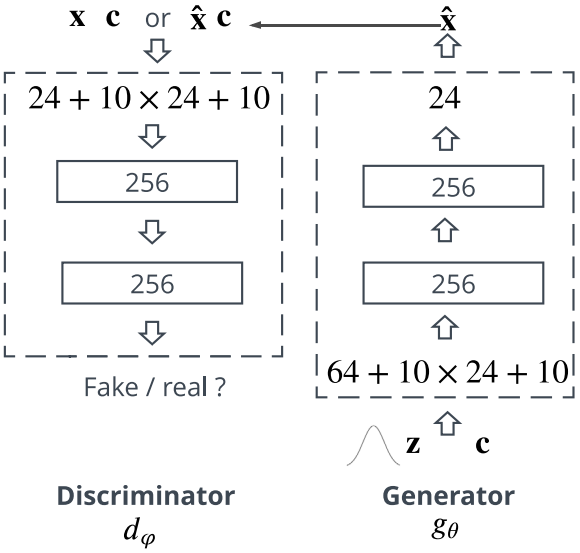


Fig. B.13. Generative adversarial network structure implemented for the wind dataset. Both the discriminator and generator are feed-forward neural networks composed of two hidden layers with 256 neurons. The latent space dimension is 64.

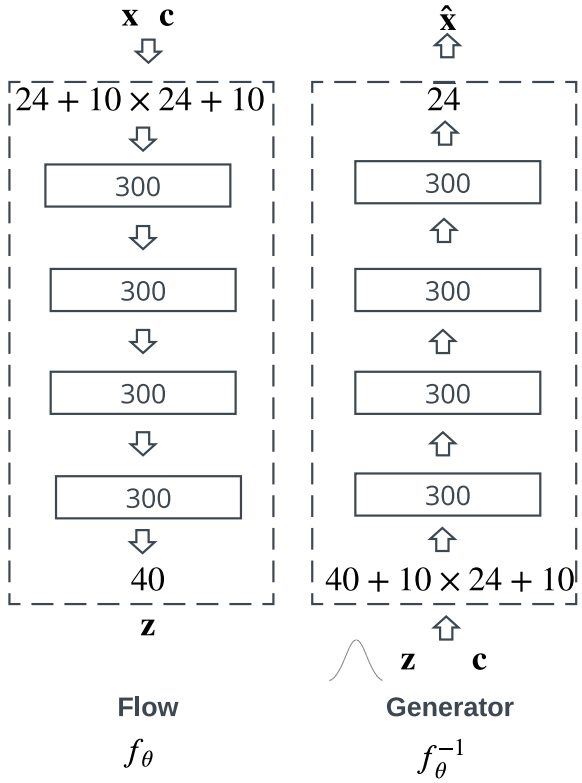


Fig. B.14. Normalizing flow structure implemented for the wind dataset. A single-step monotonic normalizing flow is implemented with a feed-forward neural network composed of four hidden layers with 300 neurons. The latent space dimension is 40. Note: for the sake of clarity the integrand network is not included but is a feed-forward neural network composed of three hidden layers with 40 neurons. Increasing the number of steps of the normalizing flow did not improve the results. The monotonic transformation is complex enough to capture the stochasticity of the variable of interest. However, when considering affine autoregressive normalizing flows the number of steps should be generally more important. Numerical experiments indicated a five-step autoregressive flow was required to achieve similar results for this dataset. Note: the results are not reported in this study for the sake of clarity.

C.1. Continuous ranked probability score

Gneiting and Raftery [60] propose a formulation called the energy form of the CRPS since it is just the one-dimensional case of the energy score, defined in negative orientation as follows

$$\mathrm{CRPS}(P, x_k) = \mathbb{E}_P[|X - x_k|] - \frac{1}{2}\mathbb{E}_P[|X - X'|], \tag{C.1}$$

where and X and X' are independent random variables with distribution P and finite first moment, and $\mathbb{E}_P$ is the expectation according to the probabilistic distribution P. The CRPS is computed over the marginals of $\hat{\mathbf{x}}$ by using the estimator of (C.1) provided by Zamo and Naveau [77]. For a given day d of the testing set, the CRPS per marginal $k = 1, \ldots, T$ is

$$\mathrm{CRPS}_{d,k} = \frac{1}{M}\sum_{i=1}^{M}|\hat{x}_{d,k}^i - x_{d,k}| - \frac{1}{2M^2}\sum_{i,j=1}^{M}|\hat{x}_{d,k}^i - \hat{x}_{d,k}^j|. \tag{C.2}$$

Then, it is averaged over the entire testing set

$$\mathrm{CRPS}_k = \frac{1}{\#TS}\sum_{d\in TS}\mathrm{CRPS}_{d,k}. \tag{C.3}$$

In Table 3, CRPS_k is averaged over all time periods

$$\overline{\mathrm{CRPS}} = \frac{1}{T}\sum_{k=1}^{T}\mathrm{CRPS}_k. \tag{C.4}$$

C.2. Energy score

Gneiting and Raftery [60] introduced a generalization of the continuous ranked probability score defined in negative orientation as follows

$$\mathrm{ES}(P, \mathbf{x}) = \mathbb{E}_P\|X - \mathbf{x}\| - \frac{1}{2}\mathbb{E}_P\|X - X'\|, \tag{C.5}$$

where and X and X' are independent random variables with distribution P and finite first moment, $\mathbb{E}_P$ is the expectation according to the probabilistic distribution P, and $\|\cdot\|$ the Euclidean norm. For a given day d of the testing set, the ES is computed following Gneiting et al. [78]

$$\mathrm{ES}_d = \frac{1}{M}\sum_{i=1}^{M}\|\hat{\mathbf{x}}_d^i - \mathbf{x}_d^i\| - \frac{1}{2M^2}\sum_{i,j=1}^{M}\|\hat{\mathbf{x}}_d^i - \hat{\mathbf{x}}_d^j\|. \tag{C.6}$$

Then, it is averaged over the testing set

$$\mathrm{ES} = \frac{1}{\#TS}\sum_{d\in TS}\mathrm{ES}_d. \tag{C.7}$$

Note: when we consider the marginals of $\mathbf{x}$, it is easy to recognize that (C.6) is the CRPS.

