Carefully Biased Data Generation Using Variational Autoencoders:

What Makes a Good Synthetic Dataset?

Simon Tindemans, Chenguang Wang, Kutay Bölat, Ensieh Sharifnia
Delft University of Technology

Session on Research and Education Efforts on Uncertainty Quantification and Modeling
IEEE PES General Meeting 2024, Seattle, 25 July 2024

The ideal "data synthesis machine"

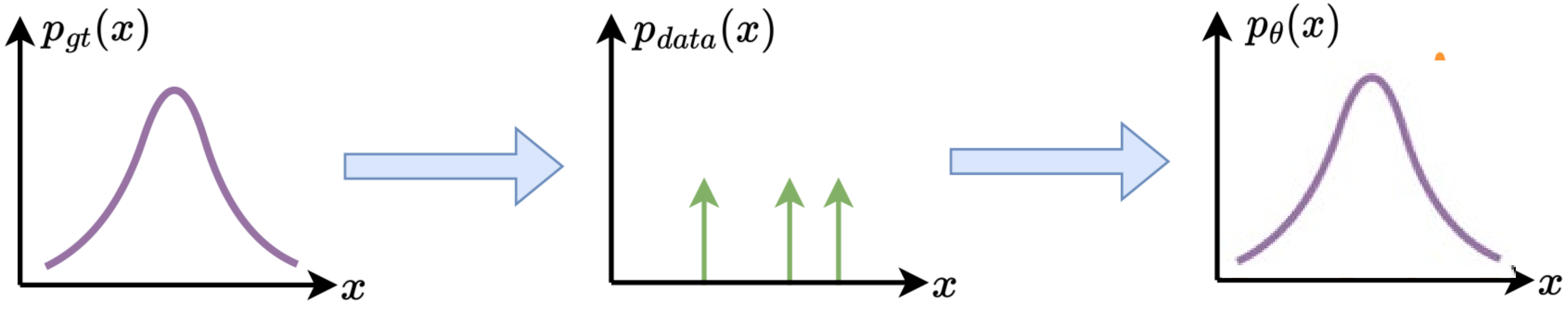
1. **General purpose utility (fidelity)**
   a. **Individually**, samples should be 'realistic'
   b. **Collectively**, samples should resemble the population

2. **Task-specific utility**: study results should be the same

tension

3. The machine should **generalise** from the training data

4. There may be **privacy/ownership concerns** over individual data points



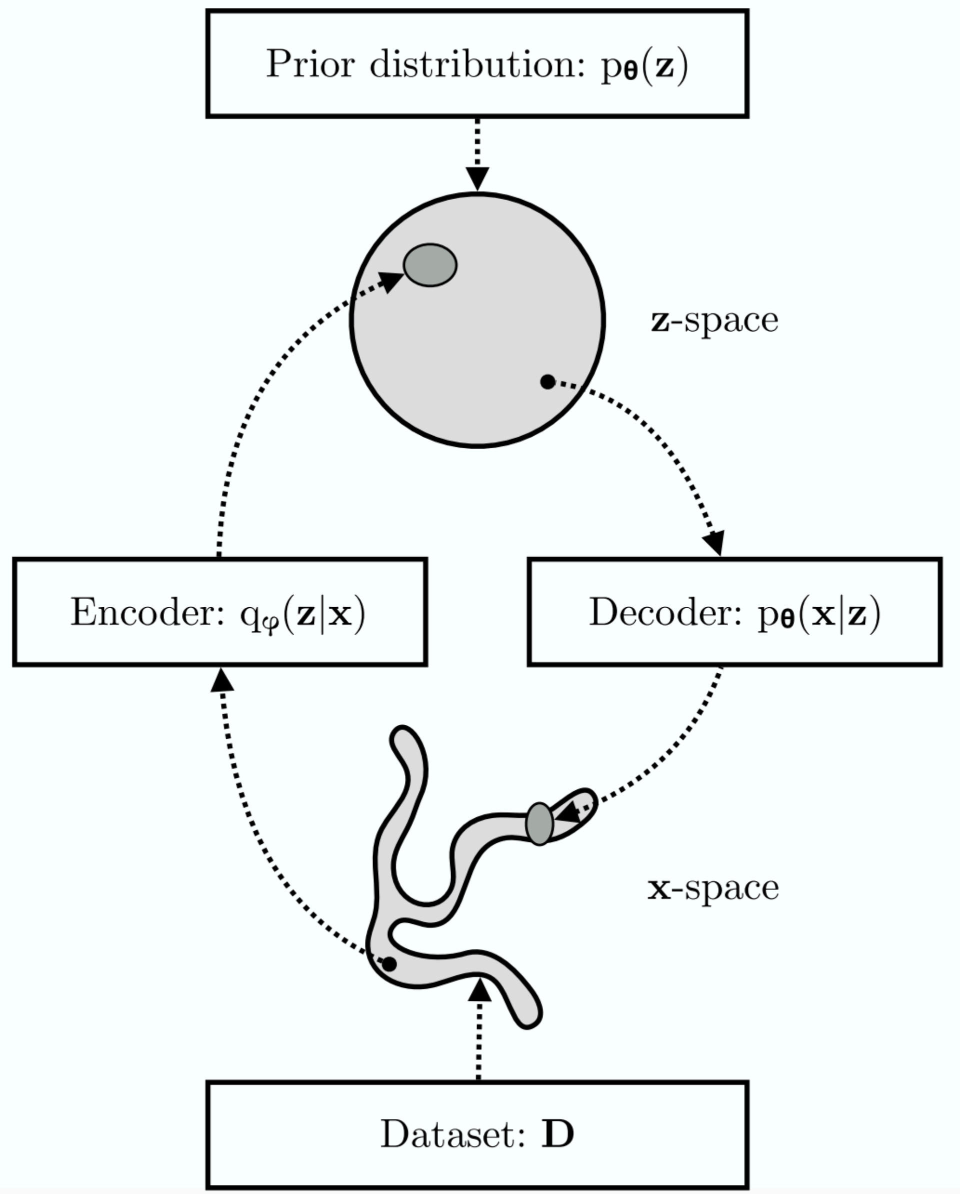
distribution inference

The variational autoencoder (VAE)

