Electric Machine Inverse Design with Variational Auto-Encoder (VAE)

Xu, Yihao; Wang, Bingnan; Sakamoto, Yusuke; Yamamoto, Tatsuya; Nishimura, Yuki; Koike-Akino, Toshiaki; Wang, Ye

TR2023-134 November 03, 2023

Abstract

Machine learning and deep learning techniques have been proposed to facilitate the design optimization of electric machines. Most of the existing research focuses on the development of surrogate models, while iterative optimization is still needed. Inverse design approach, on the other hand, can directly provide design candidates with trained deep learning model without iteration. One major challenge in deep learning based inverse design is the so-called one-to-many mapping problem. In this paper, we propose an intelligent inverse design approach for electric machines based on a variational autoencoder (VAE), which can effectively address the problem and provide desired motor design candidates for multiple design targets at the same time. We demonstrate the feasibility of the proposed strategy with multi-objective design task of a surface-mount permanent magnet motor, and show that it is generally applicable for different types of electric motors.

IEEE Energy Conversion Congress and Exposition (ECCE) 2023

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Electric Machine Inverse Design with Variational Auto-Encoder (VAE)

Yihao Xu
Mechanical and Industrial Engineering
Northeastern University
Boston, MA 02120 USA

Bingnan Wang
Mitsubishi Electric Research
Laboratories (MERL)
Cambridge, MA 02139 USA

Yusuke Sakamoto
Mitsubishi Electric Research
Laboratories (MERL)
Cambridge, MA 02139 USA

Tatsuya Yamamoto
Advanced Technology R&D Center
Mitsubishi Electric Corporation
Amagasaki, Hyogo 661-8661 Japan

Yuki Nishimura
Advanced Technology R&D Center
Mitsubishi Electric Corporation
Amagasaki, Hyogo 661-8661 Japan

Toshiaki Koike-Akino
Mitsubishi Electric Research
Laboratories (MERL)
Cambridge, MA 02139 USA

Ye Wang
Mitsubishi Electric Research
Laboratories (MERL)
Cambridge, MA 02139 USA

Abstract: Machine learning and deep learning techniques have been proposed to facilitate the design optimization of electric machines. Most of the existing research focuses on the development of surrogate models, while iterative optimization is still needed. Inverse design approach, on the other hand, can directly provide design candidates with trained deep learning model without iteration. One major challenge in deep learning based inverse design is the so-called one-to-many mapping problem. In this paper, we propose an intelligent inverse design approach for electric machines based on a variational autoencoder (VAE), which can effectively address the problem and provide desired motor design candidates for multiple design targets at the same time. We demonstrate the feasibility of the proposed strategy with multi-objective design task of a surface-mount permanent magnet motor, and show that it is generally applicable for different types of electric motors.

I. Introduction

In recent years, the demand for high-density, efficient, and cost-effective electric motors has grown significantly due to their crucial role in various societal systems, including transportation, industrial equipment, and household appliances. To achieve optimal motor designs, a common approach is multi-objective design optimization, which considers factors such as average torque generation, torque ripple, cogging torque, weight, and material cost. This optimization process involves iteratively updating design parameters using evolutionary algorithms like genetic algorithms and evaluating the performance of each design through numerical simulations with finite element analysis.

However, this optimization process presents several challenges [1]. Firstly, the design targets are interrelated, leading to trade-offs or conflicts when attempting to satisfy all targets simultaneously. Secondly, evaluating multiple design candidates through numerical simulations is computationally expensive and time-consuming, particularly when analyzing multiple rotor positions or operating points. Lastly, the optimization process relies on iterative trial and error, offering no guarantee of finding the global optimum design in a single step. Consequently, combining these challenges makes the multi-objective design optimization process highly time-consuming.

Recently, intelligent design methods utilizing contemporary numerical optimization and machine learning algorithms have gained significant traction in multiple fields, including photonics [2], acoustics [3], and mechanics [4]. For example, researchers have successfully utilized deep neural networks (DNNs) to revolutionize the design of optical devices, enabling them to adapt to different demands under varying illumination conditions [2]. Overcoming limitations faced by traditional approaches, DNN-based intelligent inverse design has expanded the number of channels for these devices, allowing for more comprehensive functionality. Consequently, there is immense potential for applying these approaches to address the challenges encountered in electric motor design, offering a highly efficient and effective solution.