C.3. Variogram score

For a given day d of the testing set and a T-variate observation $\mathbf{x}_d \in \mathbb{R}^T$, the Variogram score metric of order γ is formally defined as

$$\mathrm{VS}_d = \sum_{k,k'}^{T} w_{kk'} \left(|x_{d,k} - x_{d,k'}|^\gamma - \mathbb{E}_P |\hat{x}_{d,k} - \hat{x}_{d,k'}|^\gamma \right)^2, \tag{C.8}$$

where $\hat{x}_{d,k}$ and $\hat{x}_{d,k'}$ are the kth and k'th components of the random vector $\hat{\mathbf{x}}_d$ distributed according to P for which the γth absolute moment exists, and $w_{kk'}$ are non-negative weights. Given a set of M scenarios $\{\hat{\mathbf{x}}_d^i\}_{i=1}^M$ for this given day d, the forecast variogram $\mathbb{E}_P|\hat{x}_{d,k} - \hat{x}_{d,k'}|^\gamma$ can be approximated $\forall k, k' = 1, \ldots, T$ by

$$\mathbb{E}_P|\hat{x}_{d,k} - \hat{x}_{d,k'}|^\gamma \approx \frac{1}{M} \sum_{i=1}^{M} |\hat{x}_{d,k}^i - \hat{x}_{d,k'}^i|^\gamma. \tag{C.9}$$

Then, it is averaged over the testing set

$$\mathrm{VS} = \frac{1}{\#TS} \sum_{d \in TS} \mathrm{VS}_d. \tag{C.10}$$

In this study, we evaluate the Variogram score with equal weights across all hours of the day $w_{kk'} = 1$ and using a γ of 0.5, which for most cases provides a good discriminating ability as reported in Scheuerer and Hamill [63].

C.4. Quantile score

For a given day d of the testing set, a set of 99 quantiles (1, 2, …, 99th quantile) $\{\hat{\mathbf{x}}_d^q\}_{q=1}^{99}$ are computed from the set of M scenarios $\{\hat{\mathbf{x}}_d^i\}_{i=1}^M$, with q the quantile index ($q = 0.01, \ldots, 0.99$). For a given day d of the testing set, the quantile score, per marginal, is defined by

$$\rho_q(\hat{x}_{d,k}^q, x_{d,k}) = \begin{cases} (1-q) \times (\hat{x}_{d,k}^q - x_{d,k}) & x_{d,k} < \hat{x}_{d,k}^q \\ q \times (x_{d,k} - \hat{x}_{d,k}^q) & x_{d,k} \geq \hat{x}_{d,k}^q. \end{cases} \tag{C.11}$$

Then, it is averaged over all time periods and the testing set

$$\mathrm{QS}_q = \frac{1}{\#TS} \sum_{d \in TS} \frac{1}{T} \sum_{k=1}^{T} \rho_q(\hat{x}_{d,k}^q, x_{d,k}). \tag{C.12}$$

In Table 3, QS_q is averaged over all quantiles

$$\overline{\mathrm{QS}} = \frac{1}{99} \sum_{q=1}^{99} \mathrm{QS}_q. \tag{C.13}$$

C.5. Classifier-based metric

Modern binary classifiers can be easily turned into powerful two-sample tests where the goal is to assess whether two samples are drawn from the same distribution [79]. In other words, it aims at assessing whether a generated scenario can be distinguished from an observation. To this end, the generator is evaluated on a held-out testing set that is split into a testing-train and testing-test subsets. The testing-train set is used to train a classifier, distinguishing generated scenarios from the actual distribution. Then, the final score is computed as the performance of this classifier on the testing-test set.

In principle, any binary classifier can be adopted for computing classifier two-sample tests (C2ST). A variation of this evaluation methodology is proposed by Xu et al. [59] and is known as the 1-Nearest Neighbor (NN) classifier. The advantage of using 1-NN over other classifiers is that it requires no special training and little hyper-parameter tuning. This process is conducted as follows. Given two sets of observations S_r and generated S_g samples with the same size, *i.e.*, $|S_r| = |S_g|$, it is possible to compute the leave-one-out (LOO) accuracy of a 1-NN classifier trained on S_r and S_g with positive labels for S_r and negative labels for S_g. The LOO accuracy can vary from 0% to 100%. The 1-NN classifier should yield a 50% LOO accuracy when $|S_r| = |S_g|$ is large. It is achieved when the two distributions match. Indeed, the level 50% happens when a label is randomly assigned to a generated scenario. It means the classifier is not capable of discriminating generated scenarios from observations. If the generative model over-fits S_g to S_r, *i.e.*, $S_g = S_r$, and the accuracy would be 0%. On the contrary, if it generates widely different samples than observations, the performance should be 100%. Therefore, the closer the LOO accuracy is to 1, the higher the degree of under-fitting of the model. The closer the LOO accuracy is to 0, the higher the degree of over-fitting of the model. The C2ST approach using LOO with 1-NN is adopted by Qi et al. [37] to assess the PV and wind power scenarios of a β VAE.

However, this approach has several limitations. First, it uses the testing set to train the classifier during the LOO. Second, the 1-NN is very sensitive to outliers as it simply chooses the closest neighbor based on distance criteria. This behavior is amplified when combined with the LOO, where the testing-test set is composed of only one sample. Third, the euclidian distance cannot deal with a context such as weather forecasts. Therefore, we cannot use a conditional version of the 1-NN using weather forecasts to classify weather-based renewable generation and the observations. Fourth, C2ST with LOO cannot provide ROC curve but only accuracy scores. An essential point about ROC graphs is that they measure the ability of a classifier to produce good relative instance scores. In our case, we are interested in discriminating the generated scenarios from the observations, and the ROC provides more information than the accuracy metric to achieve this goal. A standard method to reduce ROC performance to a single scalar value representing expected performance is to calculate the area under the ROC curve abbreviated AUC. The AUC has an essential statistical property: it is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance [65].

To deal with these issues, we decided to modify this classifier-based evaluation by conducting the C2ST as follows: (1) the scenarios generated on the learning set are used to train the classifier using the C2ST. Therefore, the classifier uses the entire testing set and can compute ROC; (2) the classifier is an Extra-Trees classifier that can deal with context such as weather forecasts.

