Functional Response Conditional Variational Auto-Encoders for Inverse Design of Metamaterials

Anonymous Author(s)
Affiliation
Address
email

Abstract

Metamaterials are emerging as a new paradigmatic material system, providing unprecedented and customizable properties for various engineering applications. However, the inverse design of metamaterials, which aims to retrieve the metamaterial microstructure according to a given electromagnetic response, is very challenging as it is non-trivial to unveil the nonintuitive and intricate relationship between the microstructures, and their functional responses. In this study, we resolve this critical problem by extending the classic conditional variational autoencoder for discrete responses to a more general version that can handle functional responses. By encoding microstructures and their electromagnetic response curves into common latent spaces via deep neural networks and aligning them via a specific loss function, the proposed functional response conditional variational autoencoder can unveil the implicit relationship between microstructures and their electromagnetic responses efficiently. The proposed novel learning framework not only facilitates metamaterial design greatly by avoiding the time-consuming case-by-case numerical simulations in the traditional forward design, but also has the potential to resolve other problems with similar structures.

1 Introduction

Metamaterials are macroscopic composites that contain artificial, three-dimensional, periodic (or not) unit-cell patterns engineered to produce optimized responses to a specific excitation that is unseen in natural materials [1–4]. Due to their great potentials to manipulate electromagnetic waves, metamaterials have drawn great interests in achieving novel physics phenomenon [5], and become a breakthrough technology to realize unique functionality in various fields [6–12].

Like atoms forming a molecule in natural materials, metamaterials with various microstructures (i.e, facility topologies) can lead to different response curves. To be concrete, for a microstructure with facility topology $\mathbf{x}$, its responses to electromagnetic wave of different frequencies form a complex response curve $\mathbf{y}$. The laws of physics determine that there exists a deterministic function $\mathbf{y} = f(\mathbf{x})$ that maps the facility topology $\mathbf{x}$ to its response curves $\mathbf{y}$. In the forward design of metamaterials, we aim to learn the unknown mapping function $f(\cdot)$, i.e., predict $\mathbf{y}$ for a given $\mathbf{x}$; in the inverse design, we focus on learning the inverse mapping $f^{-1}(\cdot)$ instead, i.e., finding an appropriate $\mathbf{x}$ in a pre-given design space $\mathcal{X}$ whose response curves $\mathbf{y}$ is close enough to given target response $\mathbf{y}^*$.

In practice, researchers and designers utilize full-wave simulations via *finite element method* (FEM) to obtain mapping pairs of facility topology and its response curves in high-throughput, and try to learn the forward or inverse mapping functions, i.e., $f(\cdot)$ or $f^{-1}(\cdot)$, from the simulated data. Consider the design space $\mathcal{X}$ as the assemble of all images with $L \times L$ binary pixels, where the black pixels stand for substrate, while the white ones are metal material. Figure 1 demonstrates a typical microstructure of the I-shape and the corresponding response curves composed of four channels (two magnitude

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

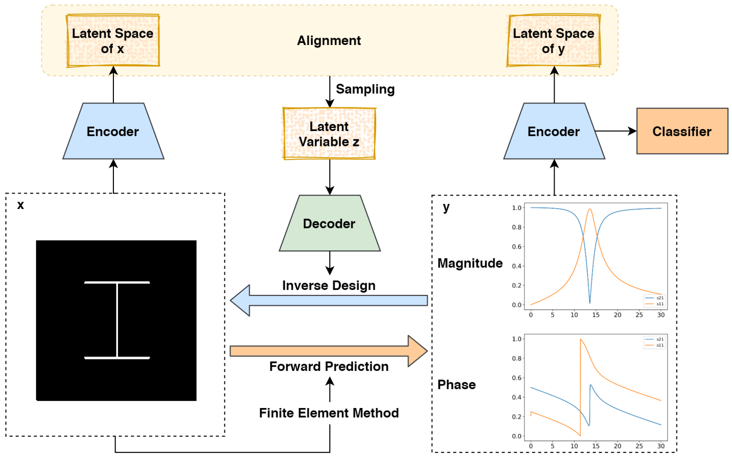

Figure 1: The proposed model for metamaterial design, characterization and classification.

