Electric Power Systems Research 200 (2021) 107465

Contents lists available at ScienceDirect

Electric Power Systems Research

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

Stochastic variational inference for probabilistic optimal power flows

Markus Löschenbrand *

Sintef Energy Research, Sem Sælands vei 11, Trondheim 7034, Norway

ARTICLE INFO

Keywords:
Probabilistic optimal power flow
Neural networks
Generative model
Variational autoencoder
Bayesian statistics

ABSTRACT

This paper applies a generative deep learning model, namely a Variational Autoencoder, on probabilistic optimal power flows. The model utilizes Gaussian approximations in order to adequately represent the distributions of the results of a system under uncertainty. These approximations are realized by applying several techniques from Bayesian deep learning, among them most notably Stochastic Variational Inference. Using the *reparameterization trick* and batch sampling, the proposed model allows for the training a probabilistic optimal power flow similar to a possibilistic process. The results are shown by application of a reformulation of the Kullback-Leibler divergence, a distance measure of distributions. Not only is the resulting model simple in its appearance, it also shows to perform well and accurate. Furthermore, the paper also explores potential pathways for future research and gives insights for practitioners using such or similar generative models.

1. Introduction

In recent years, uncertainty has gained growing importance in the context of power systems. Integration of electric vehicles, growing shares of renewable generation and emancipation of individuals as active participants in the power grid have introduced more volatility into an already uncertain system. In addition to this has the increase in computational power of hardware allowed for development of techniques dealing more efficiently with uncertainty encountered in power systems [1].

One way to approach this uncertainty is presented by probabilistic methods. Such methods distinguish themselves from other approaches in their representation of uncertainty which comes in the form of probability density functions. This stands in contrast to other methods such as possibilistic and robust methods which respectively use fuzzy membership functions or uncertainty sets to express uncertainty [2].

Utilizing a distributional representation has the advantage of covering a continuous spectrum of potential outcomes, allowing to represent a wide range of probable outcomes. This gives probabilistic methods a large number of state-of-the-art applications in power systems such as transmission expansion planning [3], line failure detection [4], generation scheduling [5] or grid operation under uncertainty [6].

The work presented here will focus on latter, grid operation under uncertainty, in specific the utilization of probabilistic load flows in the context of optimal power flows. Ref. [7] first formulated such probabilistic optimal power flows as a network of dependent variables. In this work, uncertain nodal loads affect the distribution functions of the analyzed power lines. In order to deal with tractability limitations and efficiently solve the formulated decision problem the paper applies various approximations and assumptions. This includes limiting the dispatch actions to a single slack bus and assuming distributions of nodal loads to be known initially. Approaching such tractability limitations especially aids problems under large uncertainty space and/or problems with multiple sources of uncertainties as found in e.g. security analysis problems of modern (micro)grids [8–10] or modern multi-energy systems [11].

Ref. [12] provides an overview of earlier works on probabilistic power flows, whose solution techniques can generally be classified as either *analytical* or *numerical*.

A recent example for analytical models is Ref. [13], which adds predictability as an objective to the deterministic optimal power flow and solves it as a multi-objective problem and Ref. [14] which extends convexification used to solve an AC Optimal Power Flow (OPF) to the uncertainty set. Ref. [15] is another example of an analytical model and uses batches of scenarios for moment-matching of the distributions. In similar manner, Ref. [16] proposes using a batch sampling technique in order to update the probabilistic model. The model presented in subsequent sections of this paper will similarly focus on the strength of batch training.

Compared to analytical are purely numerical models, often based on Monte-Carlo approaches, more difficult to scale. This has resulted in a family of hybrids between analytical and numerical models which most

* Corresponding author.
E-mail address: markus.loschenbrand@sintef.no.

https://doi.org/10.1016/j.epsr.2021.107465
Received 25 September 2020; Received in revised form 16 June 2021; Accepted 9 July 2021
Available online 27 July 2021

of the recent publications on numerical models can be classified as. These models are also referred to as *approximation-based*.

To be specific on these approximations, Gaussians are central to most of the work on probabilistic power flows. Ref. [17] discussed these Gaussian approximations in comparison to a traditional Monte-Carlo approach. A recent example for an approximation-based approach is given by Ref. [18], which uses polynomials as approximations. Another recent example is provided by Ref. [19], which was later extended in Ref. [20]. The paper proposes utilization of Graphical Processor Unit (GPU) acceleration technology in order to improve approximation efficiency. In similar manner, approaches making use of state-of-the-art machine learning techniques, specifically deep learning, have been proposed. These models are specifically designed to most optimally utilize GPU acceleration. Recent examples of such are Ref. [21] and Ref. [22] which both use standard neural networks and mean squared error as loss functions to train these neural networks.

Ref. [23] and Ref. [24] both utilize Kullback-Leibler divergence, a distributional distance function, as a loss function instead of mean squared error, allowing for distributional fit of the neural networks instead of comparing point estimates. However, and as shown later in this paper, the tractability of this loss function is poor, opening the possibility for methods improving on such. This is the work that the here presented paper and the model below expands on. In addition to the models presented in the deep learning literature on probabilistic power flows there exist other types of neural networks that are specifically designed for distribution fitting. An example of such is deep generative models [25].Implementing such a neural network for probabilistic optimal power flows is the research gap the here presented work aims to fill.

The model presented in the proposed work combines the insights of the mentioned previous literature, by using a Monte Carlo based sampling technique and a loss function based on the Kullback-Leibler divergence to train a nonlinear Gaussian approximator (namely a Variational Autoencoder). This allows a better scalability of the Kullback-Leibler divergence compared to traditional approaches. Also, it opens probabilistic optimal power flow models up to implementation of new insights from the field of deep learning, such as the presented noise filtering via decoder-encoder models or utilization of batch gradients. Application of latter, i.e. batch gradient descent methods, can be seen similar to Ref. [16] (with the change that in the presented batch gradient descent these groups are random), which postulates that such grouping can drastically increase the performance of probabilistic optimal power flows. In addition and similar to the previously presented papers on deep learning, the model also profits from the state-of-the-art GPU acceleration techniques that modern machine learning methods use.

In its essence, the suggested model can be considered an automatization technique for analytical probabilistic models such as Refs. [15, 26] using non-linear deep-learning approximations instead of linear approximations, which are not scalable to train analytically. Conversely, the method compares to traditional stochastic power flow approaches such as Refs. [27,28] in that the proposed method is model-free (and thus doesn't need to be adjusted to new constraints added to the optimization problem) and uses output distributions whereas these traditional models are model-based and utilize fixed percentiles within the chance constraints.

In summary, this paper provides the following contributions:

- a model-free framework for probabilistic optimal power flows using deep learning.
- an introduction on how to utilize Stochastic Variational Inference using Gaussian (or other) distributions to represent the uncertain results of solving a probabilistic optimal power flow model.
- a demonstration on how this framework allows for parallelization in its training process
- an introduction on how to utilize a Variational Autoencoder in order to solve test systems with large uncertainty space.

The novelty of the proposed framework thus not only consists of the potential to train a probabilistic power flow, independent of the generative model, the utilized distributions or the topology of the power flow problem, the paper also proposes a number of methods that allow for parallelization in the training process. All of this is achieved via using a reformulation method, the so-called *reparameterization trick*, in order to train a probabilistic model in possibilistic manner.