As a powerful tool that recently achieves great success, DNNs are machine learning models that have gained significant popularity due to their ability to solve complex tasks. Inspired by the functioning of neurons in the human brain, DNNs aim to mimic the information processing capabilities of biological neural networks. DNNs consist of interconnected layers of artificial neurons. Each layer performs linear

This work was done when Y. Xu are with Mitsubishi Electric Research Laboratories. Corresponding Author: Bingnan Wang (bwang@merl.com).

transformations by computing weighted sums of inputs using learnable parameters. Nonlinear activation functions introduce nonlinearity to capture complex patterns [5]–[7]. Stacking multiple layers enables learning hierarchical representations: initial layers learn simple features, deeper layers learn complex features, and the final layer performs task-specific computations such as classification or regression. DNNs can approximate any function by stacking layers of neurons, making them ideal data-driven tools for a wide range of applications. Consequently, DNNs serve as valuable tools for constructing forward surrogate models that accurately predict the response of specific physical designs, eliminating the need for time-consuming physical simulations and significantly expediting the design optimization process.

Numerous deep learning models have been proposed as surrogate models for electric motors. These models have the capability to predict responses for a given motor design candidate within a fraction of second. Such trained models can be used to replace time-consuming finite-element simulations (which typically takes several minutes or even hours for single simulation) during the optimization process [8]–[12]. For such surrogate model approach, the iterative process is still required in order to find a suitable design candidate that meets the objectives.

An even more fascinating application is to utilize deep learning models for the inverse design process. These inverse models have the ability to directly generate design candidates that fulfill specific requirements, thus bypassing the conventional iterative optimization process. However, achieving optimal performance with inverse models necessitates more intricate configurations tailored to specific tasks. This is due to the fact that inverse models utilizing DNNs face a challenge known as the one-to-many mapping problem. Unlike surrogate models, which yield deterministic responses for a given design, inverse models often encounter situations where multiple design candidates satisfy the specified requirements.

In this paper, we propose the use of a variational autoencoder (VAE) [13] as an inverse design and generation strategy for electric motors, which addresses the one-to-many problem in motor design tasks. We apply this method to a surface-mount permanent magnet (SPM) motor design problem and show that it is capable of providing motor designs that meet a set of design objectives without iteration. The results are validated with finite-element simulations.

II. Problem Setting & Dataset Generation

In this paper, we investigate the design optimization of surface-mount permanent magnet (SPM) motor, and use it to demonstrate the effectiveness of the proposed VAE based inverse design method. This particular motor consists of 10 magnetic poles in the rotor and 12 slots for the stator winding, as shown in Fig. 1. A total of 9 specific design parameters is subject to design optimization, while all other parameters are fixed. Table I list the design parameters, the corresponding range of values and the step size used when generating design candidates during dataset preparation. For each design candidate, we evaluate the following responses: slot area r_1, 12^{th} Fourier order of cogging torque r_2, 1^{st} Fourier order of induced voltage r_3, and its total harmonic distortion r_4.

For dataset generation, we employed finite-element simulations using JMAG, a commercial simulation software. Each individual design is characterized by a specific set of geometrical parameters, denoted as $D : [d_1, d_2, ..., d_9]$, along with a corresponding set of motor responses, denoted as $R : [r_1, r_2, ..., r_4]$. Table I provides a comprehensive representation of each design parameter (d_i) for reference. The range and step size of each parameter are set to generate design candidates with parameter sweeping. Removing the non-physical designs created in the process, these simulations enabled us to create a comprehensive SPM dataset comprising a total of $8,916$ motor designs.

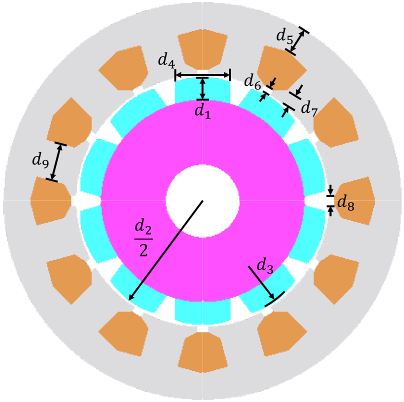


Fig. 1. The schematic of the SPM model and the representation of the design parameters.