Introduced in Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*

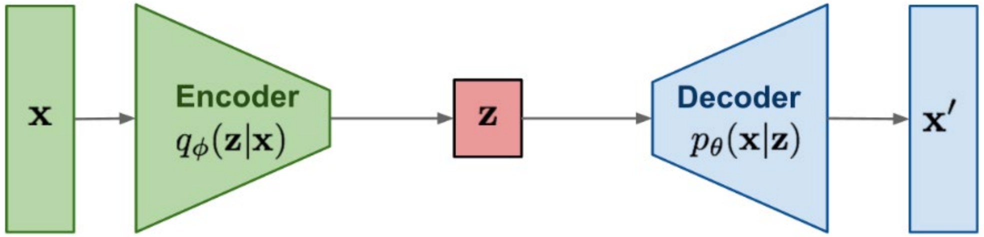
- Common assumptions:
  - Prior distribution is Gaussian
  - Probabilistic mappings are Gaussian

- Three interpretations:
  - A probabilistic autoencoder neural network
  - Latent variable models parametrized by NNs
  - An infinite Gaussian mixture model

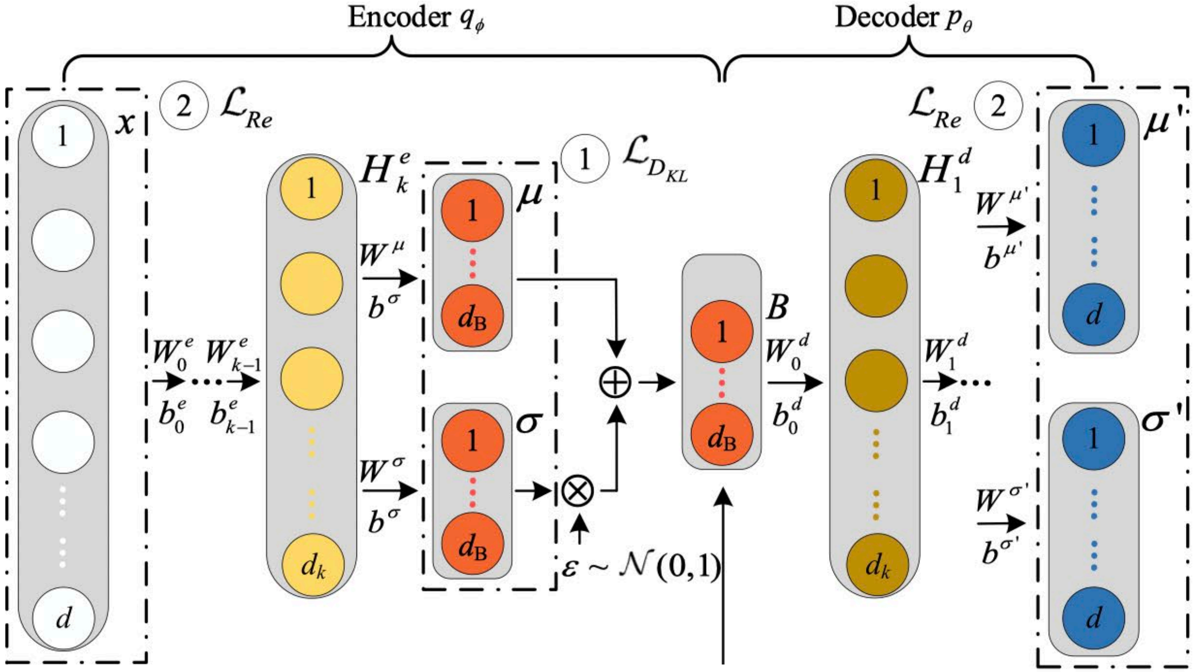


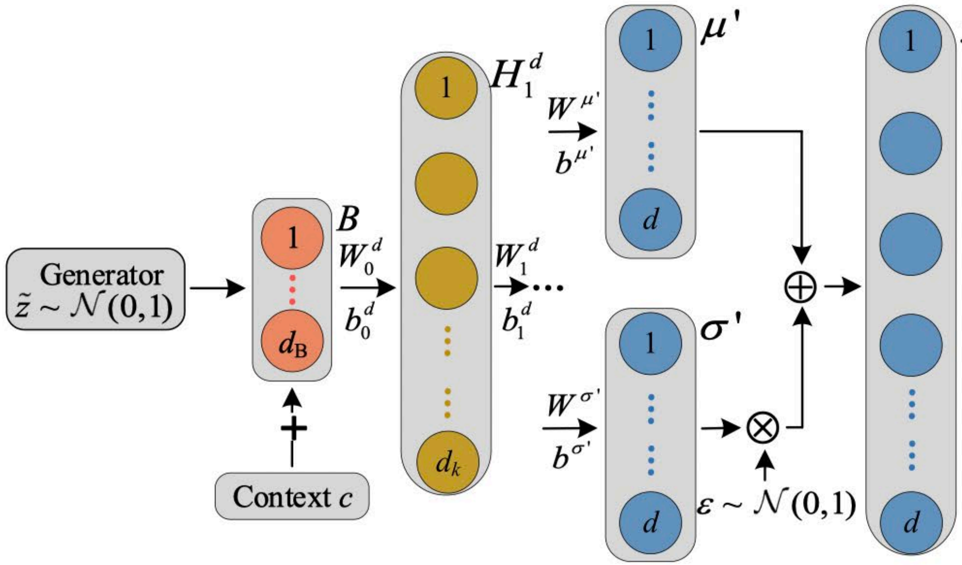
Kingma, D. P., & Welling, M. (2019). An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, *12*(4), 307–392.

A VAE neural network



https://lilianweng.github.io/posts/2021-07-11-diffusion-models/





(b)

Encoding

$$\begin{pmatrix}\mu\\ \sigma\end{pmatrix} = \begin{pmatrix}W^\mu\\ W^\sigma\end{pmatrix} \left(a(W_k^e(\dots a(W_1^e(x,c)+b_1^e)\dots)+b_k^e)\right) + \begin{pmatrix}b^\mu\\ b^\sigma\end{pmatrix}, \tag{1a}$$

$$z = \mu + \epsilon \odot \sigma\,, \tag{1b}$$

Decoding

$$\begin{pmatrix}\mu'\\ \sigma'\end{pmatrix} = \begin{pmatrix}W^{\mu'}\\ W^{\sigma'}\end{pmatrix} (\dots a(W_1^d(z,c)+b_1^d)\dots) + \begin{pmatrix}b^{\mu'}\\ b^{\sigma'}\end{pmatrix}, \tag{2a}$$

$$\hat{x} = \mu' + \epsilon \odot \sigma'\,, \tag{2b}$$