Nomenclature

Index	
g	generation
L	load
n	sample index
E	encoder network
D	decoder network
e	training epoch
b	bus index
d	encoding dimension
Variables	
x	observations
U	voltage magnitude
θ	voltage angle
P_g	active power of generator
Q_g	reactive power of generator
z	latent variable
Functions	
C	system cost
c	generator cost
P	active power injection
Q	reactive power injection
F	line flow
p, q	distributions
$ELBO$	Evidence Lower BOund
MSE	Mean Squared Error
Parameters	
$\overline{F}$	line limit
$\underline{U}, \overline{U}$	voltage magnitude limits
$\underline{\theta}, \overline{\theta}$	voltage angle limits
$\underline{P}_g, \overline{P}_g$	active power generation limits
$\underline{Q}_g, \overline{Q}_g$	reactive power generation limits
N	total number of scenarios
φ	distribution parameters
$e^{\max}$	total number of training epochs
Uncertainties	
ξ	stochastic parameter
ξ'	scenario
ϵ	noise

2. Problem

The proposed probabilistic optimal power flow (OPF),problem is a, variation of the traditional deterministic AC optimal power flow:

$$C(\xi) = \min_{U,\theta,P_g,Q_g} c\left(P_g, \xi\right) \tag{1a}$$

$$\text{s.t. } P_g - P_L(\xi) = P(U,\theta) \tag{1b}$$

$$Q_g - Q_L(\xi) = Q(U,\theta) \tag{1c}$$

$$|F(U,\theta)| \leq \overline{F}(\xi) \tag{1d}$$

$$\underline{U} \leq U \leq \overline{U} \tag{1e}$$

$$\underline{\theta} \leq \theta \leq \overline{\theta} \tag{1f}$$

$$\underline{P}_g(\xi) \leq P_g \leq \overline{P}_g(\xi) \tag{1g}$$

$$\underline{Q}_g(\xi) \leq Q_g \leq \overline{Q}_g(\xi) \tag{1h}$$

The problem objective (1a) is the minimization of the system cost of generation. In this variation, the cost function considers uncertainty, e.g.

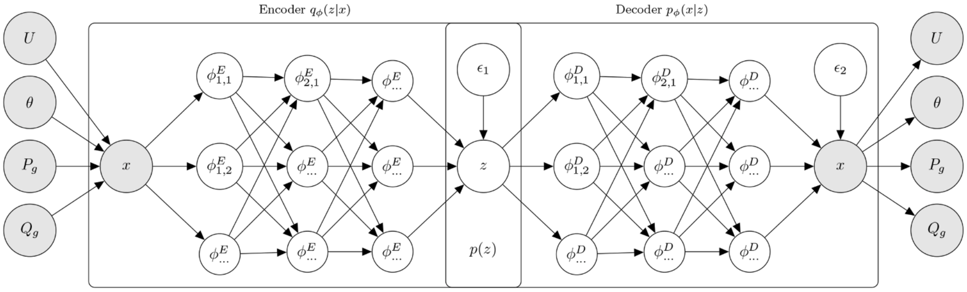

Fig. 1. Generative Model - Variational Autoencoder.

in the form of fuel price fluctuations.

The injections at a bus are defined via the balance of generation (demand) and load (supply). This is displayed for active power in constraint (1b) and for reactive power in (1c). In the considered example these inflexible loads are subject to uncertainty.

The limits of line flows are considered in constraint (1d). In the proposed probabilistic case, these line flows are considered subject to uncertainties such as line outages.

Constraint (1e) and (1f) respectively formulate the voltage magnitude and voltage angle limits which are kept deterministic in the presented model.

The generation limits formulated in constraint (1g) and constraint (1h) are considered subject to uncertainty such as fluctuations of renewable power generation or generator outages.

In the deterministic case, i.e. for a fixed ξ, this problem has been approached in a wide range of literature, which will not be expanded on further here. Instead, it will be assumed that the application of the AC OPF solver from the package Matpower [29] using Newton-Rhapson for stepwise improvement and Lagrangian relaxation to deal with constraints as seen in Ref. [30] provides an adequate platform to obtain solutions for the problem without uncertainty. Even though these solutions are not global, the result of the decision variables are here assumed to give an appropriate observation for the outcome of the cost minimal point in case of a single, deterministic outcome denoted as $\xi^{'}$:

$$C(\xi^{'}) = \min_{x} c(x|\xi^{'}) \tag{2}$$

Here, x represents a vector of the observations of the results for this specific scenario $\xi^{'}$:

$$x = \begin{bmatrix} U \\ \theta \\ P_g \\ Q_g \end{bmatrix} \tag{3}$$

Obtaining a possibilistic solution of problem (1), i.e. an expected value of the system cost, can be achieved in equal manner:

$$C(\xi^{'}) = \sum_{n=1}^{N} p\left(\xi^{'}_n\right) \min_{x_n} c\left(x_n|\xi^{'}_n\right) \tag{4}$$

Here, $p(\xi^{'}_n)$ symbolizes the probability of scenario $\xi^{'}_n$ whereas x_n displays the optimal solution of this given scenario. Due to each such solution x_n requiring a solution of its respective subproblem (1), solving for this expected value with a significantly large number of potential scenarios becomes intractable fast. This is especially problematic considering the AC power flow problem (1) itself is an NP-hard problem. Considering (continuous) distributions in parameters as the uncertainty set thus would lead to an unlimited number of potential scenarios. Thus, this means the extension of a computationally hardly tractable problem to an impossible problem:

$$C(\xi) = \int p(\xi) \min_{x_n} c(x_n|\xi) d\xi \tag{5}$$

Utilizing approximations as representations of these distributions has been at the core of probabilistic load flows and subsequently probabilistic optimal power flows since its initial appearance in literature [7]. In the presented method, function approximators from the family of deep learning, i.e. neural networks, will be utilized. Specifically, this paper explores how a generative, noise-filtering model can be applied to yield distribution approximations for the results of a probabilistic AC OPF with vast uncertainty space.

3. Model

Goal of the proposed model framework is to utilize a limited number of observations $x_n \forall n = 1, ..., N$ in order to step-wise approximate the distributions of the system results as shown in Eq. (5).

In this paper, this approximation is provided by a Variational Autoencoder (VAE) model [31]. This choice was made on one hand due to VAEs providing an established type of generative models in machine learning and on the other hand due to their noise-filtering capabilities. Nonetheless, future research expanding on this topic might explore other models from the family of generative machine learning models such as Generative Adversarial Networks (GANs). In addition to this, non-deep-learning approaches such as Bayesian networks could also be trained in similar manner [32].

This paper will however, due to the popularity of deep-learning based models, explore the proposed VAE model. The reason for this is that it provides a generalization of other deep generative models such as GAN style networks [33]. This model consists of three components:

- an encoder neural network that creates a noiseless representation of the observations.
- a latent variable z that is this noiseless representation.
- a decoder neural network that recreates observations under consideration of external noises ϵ.

3.1. Distributional representation

The proposed schematic for the probabilistic AC OPF problem is shown in Fig. 1. The output of the model is a parametric generative model $p_\varphi(x|z)$ for the distributions of the observations of the decision variables x depending on the latent distribution of the encoding $p(z)$. The basis for this distribution is the posterior distribution $q_\varphi(z|x)$ that defines the distribution of the latent variables depending on given observations x.

