Nanoscale Advances

Supplementary Materials

Leveraging Generative Neural Networks for Accurate, Diverse, and Robust Nanoparticle Design†

Tanzim Rahman*[a], Ahnaf Tahmid*[a], Shifat E. Arman[b], Tanvir Ahmed[c], Zarin Tasnim Rakhy[c], Harinarayan Das[d], Mahmudur Rahman[e], Abul Kalam Azad[a], Md. Wahadoszamen[c], and Ahsan Habib[a]

Supplementary Text 1: Tandem Neural Network

The tandem neural network has two parts, it utilizes a forward network and an inverse network during training. In inverse design applications specifically, a forward model is initially trained to be able to predict the response or output for any design within the parameters of the problem. In case of training the forward model, the loss MSE was used, as shown in the equation:

$$L_{forward} = \frac{1}{n}\sum_{i=1}^{n} \|y(i) - \hat{y}(i)\|^2 \tag{1}$$

where $y(i)$ is the true output, $\hat{y}(i)$ is the predicted output for the i-th training sample and n is the number of data points taken for the spectra.

Instead of directly training an inverse network to predict design based on desired responses, the two models operate in tandem as a whole during training the inverse model. The loss is not calculated directly based on the predictions of the inverse model. Rather the output from the inverse model is passed to the forward model and the loss is calculated by comparing the response of predicted particles with the desired response. MSE was also used in this case as the loss as shown in the equation:

$$L_{tandem} = \frac{1}{n}\sum_{i=1}^{n} \|y(i) - \hat{y}(\hat{x}(i))\|^2 \tag{2}$$

where $y(i)$ is the desired output, $\hat{x}(i)$ is the predicted output of the inverse model and $\hat{y}(\hat{x}(i))$ is the predicted response by the forward model for the i-th training sample and n is the number of data points taken for the spectra. For the inverse model, the input was the spectral response, provided as the condition input and the output was predicting the particle design.

[a] *Department of Electrical and Electronic Engineering, University of Dhaka, Dhaka-1000, Bangladesh; E-mail: mahabib@du.ac.bd*
[b] *Department of Robotics and Mechatronics Engineering, University of Dhaka, Dhaka-1000, Bangladesh*
[c] *Department of Physics, University of Dhaka, Dhaka-1000, Bangladesh*
[d] *Bangladesh Atomic Energy Commission, Dhaka-1000, Bangladesh*
[e] *Dhaka University of Engineering & Technology, Gazipur-1707, Bangladesh*
* These authors contributed equally to this work.

Supplementary Text 2: Conditional Variational Autoencoder Neural Network

Variational autoencoder is a type of generative model. In a generative model, a low dimensional latent vector is converted into a high dimensional output by a neural network. A simple generative model can be expressed by the following equation:

$$x = G_\theta(z) \tag{3}$$

Here, x is the high dimensional output, z is the low dimensional latent vector and G_θ is a generative model parameterized by θ.

An autoencoder consists of an encoder network and a decoder network. The encoder network converts the high dimensional input into a low dimensional latent vector, whereas the decoder network converts the low dimensional latent vector into a high dimensional reconstructed output.

$$x' = D_\theta(E_\phi(x)) \tag{4}$$

Here, x is the input, x' is the reconstructed output, D_θ is the decoder network parameterized by θ and $E_\phi(x)$ is the encoder network parameterized by ϕ.

During training an encoder, the parameters are updated such that, the reconstructed output becomes similar to the original input given to the network.

$$\min_{\theta,\phi} \sum_{i=1}^{n} \left\| D_\theta\left(E_\phi\left(x_i\right)\right) - x_i \right\|^2 \quad \text{w/ } \{x_i\}_{i=1-n} \tag{5}$$

A plain autoencoder is not a generative model, as it does not define a distribution.