channels and two phase channels). More microstructures of different topology types are illustrated in Figure 2 and Figure 3. For a collection of design points $\mathbf{x}_1, .., \mathbf{x}_n \in \mathcal{X}$, let τ_i be the topology type of $\mathbf{x}_i$ (e.g., I-shape, hexagon-shape and so on), and $\mathbf{y}_i = f(\mathbf{x}_i)$ being the corresponding response curves obtained via FEM simulation. Our goal in this study is to learn the inverse mapping function $f^{-1}(\cdot)$ from a collection of triplets $\{(\tau_i, \mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$.

Leveraging on the quick development of *deep neural network* (DNN) in recent years, DNN-based inverse design via *variational auto-encoder* (VAE) [13] and *conditional variational auto-encoder* (CVAE) [14] has gained great successes in a broad range of applications [15–20]. However, available methods for inverse design based on CVAE assume that the responses are discrete classification labels, and cannot handle complex response curves as encountered here. In this work, we fill in this gap by proposing a novel CVAE framework with functional responses as conditional input (referred to as FR-CVAE) that can successfully map the unstructured design space $\mathcal{X}$ and the complex functional response space $\mathcal{Y} = \{\mathbf{y} = f(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$. A series of simulation experiments confirm that the proposed method is effective to achieve high-quality inverse design for metamaterials.

2 Method

Figure 1 illustrates the overall architecture of the proposed FR-CVAE, which is composed of four components: (1) an encoding network of $\mathbf{x}$, $\phi_{\boldsymbol{\alpha}} : \mathbf{x} \to \mathbf{z}$, that maps a design $\mathbf{x} \in \mathcal{X}$ to a lower dimension latent space representation $\mathbf{z} \in \mathcal{Z}$ ($\mathcal{Z} \in \mathcal{R}^p$), which can also be expressed as an encoding distribution $q_{\boldsymbol{\alpha}}(\mathbf{z}|\mathbf{x}) = \mathbf{N}(\boldsymbol{\mu}_{\mathbf{z}}(\mathbf{x}, \phi_{\boldsymbol{\alpha}}), \sigma^2_{\mathbf{z}}(\mathbf{x}, \phi_{\boldsymbol{\alpha}}) \cdot \mathbf{I}_p)$, (2) an encoding network of $\mathbf{y}$ referred to as $\phi_{\boldsymbol{\beta}} : \mathbf{y} \to \mathbf{z}$, that embeds the functional response $\mathbf{y}$ into the same latent space $\mathcal{Z}$ via another encoding distribution $q_{\boldsymbol{\beta}}(\mathbf{z}|\mathbf{y}) = \mathbf{N}(\boldsymbol{\mu}_{\mathbf{z}}(\mathbf{y}, \phi_{\boldsymbol{\beta}}), \sigma^2_{\mathbf{z}}(\mathbf{y}, \phi_{\boldsymbol{\beta}}) \cdot \mathbf{I}_p)$, (3) a decoding network $\phi_{\boldsymbol{\gamma}} : \mathbf{z} \to \mathbf{x}$, that generates an image $\hat{\mathbf{x}} \in \mathcal{X}$ from $\mathbf{z} \in \mathcal{Z}$ via a decoding distribution $q_{\boldsymbol{\gamma}}(\mathbf{x}|\mathbf{z})$ over the design space $\mathcal{X}$, and (4) a classifier $\phi_{\boldsymbol{\psi}} : y \to p_\tau$ which shares the network of $\phi_{\boldsymbol{\beta}}$ except its last layer and utilize a linear layer parameterized by ψ and softmax function to generate the classification probability of topology types, p_τ. Let $\boldsymbol{\Theta} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\psi})$ denote all parameters involved in the model.

