

---

# Supplemental Material for “*Relational VAE: A Continuous Latent Variable Model for Graph Structured Data*”

---

Anonymous Author(s)

Affiliation

Address

email

## 1 A Appendix

### 2 A.1 RVAE on wind farm monitoring data

- 3 The steady-state wind farm wake simulator FLORIS was used [1] for the simulated dataset. An example output of a simulation from FLORIS is shown in Figure 1.

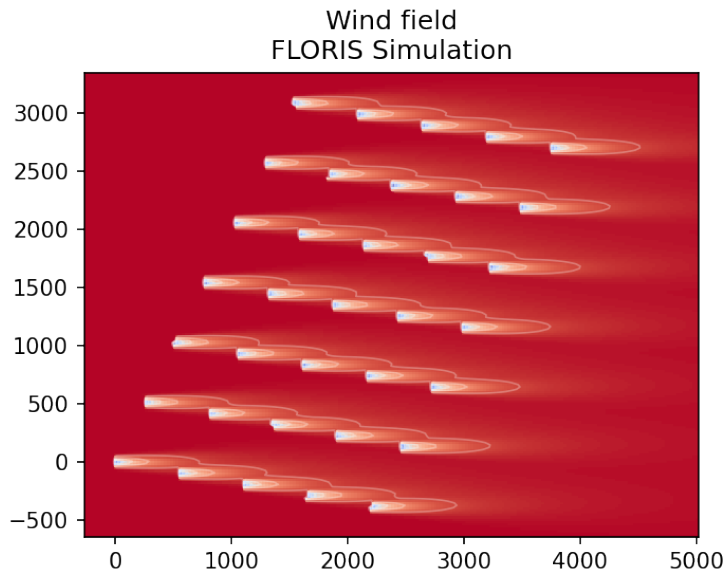


Figure 1: A representative simulated wind field from FLORIS. Lighter colors represent lower wind speeds.

5 **Additional details on farm graph construction** A graph is constructed where turbines are the  
6 vertices and directed edges are created between the turbines by truncating an all-to-all graph to a  
7 cut-off distance of  $100 \cdot d$  where  $d$  is the turbine rotor diameter. The 5MW NREL prototype turbine  
8 was used for the simulations [2] which has a diameter  $d = 126m$ . In order to test the generalization  
9 capabilities of the model, and most importantly the ability to implicitly learn how to use the relative  
10 position of the turbines and the nacelle wind orientation and power production of up-wind turbines,  
11 the model is tested on a different farm configuration than the one it is trained on. It is noted that the  
12 wake simulator used is not stochastic. The yaw directions of the turbines were randomly perturbed  
13 with  $\mathcal{N}(0., 5.^\circ)$  around the global wind orientation in order to introduce some stochasticity. The  
14 training dataset consists of 4462 farm SCADA readings for the whole farm. Both farms are simulated  
15 with the same randomly sampled mean wind and wind orientation global conditions ( $u_h$ ). Due to the  
16 different arrangement of the turbines in the two farms and their wake interactions the wind speed and  
power production of the two farms is very different. The farm configurations are shown in Figure 2.

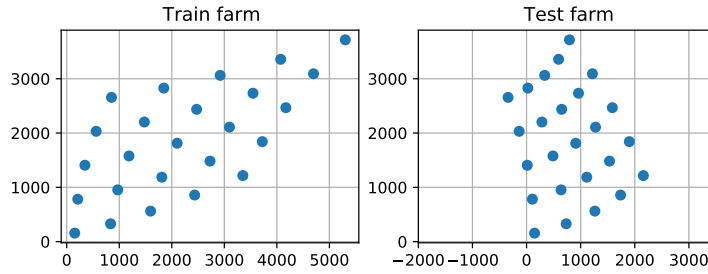


Figure 2: **Left:** The wind farm creating the training set. **Right** The wind farm used for the test set.

17

18 **Treatment of angles** As shown in Figure 3, the angle of the directional vector defined by the sender  
19 turbine  $i$  to the receiver turbine  $j$   $\phi_{ij}$ , and distance  $d_{ij}$  are used as an edge feature. In real farm  
20 monitoring data, the turbine yaw is adjusted by a per-turbine controller during operation according to  
21 the wind conditions and may be different for each turbine. The yaw angle  $\theta_i$  with respect to north is  
22 used as a node feature together with mean and standard deviation of hub-height wind speed when  
23 available, and power production.