Training the neural network

ELBO: $$\phi^*, \theta^* = \underset{\phi,\theta}{\operatorname{argmax}} \, \mathbb{E}_{p_{data}(\boldsymbol{x})}\left[\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \log\big(p_\theta(\boldsymbol{x}|\boldsymbol{z})\big) - D_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \parallel p_\theta(\boldsymbol{z})\right)\right]$$

Use gradient descent to minimise the loss

$$\mathcal{L} = \mathcal{L}_{D_{KL}} + \mathcal{L}_{Re}.$$

$$\mathcal{L}_{D_{KL}} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{d}(-1 + \sigma_{i,j}^2 + \mu_{i,j}^2 - \log \sigma_{i,j}^2),$$

$$\mathcal{L}_{Re} = -\sum_{i=1}^{n} \mathbb{E}_{Z \sim q_\phi(z|x_i)}[\log_{P_\theta}(x_i|Z)]$$

$$\approx \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{d}((x_{i,j} - \mu'_{i,j})^2/\sigma'^2_{i,j} + \log \sigma'^2_{i,j}) + \frac{nd}{2}\log 2\pi,$$

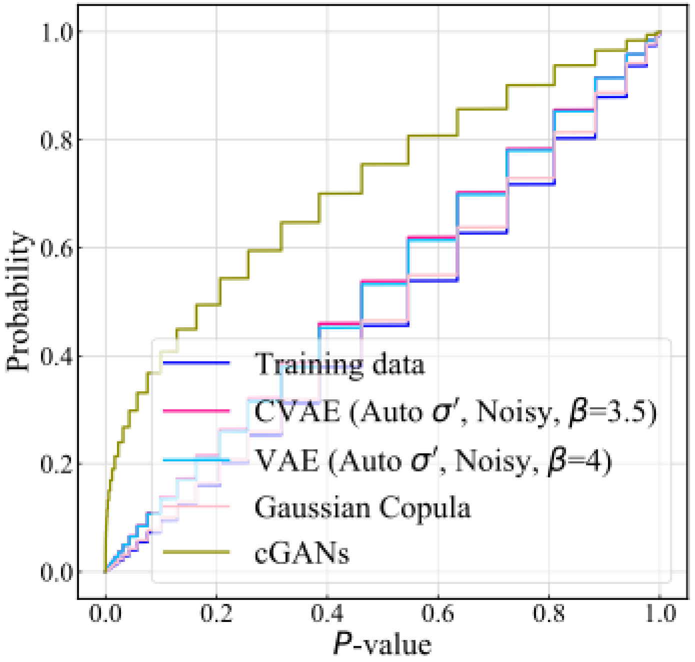
Results: measure sample quality

1. **General purpose utility (fidelity)**
   a. **Individually**, samples should be 'realistic'
   b. **Collectively**, samples should resemble the population

Dataset: hourly electricity demand of 32 European countries (5 years)