Among the four involved networks, the encoding network $\phi_{\boldsymbol{\alpha}}$, and the decoding network $\phi_{\boldsymbol{\gamma}}$, are exactly same as in the classic CVAE [14]. However, unlike the traditional CVAE using discrete classification label as condition input [14], the proposed FR-CVAE introduces an extra encoder network $\phi_{\boldsymbol{\beta}}$, to take care of the complex responses, which are continuous curves, and an additional classifier, i.e., $\phi_{\boldsymbol{\psi}}$, to format multi-task learning to guide the encoding process of $\phi_{\boldsymbol{\beta}}$. In principle, we can specify $\phi_{\boldsymbol{\beta}}$ with any DNN that can convert a high-dimensional response curve $\mathbf{y}$ into the latent space $\mathcal{Z}$. Here, we choose the Swin-Transformer [21] as the encoding network $\phi_{\boldsymbol{\beta}}$, considering its effectiveness to capture complicated patterns from sequence data due to its attention mechanism.

The loss function of FR-CVAE is composed of three components. The first component is the reconstruction loss $\mathcal{L}_x(\boldsymbol{\alpha}, \boldsymbol{\gamma})$, which plays exactly the same role as in the classic VAE or CVAE. For the i-th data point $(\tau_i, \mathbf{x}_i, \mathbf{y}_i)$, the $\mathcal{L}_x$ loss has the the following form:

$$\mathcal{L}_x(\boldsymbol{\alpha}, \boldsymbol{\gamma}; \mathbf{x}_i) = -\int \Big[\log q_{\boldsymbol{\gamma}}(\mathbf{x}_i|\mathbf{z})\Big] dq_{\boldsymbol{\alpha}}(\mathbf{z}|\mathbf{x}_i), \tag{1}$$

where the integration is about the latent vector $\mathbf{z}$ over the whole latent space $\mathcal{Z}$. The second component $\mathcal{L}_y(\boldsymbol{\beta}, \boldsymbol{\psi})$ is classification loss for $\mathbf{y}$, the cross-entropy loss $\mathcal{L}_{CE}$ [22] enhanced by an additional triplet loss $\mathcal{L}_{Triplet}$ [23], i.e.,

$$\mathcal{L}_y(\boldsymbol{\beta}, \boldsymbol{\psi}; \tau_i, \mathbf{y}_i) = \mathcal{L}_{CE}(\boldsymbol{\beta}, \boldsymbol{\psi}; \tau_i, \mathbf{y}_i) + \mathcal{L}_{Triplet}(\boldsymbol{\beta}; \tau_i, \mathbf{y}_i). \tag{2}$$

The third component $\mathcal{L}_{x\sim y}(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is applied to stabilize and align the stochastic encoding of $\mathbf{x}$ and $\mathbf{y}$ via

$$\mathcal{L}_{x\sim y}(\boldsymbol{\alpha}, \boldsymbol{\beta}; \mathbf{x}_i, \mathbf{y}_i) = w_1 \cdot KL\big(q_{\boldsymbol{\alpha}}(\cdot|\mathbf{x}_i)||\pi_0(\cdot)\big) + w_2 \cdot KL\big(q_{\boldsymbol{\alpha}}(\cdot|\mathbf{x}_i)||q_{\boldsymbol{\beta}}(\cdot|\mathbf{y}_i)\big), \tag{3}$$

where the first KL divergence plays the role of stabilization as in the ordinary VAE, since it forces the stochastic encoding function $q_{\boldsymbol{\alpha}}(\cdot|\mathbf{x}_i)$ of every $\mathbf{x}_i$ to be close to a pre-given distribution π_0 (which is typically the stand normal distribution on $\mathcal{Z}$), while the second one connects the encoding of $\mathbf{x}_i$ and $\mathbf{y}_i$ via distribution alignment. Assembling all these components together, we come up with the following joint loss function:

$$\mathcal{L}\left(\boldsymbol{\Theta} \mid \{(\tau_i, \mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n\right) = \sum_{i=1}^n \{\mathcal{L}_x(\boldsymbol{\alpha}, \boldsymbol{\gamma}; \mathbf{x}_i) + \mathcal{L}_y(\boldsymbol{\beta}, \boldsymbol{\psi}; \tau_i, \mathbf{y}_i) + \mathcal{L}_{x\sim y}(\boldsymbol{\alpha}, \boldsymbol{\beta}; \mathbf{x}_i, \mathbf{y}_i)\}. \tag{4}$$