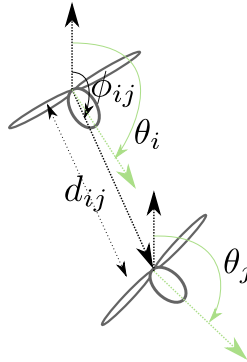


Figure 3: The edge attributes  $\mathcal{E}_h$  for edge with  $s = i, r = j$  is  $(\cos(\phi_{i,j}), \sin(\phi_{i,j}), d_{ij})$ . The nacelle angle  $\theta_i$  is included as a node (turbine) attribute.

24 Since the relative angle of the line defined between the turbines and the yaw angles is important, and  
25 not the absolute angles, a simple data augmentation procedure is performed during training where  
26 both yaw angle and the farm positions are randomly rotated with the same angle. A more general  
27 solution to the problem of learning networks that respect transformations to such symmetries would  
28 be to use GNs which are equivariant by design as proposed in [3] but this is left for future work.

29 **Further qualitative results on the simulated farm dataset** In Figure 4 some RVAE predictions  
30 and actual data from the simulated farm test set are shown. Note that the configuration of the test

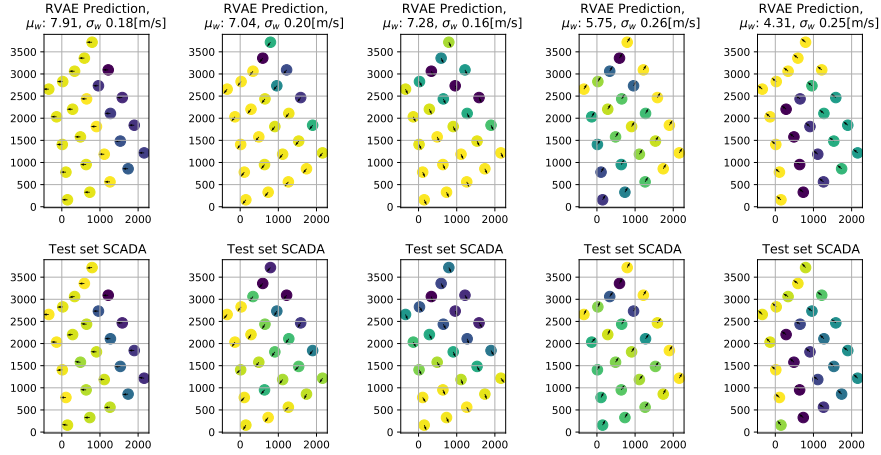


Figure 4: Examples of mean wind from the RVAE when conditioning with the global latent  $\mathbf{u}_h$  for an unseen farm (top) and corresponding mean wind from the test dataset (bottom). The arrows correspond to the incoming wind direction, and lighter colors correspond to higher wind speed.

31 farm is unseen and no training is performed. The RVAE seems to capture the wake-associated wind  
 32 deficits of the farm in all but one case (second from the right).

## 33 A.2 Anholt farm dataset

34 **Post-hoc model interpretation with gradient sensitivities** In order to identify which nodes are  
 35 important for the imputed values for a particular turbine, a simple gradient sensitivity technique was  
 36 employed. For a particular node prediction  $\mathbf{v}_i^T \in \mathcal{V}^T$ , and a RVAE model with recognition model  $q_\phi$   
 37 and generator  $p_\theta$ , we have  $\hat{\mathbf{v}}_i^T \sim p_\phi(\mathcal{V}_x | G_z; G_h) q_\phi(G_z | G_x^{C \cup T \setminus b}; G_h)$  where  $G_x^{C \cup T \setminus b}$  is the graph  
 38 that contains the masked target nodes and all the edges between context and target nodes. Since the  
 39 model is end-to-end differentiable, one can straightforwardly compute an absolute gradient sensitivity  
 40 as

$$I_{\mathbf{v}_i} = \left| \frac{\partial \hat{\mathbf{v}}_i^T}{\partial G_x^{C \cup T \setminus b}} \right| \quad (1)$$

41 An instance of such maps is visualized for a farm state snapshot and for a representative set of  
 42 turbines in Figure 5. Wind orientation is inferred from turbine nacelle orientation and it is visualized  
 43 as arrows centered at each turbine and pointing towards the incoming wind. The upwind turbines  
 44 seem to contribute more to the imputation value of the masked turbines. This is aligned with the  
 45 physics of the problem and in particular with the directionality of the wake effects which is always  
 46 from upstream to downstream turbines. The values of the turbines in the undisturbed boundaries  
 47 of the farm as in plots indexed 0 and 4 rely on non-upstream neighbors for imputation. This also is  
 48 expected, as there is no up-wind information for the imputation so the network needs to rely on any  
 49 neighbors to impute the wind velocity values.