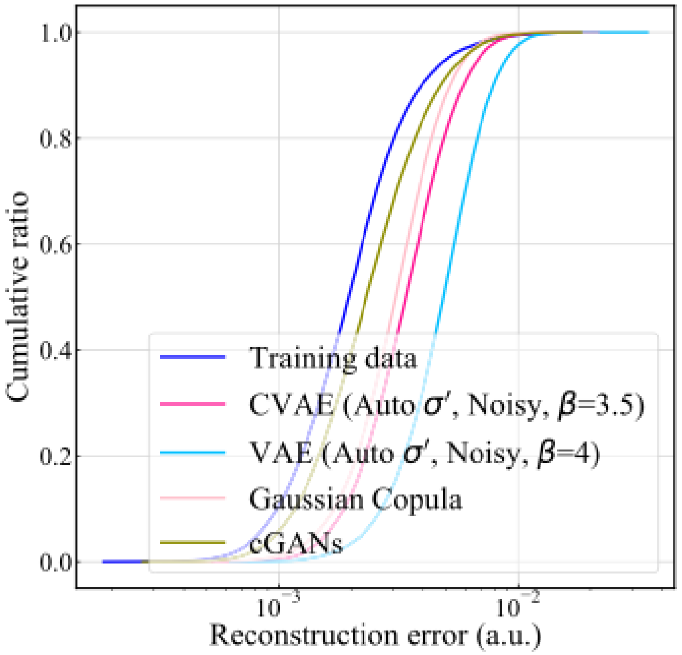
Univariate marginals

two-sample Kolmogorov-Smirnov test



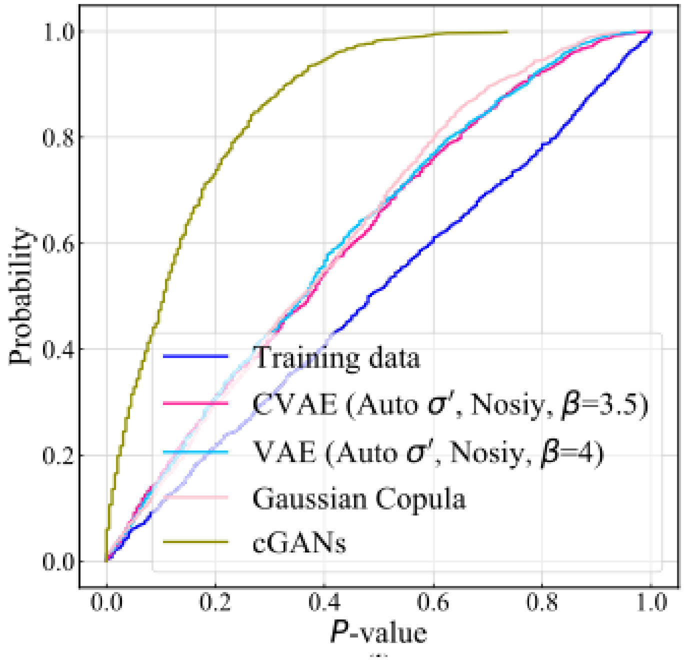
Multivariate snapshots

autoencoder reconstruction



Multivariate distribution

two-sample energy test (200 permutations)



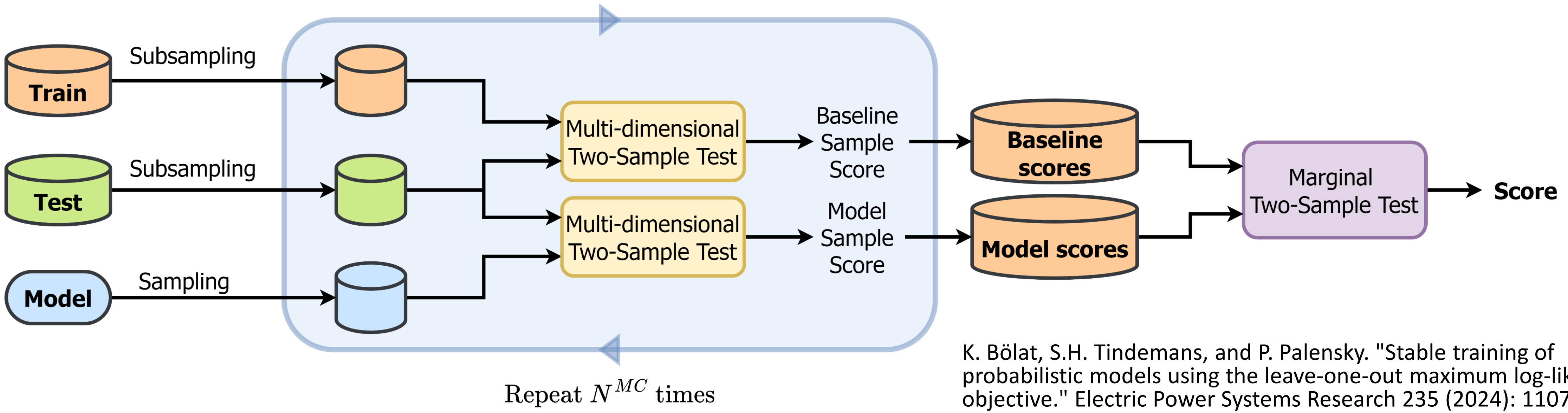
Wang, C., Sharifnia, E., Gao, Z., Tindemans, S. H., & Palensky, P. (2022). Generating multivariate load states using a conditional variational autoencoder. *Electric Power Systems Research*, *213*, 108603

Two-sample test (v2)

Quantifying similarity between real and synthetic data

Ingredients

- Multi-dimensional two-sample tests: MMD, Energy
- Marginal two-sample tests: KS, CvM, ΔMean



K. Bölat, S.H. Tindemans, and P. Palensky. "Stable training of probabilistic models using the leave-one-out maximum log-likelihood objective." Electric Power Systems Research 235 (2024): 110775.