In practice, w_1 and w_2 in Eq. (3) need to be properly specified to adjust the relative weight of the $\mathcal{L}_{x\sim y}$ loss. We simply set $w_1 = w_2 = 1$ in this study. We also note that the proposed FR-CVAE would degenerate to VAE of $\mathbf{x}$ if we remove the alignment loss by setting $w_2 = 0$. The proposed FR-CVAE can be trained in a similar way as CVAE. The complete training procedure is detailed in Algorithm 1.

Algorithm 1 Conditional Variational Auto-Encoding for Functional Responses Optimization

Input: training data set $\{\tau_i, \mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$, batch size M, and loss weights (w_1, w_2)
Initialization: random initialized $\boldsymbol{\Theta}_0 = (\boldsymbol{\alpha}_0, \boldsymbol{\beta}_0, \boldsymbol{\psi}_0, \boldsymbol{\gamma}_0)$.
Output: parameters $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, \boldsymbol{\psi}^*, \boldsymbol{\gamma}^*)$.

repeat
- Sample $(\boldsymbol{\tau}, \mathbf{X}, \mathbf{Y}) \leftarrow$ Random minibatch drawn from full dataset;
- Encoder of $\mathbf{X}$: $\boldsymbol{\mu}_{\mathbf{z}}(\mathbf{X}), \boldsymbol{\Sigma}_{\mathbf{z}}(\mathbf{X}) \leftarrow \phi_{\boldsymbol{\alpha}}$;
- Encoder of $\mathbf{Y}$ (Swin-Transformer): $\boldsymbol{\mu}_{\mathbf{z}}(\mathbf{Y}), \boldsymbol{\Sigma}_{\mathbf{z}}(\mathbf{Y}) \leftarrow \phi_{\boldsymbol{\beta}}$;
- Classifier: $\hat{\boldsymbol{\tau}} \leftarrow \phi_{\boldsymbol{\psi}}$;
- Sample $\mathbf{z}^x \leftarrow \boldsymbol{\mu}_{\mathbf{z}}(\mathbf{X}) + \boldsymbol{\epsilon} \odot (\boldsymbol{\Sigma}_{\mathbf{z}}(\mathbf{X}))^{\frac{1}{2}}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
- Sample $\mathbf{z}^y \leftarrow \boldsymbol{\mu}_{\mathbf{z}}(\mathbf{Y}) + \boldsymbol{\epsilon} \odot (\boldsymbol{\Sigma}_{\mathbf{z}}(\mathbf{Y})))^{\frac{1}{2}}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
- Decoder: $\hat{\mathbf{X}} \leftarrow \phi_{\boldsymbol{\gamma}}$;
- Compute the loss $\mathcal{L}(\boldsymbol{\Theta} \mid \boldsymbol{\tau}, \mathbf{X}, \mathbf{Y})$ according to Eq. (4)
- Back-propagate the gradients.

until maximum iteration reached

3 Experiments

3.1 Experimental Setup

Dataset Simulated response curves of 61,992 microstructure patterns belonging to 30 topology types (e.g., I-shape, cross shaped, split ring, circular, etc.) were collected to support this study. On average, about 2,000 simulated response curves were collected for each topology type at different scales. The image of every involved microstructure pattern was encoded into a 200×200 binary matrix, and the corresponding response curves are the real and imaginary part of scatter parameters, S11 and S21, over the frequency region of 0.1-30GHz, formatting as a 4×1001-dimensional vector.

Implementation Details We randomly selected 80% of the collected data (i.e., 49,594 microstructures with their corresponding response curves) to train the proposed FR-CVAE, and used the rest 20% for testing. The training is performed via Adam optimizer [24] through minibatch gradient descent for 1,000 epochs with the batch size set to be 256, which takes about fifteen hours by using 2 Nvidia Telsa P100 16GB GPU cards.

