npj Computational Materials

www.nature.com/npjcompumats

ARTICLE OPEN

Check for updates

Inverse design of two-dimensional materials with invertible neural networks

Victor Fung[1]✉, Jiaxin Zhang[2]✉, Guoxiang Hu[3], P. Ganesh[1] and Bobby G. Sumpter[1]

The ability to readily design novel materials with chosen functional properties on-demand represents a next frontier in materials discovery. However, thoroughly and efficiently sampling the entire design space in a computationally tractable manner remains a highly challenging task. To tackle this problem, we propose an inverse design framework (MatDesINNe) utilizing invertible neural networks which can map both forward and reverse processes between the design space and target property. This approach can be used to generate materials candidates for a designated property, thereby satisfying the highly sought-after goal of inverse design. We then apply this framework to the task of band gap engineering in two-dimensional materials, starting with MoS_2. Within the design space encompassing six degrees of freedom in applied tensile, compressive and shear strain plus an external electric field, we show the framework can generate novel, high fidelity, and diverse candidates with near-chemical accuracy. We extend this generative capability further to provide insights regarding metal-insulator transition in MoS_2 which are important for memristive neuromorphic applications, among others. This approach is general and can be directly extended to other materials and their corresponding design spaces and target properties.

npj Computational Materials (2021)7:200; https://doi.org/10.1038/s41524-021-00670-x

INTRODUCTION

In materials discovery problems, it is desirable to select and test candidates which hold the most promise for satisfying a particular functional target, while maintaining as broad a diversity in the search space as possible. To this end, a data-driven approach is often used to meet these needs, whereby materials are first rapidly screened via high-throughput experimentation or computational modeling to identify potential candidates[1–3]. However, the vastness of the accessible chemical search spaces can present serious challenges for current experimental or even computational methods due to the significant evaluation time and resource demands. Machine learning provides promising solutions to this problem by providing a cheaper surrogate for the computational calculations, or by producing new candidates which have a specified target property[4–9]. The latter approach may involve the use of generative models, which allows for sampling within a continuous chemical or materials latent space which can map to unique and undiscovered materials[10–14]. Although more challenging to implement than discriminative models, generative modeling is highly appealing for its potential to realize the "inverse design" of materials and to efficiently "close the loop" between modeling and experiments[4,15–18].

In general inverse problems, given a forward process $y = f(x)$, the goal is to then find a suitable inverse model $x = f^{-1}(y)$ to map the reverse process. In the context of materials discovery, the forward mapping from materials design parameters to target property can take the form of experimental measurements or computational calculations, such as density functional theory (DFT). However, the reverse process from target property to design parameters cannot be obtained directly via the same methods but can be inferred with machine learning. One such approach uses variational autoencoders[19] (VAEs), where the encoder and decoder models learn to approximate inverse solutions upon convergence. Instead, we propose to use invertible neural networks[20] (INNs) and conditional INNs[21] (cINNs) where a single model can be trained on a forward process and the exact inverse solution can then be obtained for free. The intrinsic invertibility of INNs offers potential advantages in stability and performance[20–23] over VAEs and the popular generative adversarial networks[24] (GANs), which suffer from difficulties in training due to mode collapse.

In this work, we leverage the INN architecture to solve materials design problems and develop a framework utilizing these models to generate high-quality materials candidates with the targeted properties, outlined in Fig. 1. Starting with a given materials design space, we begin by generating training data sampling within this space using DFT. We use the data to train the invertible neural network to obtain forward and reverse mappings. The trained network is then evaluated in the reverse direction to generate samples given a target property. We note the generated samples may not directly yield sufficiently high-fidelity candidates within chemical accuracy of the target. We therefore include additional (1) down-selection based on fitness criteria and (2) optimization to localize generated samples to the exact solutions using the initialized samples from the INN. In this work, the fitness criteria for down-selection limits samples to those which are close to the desired property and with parameters within the training data distribution, but this can be further expanded to include additional criteria such as experimental feasibility as needed. Selected samples are then optimized by gradient descent with automatic differentiation[25]. For down-selection and optimization, the forward mode of the INN can be conveniently used as the property prediction surrogate and provide gradients via back-propagation. These final optimized samples may be further validated by DFT, or if the performance is sufficiently high, may be omitted and the generated samples then analyzed directly. Here we will show our framework, MatDesINNe (Materials Design