TABLE I
List of Design Parameters and Data Range in the Dataset

	Name	min. [mm]	max. [mm]	step. [mm]
d_1	Magnet height at the center	2.0	5.0	0.2
d_2	Stator inner diameter	45.0	50.0	0.5
d_3	Magnet curvature radius	8.0	20.0	1.0
d_4	Magnet width	5.0	13.0	0.5
d_5	Back yoke width	3.0	6.0	0.5
d_6	Tooth shoe height 1	0.2	1.6	0.2
d_7	Tooth shoe height 2	1.0	3.0	0.5
d_8	Slot opening	0.0	2.0	0.2
d_9	Tooth width	5.0	10.0	0.25

III. Inverse Design Based on VAE

DNN-based surrogate models can learn the relationship between motor design and its performance metrics. These models take in a motor design as input and predict the corresponding responses or performance metrics. The mapping in this scenario is straightforward as it is a one-to-one relationship. In other words, for any given inputs (which are motor design parameters in our case) to the model, there is only one possible and deterministic output (which are motor responses in our case).

TABLE II
DIFFERENT MOTOR WITH SIMILAR RESPONSES

Unit: mm	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
Motor 1	3.4	50.0	9.0	5.5	3.0	1.0	2.6	0.6	7.75
Motor 2	3.6	47.0	10.0	5.5	5.0	0.4	2.8	0.6	6.50

Response:	r_1 (mm^2)	r_2 (N$\cdot$ m)	r_3 (V)	r_4 (1)
Motor 1	30.73	0.090	18.78	0.279
Motor 2	30.72	0.094	18.77	0.249

However, the situation is totally different in motor design tasks, where the objective is often to find the best motor design that meets specific design targets among quite a lot of candidates. This process, known as inverse design, can result in multiple motor designs with different parameters producing similar or identical responses. In this case, when generating a large dataset using finite element analysis (FEA) simulation, it is possible to encounter "conflicting" data. For instance, we may discover two designs, denoted as D_1 and D_2, that exhibit very similar or even identical responses, R. This situation creates a scenario of one-to-many mapping. To illustrate this problem, we consider two different motor designs from the dataset, as shown in Table II, along with their corresponding responses. It is evident that these two motors display nearly identical response sets despite having distinct parameters. This represents just one instance of "conflicting" data, and there may be more occurrences of this nature when generating a very large dataset. The presence of this one-to-many mapping problem can pose challenges in training deep neural networks (DNNs), potentially leading to convergence issues.

Identifying the cause of this confusion helps in finding appropriate solutions. To avoid the neural network becoming confused by the different branches of the design, it is possible to employ two different approaches: (i) Allow the neural network to converge to a single branch, disregarding the conflicting data. By forcing a convergence towards one branch over the other, the network can focus on learning and generating designs that align with the branches. (ii) Alternatively, let the neural network converge at distinct branches based on different circumstances. This approach acknowledges that different situations may require different design solutions. By training the network to adapt its convergence behavior, it becomes capable of generating designs that are better suited for specific requirements.

In our previous study [14], we employed a tandem neural network to tackle the one-to-many mapping problem. This kind of neural network belongs to the aforementioned approach (i), which aimed to converge on only one single branch for each input. This was achieved by combining the inverse design model with a pretrained surrogate model. Although this approach demonstrated effective convergence, it only provided a single motor design candidate for a given target response. However, the tandem network has the drawbacks. While it can overcome the one-to-many mapping problem and exhibit excellent inverse design performance, it can only offer one solution for the target response. In reality, there are often additional constraints, such as material cost, feature size and robustness, that restrict motor designs. This deterministic nature of the tandem neural network fails to account for multiple available solutions that exist and may be better suited. Therefore, it is crucial for an inverse design model to have the ability to generate multiple candidates for a single target response, adopting the idea of approach (ii). Essentially, directly resolving the one-to-many mapping issue by exploring all design branches would be highly desirable instead of settling for the solution to one single branch.