## 50 B Supplemental material for the NP section

51 **Additional details on the training setup** All experiments were performed using the Adam opti-  
 52 mizer [4] with a learning rate of  $10^{-4}$  and default parameters. All RVAE models were trained for  
 53  $4 \cdot 10^4$  steps and NP models up to  $5 \cdot 10^4$  steps with batches of size 16. Each batch contains a random  
 54 number of context and target points which varies between 3 and 50. All GN functions involved are  
 55 feed-forward 3-layer Multi-Layer Perceptron (MLPs) with rectified linear unit non-linearities and no  
 56 activation in the last layer. The GN block used is encode-process-decode architecture as in [5] with  
 57 residual connections. The encoder and decoder contain layers that do not perform message passing  
 58 but only cast the inputs to a predefined dimension (Graph Independent layers).

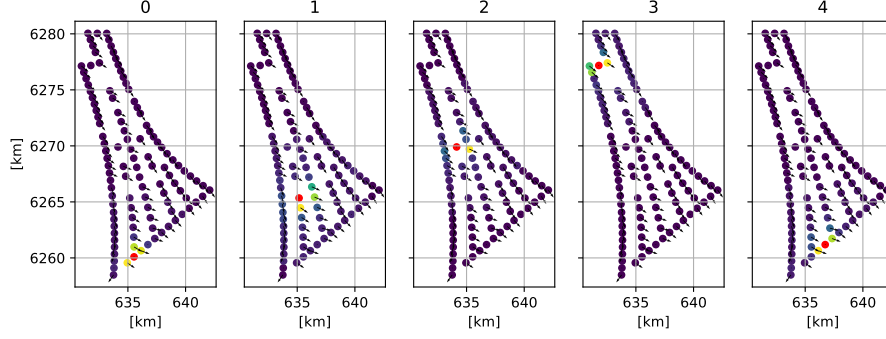


Figure 5: Absolute gradient sensitivity maps for data imputation of SCADA for different turbines, for a single snapshot of the farm conditions. Arrows point against incoming wind. Imputation target turbines in red. The value of neighbors and in particular of upstream neighbors are important.

59 **Qualitative results on 1D regression meta-learning** In Figure 7 qualitative results for the behavior  
60 of the RVAE for varying numbers of context points are shown. In the absence of context points, the  
61 model tends to predict the mean which is zero for the training dataset. Please refer directly to the  
62 notebook for further details on the implementation.

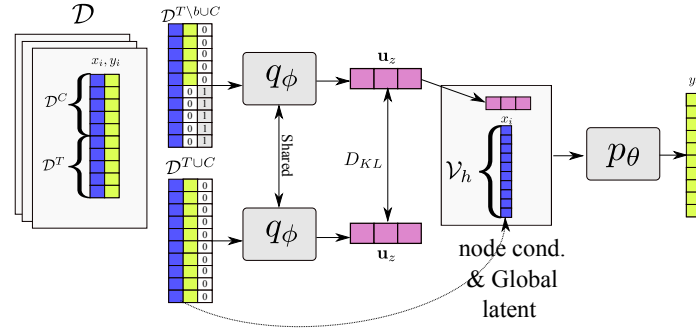


Figure 6: Implementation of NP as a conditional RVAE. The  $q_\phi$  is implemented as a deep set, and  $p_\theta$  is a node block. The node block is a function that contains a node update function which uses the node inputs and the global context (here the  $\mathbf{u}^z$  variable).

### 63 B.1 Implementation of NP as a VAE with arbitrary conditioning

64 In order to make the exposition of the main text clearer in what follows the ELBO objective for a  
65 RVAE model with arbitrary conditioning and a DeepSet encoder is detailed. In the NP implementation  
66 a set of context  $\mathcal{D}^C$  and target  $\mathcal{D}^T$  points are encoded through  $q_\phi$  to a global Gaussian latent variable  
67  $\mathbf{u}_z$ . Now consider the NP problem as a missing data imputation problem, where the context points,  
68  $\mathcal{D}^C : \{x_C, y_C\}$  are observed sets of points from a function, and the target points  $\mathcal{D}^T : \{x_T, y_T\}$  are  
69 points coming from the same function where we only have their  $x$  coordinate ( $x_T$ ) and we seek to  
70 evaluate the  $y_T$  values. The NP approach to this problem is to separately encode  $\mathcal{D}^{C \cup T}$  and  $\mathcal{D}^C$  to a  
71 Gaussian latent space  $\mathbf{u}_z$  using a DeepSet encoder, use the known  $x_T$  as conditioning, and compute  
72  $y_T$  using  $\mathbf{u}_z$  and  $x_T$  (and optionally a deterministic  $r$  introduced in [6]). The loss function for NPs  
73 given in Equation 2 can be cast as an arbitrary conditioning objective as in [7].