Challenge: importance sam

Regular Monte Carlo

$$\hat{r}_{MC} = \frac{1}{m}\sum_{i=1}^{m} h(x_i) \qquad X_i \sim p(x)$$

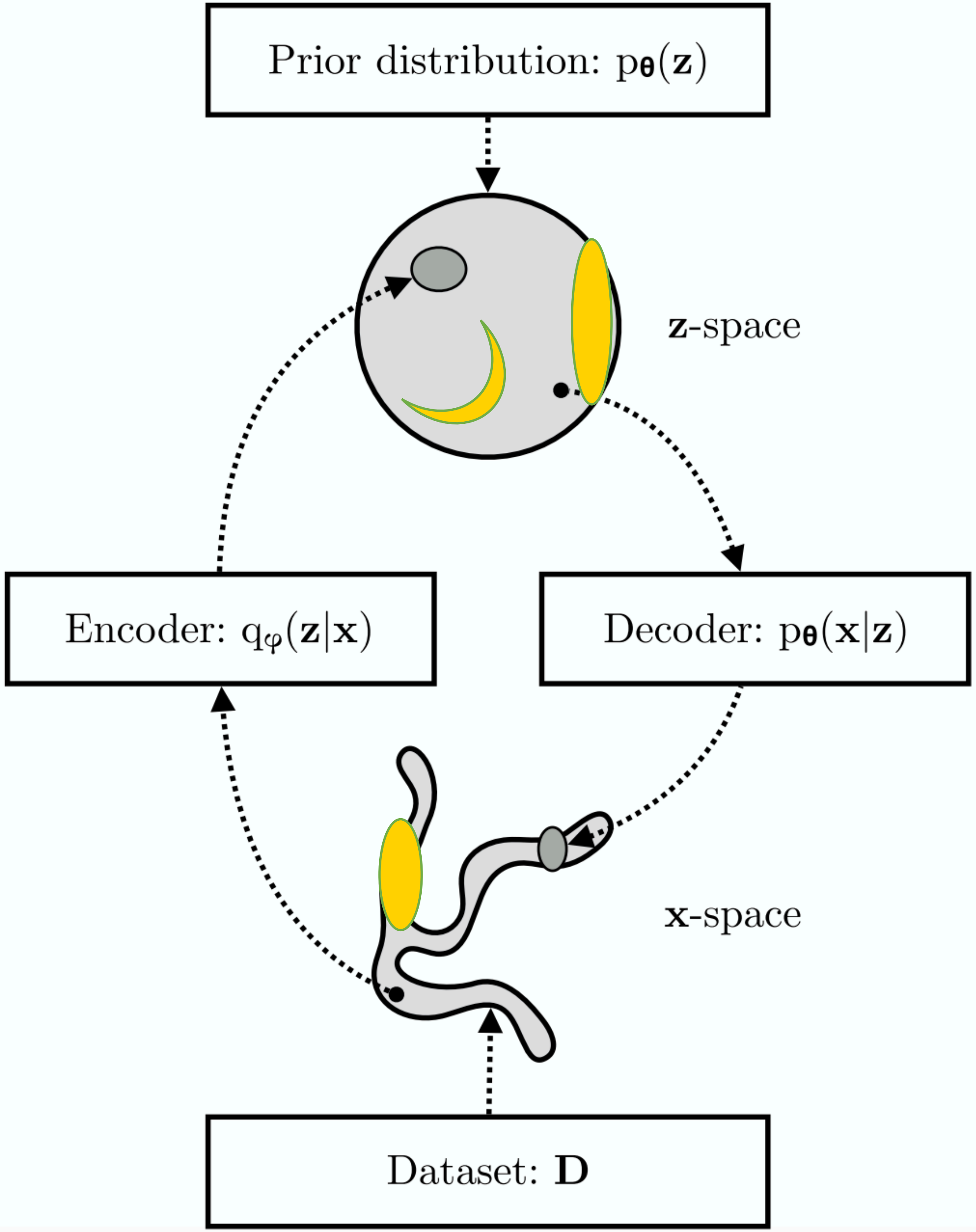
Importance sampling Monte Carlo

$$\hat{r}_{IS} = \frac{1}{m}\sum_{i=1}^{m} h(x_i')w(x_i') \qquad X'_i \sim q(x)$$