The goal of this problem is to define parameters for both distributions adequately to receive close to a perfect match $q_\varphi(z|x) = p_\varphi(z|x)$. However, the VAE will not reach this perfect match for all possible outcomes of x. This is by design, as the model aims to approximate the output to the input after accounting for the noise, resulting in a match of $q_\varphi(z|x) \approx p_\varphi(z|x)$ where the difference between the distributions is the random noise, which is thus filtered out.

```
(a) initialize $\phi$ randomly;
for $e = 1, \dots, e^{\max}$ do
    (b) sample $\xi_n \quad \forall n = 1, ..., N$;
    (c) solve problem (1) to receive $x_n \quad \forall n = 1, ..., N$;
    (d) sample noises $\epsilon_1$ and $\epsilon_2$;
    (e) sample $z_n \sim q_\phi(z_n|x_n)$;
    (f) sample $x_n \sim p_\phi(x_n|z_n)$;
    (g) calculate gradient estimators $\nabla_\phi ELBO(\phi)$;
    (h) based on this estimate update the weights and biases $\phi$ via an optimizer;
end
```

Algorithm 1. Stochastic Variational Inference for Probabilistic ACOPF

3.2. Loss function

This distribution fitting problem becomes the problem of finding the optimal distribution parameters φ which consist of the weights and biases of the neural networks that are the encoder and the decoder. As shown in Ref. [34], maximization of a bound based on Jensens inequality named the Evidence Lower BOund (ELBO) can be utilized to solve this problem:

$$\max_{\varphi} ELBO(\varphi) = \max_{\varphi} \mathbb{E}_{z\sim q_{\varphi}(z|x)}\left[\log p_{\varphi}(x|z)\right] - KL\Big(p(z)|q_{\varphi}(z|x)) \tag{6}$$

The ELBO is based on a non-negative distance measure of distributions, referred to as the Kullback-Leibler (KL) divergence:

$$KL(p(z)|q_{\varphi}(z|x)) = \mathbb{E}_{z\sim q_{\varphi}(z|x)}\left[\log p(z) - \log q_{\varphi}(z|x)\right] \tag{7}$$

In fact, and as described in detail in Ref. [34], the ELBO is a reformulation of the KL divergence that circumvents the intractability caused by term $\log q_{\varphi}(z|x)$. This is because in order to evaluate the KL divergence, the distribution $q_{\varphi}(z|x)$ would have to been known. Having this information would mean that the result of the VAE is already known, rendering the model redundant. Instead, however, the only information given is samples of x (and thus samples of $z \sim q_{\varphi}(z|x)$). As the decision vector x contains continuous variables at an infinite amount of samples of x, the distribution would be known and taking the KL divergence would be possible. As every sample is the solution of a deterministic optimal power flow and thus requires computation (i.e. computational time), instead of the KL divergence an approximation of the distribution in form of the VAE is derived.

3.3. Training process

Yielding the parameters of this approximation based on the ELBO (as a negative loss function), in specific optimization of the distribution parameters φ, is conducted via Stochastic Variational Inference (SVI) [34]. SVI is a Monte-Carlo method which allows to conduct sub-sampling of data, here observations x, in order to update these distribution parameters. In other words, it allows to utilize a limited number of N observations in order to update the posterior distributions. In order to simplify training, the algorithm can make use of the *reparameterization trick* [31]. In this concept, the noise parameters ϵ are considered known and thus deterministic for a given observation. This separates the noise and the parameters of the distributions from each other, allowing to learn the parameters via back-propagation similar to traditional machine learning applications. An illustrative example in the appendix explains this technique in detail.

Algorithm 1 sums the training process for the decision problem up. As it can be observed, this algorithm makes use of training on a number of N samples at the same time, via batch gradient descent [35].

3.4. Computational efficiency

The computational efficiency of the proposed framework is its strength: training the network weights and biases φ is a GPU-centric task, which profits from recent advantages in GPU-acceleration technology in step (g) and (h). Opposing to this, finding the AC OPF solutions x is a CPU-centric task. However, as the framework allows for training on a number observations in parallel via batch gradient descent, these N solutions of step (c) can be computed in parallel, allowing the proposed framework to profit from multiprocessing/multithreading as well. This is highlighted by the sampling and training process as shown in Fig. 2. This also shows that the limit of the performance of this algorithm is provided by the CPU and GPU memory capacities and not the tact speed of the processing units as it is the case in traditional Monte Carlo methods. In summary, Algorithm 1 thus utilizes the potential of modern

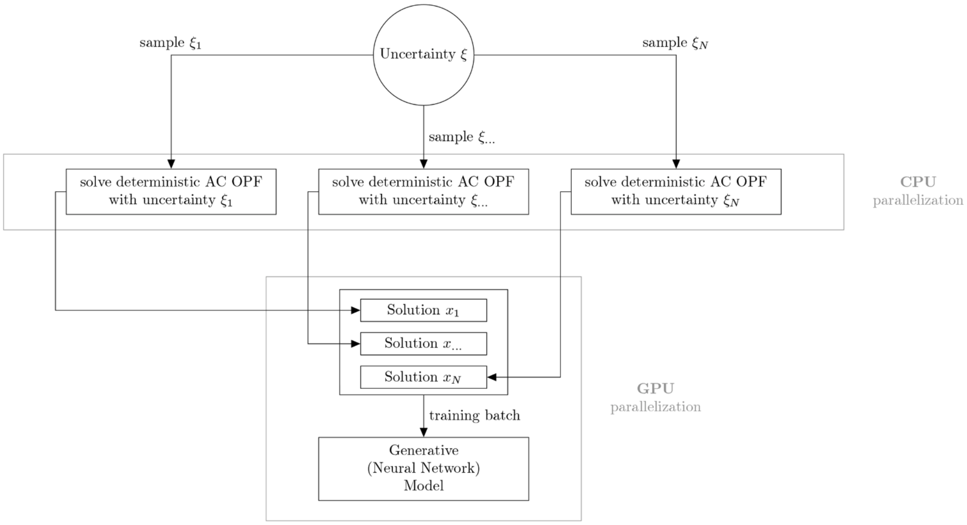
Uncertainty ξ
sample ξ_1
sample ξ_N
sample $\xi_{...}$
solve deterministic AC OPF with uncertainty ξ_1
solve deterministic AC OPF with uncertainty $\xi_{...}$
solve deterministic AC OPF with uncertainty ξ_N
CPU parallelization
Solution x_1
Solution $x_{...}$
Solution x_N
training batch
Generative (Neural Network) Model
GPU parallelization

Fig. 2. Generative Model Training Step.

neural network principles such as batch gradient descent to allow making use of multi processor architecture and transforming the iterative distribution fitting process into a parallelized version. This is supported by using an optimizer that is specifically designed for batch gradient descent (such as Adam [36], the optimizer used in the case study below) over traditional methods such as stochastic gradient descent in order to derive the parameters of the neural network.

Verbally, the training process can be summarized in short as 'sample stochastic parameters and noises' → 'solve AC OPF for the given samples' → 'update weights of the approximator using the ELBO function and a gradient descent method'.

After yielding the weights, the decoder network can be utilized to generate samples for x. This is done by only using the right side of the VAE as shown in Fig. 1. Algorithm 2 demonstrates how to draw n samples from the resulting generator model.

It has to be remarked that the two distribution noises ϵ_1 and ϵ_2 relate to the specific distributions for the encoding and the output. These two distributions might be different. In the here presented case studies the distributions were chosen to be normal for the encoding and truncated normal for the generated output.

4. Case studies

The case studies utilize stochastic power flow problems based on the IEEE test cases and additional assumptions such as line failure rates in systems with high shares of renewables [37]. The **case configuration**, i. e. the adjustments on the original IEEE test cases, was the following:

1. **variable fuel prices** - all cost factors for the first two thirds of the generation units was subject to a uniform distribution between 95% and 105% of its original values.
2. **renewable generation** - the maximum generation capacities of the last third of the generation units was normally distributed with a mean of 125% and a variance of 7.5% of its original values. In case a sampled value would dip below the minimum generation would result in this plant being shut down (in this specific sample). All cost factors for these plants were set to 0.
3. **line outages** - each line was assigned an outage probability of 0.1% (the line was removed for this case).
4. **generation unit outage** - in similar manner, each generation unit was assigned an outage probability of 0.1%.
5. **demand fluctuations** - both active and reactive loads were assumed to fluctuate according to a normal distribution with a mean of 100% and a variance of 10% of its original values.