More formally, for a given generative model g, the following steps are conducted:

1. Initialization step: the generative model g has been trained on the LS and has generated M weather-based scenarios per day of both the LS and TS: $\{\hat{\mathbf{x}}_{\mathrm{LS}}^i\}_{i=1}^M := \cup_{d \in \mathrm{LS}} \{\hat{\mathbf{x}}_d^i\}_{i=1}^M$ and $\{\hat{\mathbf{x}}_{\mathrm{TS}}^i\}_{i=1}^M := \cup_{d \in \mathrm{TS}} \{\hat{\mathbf{x}}_d^i\}_{i=1}^M$. For the sake of clarity the index g is omitted, but both of these sets are dependent on model g.
2. M pairs of learning and testing sets are built with an equal proportion of generated scenarios and observations: $\mathcal{D}_{\mathrm{LS}}^i := \left\{ \{\hat{\mathbf{x}}_{\mathrm{LS}}^i, 0\} \cup \left\{ \{\mathbf{x}_{\mathrm{LS}}^i, 1\} \right\} \right.$ and $\mathcal{D}_{\mathrm{TS}}^i = \left\{ \{\hat{\mathbf{x}}_{\mathrm{TS}}^i, 0\} \cup \left\{ \{\mathbf{x}_{\mathrm{TS}}^i, 1\} \right\} \right.$. Note: $|\mathcal{D}_{\mathrm{LS}}^i| = 2|\mathrm{LS}|$ and $|\mathcal{D}_{\mathrm{TS}}^i| = 2|\mathrm{TS}|$.
3. For each pair of learning and testing sets $\{\mathcal{D}_{\mathrm{LS}}^i, \mathcal{D}_{\mathrm{TS}}^i\}_{i=1}^M$ a classifier d_g^i is trained and makes predictions.
4. The ROC_g^i curves and corresponding AUC_g^i are computed for $i = 1, \ldots, M$.

This classifier-based methodology is conducted for all models g, and the results are compared. Fig. C.15 depicts the overall approach. The classifiers d_g^i are all Extra-Trees classifier made of 1000 unconstrained trees with the hyper-parameters "max_depth" set to "None", and "n_estimators" to 1000.

C.6. Diebold–Mariano test

For a given day d of the testing set, let $\epsilon_d \in \mathbb{R}$ be the error computed by an arbitrary forecast loss function of the observation and scenarios.

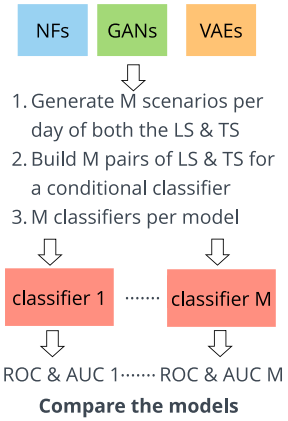


Fig. C.15. Classifier-based metric methodology.
Each generative model generates M scenarios per day of the learning and testing sets. They are used to build M pairs of learning and testing sets for a conditional classifier by including an equal proportion of observations and weather forecasts. M conditional classifiers, per model, are trained and make predictions. The M ROC and AUC are computed per model, and the results are compared.

The test consists of computing the difference between the errors of the pair of models g and h over the testing set

$$\Delta(g,h)_d = \epsilon_d^g - \epsilon_d^h, \quad \forall d \in \text{TS}, \tag{C.14}$$

and to perform an asymptotic z-test for the null hypothesis that the expected forecast error is equal and the mean of differential loss series is zero $\mathbb{E}[\Delta(g,h)_d] = 0$. It means there is no statistically significant difference in the accuracy of the two competing forecasts. The statistic of the test is deduced from the asymptotically standard normal distribution as follows

$$\text{DM}(g,h) = \sqrt{\#\text{TS}}\frac{\hat{\mu}}{\hat{\sigma}}, \tag{C.15}$$

with #TS the number of days of the testing set, $\hat{\mu}$ and $\hat{\sigma}$ the sample mean and the standard deviation of $\Delta(g,h)$. Under the assumption of covariance stationarity of the loss differential series $\Delta(g,h)_d$, the DM statistic is asymptotically standard normal. The lower the p-value, *i.e.*, the closer it is to zero, the more the observed data is inconsistent with the null hypothesis: $\mathbb{E}[\Delta(g,h)_d] < 0$ the forecasts of the model h are more accurate than those of model g. If the p-value is less than the commonly accepted level of 5%, the null hypothesis is typically rejected. It means that the forecasts of model g are significantly more accurate than those of model h.

When considering the ES or VS scores, there is a value per day of the testing set ES_d or VS_d. In this case, $\epsilon_d = \text{ES}_d$ or $\epsilon_d = \text{VS}_d$. However, when considering the CRPS or QS, there is a value per marginal and per day of the testing set $\text{CRPS}_{d,k}$ or $\text{QS}_{d,k}$. A solution consists of computing 24 independent tests, one for each hour of the day. Then, to compare the models based on the number of hours for which the predictions of one model are significantly better than those of another. Another way consists of a multivariate variant of the DM-test with the test performed jointly for all hours using the multivariate loss differential series. In this case, for a given day d, $\epsilon_d^g = [\epsilon_{d,1}^g, \dots, \epsilon_{d,24}^g]^\intercal$, $\epsilon_d^h = [\epsilon_{d,1}^h, \dots, \epsilon_{d,24}^h]^\intercal$ are the vectors of errors for a given metric of models g and h, respectively. Then the multivariate loss differential series

$$\Delta(g,h)_d = \|\epsilon_d^g\|_1 - \|\epsilon_d^h\|_1, \tag{C.16}$$

defines the differences of errors using the $\|\cdot\|_1$ norm. Then, for each model pair, the p-value of two-sided DM tests is computed as described above. The univariate version of the test has the advantage of providing a more profound analysis as it indicates which forecast is significantly better for which hour of the day. The multivariate version enables a better representation of the results as it summarizes the comparison in a single p-value, which can be conveniently visualized using heat maps arranged as chessboards. In this study, we decided to adopt the multivariate DM-test for the CRPS and QS.

Appendix D. Value assessment

D.1. Notation

Sets and indexes

Name	Description
t	Time period index.
ω	Scenario index.
T	Number of time periods per day.
$\#\Omega$	Number of scenarios.
$\mathcal{T}$	Set of time periods, $\mathcal{T} = \{1, 2, \dots, T\}$.
Ω	Set of scenarios, $\Omega = \{1, 2, \dots, \#\Omega\}$.