74 The VAE-AC[7] approach to an imputation problem such as this one, is to augment the dataset  
75 with an additional variable  $b$  which signifies whether points are observed or unobserved. The  $D_{KL}$   
76 term of the ELBO objective drives the latent representation  $\mathbf{u}_z$  of the recognition network to be  
77 similar when observing  $\mathcal{D}^C$  and  $\mathcal{D}^{C \cup T}$ . A network with parameters  $\theta$  is jointly trained to yield

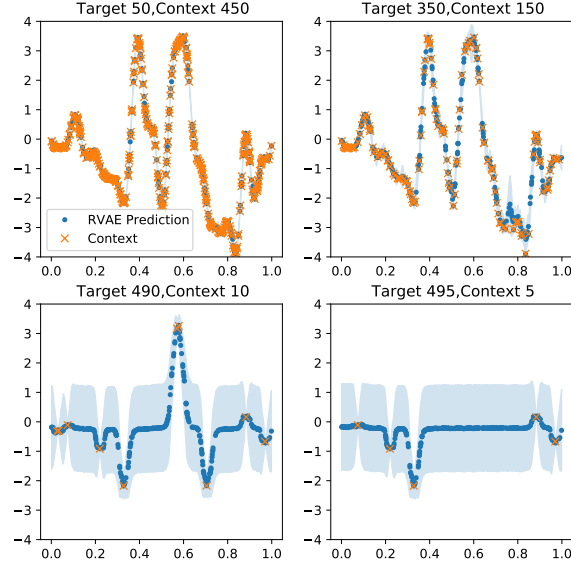


Figure 7: Qualitative results for 1D regression meta-learning with an RVAE. The model was trained using both node and edge features and with samples from GPs that had the same parameters.

78  $y_T \sim p_\theta(y_T | \mathbf{u}_z, x_T)$ . The arbitrary conditioning ELBO objective in that case reads,

$$\begin{aligned}
 \log p(y_T | x_T, x_C, y_C) &\geq \mathbb{E}_{q_\phi(z | \mathcal{D}^{C \cup T})} \left[ \sum_{i \in T} \log p_\theta(y_i | z, x_i) + \log \frac{q(z | \mathcal{D}^{C \cup T \setminus b})}{q(z | \mathcal{D}^{C \cup T})} \right] \\
 &\quad \mathbb{E}_{q_\phi(z | \mathcal{D}^{C \cup T})} \left[ \sum_{i \in T} \log p_\theta(y_i | z, x_i) \right] - D_{KL}(q_\phi(z | \mathcal{D}^{C \cup T}) || q_\phi(z | \mathcal{D}^{C \cup T \setminus b}))
 \end{aligned}
 \tag{2}$$

79 The VAE-AC approach is particularly convenient for the RVAE which contains message passing  
80 layers, since no special message passing needs to be performed between observed and unobserved  
81 points. In Figure 6 a more detailed computational diagram is shown. Please also refer to the  
82 corresponding notebook provided with the supplemental materials.

## 83 References

- 84 [1] NREL. FLORIS. Version 2.2.0, 2020.
- 85 [2] Brian R Resor. Definition of a 5mw/61.5 m wind turbine blade reference model. *Albuquerque,*  
86 *New Mexico, USA, Sandia National Laboratories, SAND2013-2569*, 2013, 2013.
- 87 [3] Victor Garcia Satorras, Emiel Hooeboom, and Max Welling. E (n) equivariant graph neural  
88 networks. *arXiv preprint arXiv:2102.09844*, 2021.
- 89 [4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*  
90 *arXiv:1412.6980*, 2014.
- 91 [5] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zam-  
92 baldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al.  
93 Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*,  
94 2018.

- 95 [6] Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum,  
96 Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. *arXiv preprint arXiv:1901.05761*,  
97 2019.
- 98 [7] Oleg Ivanov, Michael Figurnov, and Dmitry Vetrov. Variational autoencoder with arbitrary  
99 conditioning. *arXiv preprint arXiv:1806.02382*, 2018.
- 100 [8] Ben Day, Cătălina Cangea, Arian R Jamasb, and Pietro Liò. Message passing neural processes.  
101 *arXiv preprint arXiv:2009.13895*, 2020.