Toward this end, it is more beneficial to use a generative model which can generate multiple design candidates instead of a deterministic inverse design model. Some popular examples of generative neural networks include VAEs [15], [16], Generative Adversarial Networks (GANs) [17] and Transformer-based Models [18]. VAEs are generative models that combine elements from autoencoders and probabilistic models. They consist of an encoder network that maps input data to a latent space and a decoder network that reconstructs the input data from the latent space. VAEs are trained using a probabilistic approach that maximizes the evidence lower bound (ELBO), which consists of two main loss terms, namely reconstruction loss and Kullback–Leibler (KL)-divergence, allowing them to generate new design candidates by sampling from the learned latent space. The advantage of VAEs is their ability to learn meaningful latent representations and offer a principled framework for generating new data while allowing for interpolation and controlled synthesis. GANs, on the other hand, consist of a generator and a discriminator network that are trained simultaneously. The generator learns to produce synthetic data samples to deceive the discriminator, while the discriminator learns to distinguish between real and fake samples. GANs can generate realistic and diverse samples by effectively capturing the underlying data distribution. However, they can be challenging to train and may suffer from model collapse and/or instability issues. Lastly, Transformer-based models, known for their success in natural language processing tasks, have also been applied to generative modeling. By leveraging self-attention mechanisms, transformers can capture long-range dependencies in the data, making them effective in generating sequences. Transformers offer parallelizability and have achieved state-of-the-art results in tasks such as language generation and image synthesis. Overall, generative neural networks provide a range of approaches for data generation. While GANs and transformer-based models each have their own strengths and applications in generating diverse and realistic data samples, VAEs stand out for their ability to learn meaningful latent representations, principled probabilistic framework, and the control they offer through the latent space. They find applications in image generation, data compression, and anomaly detection.

In this study, we will focus on DNN-based inverse design models for motor design tasks by utilizing a VAE that presents a solution to fully resolve the one-to-many mapping problem. The VAE architecture encodes the information about the motor

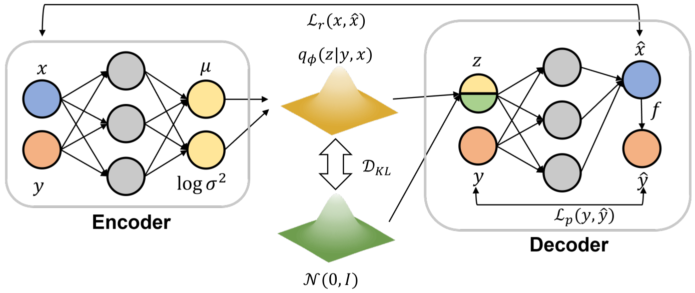


Fig. 2. The schematic of a variational auto-encoder (VAE) model.

design (x) and its response (y) into a latent distribution ($q_\phi(z|y,x)$), as illustrated in Fig. 2. The latent distribution is typically chosen to be a multivariate Gaussian distribution for two main reasons: first, it can be simply represented by a mean vector (μ) and a standard deviation vector (σ), and the dimension of this distribution is a hyper-parameter in the VAE referred to as the latent dimension (N_l). Second, it is straightforward to evaluate the similarity between two Gaussian distributions using the analytical formula for the KL-divergence loss.

The goal of the VAE is to encourage the distribution produced by the encoder to approximate a prior distribution, denoted as $p_\theta(z)$. This prior distribution is often chosen to be a standard Gaussian distribution $\mathcal{N}(0,I)$ with the reasons discussed before, where I represents the N_l dimensional identity matrix. By imposing this constraint, the VAE ensures that the latent space is well-structured and allows for meaningful interpolation and sampling. To achieve this, the VAE employs an encoding process that maps the input data to a distribution in the latent space, characterized by the mean (μ) and standard deviation (σ). The encoder network takes the input data and maps it to these parameters, which define the shape of the distribution. Then, a random sampling procedure is performed to generate a latent vector (z) from this distribution. It is important to note that multiple latent vectors can be generated from the same latent distribution, specified by the same μ and σ, through a random process. This stochasticity is crucial in addressing the problem of one-to-many mappings, allowing the VAE to capture the underlying variability in the data. Once the latent vector is obtained, it serves as an input to the decoder network along with the target response y. The decoder reconstructs the original input data by decoding the information contained in the latent vector and incorporating the target response. By feeding both the latent vector and the target response to the decoder, the VAE can generate a motor design that not only captures the characteristics of the latent distribution but also fulfills the desired target response. This fusion of information enables the VAE to produce meaningful and context-aware motor designs.

IV. Model Implementation & Test Results

In this section, we first explain the details of the VAE based inverse model architecture and its implementation process, and then show the test results on the SPM dataset.