Parameters

Name	Description
e_t^{min}, e_t^{max}	Minimum/maximum day-ahead bid [MWh].
y_t^{min}, y_t^{max}	Minimum/maximum retailer net position [MWh].
$y_{\text{max}}^{\text{dis}}$, $y_{\text{max}}^{\text{cha}}$	BESS maximum (dis)charging power [MW].
η^{dis}, η^{cha}	BESS (dis)charging efficiency [-].
s^{min}, s^{max}	BESS minimum/maximum capacity [MWh].
s^{ini}, s^{end}	BESS initial/final state of charge [MWh].
π_t	Day-ahead price [€/MWh].
$\bar{q}_t$, $\bar{\lambda}_t$	Negative/positive imbalance price [€/MWh].
Δt	Duration of a time period [hour].

Variables

For the sake of clarity the subscript ω is omitted.

Name	Range	Description
e_t	$[e_t^{min}, e_t^{max}]$	Day-ahead bid [MWh].
y_t	$[y_t^{min}, y_t^{max}]$	Retailer net position [MWh].
y_t^{pv}	$[0, 1]$	PV generation [MW].
y_t^{w}	$[0, 1]$	Wind generation [MW].
y_t^{cha}	$[0, y_{\text{max}}^{\text{cha}}]$	Charging power [MW].
y_t^{dis}	$[0, y_{\text{max}}^{\text{dis}}]$	Discharging power [MW].
s_t	$[s^{\text{min}}, s^{\text{max}}]$	BESS state of charge [MWh].
d_t^-, d_t^+	$\mathbb{R}_+$	Short/long deviation [MWh].
y_t^b	$\{0, 1\}$	BESS binary variable [-].

D.2. Problem formulation

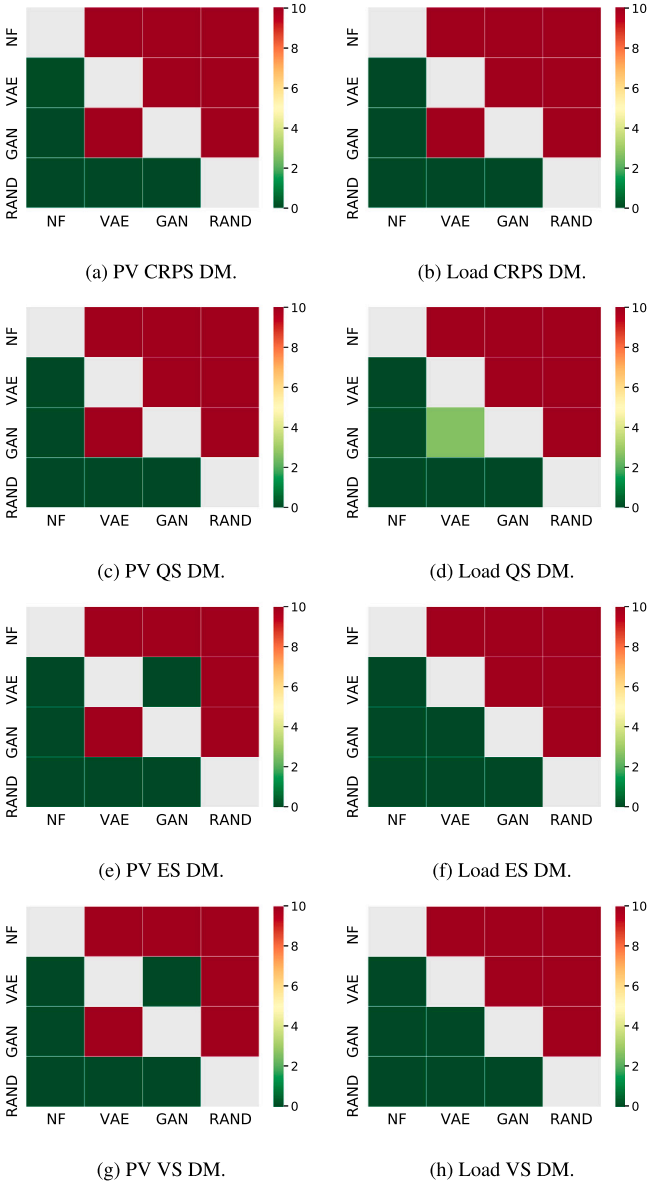
The mixed-integer linear programming (MILP) optimization problem to solve is

$$\max_{e_t \in \mathcal{X}, y_{t,\omega} \in \mathcal{Y}(e_t)} \sum_{\omega \in \Omega} \alpha_\omega \sum_{t \in \mathcal{T}} \left[\pi_t e_t - \bar{q}_t d_{t,\omega}^- - \bar{\lambda}_t d_{t,\omega}^+\right], \tag{D.1a}$$

$$\mathcal{X} = \left\{ e_t : e_t \in [e_t^{min}, e_t^{max}] \right\}, \tag{D.1b}$$

$$\mathcal{Y}(e_t) = \left\{ y_{t,\omega} : \text{(D.2a)- (D.2m)} \right\}. \tag{D.1c}$$

The optimization variables are e_t, day-ahead bid of the net position, $\forall \omega \in \Omega$, $y_{t,\omega}$, retailer net position in scenario ω, $d_{t,\omega}^-$, short deviation, $d_{t,\omega}^+$, long deviation, $y_{t,\omega}^{\text{pv}}$, PV generation, $y_{t,\omega}^{\text{w}}$, wind generation, $y_{t,\omega}^{\text{cha}}$, battery energy storage system (BESS) charging power, $y_{t,\omega}^{\text{dis}}$, BESS discharging power, $s_{t,\omega}$, BESS state of charge, and $y_{t,\omega}^b$ a binary variable to prevent from charging and discharging simultaneously. The imbalance penalty is modeled by the constraints (D.2a)–(D.2b) $\forall \omega \in \Omega$, that define the short and long deviations variables $d_{t,\omega}^-, d_{t,\omega}^+ \in \mathbb{R}_+$. The energy balance is provided by (D.2c) $\forall \omega \in \Omega$. The set of constraints that bound $y_{t,\omega}^{\text{pv}}$ and $y_{t,\omega}^{\text{w}}$ variables are (D.2d)–(D.2e) $\forall \omega \in \Omega$ where $\hat{y}_{t,\omega}^{\text{pv}}$ and $\hat{y}_{t,\omega}^{\text{w}}$ are PV and wind generation scenarios. The load is assumed



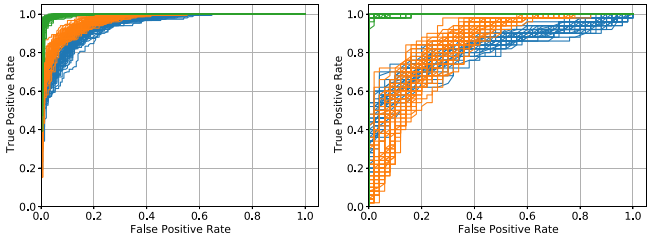
(a) PV CRPS DM. (b) Load CRPS DM.