Other than these adjustments introducing a large uncertainty space, the other parameters of the respective IEEE test systems were kept similar to the respective original, with no additions such as ramping included. In regards to the model, drawing a single sample from these uncertainties would give an instance of ξ, drawing several would result in a batch of samples.

In order to deal with infeasible solutions caused by a specific instance of ξ (as could be the case caused by e.g. too many line or generator failures), infeasible solutions for specific parameters have to be filtered out. This is because for an infeasible state, there is no information on the optimal results of the variables. The Gaussian distributions yielded by the model thus only represents the distribution over the feasible space. Future research might build on this understanding and utilize a classification model to define which inputs lead to infeasible outputs and thus nest the proposed model within this classification model. Similar could be achieved with a Bernoulli distribution in parallel to the proposed Gaussian.

The **hyperparameters** used were dependent on the neural networks used. These decoder and encoder networks were both traditional feedforward neural networks with a layer size of 700 and six layers (with sigmoidal function units as activation functions). The encoding distributions $p(z)$ was assumed to be Gaussian and the distribution of the generated samples $p_\varphi(x_n|z_n)$ a truncated Gaussian with the limits of the observation minimums and maximums. Both neural networks thus also each required two separate linear output layers as their seventh layers, dedicated respectively to the location and scale. As described above, the chosen optimizer for step (h) of Algorithm 1 was *Adam* [36], a second order batch gradient descent algorithm. In addition to that, a so-called scheduler was applied in order to apply a learning rate decay of 33% after each 33th episode.

In traditional neural network models, inputs are normalized to a mean of zero and a variance of one. In the given example, however, such a normalization is not possible as the total potential range of the variables is uncertain. To circumvent this, the inputs were scaled by a fixed scaler. This scaler was calculated by drawing an initial sample and calculating the means of the variables. This is displayed in the flowchart of the training process presented in Fig. 3.

Two different cases were analyzed, based on the IEEE 9 bus and 118 bus test system with the previously mentioned modifications. The convergence curves are shown in Figs. 4 and 5 for the respective cases and a batch size of three. The comparison baseline chosen was a traditional Monte Carlo (MC) approach which used the same sample batches to fit truncated Gaussians. Comparing to traditional stochastic power flow techniques, this baseline can be considered an adaption of the

Algorithm 2. Output Generator for Probabilistic ACOPF

```
(a) sample noises $\epsilon_1$ and $\epsilon_2$;
(b) sample $z_n \sim p(z_n)$;
(c) sample $x_n \sim p_\phi(x_n|z_n)$;
```

Monte Carlo approach introduced in Ref. [27] but using parameters of non-linear distributions instead of uncertainty margins. Yielding a single sample from the AC OPF problem required approximately 2 seconds on an Intel i7-8850H CPU@ 2.60 GHz and training the neural networks around 1 second per episode on a Nvidia Quadro P2000. Using a VAE over the traditional model had no measurable impact on the model training times. s the model allows to be updated with batches of samples at the same time, it can be stated that the bottleneck in training can be attributed to the speed of obtaining a single AC OPF solution. In fact, this is a characteristic that the proposed model has in common with similar techniques. However, compared to traditional stochastic optimization techniques, this model can be trained in batches, allowing for solving multiple instances of the AC OPF problem in parallel for multiple samples, as shown previously in Fig. 2. Thus it not only allows for more complex models than the traditional linear models used in analytical models, it also allows for parallelization of the optimization steps compared to traditional iterative methods.

The training history of this loss function, i.e. the ELBO, demonstrates that the example is able to create a nearly perfect fit of $q_\varphi(z|x) \approx p_\varphi(z|x)$, suggesting that the model is able to *learn* the distribution of the noise and adequately fit the output to the given approximation - the latent distribution.

Numerical results of the VAE model are given for the 9 bus test system in Fig. 6 and for the 118 bus test system in Fig. 7.

In both cases, introducing uncertainty has the highest impact on the voltage angles which deviate significantly from the previous deterministic results. For several variables and busses, either the mean of or the entire batch of generated samples was equivalent to the deterministic example. This shows the impact of adding the given uncertainty to the deterministic system, which does not impact the entire system evenly. For most of the cases there was a significant deviation from the deterministic results. This suggests a significant impact of adding uncertainty to the problem over the given deterministic formulation.

The results also underline the importance of a deep-learning approach. Table 1 illustrates that there are correlations within the different variables. These correlation coefficients are higher for the smaller system, meaning that uncertainty on one end of the system has a higher impact on the rest of the system if the system is smaller. This result is intuitive, as a fluctuation in a generator is less significant for the results of the system if the generator is small compared to the total system capacity. Appropriately modeling these effects of the systems is taken care of by deep-learning models, as they generate the samples in connection to the other samples, meaning that statistical dependence is considered. In a more traditional model, these effects would potentially be neglected.

Further, even though the results of the test cases is provided in form of samples, the formulation of the case study shows an additional benefit of the algorithm. As explained in the appendix, as SVI updates its parameters based on the *reparameterization trick* by sampling, no information on the distributions of the uncertain parameters is needed. Instead, it would be entirely possible to utilize only samples or scenarios instead of distributions. It is further possible to use a mixture of these uncertainty representations as inputs. In practical situations there might be information on the distribution for some parameters and only scenarios/samples for others. Thus, instead of supplying distributions, the model learns the probabilistic output of the optimal power flow problem by learning the latent distribution from the given samples.

Nonetheless, irrespective of the inputs being distributions or samples, the result of applying the proposed method is a distribution that allows single samples to be taken from. A practical application of such is provided by the line flows for the respective cases given in Figs. 8 and 9. Similar to traditional stochastic powerflows via chance constraints, quantiles can be defined by drawing samples and calculating these quantiles. Further, the percentiles can be dynamically defined. Whereas in chance constrained stochastic power flows these have to be defined prior to solving the problem, in the here presented model they can be