As shown in Fig. 2, the VAE model requires a surrogate model f and a design-response dataset (x, y) during the training phase. In the VAE model, the encoder component takes a batch of data and produces μ and $\log\sigma^2$, which define the latent distribution $q_\phi(z|y,x)$. This equation indicates that the latent distribution is conditioned on the input data pair x and y. In other words, the encoder network generates a distinct latent distribution for each input data pair. This particular VAE model is referred to as the conditional-VAE (cVAE). However, for the sake of simplicity, we will still use the abbreviation VAE to denote this model in subsequent text. During the inverse design stage, we lack information about the design x and only possess the target response y. Consequently, we cannot obtain the exact latent distribution q_ϕ. Hence, the approach is to ensure that the latent distribution for all data pairs closely approximates a fixed, user-defined prior distribution $p_\theta(z)$ that remains independent of both x and y. The KL-divergence loss is minimized to bring the distribution closer to the Gaussian prior distribution $\mathcal{N}(0,I)$, which is expressed as follows when p_θ and q_ϕ are both Gaussian distributions:

$$\mathcal{D}_{KL}(q_\phi||p_\theta) \equiv \int_z q_\phi \log\frac{q_\phi}{p_\theta}dz = \log\frac{1}{\sigma} + \frac{\sigma^2+\mu^2}{2} - \frac{1}{2} \quad (1)$$

Then, a batch of random latent vectors z is sampled from the distribution q_ϕ, combined with the target motor responses y as input to the decoder part of VAE model, generating motor designs $\hat{x}$ and a reconstruction loss is calculated using root-mean-square error (RMSE):

$$\mathcal{L}_r(x,\hat{x}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\hat{x}_i)^2} \quad (2)$$

On the other hand, the surrogate model predicts responses $\hat{y}$ and a prediction loss is calculated using RMSE:

$$\mathcal{L}_p(y,\hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2} \quad (3)$$

In the generation phase, the VAE only receives target response y as input, and a latent vector z is generated from the prior distribution $p_\theta = \mathcal{N}(0,I)$, which is decoded to produce design $\hat{x}$ and evaluated with the surrogate model to get a response $\hat{y}$. The final motor design is $\hat{x}$ for target response y.

Although the prediction loss is the most important metric to describe the performance of the inverse design model because it directly tells the difference between the target response and the response of the motor design suggested by the model, the reconstruction loss should also be included in the training phase to ensure that the reconstructed designs have a similar data range and distribution as the training data, though not necessary to be exactly minimized. Therefore, the total loss function used for backpropagation is defined as:

$$\mathcal{L} = w_r \cdot \mathcal{L}_r(x,\hat{x}) + \mathcal{L}_p(y,\hat{y}) + \mathcal{D}_{KL}(q_\phi||p_\theta) \quad (4)$$

where w_r is the weight of the reconstruction loss as a hyper-parameter.

To implement the inverse model, we first partition the dataset employing an 80:20 split ratio, allocating 80% of the dataset for training purposes and reserving the remaining 20% for testing the model's performance.

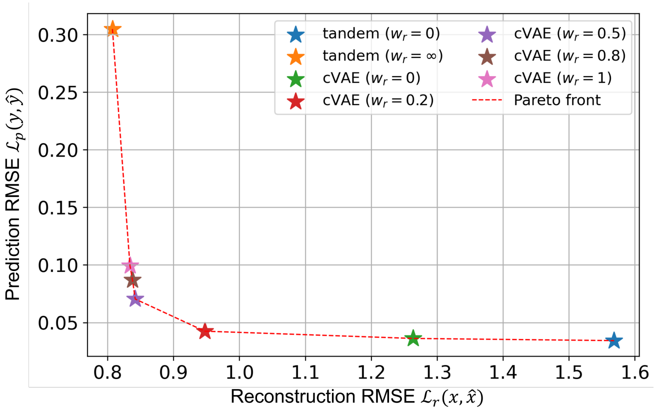


Fig. 3. Comparison of results from different loss functions in VAE and tandem network settings. The red dashed line represents the Pareto front considering the two loss functions as objectives.