The total loss function in a variational autoencoder comprises two loss functions serving different functions. The two losses are reconstruction loss and Kullback-Leibler (KL) divergence loss. The Reconstruction Loss measures the difference between the input x and the reconstructed output $hatx$. If the input data is continuous, Mean Squared Error (MSE) can be used that calculates the squared differences between x and $\hat{x}$.

$$\text{Reconstruction Loss} = ||x - \hat{x}||^2 \tag{6}$$

Here, x is the input and $\hat{x}$ is the reconstructed output.

The KL Divergence Loss acts as a regularizer that ensures that the latent space $q_\phi(z|x)$ follows a prior distribution $p(z)$, which is usually a standard normal distribution $N(0,I)$.

The conditional variational autoencoder(cVAE) extends this architecture and introduces a conditional input. This allows for the decoder to generate outputs based on a given input condition. The conditional input is given to both the encoder and decoder networks during training. The encoder network learns and establishes a relationship between the input condition and the latent variables. During generation, the condition and latent vector are given to the decoder as inputs. This allows the decoder to learn multiple solutions for the same condition as the value of the latent vector is different for each solution.

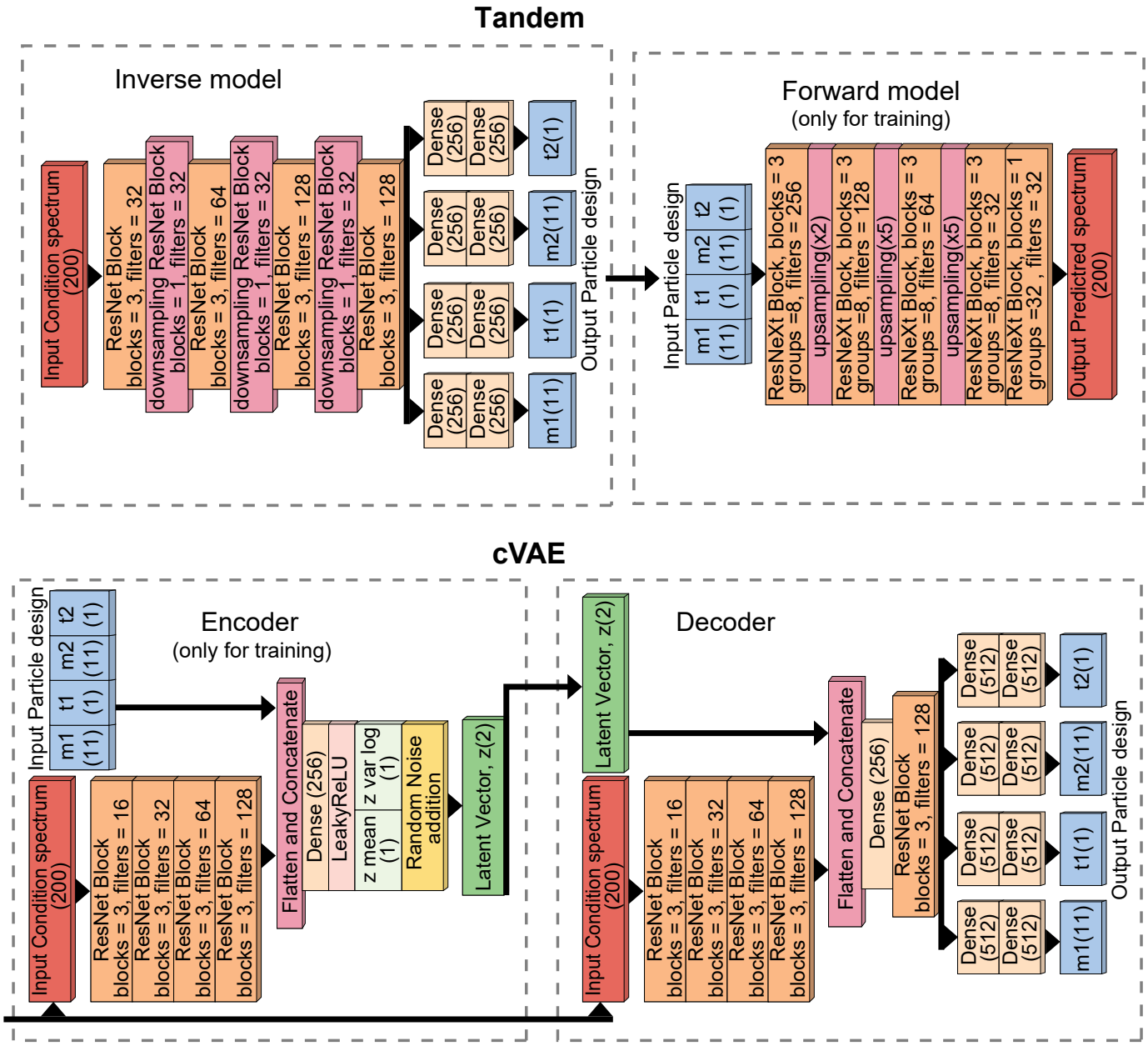

Fig. S1 Schematic of Tandem and cVAE models for inverse design of core-shell nanoparticles. The values within the brackets represent the size of a particular layer. The Tandem model consists of a forward model and an inverse model. The forward model predicts the optical spectrum from an input particle design, using a series of ResNet blocks, while the inverse model takes the desired condition spectrum as input to reconstruct the particle design using downsampling ResNet blocks and dense layers. In the cVAE model, the encoder takes both the particle design and the condition spectrum as inputs. These inputs pass through ResNet blocks. The encoder then generates a latent vector. The decoder then reconstructs the particle design based on the latent vector and input condition spectrum using downsampling ResNet blocks and dense layers.

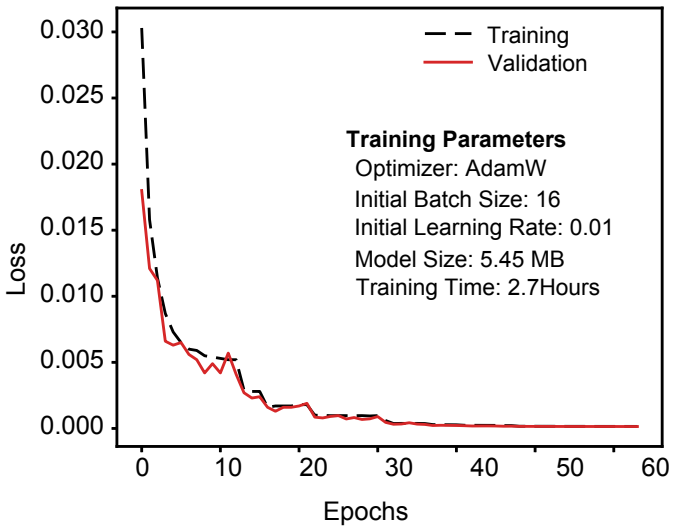

Fig. S2 Training and validation loss curves for the forward networks (for tandem) used in the study. The model was trained using the AdamW optimizer with an initial batch size of 16 and an initial learning rate of 0.01. The total training time was 2.7 hours, and the model size is 5.45 MB.