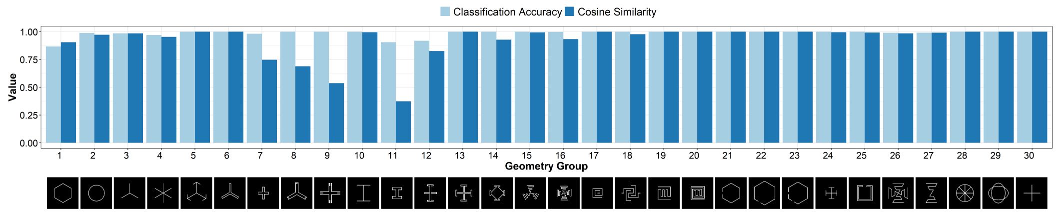

Figure 2: Numerical evaluation of the proposed model with ϕ_β being Swin-Transformer. 1-30 represent the topology types, each of which contains samples from the test data set.

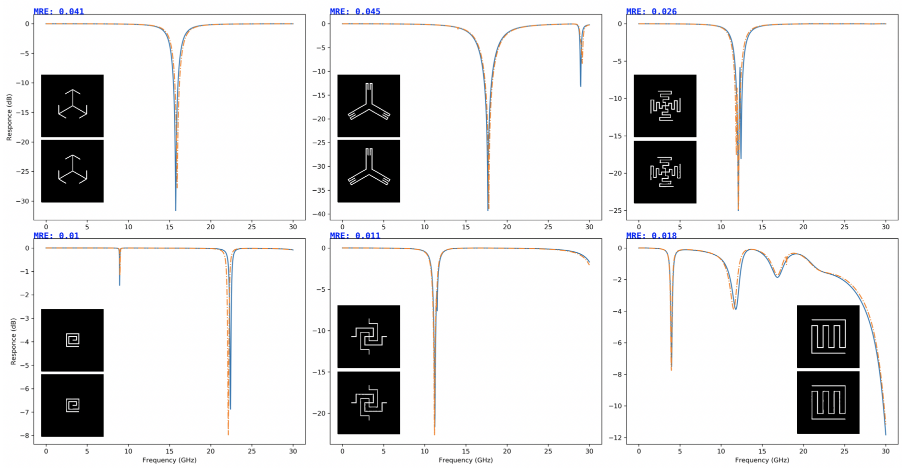

Figure 3: On-demand inverse design. The two insets are the ground-truth design patterns (up) whose response curves are solid blue and retrieved design patterns (down) whose response curves are dashed yellow.

3.2 Experimental Results

Classification and Similarity The bar-plots in Figure 2 summarize the quality of inverse design based on the proposed FR-CVAE with Swin-Transformer (results of MLP based functional response encoder shown as Figure A2) for each of the 30 topology types: the light blue bars show the classification accuracy that the generated structures belong to the ground true topology type, the dark blue bars report the average cosine similarity between the two embedding vectors of the generated and ground true microstructure. The proposed FR-CVAE achieves high-quality results in both perspectives for most topology types, suggesting that designs very close to the ideal ones can be successfully captured.

On-demand Inverse Design To further check whether the generated microstructures can indeed produce response curves that are close to the target response curves, we visualize the response curves of a few generated microstructures versus their target response curves in Figure 3, with the images of the generated and ground true microstructure showed side by side as well. From the figure we can see clearly that most of the generated microstructures have a clear and feasible configuration and the generated designs reproduce the corresponding input response curves with high fidelity, which means the trained FR-CVAE model can effectively link the microstructure design and response curve through the probabilistic representation by latent variables and even preserve some fine features. However, it can also be noted that some generated microstructures have blurred regions on the boundary, which is a common phenomenon for generative models with a log-likelihood loss function [25].

References

[1] David Schurig et al. “Metamaterial Electromagnetic Cloak at Microwave Frequencies”. In: *Science* 314.5801 (2006), pp. 977–980.

[2] R.A. Shelby, D.R. Smith, and S. Schultz. "Experimental verification of a negative index of refraction." In: *Science* 292 (2001), pp. 77–79.