In order to evaluate the performance of the VAE model and investigate the impact of the weight parameter, we trained multiple models with varying loss functions by adjusting the weight w_r from 0 to 1, while maintaining consistent model configuration and other hyper-parameters. Figure 3 illustrates the RMSE values for both prediction and reconstruction of each model, including two tandem neural network-based models: one with solely reconstruction loss ($w_r = 0$) and the other with only prediction loss ($w_r = \infty$). The red dashed line represents the Pareto front [19], indicating the optimal solutions in the two-objective optimization (minimizing both RMSEs), where enhancing one objective necessitates compromising the other. Ideally, we aim for minimal prediction loss disregarding reconstruction loss; however, a high reconstruction loss can lead to errors in the surrogate model and potentially yield physically infeasible designs. Considering both factors, the plot suggests that achieving moderate prediction and reconstruction loss is more attainable with a weight for the reconstruction loss ranging from 0.5 to 0.8.

To provide a comprehensive demonstration of the performance of a well-trained VAE model, we present a visualization of the prediction and reconstruction distributions for $w_r = 0.5$ in Figure 4. Figure 4(a) displays the ground truth motor responses (target responses R) on the horizontal axis, while the vertical axis represents the responses of the reconstructed motor designs generated by the VAE model using the surrogate model for prediction instead of FEA simulation, thereby improving efficiency. The plot exhibits a strong agreement between predicted and target responses, confirming the accuracy of the VAE-based inverse design. However, it is essential to note that the error encompasses both the surrogate model's prediction error and the discrepancy between the target and predicted response. In Figure 4(b), we illustrate the distributions of the design parameters in the dataset (blue) and those generated by the inverse model (orange) to qualitatively evaluate the prediction error of the surrogate model. The figure demonstrates that all reconstructed parameters fall within a reasonable range, indicating the effectiveness of the surrogate model in predicting motor responses. Therefore, we confidently assert that the VAE-based inverse model exhibits excellent performance in proposing new motor design candidates that align with user-defined responses.

Furthermore, one of the significant advantages of probabilistic generation models like VAE is the ability to generate multiple design candidates for a fixed target response. To showcase this capability of the VAE model, we select one specific set of target responses from the test dataset, as listed in the first row of the table in Figure 5. Subsequently, we run the VAE model to simultaneously generate over 1000 distinct sets of design parameters. To visualize this behavior, we plot the distribution of all generated parameters of the design candidates, as depicted in Figure 5. Notably, we observe that the parameters span a certain range rather than converge to only a single parameter set. Although this range may not be extensive, it still provides a finer level of variation that enables users to consider physical constraints and material costs when seeking an optimal design. It is important to highlight that all these design candidates exhibit a high level of accuracy in approximating the target responses, as demonstrated in the three examples presented in the table. Consequently, we establish that the VAE model possesses the capability to generate multiple design candidates that closely align with user-defined design targets, ensuring both accuracy and efficiency.

As a final step, we aim to validate that the motor design candidates generated by the VAE model indeed fulfill the design targets. To achieve this, we conducted FEA simulations instead of relying solely on the surrogate model trained on the same dataset. By comparing the responses of randomly selected reconstructed designs generated from the VAE model ($w_r = 0.5$) to the target responses through FEA simulations, we evaluate their validity. A total of 30 designs D and their corresponding true responses R were randomly selected from the dataset and used as target responses for the VAE. The VAE suggested designs $\hat{D}$ for each target response, and both D and $\hat{D}$ were evaluated through simulations. Figure 6 demonstrates that the majority of designs exhibited excellent agreement with the target responses, particularly for metrics such as slot area, induced voltage, and harmonic distortion. Although larger errors were expected for cogging torque, the results still closely aligned with the target response.

The capability of VAE to simultaneously match all target responses with high accuracy renders it suitable for multi-objective optimization and multitask design problems. Additionally, the ability of VAE to generate multiple designs when executed multiple times enhances its robustness in real-world applications, providing multiple candidates for a single design task.

