

Variational Autoencoder Model Order Reduction for Electric Machines Modeling

Michael Müller,¹ David Panek,¹ Pavel Karban,¹

¹University of West Bohemia, Univerzitni 26, 306 14 Pilsen, Czech Republic, michaelm{panek50,karban}@fel.zcu.cz

Modeling of electric machines is computationally intensive, especially for dynamic behavior, inverse problems, sensitivity analyses, and optimization tasks. Traditional order reduction techniques excel in linear cases, but nonlinear scenarios present greater challenges. This study explores the integration of variational autoencoders (VAEs) with finite element method (FEM) solvers to create reduced-order models for nonlinear problems. A switched reluctance motor is used as a test case, where VAEs trained on minimal snapshots predict machine behavior with high accuracy. Results demonstrate the efficacy of VAEs for data reconstruction and parameter sensitivity analysis, despite complexities in latent space optimization. These findings pave the way for efficient modeling and calibration in electric machine design.

Index Terms—proper orthogonal decomposition, variational autoencoder, principal component analysis, reduced models, electric machines.

I. INTRODUCTION

MODELING of electrical machines remains a computationally demanding task, particularly in scenarios such as modeling dynamic behavior, solving inverse problems (e.g., estimating material parameters and boundary conditions), performing sensitivity analyses, and addressing optimization challenges. There remains significant potential for developing methods to reduce system order, especially in the domain of nonlinear models.

While numerous well-established approaches exist for linear cases, as demonstrated, for instance, in the review article [1], the situation for nonlinear problems is considerably more complex. Recently, a growing number of studies have explored the use of machine learning techniques to achieve system order reduction.

One particularly promising method involves the use of a *variational auto-encoder* [2]. In our work, we aim to adopt a strategy that integrates the variational auto-encoder approach as seamlessly as possible with the finite element method (FEM) solver, while also introducing modifications to the standard auto-encoder architecture to enhance its performance.

II. METHODOLOGY AND IMPLEMENTATION

In our work, we aim to combine finite element methods (FEM) with machine learning techniques. We utilize software developed at our workplace, *Agros* [3]. This software includes a graphical user interface and the Python wrapper *agrossuite*, which is available via PyPI [4]. This setup allows access to relatively low-level structures created during the FEM solution process. These structures are used through the Python writer within standard Python scripts. The variational auto-encoder itself is implemented using the PyTorch framework [5].

The basic workflow of our solution is depicted in Fig. 1. We consider the weak formulation of the magnetostatic problem in the form:

$$\int_{\Omega} \phi \cdot \nabla \times \left(\frac{1}{\mu} \nabla \times \mathbf{A} \right) d\Omega = \int_{\Omega} \phi \cdot \mathbf{J} d\Omega, \quad (1)$$

where $\mathbf{A}$ is the vector of magnetic potential, μ is the permeability, $\mathbf{J}$ is the external current density, and ϕ is a test function. We assume that the vector potential is approximated by a product of a solution vector (coefficients) and a vector of basis functions in the form:

$$\mathbf{A} \approx \sum_i A_i \phi_i = \tilde{\mathbf{A}} \cdot \boldsymbol{\phi}, \quad (2)$$

where $\tilde{\mathbf{A}}$ represents the vector of coefficients.

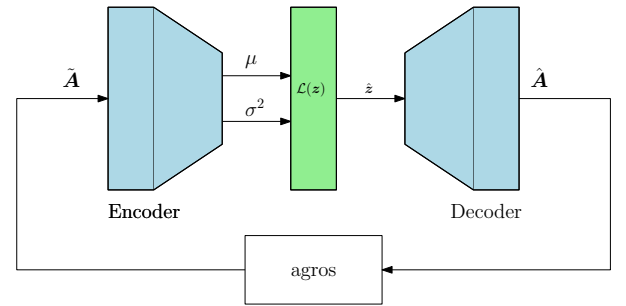


Fig. 1: The magnetic flux density obtained by the reduced model.

The solution vector $\tilde{\mathbf{A}}$ is calculated using *Agros* and subsequently used as the input to an encoder, which is represented by a fully connected neural network. Each input vector is transformed into a random variable z_i with a normal distribution $\mathcal{N}(0, 1)$ in the latent space. The loss function is based on the Kullback–Leibler divergence in its standard form:

$$D_{KL}(q(z|x)|p(z)) = \frac{1}{2} \sum_{i=1}^d \left(\log \sigma_i^2 + \frac{\mu_i^2 + \sigma_i^2}{1} - 1 \right). \quad (3)$$

A random sample is then generated from the latent space and transformed by the decoder into an estimate of the solution vector $\hat{\mathbf{A}}$. This estimate is returned to *Agros*, where it is subsequently processed.

III. ILLUSTRATIVE EXAMPLE

Our goal is to train a variational auto-encoder for a given electric machine using a minimal number of snapshots to obtain a surrogate model capable of predicting the machine's behavior with a specified level of precision. For simplicity, we consider a switched reluctance motor, as depicted in Fig. 2.