(c) PV QS DM. (d) Load QS DM.

(e) PV ES DM. (f) Load ES DM.

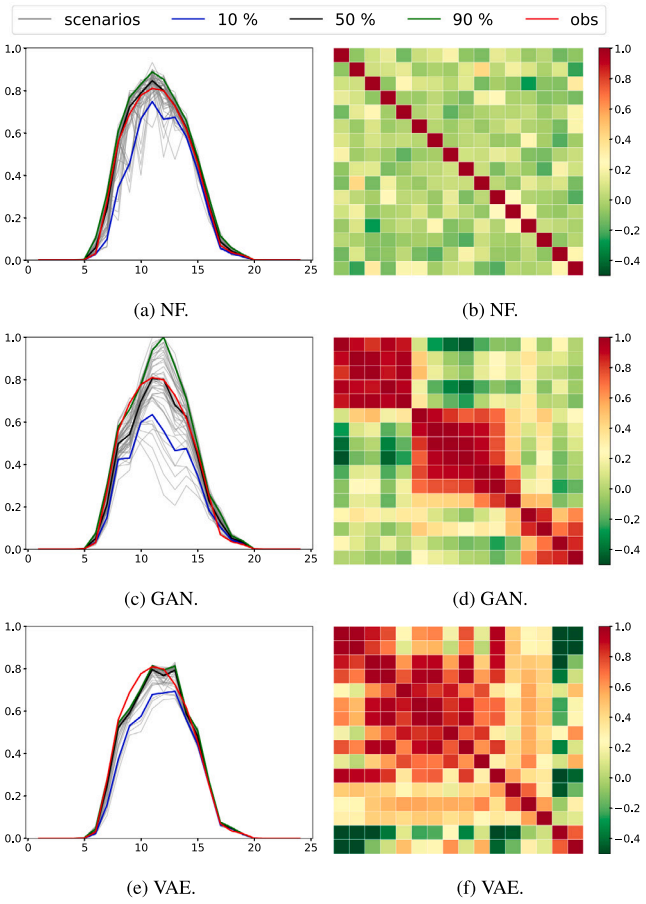
(g) PV VS DM. (h) Load VS DM.

Fig. E.16. PV and load tracks Diebold–Mariano tests.
The Diebold–Mariano tests of the CRPS, QS, ES, and VS demonstrate that the NF outperforms the VAE and GAN. Note: the GAN outperforms the VAE for both the ES and VS for the PV track. However, the VAE outperforms the GAN on this dataset for both the CRPS and QS.



(a) ROC PV. (b) ROC load.

Fig. E.17. Classifier-based metric for both the PV and load tracks.
The NF (blue) is the best to fake the classifier, followed by the VAE (orange), and the GAN (green).



(a) NF. (b) NF.

(c) GAN. (d) GAN.

(e) VAE. (f) VAE.

Fig. E.18. PV scenarios shape comparison and analysis.
Left part (a) NF, (c) GAN, and (e) VAE: 50 PV scenarios (gray) of a randomly selected day of the testing set along with the 10% (blue), 50% (black), and 90% (green) quantiles, and the observations (red). Right part (b) NF, (d) GAN, and (f) VAE: the corresponding Pearson time correlation matrices of these scenarios with the periods as rows and columns. Similar to wind power and load scenarios, NF tends to exhibit no time correlation between scenarios. In contrast, the VAE and GAN tend to be partially time-correlated over a few periods.

to be non-flexible and is a parameter (D.2f) $\forall \omega \in \Omega$ where $\hat{y}^{\mathrm{l}}_{t,\omega}$ are load scenarios. The BESS constraints are provided by (D.2g)–(D.2j), and the BESS dynamics by (D.2k)–(D.2m) $\forall \omega \in \Omega$.

$$-d^{-}_{t,\omega} \leq -(e_t - y_{t,\omega}), \forall t \in \mathcal{T} \tag{D.2a}$$

$$-d^{+}_{t,\omega} \leq -(y_{t,\omega} - e_t), \forall t \in \mathcal{T} \tag{D.2b}$$

$$\frac{y_{t,\omega}}{\Delta t} = y^{\mathrm{pv}}_{t,\omega} + y^{\mathrm{w}}_{t,\omega} - y^{\mathrm{l}}_{t,\omega} + y^{\mathrm{dis}}_{t,\omega} - y^{\mathrm{cha}}_{t,\omega}, \forall t \in \mathcal{T} \tag{D.2c}$$

$$y^{\mathrm{pv}}_{t,\omega} \leq \hat{y}^{\mathrm{pv}}_{t,\omega}, \forall t \in \mathcal{T} \tag{D.2d}$$

$$y^{\mathrm{w}}_{t,\omega} \leq \hat{y}^{\mathrm{w}}_{t,\omega}, \forall t \in \mathcal{T} \tag{D.2e}$$

$$y^{\mathrm{l}}_{t,\omega} = \hat{y}^{\mathrm{l}}_{t,\omega}, \forall t \in \mathcal{T} \tag{D.2f}$$

$$y^{\mathrm{cha}}_{t,\omega} \leq y^{b}_{t,\omega} y^{\mathrm{cha}}_{\max}, \forall t \in \mathcal{T} \tag{D.2g}$$

$$y^{\mathrm{dis}}_{t,\omega} \leq (1 - y^{b}_{t,\omega}) y^{\mathrm{dis}}_{\max}, \forall t \in \mathcal{T} \tag{D.2h}$$

$$-s_{t,\omega} \leq -s^{\min}, \forall t \in \mathcal{T} \tag{D.2i}$$

$$s_{t,\omega} \leq s^{\max}, \forall t \in \mathcal{T} \tag{D.2j}$$

$$\frac{s_{1,\omega} - s^{\mathrm{ini}}}{\Delta t} = \eta^{\mathrm{cha}} y^{\mathrm{cha}}_{1,\omega} - \frac{y^{\mathrm{dis}}_{1,\omega}}{\eta^{\mathrm{dis}}}, \tag{D.2k}$$