$$w(x) = \frac{p(x)}{q(x)}$$

optimal importance sampling distribution

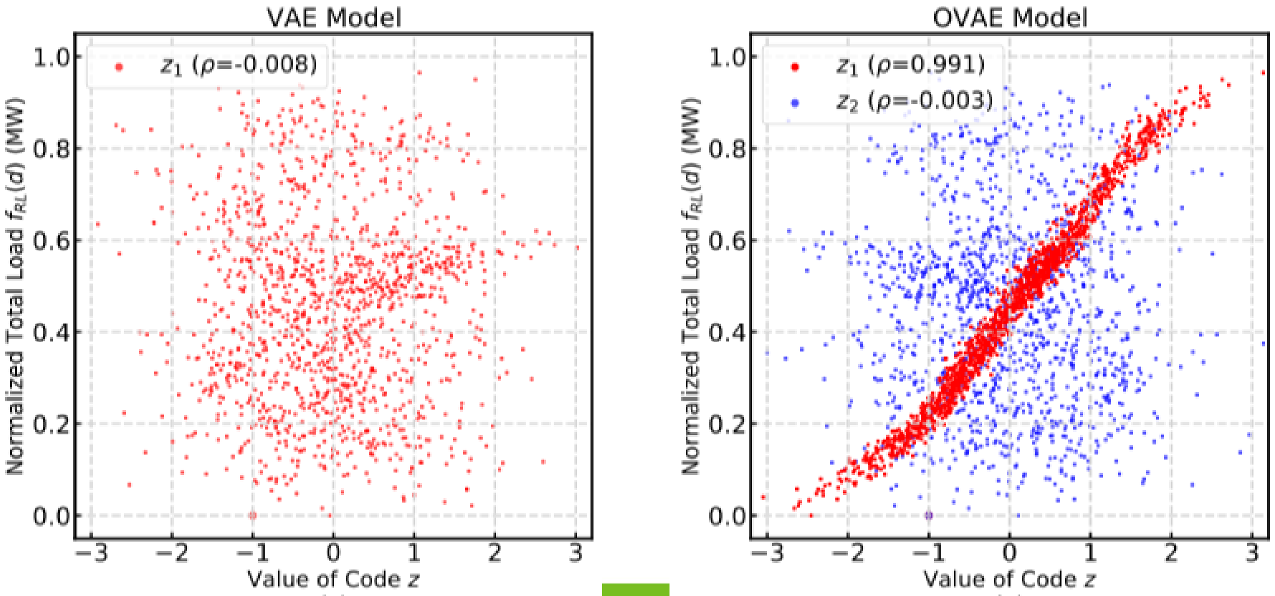
$$q^*(x) = \frac{h(x)p(x)}{E_{X\sim p(x)}[h(X)]} = \frac{h(x)p(x)}{r}$$

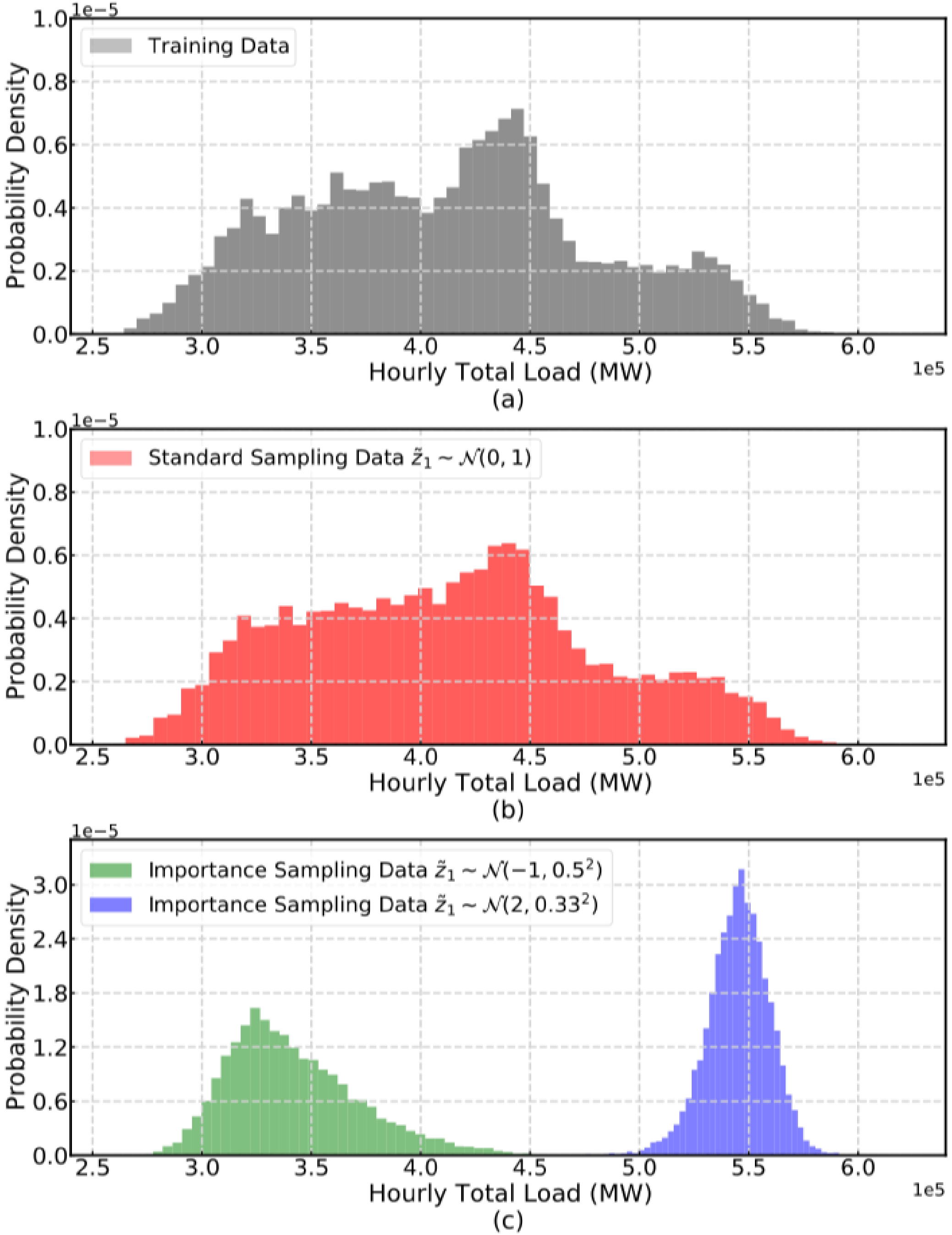


Kingma, D. P., & Welling, M. (2019). An Introduction to Va
Autoencoders. *Foundations and Trends® in Machine Learn*