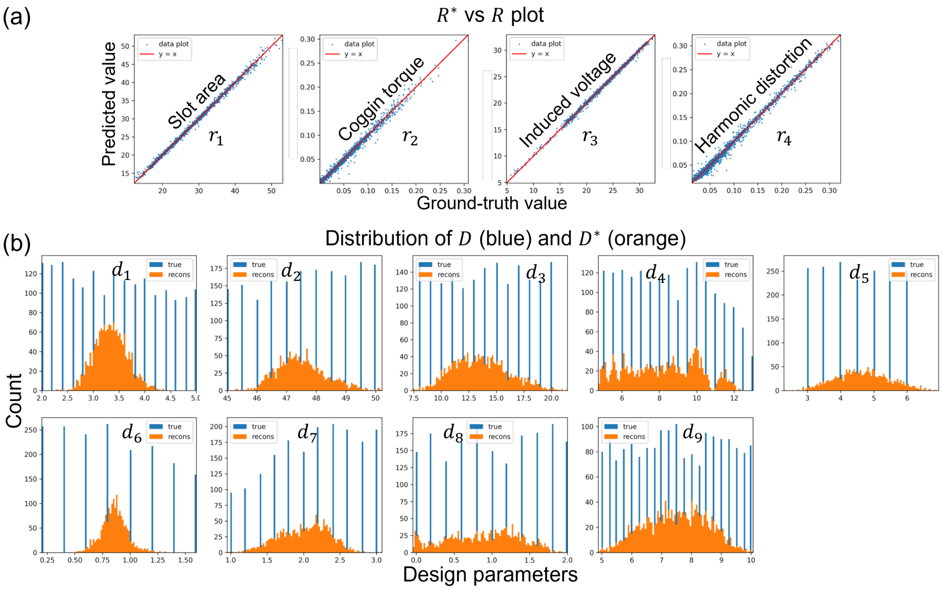
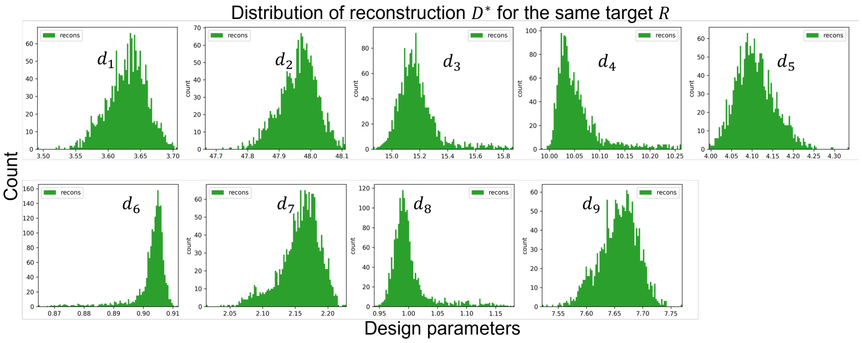


Fig. 4. Test results of the optimized VAE models.(a) The response plot shows the predicted response vs. the target response. (b) The distribution of the ground-truth design parameters (blue) and the retrieved design parameters (orange).



Motor reconstructions D^*

Unit: mm	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	r_1 (mm^2)	r_2 (N· m)	r_3 (V)	r_4 (1)
Target	/	/	/	/	/	/	/	/	/	29.49	0.018	28.76	0.03
Motor 1	3.65	48.01	15.14	10.03	4.06	0.9	2.20	0.99	7.66	29.73	0.0177	28.84	0.025
Motor 2	3.51	47.72	15.42	10.08	4.28	0.9	2.06	1.00	7.54	29.73	0.0153	28.68	0.028
Motor 3	3.59	47.82	15.77	10.20	4.17	0.9	2.08	1.13	7.63	29.64	0.0185	29.02	0.029

Fig. 5. The distribution of the reconstruction of design parameters for the same design target. Three examples of the reconstructed motors including their design parameters and their responses are shown in the table in comparison to the target responses.

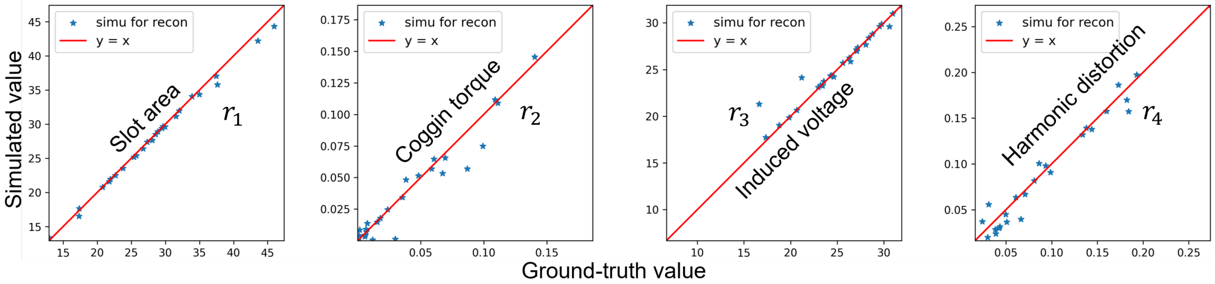


Fig. 6. Comparison between the target responses (x-axis) and the responses of the reconstructed motors from FEM simulation.