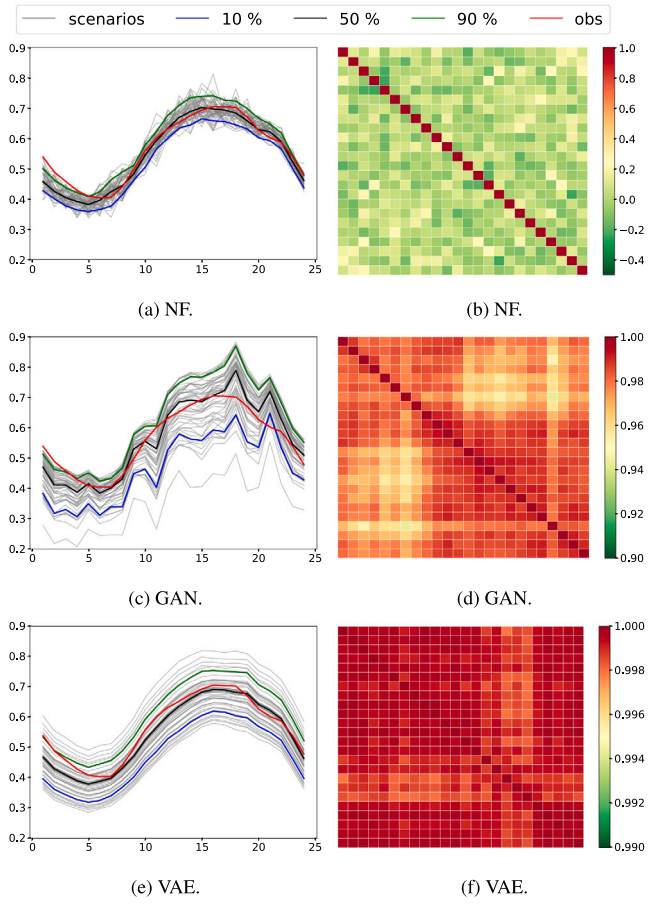


Fig. E.19. Load scenarios shape comparison and analysis.
Left part (a) NF, (c) GAN, and (e) VAE: 50 load scenarios (gray) of a randomly selected day of the testing set along with the 10% (blue), 50% (black), and 90% (green) quantiles, and the observations (red). Right part (b) NF, (d) GAN, and (f) VAE: the corresponding Pearson time correlation matrices of these scenarios with the periods as rows and columns. Similar to PV and wind power scenarios, NF tends to exhibit no time correlation between scenarios. In contrast, the VAE and GAN tend to be highly time-correlated.

$$\frac{s_{t,\omega} - s_{t-1,\omega}}{\Delta t} = \eta^{\text{cha}} y^{\text{cha}}_{t,\omega} - \frac{y^{\text{dis}}_{t,\omega}}{\eta^{\text{dis}}}, \forall t \in \mathcal{T} \setminus \{1\} \tag{D.2l}$$

$$s_{T,\omega} = s^{\text{end}} = s^{\text{ini}}. \tag{D.2m}$$

Notice that if $\bar{\lambda}_t < 0$, the surplus quantity is remunerated with a non-negative price. In practice, such a scenario could be avoided provided that the energy retailer has curtailment capabilities, and $(\bar{q}_t, \bar{\lambda}_t)$ are strictly positive in our case study. The deterministic formulation with perfect forecasts, the oracle (O), is a specific case of the stochastic formulation by considering only one scenario where $y^{\text{pv}}_{t,\omega}$, $y^{\text{w}}_{t,\omega}$, and $y^{\text{l}}_{t,\omega}$ become the actual values of PV, wind, and load $\forall t \in \mathcal{T}$. The optimization variables are e_t, y_t, d^-_t, d^+_t, y^{pv}_t, and y^{w}_t, y^{cha}_t, y^{dis}_t, s_t, and y^b_t.

D.3. Dispatching

Once the bids e_t have been computed by the planner, the dispatching consists of computing the second stage variables given observations of the PV, wind power, and load. The dispatch formulation is a specific case of the stochastic formulation with e_t as parameter and by considering only one scenario where $y^{\text{pv}}_{t,\omega}$, $y^{\text{w}}_{t,\omega}$, and $y^{\text{l}}_{t,\omega}$ become the actual values of PV, wind, and load $\forall t \in \mathcal{T}$. The optimization variables are y_t, d^-_t, d^+_t, y^{pv}_t, and y^{w}_t, y^{cha}_t, y^{dis}_t, s_t, and y^b_t.

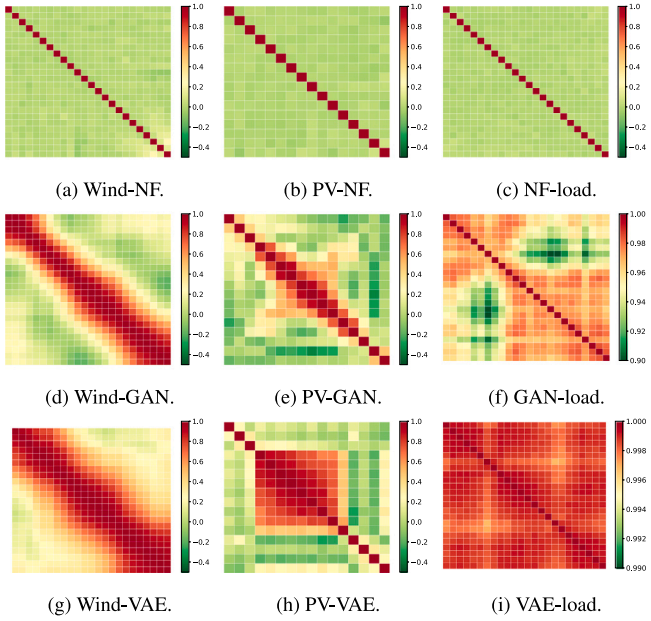


Fig. E.20. Average of the correlation matrices over the testing set for the three datasets.
Left: wind power; center: PV; right:load. The trend in terms of time correlation is observed on each day of the testing set for all the datasets. The NF scenarios are not correlated. In contrast, the VAE and GAN scenarios tend to be time-correlated over a few periods. In particular, the VAE generates highly time-correlated scenarios for the load dataset.

Appendix E. Quality results

See Figs. E.16–E.20.

References

[1] Allen M, Antwi-Agyei P, Aragon-Durand F, Babiker M, Bertoldi P, Bind M, et al. Technical summary: Global warming of 1.5 °C. An IPCC special report on the impacts of global warming of 1.5 °C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty. Technical report, Intergovernmental Panel on Climate Change; 2019.