[3] R Marqués, F Martín, and M. Sorolla. *Metamaterials with Negative Parameters*. John Wiley Sons, Ltd, 2007. ISBN: 9780470191736.

[4] T. J. Cui, D. Smith, and R. Liu. *Metamaterials*. Springer US, 2010.

[5] D. L. Mcdowell. "Integrated Design of Multiscale, Multifunctional Materials and Products". In: *Integrated Design of Multiscale Multifunctional Materials amp; Products* 91.3 (2009).

[6] Liu et al. "Broadband Ground-Plane Cloak." In: *Science* (2009).

[7] Andrea Alù and Nader Engheta. "Achieving transparency with plasmonic and metamaterial coatings". In: *Phys. Rev. E* 72 (1 July 2005), p. 016623. DOI: `10.1103/PhysRevE.72.016623`. URL: `https://link.aps.org/doi/10.1103/PhysRevE.72.016623`.

[8] Xingjie Ni et al. "An ultrathin invisibility skin cloak for visible light". In: *Science* 349.6254 (2015), pp. 1310–1314. DOI: `10.1126/science.aac9411`.

[9] Francesco Monticone, Nasim Mohammadi Estakhri, and Andrea Alù. "Full Control of Nanoscale Optical Transmission with a Composite Metascreen". In: *Phys. Rev. Lett.* 110 (20 May 2013), p. 203903. DOI: `10.1103/PhysRevLett.110.203903`. URL: `https://link.aps.org/doi/10.1103/PhysRevLett.110.203903`.

[10] C. Enkrich et al. "Magnetic metamaterials at telecommunication and visible frequencies". In: *Physical Review Letters* 95.20 (2005).

[11] C. Tao, S. Li, and S. Hui. "Metamaterials Application in Sensing". In: *Sensors* 12.3 (2012), pp. 2742–2765.

[12] Mri Faruque, M. T. Islam, and N. Misran. "Electromagnetic (EM) absorption reduction in a muscle cube with metamaterial attachment". In: *Medical Engineering amp; Physics* 33.5 (2011), pp. 646–652.

[13] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *ICLR*. 2014.

[14] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. "Learning structured output representation using deep conditional generative models". In: *Advances in neural information processing systems* 28 (2015), pp. 3483–3491.

[15] Ma et al. "Deep-Learning-Enabled On-Demand Design of Chiral Metamaterials". In: *ACS nano* (2018).

[16] Zhaocheng et al. "Generative Model for the Inverse Design of Metasurfaces." In: *Nano letters* (2018).

[17] W. Ma et al. "Probabilistic representation and inverse design of metamaterials based on a deep generative model with semi-supervised learning strategy". In: *Advanced Materials* 31.35 (2019), pp. 1901111.1–1901111.9.

[18] B Lwa et al. "Deep generative modeling for mechanistic-based learning and design of metamaterial systems". In: *Computer Methods in Applied Mechanics and Engineering* 372 (2020).

[19] T. Qiu et al. "Deep Learning: A Rapid and Efficient Route to Automatic Metasurface Design". In: *Advanced Science* (2019).

[20] X. Li et al. "Designing phononic crystal with anticipated band gap through a deep learning based data-driven method". In: *Computer Methods in Applied Mechanics and Engineering* 361.36 (2019).

[21] Z. Liu et al. "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows". In: (2021).

[22] *Cross Entropy Loss*. `https://pytorch.org/docs/master/generated/torch.nn.CrossEntropyLoss.html#crossentropyloss`.

[23] F Schroff, D. Kalenichenko, and J. Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.

[24] D. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: *Computer Science* (2014).