V. Concluding Remarks

In summary, we proposed an inverse design method for electric machines using a VAE-based deep learning model, which is capable of generating multiple motor designs based on a set of design objectives without iteration. We showed with an SPM design task that the proposed method is effective in finding optimal designs, whose performances are in good agreement with objective responses. Unlike machine learning and deep learning based surrogate models, the VAE inverse model is able to generate optimal designs without iteration. Compared with other inverse models such as tandem network, the VAE model can handle multi-objective optimization and multi-task design problems, and provides multiple candidates for user-defined design tasks, offering higher robustness in real-world applications. The reconstructed parameters also fall within a reasonable range defined by the training dataset, ensuring the accuracy of predictions from a pretrained surrogate model. The actual performance of the VAE generated motor designs have also been validated by finite-element simulations, showing the effectiveness of the approach.

References

[1] G. Bramerdorfer, J. A. Tapia, J. J. Pyrhönen, and A. Cavagnino, "Modern electrical machine design optimization: Techniques, trends, and best practices," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 7672–7684, 2018.

[2] Z. Liu, L. Raju, D. Zhu, and W. Cai, "A hybrid strategy for the discovery and design of photonic structures," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 1, pp. 126–135, 2020.

[3] B. Zheng, J. Yang, B. Liang, and J.-c. Cheng, "Inverse design of acoustic metamaterials based on machine learning using a Gauss–Bayesian model," *Journal of Applied Physics*, vol. 128, no. 13, p. 134902, 2020.

[4] Q. Zeng, Z. Zhao, H. Lei, and P. Wang, "A deep learning approach for inverse design of gradient mechanical metamaterials," *International Journal of Mechanical Sciences*, vol. 240, p. 107920, 2023.

[5] W. Gerstner and W. M. Kistler, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[6] W. Ma, Z. Liu, Z. A. Kudyshev, A. Boltasseva, W. Cai, and Y. Liu, "Deep learning for the design of photonic structures," *Nature Photonics*, vol. 15, no. 2, pp. 77–90, 2021.

[7] Y. Xu, X. Zhang, Y. Fu, and Y. Liu, "Interfacing photonics with artificial intelligence: an innovative design strategy for photonic structures and devices based on artificial neural networks," *Photonics Research*, vol. 9, no. 4, pp. B135–B152, 2021.

[8] Y. Xu, B. Wang, Y. Sakamoto, T. Yamamoto, and Y. Nishimura, "Comparison of learning-based surrogate models for electric motors."

[9] H. Sasaki and H. Igarashi, "Topology optimization accelerated by deep learning," *IEEE Transactions on Magnetics*, vol. 55, no. 6, pp. 1–5, 2019.

[10] A. Khan, V. Ghorbanian, and D. Lowther, "Deep learning for magnetic field estimation," *IEEE Transactions on Magnetics*, vol. 55, no. 6, pp. 1–4, 2019.

[11] A. Khan, M. H. Mohammadi, V. Ghorbanian, and D. Lowther, "Efficiency map prediction of motor drives using deep learning," *IEEE Transactions on Magnetics*, vol. 56, no. 3, pp. 1–4, 2020.

[12] Z. Pan, S. Fang, H. Wang, and Y. Zhong, "Accurate and efficient surrogate model-assisted optimal design of flux reversal permanent magnet arc motor," *IEEE Transactions on Industrial Electronics*, 2022.

[13] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[14] Y. Xu, B. Wang, Y. Sakamoto, T. Yamamoto, Y. Nishimura, T. Koike-Akino, and Y. Wang, "Tandem neural networks for electric machine inverse design."

[15] L. Pinheiro Cinelli, M. Araújo Marins, E. A. Barros da Silva, and S. Lima Netto, "Variational autoencoder," in *Variational Methods for Machine Learning with Applications to Deep Networks*. Springer, 2021, pp. 111–149.

[16] N. Dilokthanakul, P. A. Mediano, M. Garnelo, M. C. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with gaussian mixture variational autoencoders," *arXiv preprint arXiv:1611.02648*, 2016.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[19] E. Goodarzi, M. Ziaei, and E. Z. Hosseinipour, *Introduction to optimization analysis in hydrosystem engineering*. Springer, 2014.