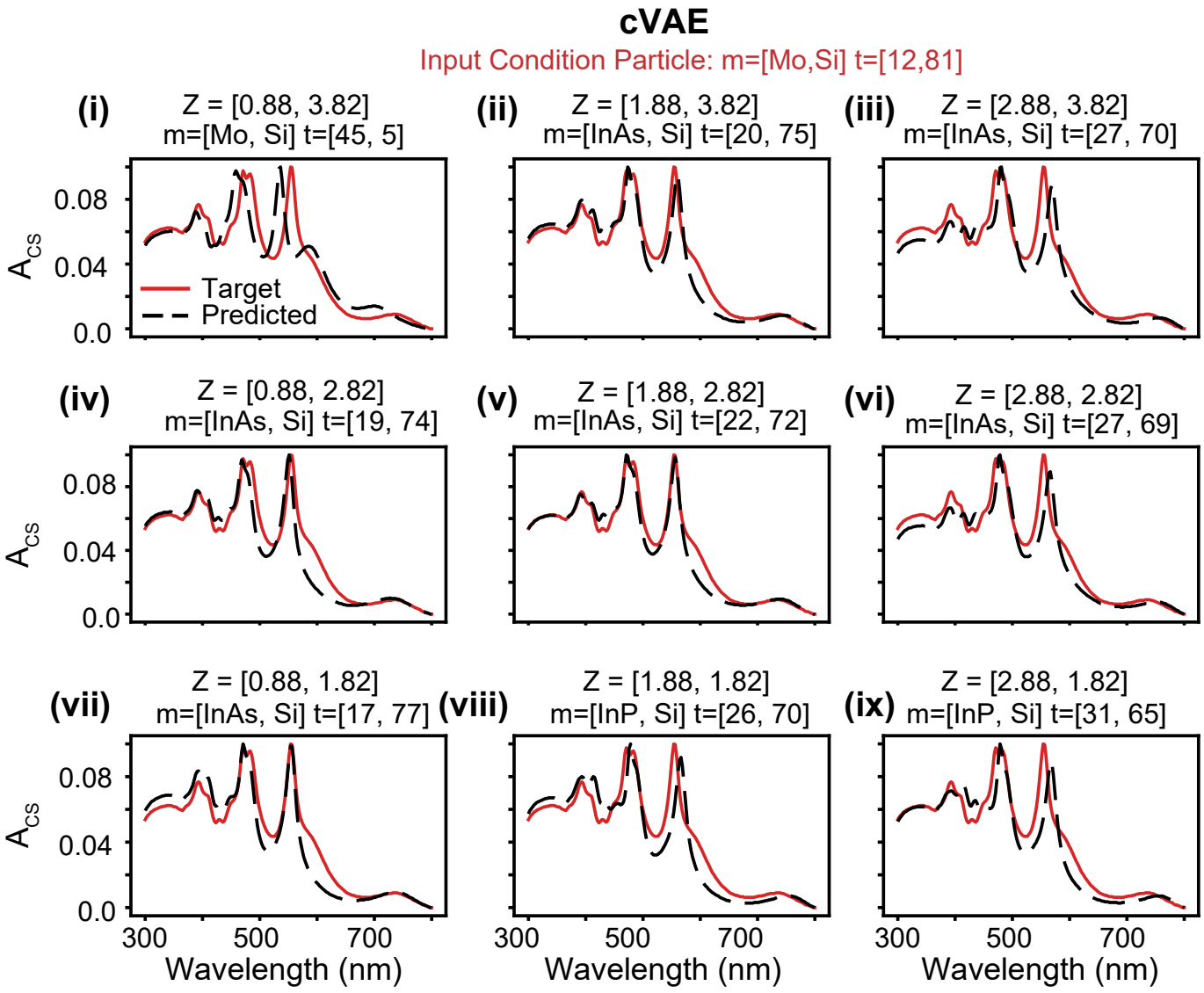

Fig. S3 Additional examples of the cVAE's capability in predicting varied designs while maintaining accurate optical properties are included.