[25] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. The MIT Press, 2016.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [No]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code and the data are proprietary.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 3.1.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 3.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] The data are proprietary.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] The data we are using contains no personally identifiable information or offensive content.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

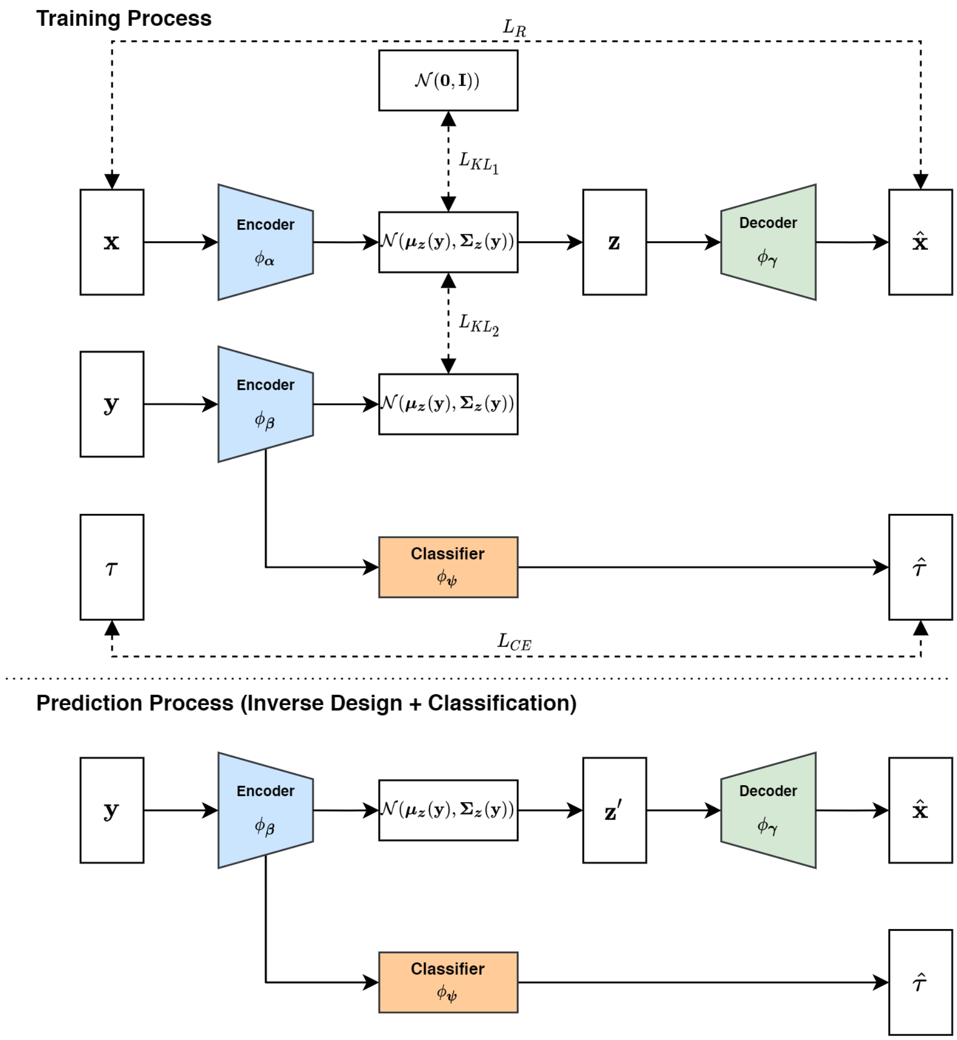

Figure A1: Architecture of the proposed deep generative model.

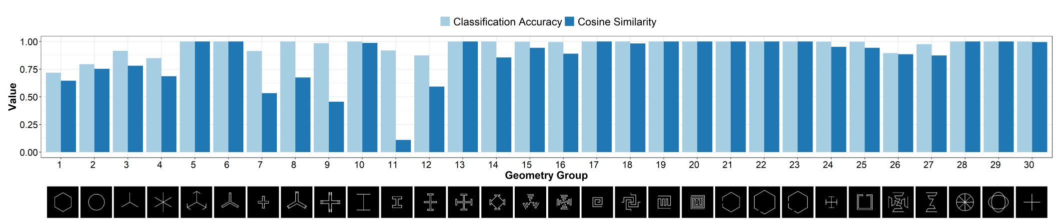

Figure A2: Numerical evaluation of the proposed model with ϕ_β being MLP.