Oriented VAE

- Add penalty $\mathcal{L}_{ori}$ to align orientation of z_1 and feature $f(x)$
- Train by minimising $\mathcal{L} = \beta\mathcal{L}_{D_{KL}} + \mathcal{L}_{Re} + \mathcal{L}_{Ori}$



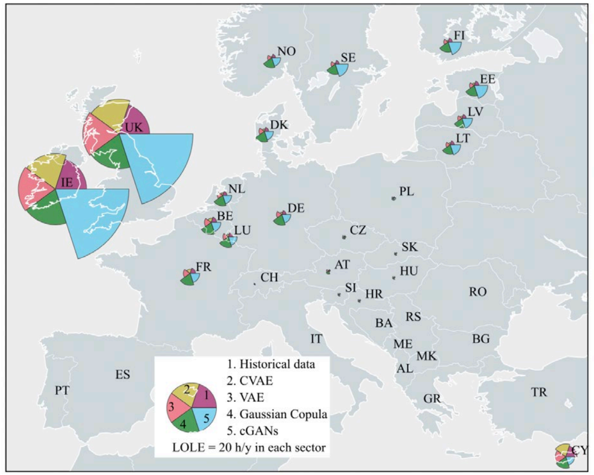


Results

2. Task-specific utility: study results should be the same

Multi-area resource adequacy assessment

- Generative models increase tail risks
- Similar results across models
- Importance sampling yields speedups



Importance sampling

$$q(z_1) = \alpha \mathcal{N}(z_1; 0, 1) + (1-\alpha)\mathcal{N}(z_1; \mu_{IS}, \sigma_{IS}^2),$$
$$q(z_{i\neq 1}) = \mathcal{N}(z_i; 0, 1),$$

RESOURCE ADEQUACY RESULTS AND IMPORTANCE SAMPLING SPEEDUP

Load model	μ_{IS}	σ_{IS}	Time (s)	LOLE (h/y)	EENS (MWh/y)	LOLE Speedup	EENS Speedup
Historical load	-	-	4319	10.79(31)	$1.190(47) \times 10^4$	n/a	n/a
OVAE-total load	0	1	4155	18.76(40)	$4.50(17) \times 10^4$	n/a	n/a
OVAE-EENS 5% training	0	1	4236	18.45(40)	$3.20(10) \times 10^4$	n/a	n/a
OVAE-EENS 20% training	0	1	4130	18.66(40)	$3.46(12) \times 10^4$	n/a	n/a
OVAE-EENS 30% training	0	1	4131	18.16(40)	$3.00(9) \times 10^4$	n/a	n/a
OVAE-total load	2.25	0.68	4666	18.43(20)	$4.137(40) \times 10^4$	3.5	14.5
OVAE-EENS 5% training	2.04	0.58	4524	18.06(20)	$3.333(35) \times 10^4$	3.8	8.7
OVAE-EENS 20% training	2.00	0.48	4512	18.55(19)	$3.530(42) \times 10^4$	4.2	7.3
OVAE-EENS 30% training	1.92	0.60	4512	18.17(25)	$3.217(43) \times 10^4$	2.3	5.0

PSA: looking for a postdoc

Generation of synthetic grid states

- 3-year postdoc (1+2) at TU Delft, the Netherlands
- Supervisors: Jochen Cremer, Simon Tindemans

- Vacancy text: https://tinyurl.com/5f5r44uz

Summary and next steps

Summary

- The variational autoencoder provides an intuitive class of generative models
- Multivariate data distributions are captured very well – according to the selected metrics
- Latent space manipulation can be used for importance sampling

Next steps

- Investigate fundamental instability of the ELBO objective
- Embed physical constraints in data generation
- Define and ensure privacy for consumption profile generation

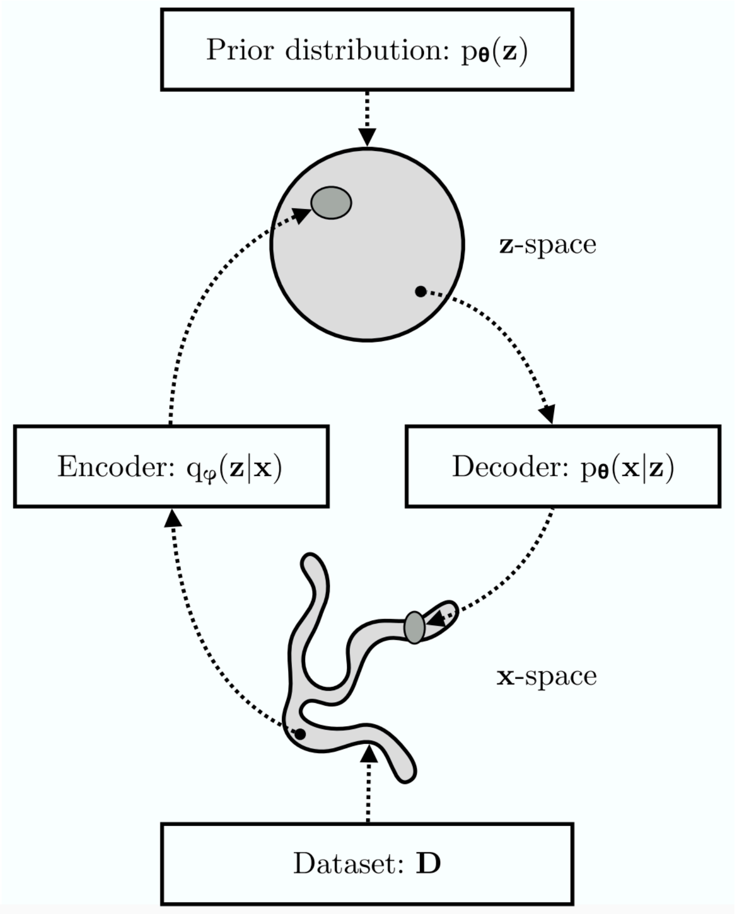
This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sktodowska-Curie grant agreement No 956433.