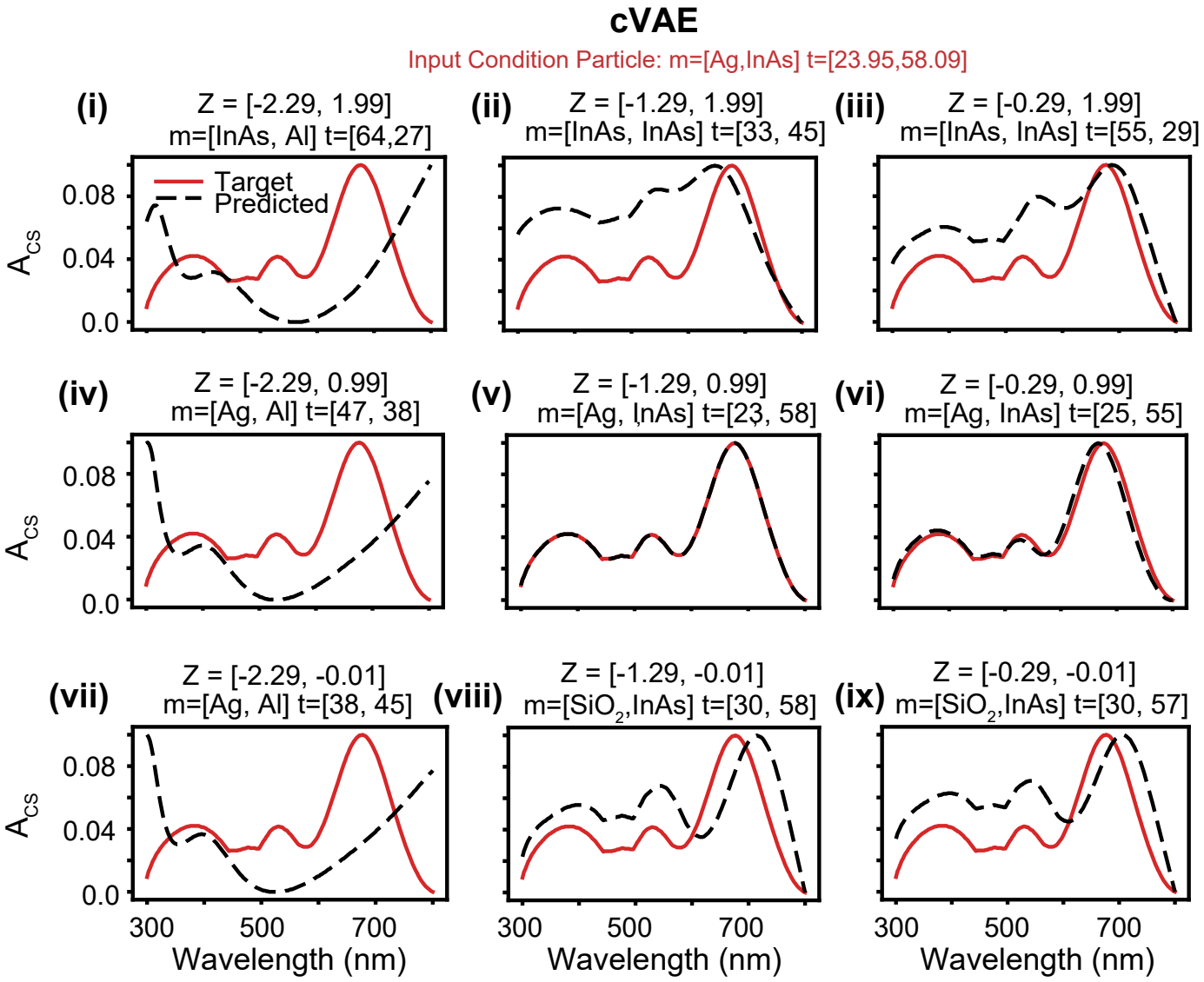

Fig. S4 Additional examples of the cVAE's capability in predicting varied designs while maintaining accurate optical properties are included. It shows the cVAE model traversing the latent space and trying multiple solutions for a given input condition.