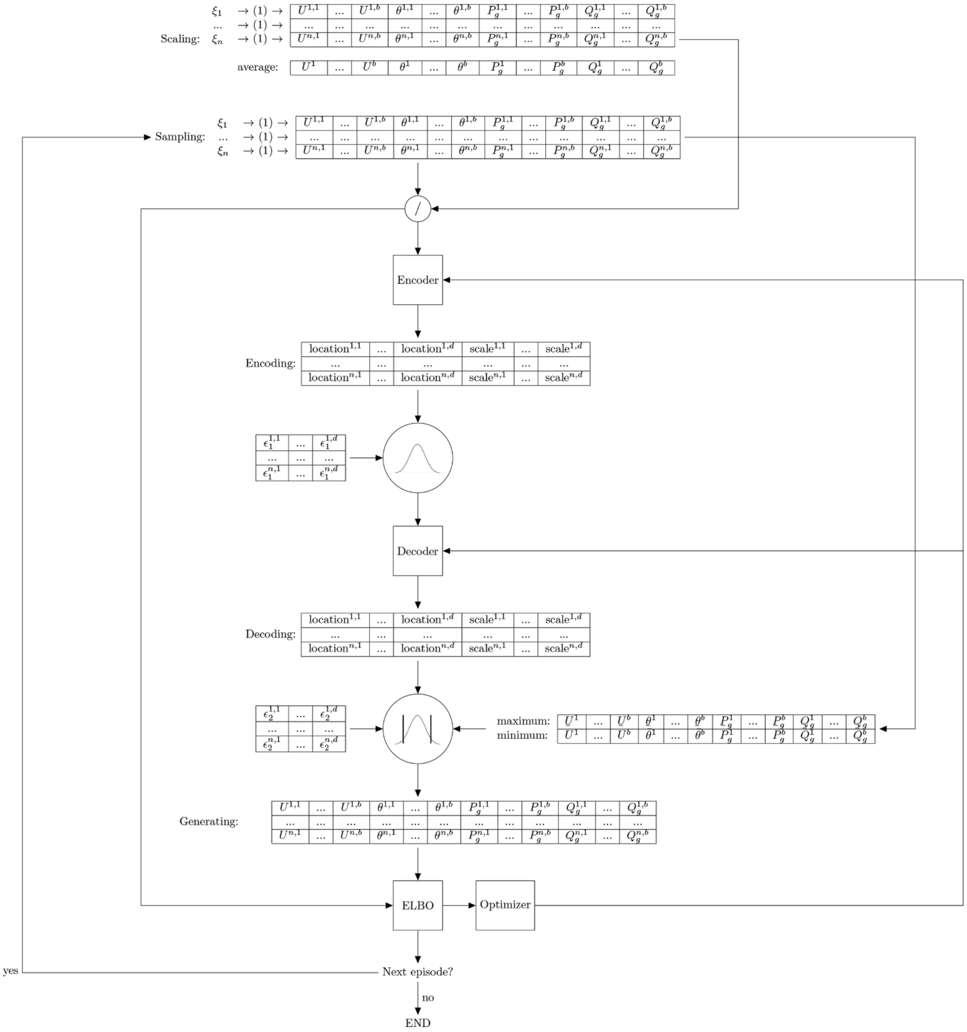
Scaling:
ξ_1 → (1) →
... → (1) →
ξ_n → (1) →
$U^{1,1}$ ... $U^{1,b}$ $\theta^{1,1}$ ... $\theta^{1,b}$ $P_g^{1,1}$ ... $P_g^{1,b}$ $Q_g^{1,1}$ ... $Q_g^{1,b}$
$U^{n,1}$ ... $U^{n,b}$ $\theta^{n,1}$ ... $\theta^{n,b}$ $P_g^{n,1}$ ... $P_g^{n,b}$ $Q_g^{n,1}$ ... $Q_g^{n,b}$
average:
U^1 ... U^b θ^1 ... θ^b P_g^1 ... P_g^b Q_g^1 ... Q_g^b
Sampling:
/
Encoder
Encoding:
location^{1,1} ... location^{1,d} scale^{1,1} ... scale^{1,d}
location^{n,1} ... location^{n,d} scale^{n,1} ... scale^{n,d}
$\epsilon_1^{1,1}$ ... $\epsilon_1^{1,d}$
$\epsilon_1^{n,1}$ ... $\epsilon_1^{n,d}$
Decoder
Decoding:
$\epsilon_2^{1,1}$ ... $\epsilon_2^{1,d}$
$\epsilon_2^{n,1}$ ... $\epsilon_2^{n,d}$
maximum:
minimum:
Generating:
ELBO
Optimizer
Next episode?
yes
no
END

Fig. 3. Model Flowchart.

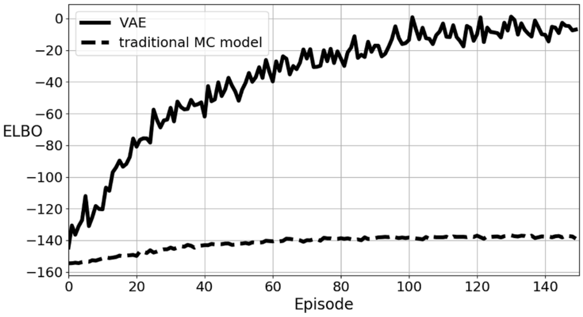

Fig. 4. Evidence LOwer Bound for IEEE 9 bus problem.

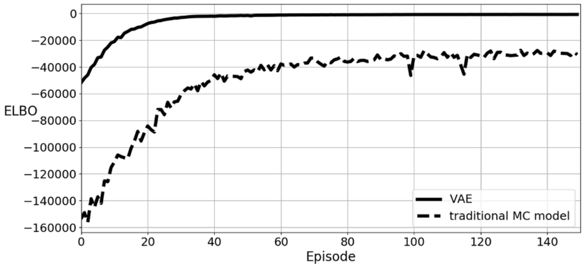

Fig. 5. Evidence LOwer Bound for IEEE 118 bus problem.

applied after solving the problem (by calculating the given percentile on the samples taken).

Similar can be done for other aspects of the solution. For example, distributions of the nodal balances in specific busses or a distribution of the total system cost (or subsets of it, i.e. a distribution of the cost of generators in a specific section of the system) can be obtained in similar manner.

5. Conclusion

This paper demonstrates how to apply Stochastic Variational Inference to train a probabilistic deep learning approximator for a stochastic power flow problem. The method allows automatic training of such non-linear approximators that allow Gaussian approximations of AC optimal power flows under high uncertainty spaces.

To do so, a loss function (or a bound, *misused* as a loss function) based on the Kullback-Leibler divergence is utilized, named the Evidence Lower BOund. By utilizing the so-called *reparameterization trick*, the resulting probabilistic model can be trained via sampling, similar to possibilistic models. This training can be achieved via traditional backpropagation, or in other words, application of methods from the family of gradient descent algorithms (also referred to as *optimizers*). The result is a model that is able to automatically update distributions based on samples of decision variables received by iteratively solving batches