[2] Gneiting T, Katzfuss M. Probabilistic forecasting. Annu Rev Stat Appl 2014;1:125–51.

[3] Hong T, Fan S. Probabilistic electric load forecasting: A tutorial review. Int J Forecast 2016;32(3):914–38.

[4] Morales JM, Conejo AJ, Madsen H, Pinson P, Zugno M. Integrating renewables in electricity markets: operational problems, vol. 205. Springer Science & Business Media; 2013.

[5] Hong T, Pinson P, Wang Y, Weron R, Yang D, Zareipour H, et al. Energy forecasting: A review and outlook. Technical report, Department of Operations Research and Business Intelligence, Wroclaw; 2020.

[6] Khoshrou A, Pauwels EJ. Short-term scenario-based probabilistic load forecasting: A data-driven approach. Appl Energy 2019;238:1258–68.

[7] Mashlakov A, Kuronen T, Lensu L, Kaarna A, Honkapuro S. Assessing the performance of deep learning models for multivariate probabilistic energy forecasting. Appl Energy 2021;285:116405.

[8] Wang P, Liu B, Hong T. Electric load forecasting with recency effect: A big data approach. Int J Forecast 2016;32(3):585–97.

[9] De Gooijer JG, Hyndman RJ. 25 years of time series forecasting. Int J Forecast 2006;22(3):443–73.

[10] Morales JM, Minguez R, Conejo AJ. A methodology to generate statistically dependent wind speed scenarios. Appl Energy 2010;87(3):843–55.

[11] Karaki SH, Salim BA, Chedid RB. Probabilistic model of a two-site wind energy conversion system. IEEE Trans Energy Convers 2002;17(4):530–6.

[12] Karaki S, Chedid R, Ramadan R. Probabilistic performance assessment of autonomous solar-wind energy conversion systems. IEEE Trans Energy Convers 1999;14(3):766–72.

[13] Pinson P, Madsen H, Nielsen HA, Papaefthymiou G, Klöckl B. From probabilistic forecasts to statistical scenarios of short-term wind power production. Wind Energy 2009;12(1):51–62.

[14] Zhang H, Lu Z, Hu W, Wang Y, Dong L, Zhang J. Coordinated optimal operation of hydro–wind–solar integrated systems. Appl Energy 2019;242:883–96.

[15] Camal S, Teng F, Michiorri A, Kariniotakis G, Badesa L. Scenario generation of aggregated Wind, Photovoltaics and small Hydro production for power systems applications. Appl Energy 2019;242:1396–406.

[16] Shi H, Xu M, Li R. Deep learning for household load forecasting—A novel pooling deep RNN. IEEE Trans Smart Grid 2017;9(5):5271–80.

[17] Dumas J, Cointe C, Fettweis X, Cornélusse B. Deep learning-based multi-output quantile forecasting of pv generation. In: 2021 IEEE Madrid PowerTech. 2021 p. 1–6. http://dx.doi.org/10.1109/PowerTech46648.2021.9494976.

[18] Hewamalage H, Bergmeir C, Bandara K. Recurrent neural networks for time series forecasting: Current status and future directions. Int J Forecast 2020;37(1):388–427.

[19] Toubeau J-F, Bottieau J, Vallée F, De Grève Z. Deep learning-based multivariate probabilistic forecasting for short-term scheduling in power markets. IEEE Trans Power Syst 2018;34(2):1203–15.

[20] Salinas D, Flunkert V, Gasthaus J, Januschowski T. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. Int J Forecast 2020;36(3):1181–91.

[21] Bond-Taylor S, Leach A, Long Y, Willcocks CG. Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models. 2021, arXiv preprint arXiv:2103.04922.

[22] Ruthotto L, Haber E. An introduction to deep generative modeling. GAMM-Mitt 2021;e202100008.

[23] Kingma DP, Welling M. Auto-encoding variational bayes. 2013, arXiv preprint arXiv:1312.6114.

[24] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. 2014, arXiv preprint arXiv:1406.2661.

[25] Ordiano JAG, Gröll L, Mikut R, Hagenmeyer V. Probabilistic energy forecasting using the nearest neighbors quantile filter and quantile regression. Int J Forecast 2020;36(2):310–23.

[26] Sun M, Feng C, Zhang J. Probabilistic solar power forecasting based on weather scenario generation. Appl Energy 2020;266:114823.

[27] Zhanga H, Hua W, Yub R, Tangb M, Dingc L. Optimized operation of cascade reservoirs Considering Complementary Characteristics between wind and photovoltaic based on variational auto-encoder. In: MATEC web of conferences, vol. 246. EDP Sciences; 2018, p. 01077.

[28] Dairi A, Harrou F, Sun Y, Khadraoui S. Short-term forecasting of photovoltaic solar power production using variational auto-encoder driven deep learning approach. Appl Sci 2020;10(23):8400.

[29] Chen Y, Wang Y, Kirschen D, Zhang B. Model-free renewable scenario generation using generative adversarial networks. IEEE Trans Power Syst 2018;33(3):3265–75.

[30] Chen Y, Li P, Zhang B. Bayesian renewables scenario generation via deep generative networks. In: 2018 52nd annual conference on information sciences and systems. IEEE; 2018, p. 1–6.

[31] Yuan R, Wang B, Mao Z, Watada J. Multi-objective wind power scenario forecasting based on PG-GAN. Energy 2021;120379.

[32] Chen Z, Jiang C. Building occupancy modeling using generative adversarial network. Energy Build 2018;174:372–9.

[33] Lan J, Guo Q, Sun H. Demand side data generating based on conditional generative adversarial networks. Energy Procedia 2018;152:1188–93.

[34] Zhang Y, Ai Q, Xiao F, Hao R, Lu T. Typical wind power scenario generation for multiple wind farms using conditional improved wasserstein generative adversarial network. Int J Electr Power Energy Syst 2020;114:105388.

[35] Wang Y, Hug G, Liu Z, Zhang N. Modeling load forecast uncertainty using generative adversarial networks. Electr Power Syst Res 2020;189:106732.

[36] Jiang C, Mao Y, Chai Y, Yu M. Day-ahead renewable scenario forecasts based on generative adversarial networks. Int J Energy Res 2021;45(5):7572–87.

[37] Qi Y, Hu W, Dong Y, Fan Y, Dong L, Xiao M. Optimal configuration of concentrating solar power in multienergy power systems with an improved variational autoencoder. Appl Energy 2020;274:115124.