Related papers

- C. Wang, S. H. Tindemans and P. Palensky, "Generating Contextual Load Profiles Using a Conditional Variational Autoencoder," ISGT Europe 2022, Novi Sad.
- C. Wang, E. Sharifnia, Z. Gao, S. Tindemans, P. Palensky, “Generating Multivariate Load States Using a Conditional Variational Autoencoder,” Electric Power Systems Research 213, 108603 (2022).
- C. Wang, E. Sharifnia, S.H. Tindemans, P. Palensky, “Targeted Analysis of High-Risk States Using an Oriented Variational Autoencoder”, arXiv:2303.11410
- K. Bölat, S.H. Tindemans, and P. Palensky. "Stable training of probabilistic models using the leave-one-out maximum log-likelihood objective." Electric Power Systems Research 235 (2024): 110775.

Bonus slides

Evidence lower bound (ELBO)



for some z

$$\begin{aligned}\log p_\theta(\boldsymbol{x}) &= \log\left(\frac{p_\theta(\boldsymbol{x}|\mathbf{z})p_\theta(\mathbf{z})}{p_\theta(\mathbf{z}|\boldsymbol{x})}\right)\\ &= \log\left(\frac{p_\theta(\boldsymbol{x}|\mathbf{z})p_\theta(\mathbf{z})}{p_\theta(\mathbf{z}|\boldsymbol{x})}\frac{q_\phi(\mathbf{z}|\boldsymbol{x})}{q_\phi(\mathbf{z}|\boldsymbol{x})}\right)\\ &= \log(p_\theta(\boldsymbol{x}|\mathbf{z})) - \log\left(\frac{q_\phi(\mathbf{z}|\boldsymbol{x})}{p_\theta(\mathbf{z})}\right) + \log\left(\frac{q_\phi(\mathbf{z}|\boldsymbol{x})}{p_\theta(\mathbf{z}|\boldsymbol{x})}\right)\end{aligned}$$

$$\begin{aligned}\log p_\theta(\boldsymbol{x}) &= \mathbb{E}_{q_\phi(\mathbf{z}|\boldsymbol{x})}\log p_\theta(\boldsymbol{x})\\ &= \mathbb{E}_{q_\phi(\mathbf{z}|\boldsymbol{x})}\log(p_\theta(\boldsymbol{x}|\mathbf{z})) - \mathbb{E}_{q_\phi(\mathbf{z}|\boldsymbol{x})}\log\left(\frac{q_\phi(\mathbf{z}|\boldsymbol{x})}{p_\theta(\mathbf{z})}\right) + \mathbb{E}_{q_\phi(\mathbf{z}|\boldsymbol{x})}\log\left(\frac{q_\phi(\mathbf{z}|\boldsymbol{x})}{p_\theta(\mathbf{z}|\boldsymbol{x})}\right)\\ &= \mathbb{E}_{q_\phi(\mathbf{z}|\boldsymbol{x})}\log(p_\theta(\boldsymbol{x}|\mathbf{z})) - D_{KL}\left(q_\phi(\mathbf{z}|\boldsymbol{x}) \parallel p_\theta(\mathbf{z})\right) + D_{KL}\left(q_\phi(\mathbf{z}|\boldsymbol{x}) \parallel p_\theta(\mathbf{z}|\boldsymbol{x})\right)\\ &\geq \mathbb{E}_{q_\phi(\mathbf{z}|\boldsymbol{x})}\log(p_\theta(\boldsymbol{x}|\mathbf{z})) - D_{KL}\left(q_\phi(\mathbf{z}|\boldsymbol{x}) \parallel p_\theta(\mathbf{z})\right) \equiv \boldsymbol{ELBO}\end{aligned}$$

Maximum likelihood estimator

$$\begin{aligned}\phi^*, \theta^* &= \operatorname*{argmax}_{\phi,\theta} \log p_\theta(\boldsymbol{x} = \boldsymbol{X})\\ &= \operatorname*{argmax}_{\theta} \mathbb{E}_{p_{data}(\boldsymbol{x})}\log p_\theta(\boldsymbol{x})\\ &= \operatorname*{argmax}_{\phi,\theta} \mathbb{E}_{p_{data}(\boldsymbol{x})}\left[\mathbb{E}_{q_\phi(\mathbf{z}|\boldsymbol{x})}\log(p_\theta(\boldsymbol{x}|\mathbf{z})) - D_{KL}\left(q_\phi(\mathbf{z}|\boldsymbol{x}) \parallel p_\theta(\mathbf{z})\right)\right]\end{aligned}$$