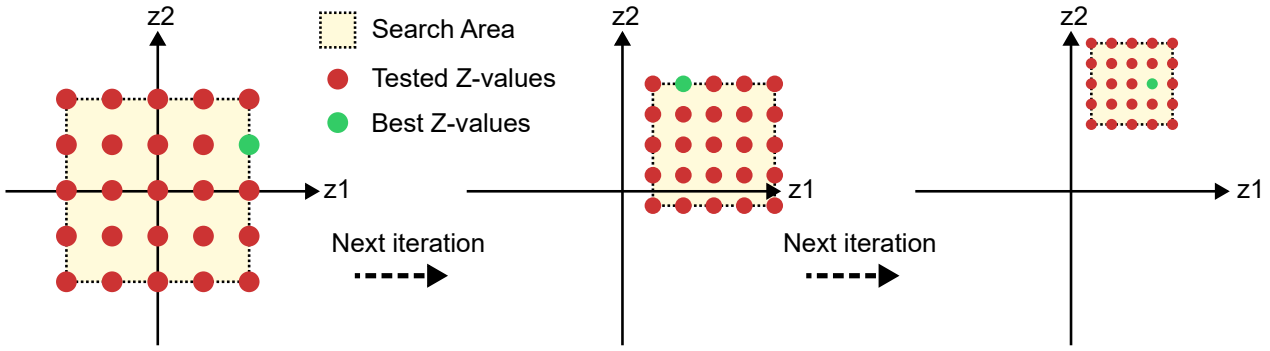

Fig. S5 Illustration of the method for choosing z values. The search area is centered on the origin (0, 0), and a 5x5 grid of the latent space is examined. The z-value with the lowest MAE is used as the new origin in subsequent iterations, with the search area size halved each time. This process is repeated five times to find the best z-value for the final model prediction.