[38] Ge L, Liao W, Wang S, Bak-Jensen B, Pillai JR. Modeling daily load profiles of distribution network for scenario generation using flow-based generative network. IEEE Access 2020;8:77587–97.

[39] Rezende D, Mohamed S. Variational inference with normalizing flows. In: International conference on machine learning. PMLR; 2015, p. 1530–8.

[40] Oord A, Li Y, Babuschkin I, Simonyan K, Vinyals O, Kavukcuoglu K, et al. Parallel wavenet: Fast high-fidelity speech synthesis. In: International conference on machine learning. PMLR; 2018, p. 3918–26.

[41] Green SR, Gair J. Complete parameter inference for GW150914 using deep learning. Mach Learn: Sci Technol 2021;2(3):03LT01.

[42] Albergo MS, Boyda D, Hackett DC, Kanwar G, Cranmer K, Racanière S, et al. Introduction to normalizing flows for lattice field theory. 2021, arXiv preprint arXiv:2101.08176.

[43] Dumas J, Cointe C, Wehenkel A, Sutera A, Fettweis X, Cornélusse B. A probabilistic forecast-driven strategy for a risk-aware participation in the capacity firming market. IEEE Trans Sustain Energy 2021. Manuscript [submitted for publication].

[44] Huang C-W, Krueger D, Lacoste A, Courville A. Neural autoregressive flows. In: International conference on machine learning. PMLR; 2018, p. 2078–87.

[45] Hong T, Pinson P, Fan S, Zareipour H, Troccoli A, Hyndman RJ. Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. Elsevier; 2016.

[46] Silva JAA, López JC, Arias NB, Rider MJ, da Silva LC. An optimal stochastic energy management system for resilient microgrids. Appl Energy 2021;300:117435.

[47] Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning, vol. 1, no. 2. MIT Press Cambridge; 2016.

[48] Zhang A, Lipton ZC, Li M, Smola AJ. Dive Into Deep Learning. 2020, https://d2l.ai.

[49] Kingma DP, Salimans T, Jozefowicz R, Chen X, Sutskever I, Welling M. Improving variational inference with inverse autoregressive flow. 2016, arXiv preprint arXiv:1606.04934.

[50] Wehenkel A, Louppe G. Unconstrained monotonic neural networks. In: Advances in neural information processing systems. 2019, p. 1545–55.

[51] Papamakarios G, Pavlakou T, Murray I. Masked autoregressive flow for density estimation. In: Advances in neural information processing systems. 2017, p. 2338–47.

[52] Pérez-Cruz F. Kullback–Leibler divergence estimation of continuous distributions. In: 2008 IEEE international symposium on information theory. IEEE; 2008, p. 1666–70.

[53] Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville A. Improved training of wasserstein gans. 2017, arXiv preprint arXiv:1704.00028.

[54] Wehenkel A, Louppe G. Graphical normalizing flows. 2020, arXiv preprint arXiv:2006.02548.

[55] Zhao S, Song J, Ermon S. Towards deeper understanding of variational autoencoding models. 2017, arXiv preprint arXiv:1702.08658.

[56] Arjovsky M, Bottou L. Towards principled methods for training generative adversarial networks. 2017, arXiv preprint arXiv:1701.04862.

[57] Theis L, Oord Avd, Bethge M. A note on the evaluation of generative models. 2015, arXiv preprint arXiv:1511.01844.

[58] Borji A. Pros and cons of gan evaluation measures. Comput Vis Image Underst 2019;179:41–65.

[59] Xu Q, Huang G, Yuan Y, Guo C, Sun Y, Wu F, et al. An empirical study on evaluation metrics of generative adversarial networks. 2018, arXiv preprint arXiv:1806.07755.

[60] Gneiting T, Raftery AE. Strictly proper scoring rules, prediction, and estimation. J Amer Statist Assoc 2007;102(477):359–78.

[61] Lauret P, David M, Pinson P. Verification of solar irradiance probabilistic forecasts. Sol Energy 2019;194:254–71.

[62] Golestaneh F, Gooi HB, Pinson P. Generation and evaluation of space–time trajectories of photovoltaic power. Appl Energy 2016;176:80–91.

[63] Scheuerer M, Hamill TM. Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities. Mon Weather Rev 2015;143(4):1321–34.

[64] Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. Mach Learn 2006;63(1):3–42.

[65] Fawcett T. ROC graphs: Notes and practical considerations for researchers. Mach Learn 2004;31(1):1–38.

[66] Diebold FX, Mariano RS. Comparing predictive accuracy. J Bus Econom Statist 2002;20(1):134–44.

[67] Ziel F, Weron R. Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks. Energy Econ 2018;70:396–420.

[68] Landry M, Erlinger TP, Patschke D, Varrichio C. Probabilistic gradient boosting machines for GEFCom2014 wind forecasting. Int J Forecast 2016;32(3):1061–6.

[69] Buitinck L, Louppe G, Blondel M, Pedregosa F, Mueller A, Grisel O, et al. API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD workshop: languages for data mining and machine learning. 2013. p. 108–22.

[70] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An imperative style, high-performance deep learning library. In: Advances in neural information processing systems, vol. 32. Curran Associates, Inc.; 2019, p. 8024–35.

[71] Biewald L. Experiment tracking with weights and biases. 2020, URL: https://www.wandb.com/. Software available from wandb.com.

[72] Papamakarios G, Nalisnick E, Rezende DJ, Mohamed S, Lakshminarayanan B. Normalizing flows for probabilistic modeling and inference. 2019, arXiv preprint arXiv:1912.02762.

[73] Kobyzev I, Prince S, Brubaker M. Normalizing flows: An introduction and review of current methods. IEEE Trans Pattern Anal Mach Intell 2020.

[74] Arjovsky M, Chintala S, Bottou L. Wasserstein generative adversarial networks. In: International conference on machine learning. PMLR; 2017, p. 214–23.

[75] Villani C. Optimal transport: old and new, vol. 338. Springer Science & Business Media; 2008.

[76] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, arXiv preprint arXiv:1412.6980.
[77] Zamo M, Naveau P. Estimation of the continuous ranked probability score with limited information and applications to ensemble weather forecasts. Math Geosci 2018;50(2):209–34.
[78] Gneiting T, Stanberry LI, Grimit EP, Held L, Johnson NA. Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds. Test 2008;17(2):211–35.
[79] Lehmann EL, Romano JP. Testing statistical hypotheses. Springer Science & Business Media; 2006.