Since we aim to explore automatic calibration as one of the possible applications of variational auto-encoder-based order reduction, we introduce a parameterization where one parameter represents the current density in the windings, and the other parameter characterizes the material properties of the rotor and stator. Each point in the parameter space is thus defined by a specific current density and a corresponding H-B curve. The latent space of the variational auto-encoder is two-dimensional.

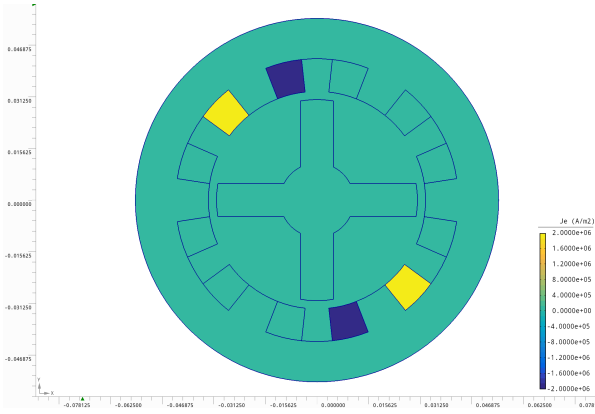


Fig. 2: Magnetic flux density obtained using COMSOL Multiphysics.

IV. RESULTS AND DISCUSSION

The solution vector consisted of 2000 items. Using a three-layer encoder, it was transformed into a latent space with a dimension of $n = 4$. The activation functions used were ReLU and tanh. The parametrization was based on a grid with ten different current densities and ten different material characteristics. The model was trained over 500 batches, with 500 datasets in each batch. As the optimization algorithm Adam was chosen.

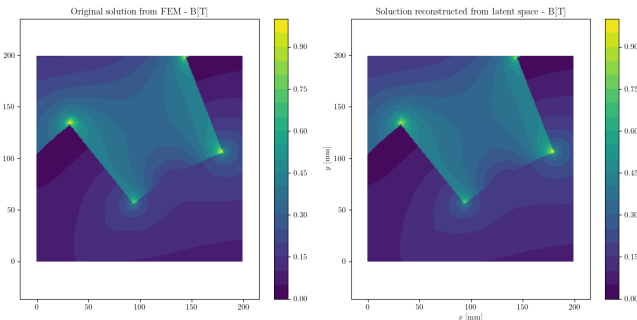


Fig. 3: Comparison between flux density obtained by FEM and reconstructed flux density from the VAE.

It was observed that a variational auto-encoder can be trained successfully to reconstruct data both at and "near" the training points. A comparison of the results obtained by the FEM solution and the reconstruction from the latent space for a point not included in the training set is shown in Fig. 3. Note that the figure shows only a section of the motor containing the tooth with the highest magnetic induction values. However, as demonstrated in Fig. 4, even for a simple two-dimensional example, the structure of the latent space can be complex and suboptimal for optimization tasks.

The structure of the latent space is the result of a compromise between the reconstruction loss and the divergence loss (appropriate probability distribution), representing a particular optimization task. The key challenge is that FEM calculations are time-consuming, and therefore the goal is to reduce the number of necessary FEM calculations. A promising approach might be to consider the latent space as a Gaussian process rather than a set of independent random variables.

The combination of the variational auto-encoder with FEM has several potential applications. As mentioned above, one such application could be the automatic calibration of models, such as the automatic correction of boundary conditions or material properties. Another possible application is the use of fast estimates provided by the VAE to enable automatic adaptivity in FEM simulations.

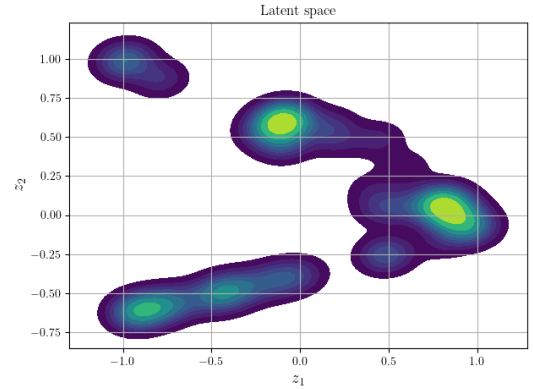


Fig. 4: Visualization of the latent space

ACKNOWLEDGEMENT

This work was supported by the internal project SGS-2024-025 of the University of West Bohemia

REFERENCES

- [1] K. L. Kiss and T. Orosz, "Model order reduction methods for rotating electrical machines: A review," *Energies*, vol. 17, p. 5145, Oct. 2024.
- [2] V. Parekh, D. Flore, and S. Schöps, "Variational autoencoder-based metamodeling for multi-objective topology optimization of electrical machines," *IEEE Transactions on Magnetics*, vol. 58, p. 1–4, Sept. 2022.
- [3] P. Karban *et al.*, "agros: Free scientific tool," January 2025. Accessed: 15 January 2025.
- [4] P. Karban *et al.*, "agrossuite: agros python wrapper," January 2025. Accessed: 15 January 2025.
- [5] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," 2019. Available at <https://pytorch.org/>.