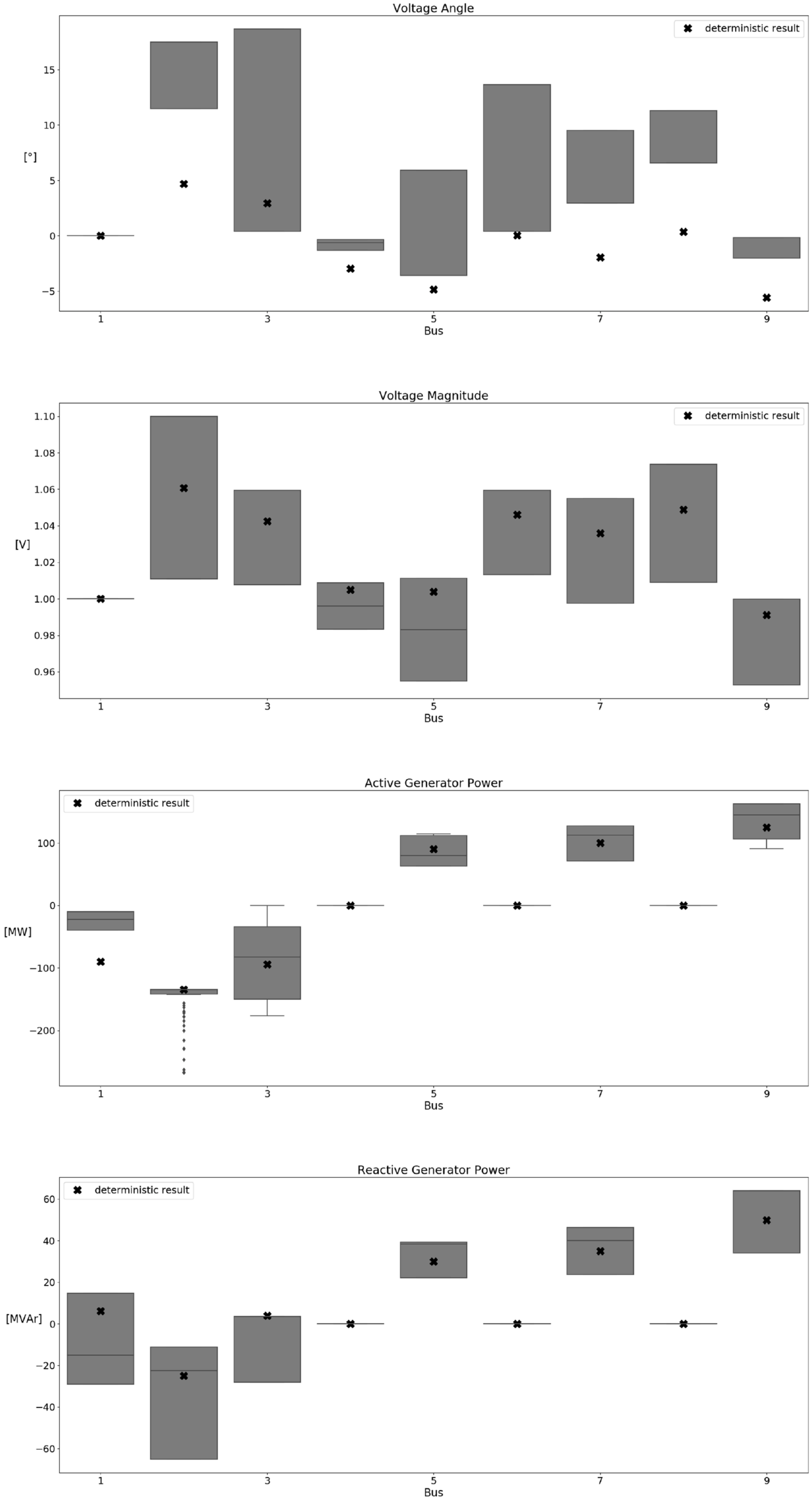

Fig. 6. Results for the 9 bus case.

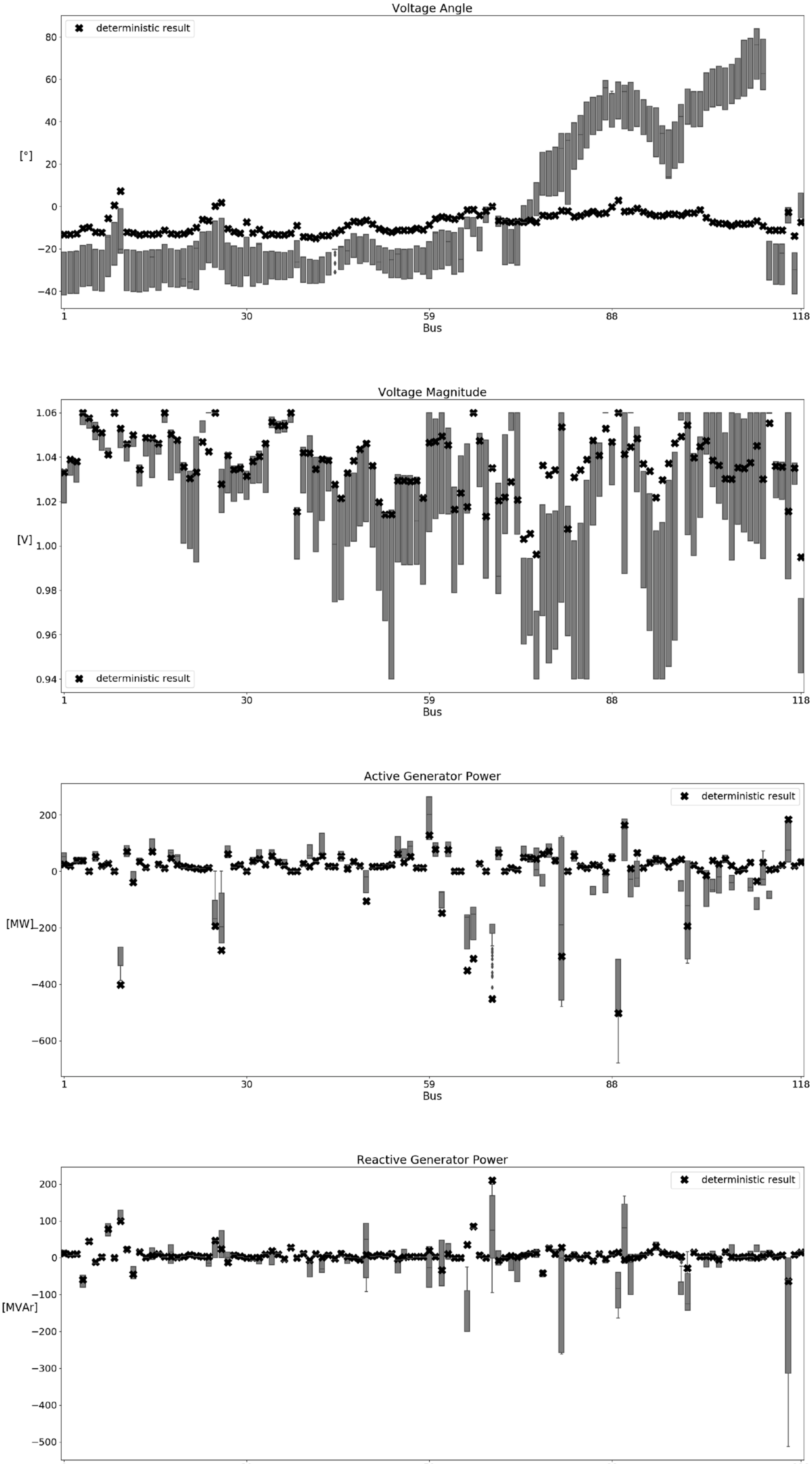

Fig. 7. Results for the 118 bus case.

Table 1
Average Correlation Coefficients of generated Samples.

	$\theta - U$	$\theta - P$	$\theta - Q$	$U - P$	$U - Q$	$P - Q$
9 bus	0.6	− 0.5	− 0.42	− 0.45	− 0.41	0.84
118 bus	− 0.1	− 0.23	− 0.05	− 0.2	− 0.06	0.06

of deterministic power flows with sampled parameters. This is demonstrated in the paper by giving examples of the IEEE 9 and 118 bus systems extended by uncertainties in cost, capacity, line availability, generator availability and demand. Compared to traditional Monte Carlo methods, this batch-wise training process allows for parallelization in solving the AC OPF with uncertainty samples as well as training the neural network used to approximate the distributions.

In addition to this, the paper suggests several avenues for further academic use. The main avenue is provided by other models from the family of generative machine learning models that could similarly be applied on the problem and tested for their potential. Another option is to alter the proposed deep learning model itself. This could be done via e.g. changing the type of distributions used in the latent approximation or changing the utilized networks itself (as e.g. application of multi-period power flows might consider recurrence in the utilized neural networks). The case study itself focuses on large uncertainty spaces where the model proves to be capable of approximating the non-linear relations within the AC OPF and provide a distribution of the probabilistic solution to the given system that allows individual samples to be taken from. In summary, the proposed framework provides a technique to efficiently train non-linear function approximations that allow deriving Gaussian distributions of the results of non-linear stochastic optimal power flows. Based on the insights from this paper, the proposed starting points for future research can be formulated as:

- extend the problem to multi-period power flows
- consider non-linear scheduling and associated ramping constraints
- develop a tailored deep learning formulation (e.g. evaluate the capabilities of Generative Adversarial Networks against the proposed model)
- reformulate the given model to a model-based formulation (and potentially combine this with other approximation techniques for the non-linear AC power flow equations)
- expand the proposed model to the infeasible space of the uncertainty set