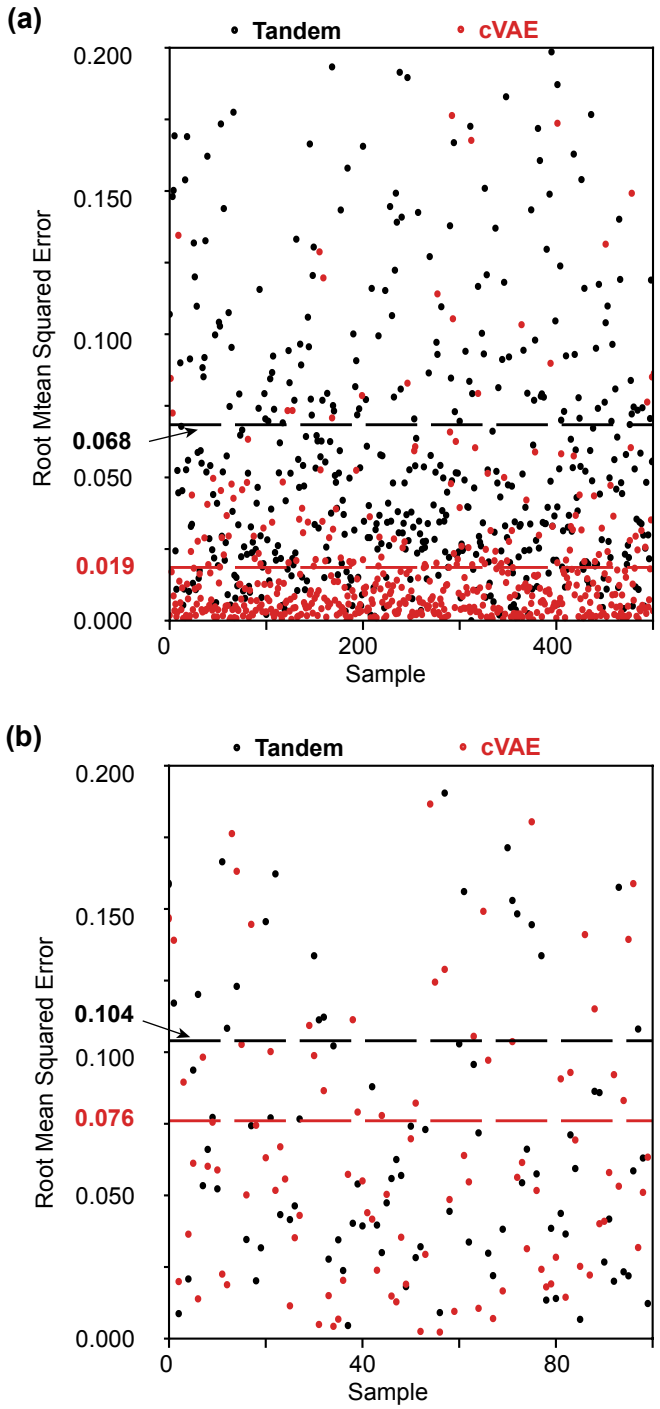

Fig. S6 Comparison of root mean squared error (RMSE) between the spectral responses of the predicted designs and the input condition spectra for both the tandem model and the cVAE model. (a) Scatter plot showing the RMSE for 500 randomly generated nanospheres. The cVAE model demonstrates better performance with a mean RMSE of 0.019 compared to 0.068 for the tandem model. (b) Scatter plot showing the robustness analysis with 100 test spectra. The cVAE model achieves a mean RMSE of 0.076 compared to 0.104 for the tandem model, further confirming its superior robustness.

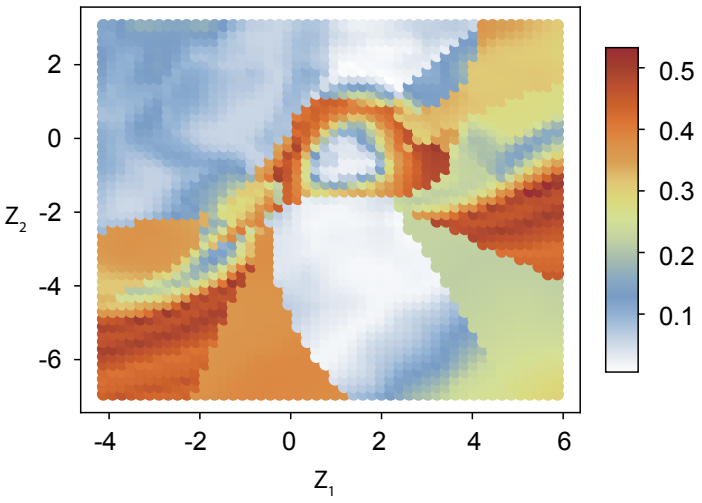

Fig. S7 RMSE distributions in the latent space for a particular test target.