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence

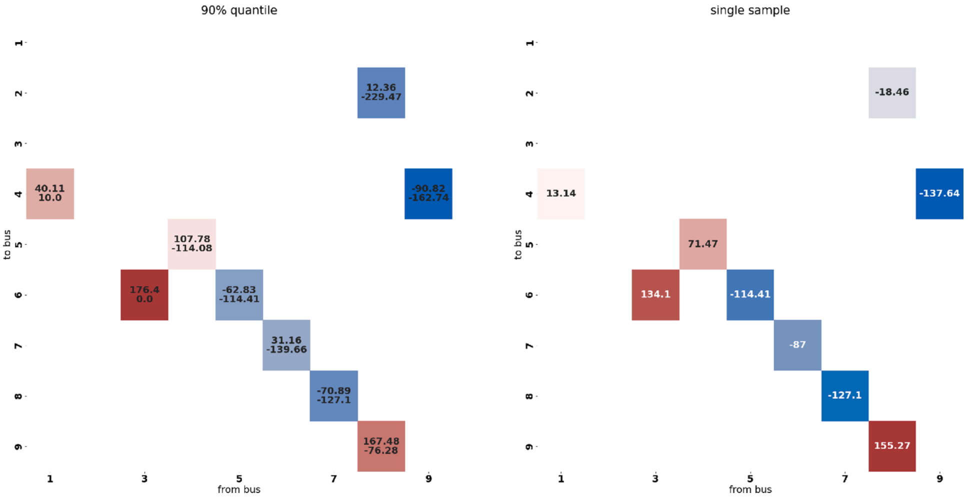

Fig. 8. Line flows [MW] for the 9 bus case.

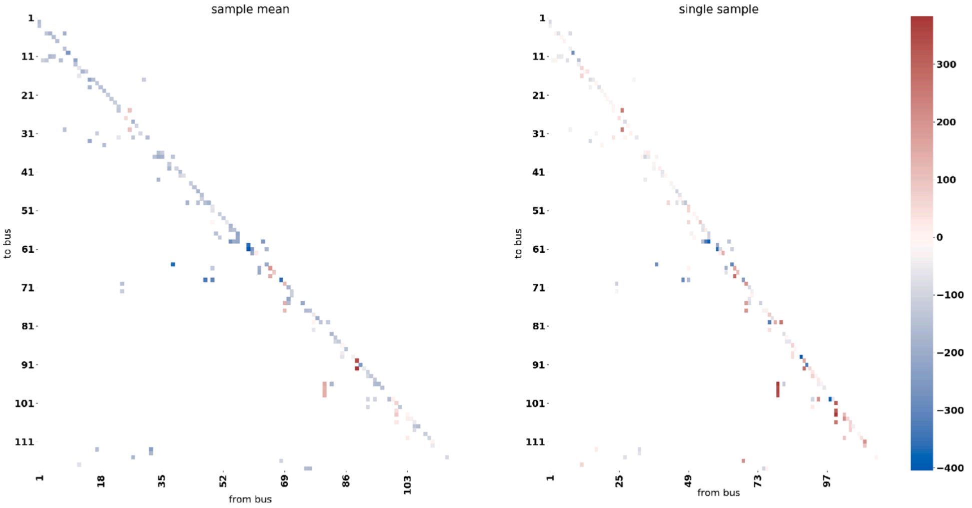

Fig. 9. Line flows [MW] for the 118 bus case.

the work reported in this paper.

Acknowledgment

The author would like to thank the project consortium of the SINTEF research project “Modeling Flexible Resources in Smart Distribution Grids - ModFlex” (255209/E20) and the Norwegian Research Council for supporting this work.

Appendix A. The reparameterization trick

The so-called *reparameterization trick* as introduced in Ref [31] provides a core concept of the proposed framework that bridges the possibilistic result in Eq. (4) with the probabilistic result in Eq. (5). The concept will therefore be explained here based on an illustrative example of a simple linear regression problem using the mean squared error as a loss function. The example can be trained via any gradient descent method. Further, and despite the examples' simplicity, the concept can be expanded in similar manner to the VAE model presented in this paper that instead uses a Bayesian neural network model, the ELBO as a loss function and a second order stochastic gradient descent algorithm.

Assumed be the problem of fitting latent variable z to a linear function of variable x, whereas both allow for sampling denoted as z_n and x_n respectively. In addition, the linear approximation considers no deterministic constant and a standard-normally distributed noise ϵ as a stochastic constant:

$$z_n \approx \varphi x_n + \epsilon \quad \forall n \tag{A.1}$$

The problem is that of finding the most fitting model parameter φ that gives the best approximation for this linear regression. This can be done by defining a loss function, which for this example will be the Mean Squared Error (MSE):

$$MSE_\varphi(z,x) = \sum_n (z_n - \varphi x_n - \epsilon_n)^2 \tag{A.2}$$

In traditional linear regression, finding this optimal fit $\underset{\varphi}{argmin} MSE_\varphi(z,x)$ could thus be achieved via gradient steps in the form of:

$$\frac{\partial MSE_\varphi(z,x)}{\partial\varphi} = \frac{-\sum_n 2x_n(-\epsilon - \varphi x_n + z_n)}{n} \tag{A.3}$$

In the given problem, however, this gradient is not deterministic but instead shows a normally distributed noise. However, sampling the noise (in the given example by $\epsilon \sim \mathcal{N}(0,1)$) and considering the resulting noise as a constant leads to a deterministic result for the loss function *MSE* and thus gives a deterministic gradient $\frac{\partial MSE_\varphi(z,x)}{\partial\varphi}$. For a large enough *minibatch size* of n and drawing as many samples ϵ_n, this gradient can thus be approximated as the following:

$$\frac{\partial MSE_\varphi(z,x)}{\partial\varphi} \approx \frac{-\sum_n 2x_n(-\epsilon_n - \varphi x_n + z_n)}{n} \tag{A.4}$$

In its essence this is nothing else than approximating the stochastic gradient by its expected value. In the provided example, this then means that the weight φ of the linear approximation can then be updated by deterministic gradient descent, for example with a learning rate of 0.3:

$$\varphi := \varphi + 0.3\frac{\partial MSE_\varphi(z,x)}{\partial\varphi} \tag{A.5}$$

References

[1] K.N. Hasan, R. Preece, J.V. Milanović, Existing approaches and trends in uncertainty modelling and probabilistic stability analysis of power systems with renewable generation, Renew. Sustain. Energy Rev. 101 (2019) 168–180, https://doi.org/10.1016/j.rser.2018.10.027.

[2] U.H. Ramadhani, M. Shepero, J. Munkhammar, J. Widén, N. Etherden, Review of probabilistic load flow approaches for power distribution systems with photovoltaic generation and electric vehicle charging, Int. J. Electr. Power Energy Syst. 120 (2020) 106003, https://doi.org/10.1016/j.ijepes.2020.106003.

[3] Z. Zhuo, E. Du, N. Zhang, C. Kang, Q. Xia, Z. Wang, Incorporating massive scenarios in transmission expansion planning with high renewable energy penetration, IEEE Trans. Power Syst. 35 (2) (2020) 1061–1074, https://doi.org/10.1109/TPWRS.2019.2938618.

[4] S. Soltan, P. Mittal, H.V. Poor, Line failure detection after a cyber-physical attack on the grid using Bayesian regression, IEEE Trans. Power Syst. 34 (5) (2019) 3758–3768, https://doi.org/10.1109/TPWRS.2019.2910396.

[5] T. Liu, Y. Liu, J. Liu, L. Wang, L. Xu, G. Qiu, H. Gao, A Bayesian learning based scheme for online dynamic security assessment and preventive control, IEEE Trans. Power Syst. 35 (5) (2020) 4088–4099, https://doi.org/10.1109/TPWRS.2020.2983477.

[6] H. Wang, Z. Yan, X. Xu, K. He, Probabilistic power flow analysis of microgrid with renewable energy, Int. J. Electr. Power Energy Syst. 114 (2020) 105393, https://doi.org/10.1016/j.ijepes.2019.105393.

[7] B. Borkowska, Probabilistic load flow, IEEE Trans. Power Apparatus Syst. (3) (1974) 752–759, https://doi.org/10.1109/TPAS.1974.293973.

[8] S. Mashayekh, M. Stadler, G. Cardoso, M. Heleno, S.C. Madathil, H. Nagarajan, R. Bent, M. Mueller-Stoffels, X. Lu, J. Wang, Security-constrained design of isolated multi-energy microgrids, IEEE Trans. Power Syst. 33 (3) (2018) 2452–2462, https://doi.org/10.1109/TPWRS.2017.2748060.

[9] J. Jithendranath, D. Das, J.M. Guerrero, Probabilistic optimal power flow in islanded microgrids with load, wind and solar uncertainties including intermittent generation spatial correlation, Energy 222 (2021) 119847, https://doi.org/10.1016/j.energy.2021.119847.

[10] J.S. Ferreira, E.J. de Oliveira, A.N. de Paula, L.W. de Oliveira, J.A. Passos Filho, Optimal power flow with security operation region, Int. J. Electr. Power Energy Syst. 124 (2021) 106272, https://doi.org/10.1016/j.ijepes.2020.106272.

[11] J. Wei, Y. Zhang, J. Wang, X. Cao, M.A. Khan, Multi-period planning of multi-energy microgrid with multi-type uncertainties using chance constrained information gap decision method, Appl. Energy 260 (2020) 114188, https://doi.org/10.1016/j.apenergy.2019.114188.

[12] P. Chen, Z. Chen, B. Bak-Jensen, Probabilistic load flow: a review. 2008 Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, IEEE, Nanjing, China, 2008, pp. 1586–1591, https://doi.org/10.1109/DRPT.2008.4523658.

[13] S. Galvani, S. Rezaeian Marjani, Optimal power flow considering predictability of power systems, Electr. Power Syst. Res. 171 (2019) 66–73, https://doi.org/10.1016/j.epsr.2019.02.011.

[14] Q. Li, Uncertainty-aware three-phase optimal power flow based on data-driven convexification, IEEE Trans. Power Syst. 36 (2) (2021) 1645–1648, https://doi.org/10.1109/TPWRS.2021.3050926.

[15] C. Wang, C. Liu, F. Tang, D. Liu, Y. Zhou, A scenario-based analytical method for probabilistic load flow analysis, Electr. Power Syst. Res. 181 (2020) 106193, https://doi.org/10.1016/j.epsr.2019.106193.

[16] W. Sun, M. Zamani, H.-T. Zhang, Y. Li, Probabilistic optimal power flow with correlated wind power uncertainty via Markov Chain Quasi-Monte-Carlo sampling, IEEE Trans. Ind. Inf. 15 (11) (2019) 6058–6069, https://doi.org/10.1109/TII.2019.2928054.

[17] A. Schellenberg, W. Rosehart, J. Aguado, Cumulant-based probabilistic optimal power flow (P-OPF) with Gaussian and gamma distributions, IEEE Trans. Power Syst. 20 (2) (2005) 773–781, https://doi.org/10.1109/TPWRS.2005.846184.

[18] H. Sheng, X. Wang, Probabilistic power flow calculation using non-intrusive low-rank approximation method, IEEE Trans. Power Syst. 34 (4) (2019) 3014–3025, https://doi.org/10.1109/TPWRS.2019.2896219.

[19] M. Abdelaziz, GPU-OpenCL accelerated probabilistic power flow analysis using Monte-Carlo simulation, Electr. Power Syst. Res. 147 (2017) 70–72, https://doi.org/10.1016/j.epsr.2017.02.022.

[20] M. Abdelaziz, OpenCL-accelerated probabilistic power flow for active distribution networks, IEEE Trans. Sustain. Energy 9 (3) (2018) 1255–1264, https://doi.org/10.1109/TSTE.2017.2781148.

[21] M. Xiang, J. Yu, Z. Yang, Y. Yang, H. Yu, H. He, Probabilistic power flow with topology changes based on deep neural network, Int. J. Electr. Power Energy Syst. 117 (2020) 105650, https://doi.org/10.1016/j.ijepes.2019.105650.

[22] Y. Yang, Z. Yang, J. Yu, B. Zhang, Y. Zhang, H. Yu, Fast calculation of probabilistic power flow: a model-based deep learning approach, IEEE Trans. Smart Grid 11 (3) (2020) 2235–2244, https://doi.org/10.1109/TSG.2019.2950115.

[23] Y. Xu, Z. Hu, L. Mili, M. Korkali, X. Chen, Probabilistic power-flow calculation based on a novel Gaussian process emulator, IEEE Trans. Power Syst. (2020) 1–4, https://doi.org/10.1109/TPWRS.2020.2983603.

[24] M. Xiang, N. Feng, W. Dai, J. Yu, Z. Yang, Y. Yang, H. Yu, H. Xiang, Improving the scalability of deep neural network for probabilistic power flow, Int. Trans. Electr. Energy Syst. 30 (5) (2020), https://doi.org/10.1002/2050-7038.12322.

[25] D.P. Kingma, S. Mohamed, D.J. Rezende, M. Welling, Semi-supervised learning with deep generative models, Adv. Neural Inf. Process. Syst. (2014) 3581–3589.

[26] H. Wang, Z. Yan, M. Shahidehpour, X. Xu, Q. Zhou, Quantitative evaluations of uncertainties in multivariate operations of microgrids, IEEE Trans. Smart Grid 11 (4) (2020) 2892–2903, https://doi.org/10.1109/TSG.2020.2971689.

[27] L. Roald, G. Andersson, Chance-constrained AC optimal power flow: reformulations and efficient algorithms, IEEE Trans. Power Syst. 33 (3) (2018) 2906–2918, https://doi.org/10.1109/TPWRS.2017.2745410.

[28] M. Vrakopoulou, M. Katsampani, K. Margellos, J. Lygeros, G. Andersson, Probabilistic security-constrained AC optimal power flow. 2013 IEEE Grenoble Conference, IEEE, Grenoble, France, 2013, pp. 1–6, https://doi.org/10.1109/PTC.2013.6652374.

[29] R.D. Zimmerman, C.E. Murillo-Sanchez, R.J. Thomas, MATPOWER: steady-state operations, planning, and analysis tools for power systems research and Education, IEEE Trans. Power Syst. 26 (1) (2011) 12–19, https://doi.org/10.1109/TPWRS.2010.2051168.

[30] H. Wang, R.D. Zimmerman, R.J. Thomas, On computational issues of market-based optimal power flow, IEEE Trans. Power Syst. 22 (3) (2007) 1185–1193.

[31] D.P. Kingma, M. Welling, Auto-Encoding Variational Bayes, arXiv preprint arXiv:1312.6114 (2014).

[32] D. Koller, N. Friedman, Probabilistic graphical models: principles and techniques, in: Adaptive computation and machine learning, MIT Press, Cambridge, MA, 2009.

[33] L. Mescheder, S. Nowozin, A. Geiger, Adversarial variational bayes: unifying variational autoencoders and generative adversarial networks, arXiv preprint arXiv:1701.04722 [cs] (2018).

[34] M.D. Hoffman, Stochastic variational inference, J. Mach. Learn. Res. 14 (1) (2013) 1303–1347.

[35] S. Ruder, An overview of gradient descent optimization algorithms, arXiv preprint arXiv:1609.04747 (2017).

[36] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980 [cs] (2014).

[37] O.R. Solheim, T. Trotscher, G. Kjolle, Wind dependent failure rates for overhead transmission lines using reanalysis data and a Bayesian updating scheme. 2016 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), IEEE, Beijing, 2016, pp. 1–7, https://doi.org/10.1109/PMAPS.2016.7764104.