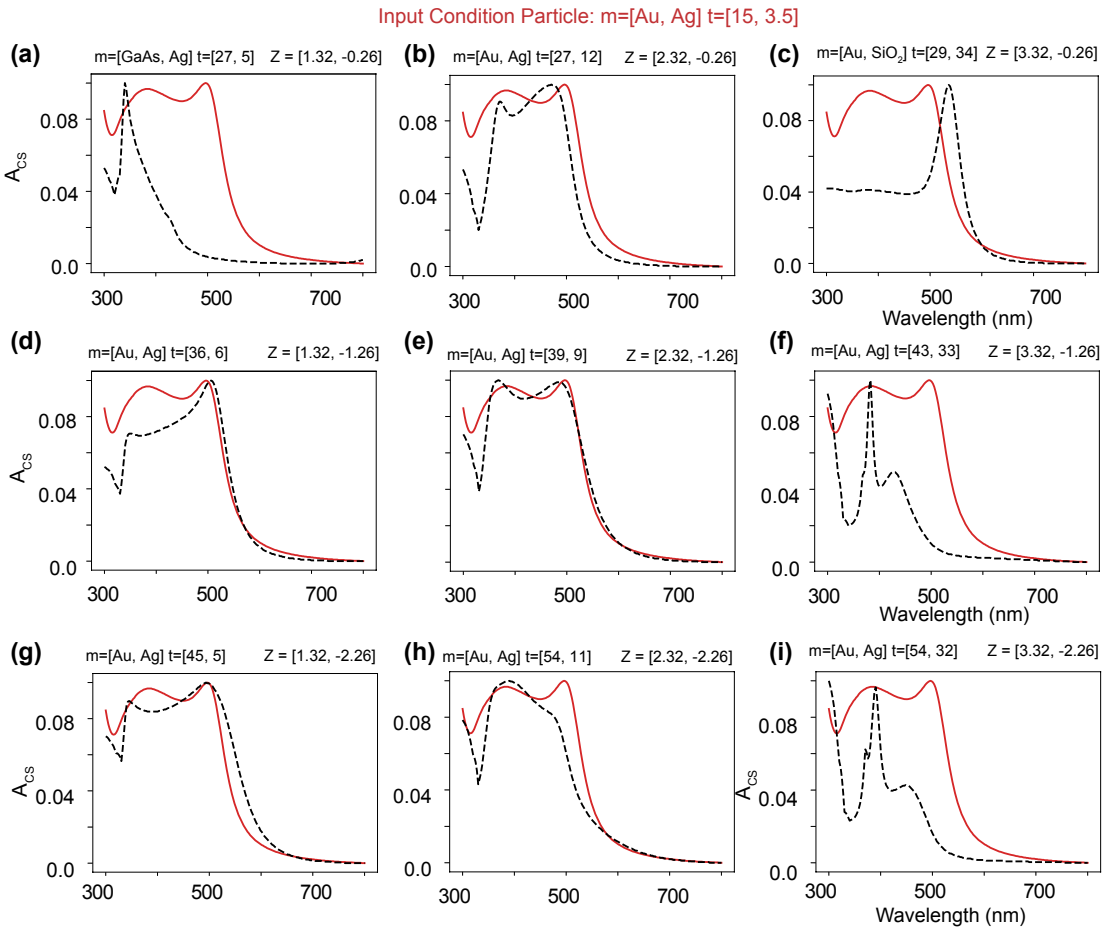

Fig. S8 Comparison of the experimental absorption cross-section (red curve) given to the models as the input condition and the simulated response from the predicted inverse design particles from the cVAE models (dashed black curve). The cVAE model, due to its advantageous diversity and robustness, shows a close match to the target spectrum, resulting in a low MAE. The figure also illustrates that multiple solutions exist where the predicted spectrum closely matches the target spectrum with varied geometrical parameters.

Table S1 Material IDs and sources for refractive indices for the materials used during this work.

Material	Ag	Al	Au	Cu	GaAs	InAs	InP	Mo	Si	SiO_2
ID	1	2	3	4	5	6	7	8	9	10
Ref.	1	2	1	3	4	5	5	6	5	7

Notes and references

1 K. M. McPeak, S. V. Jayanti, S. J. P. Kress, S. Meyer, S. Iotti, A. Rossinelli, and D. J. Norris, "Plasmonic Properties of Gold Nanoparticles: A Comprehensive Study," ACS Photonics **2**(3), 326-333 (2015).

2 F. Cheng, P.-H. Su, J. Choi, S. Gwo, X. Li, and C.-K. Shih, "Epitaxial Growth of Atomically Smooth Aluminum on Silicon and Its Intrinsic Optical Properties," ACS Nano **10**(11), 9852-9860 (2016).

3 P. B. Johnson and R. W. Christy, "Optical Constants of the Noble Metals," Phys. Rev. B **6**(12), 4370-4379 (1972).

4 K. Papatryfonos, T. Angelova, A. Brimont, B. Reid, S. Guldin, P. R. Smith, M. Tang, K. Li, A. J. Seeds, H. Liu, and D. R. Selviah, "Refractive indices of MBE-grown AlxGa(1-x)As ternary alloys in the transparent wavelength region," AIP Adv. **11**(2), 025327 (2021).

5 D. E. Aspnes and A. A. Studna, "Dielectric functions and optical parameters of Si, Ge, GaP, GaAs, GaSb, InP, InAs, and InSb from 1.5 to 6.0 eV," Phys. Rev. B **27**(2), 985-1009 (1983).

6 W. S. M. Werner, K. Glantschnig, and C. Ambrosch-Draxl, "Optical Constants and Inelastic Electron-Scattering Data for 17 Elemental Metals," J. Phys. Chem. Ref. Data **38**(4), 1013-1092 (2009).

7 I. H. Malitson, "Interspecimen Comparison of the Refractive Index of Fused Silica," J. Opt. Soc. Am. **55**(10), 1205 (1965).