[1]Center for Nanophase Materials Sciences, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA. [2]Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA. [3]Department of Chemistry and Biochemistry, Queens College of the City University of New York, Queens, NY 11367, USA.
✉email: fungv@ornl.gov; zhangj@ornl.gov

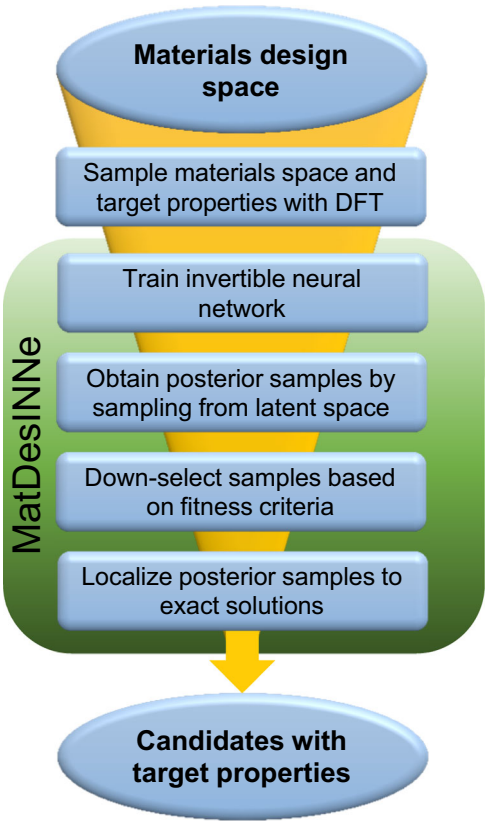

Fig. 1 Inverse materials design workflow. Starting with a specified design space, training data is obtained with DFT which is then fed to train the MatDesINNe framework using invertible neural networks. Samples are generated, down-selected and localized to ensure high quality candidates. An optional validation step with DFT ensures the candidates have the intended target properties.

with Invertible Neural Networks), can successfully achieve inverse materials design with a high accuracy.

We apply MatDesINNe to the band gap engineering problem for monolayer MoS_2, the archetypical 2D dichalcogenide, where the design parameters are applied strain and external electric field and the target property is the electronic band gap (E_g). Strain can been applied to 2D materials to monotonically tune their band gaps, which can be experimentally accomplished via various methods including bending or stretching the substrate, thermal expansion, or through the application of local stress from an atomic force microscope tip[26–30]. Similarly, an external electric field can also be used to further modulate the band gap due to band-bending, and provides an additional degree of freedom which is not constrained by the elastic strength of the material[31,32]. The ability to tune the band gap freely allows the material to be designed for a target application, including in photocatalysis, electronics, sensors, and neuromorphic devices[26,33–36]. Here, we will represent applied strain as deviations in the equilibrium lattice constants a, b, c, α, β, γ, and an electric field E normal to the monolayer as the seventh dimension in the design space. Our focus is to demonstrate our framework on a model system that is sufficiently complex to compare its merits over existing approaches, yet simple enough to yield useful physical insights for a technologically relevant problem.

RESULTS

Data generation

To generate the training data, we performed approximately 11,000 DFT calculations by sampling over the entire range of the design space (Fig. 2a). In this work, we use a sampling range of 20% above and below the equilibrium for the six lattice parameters. While this may exceed realistic achievable ranges in some cases, we focus first on achieving an accurate mapping of the forward and reverse process from the full design space, and experimental limitations can be applied later at the down-selection stage. In addition to strain, an external electric field is applied in a range from −1 to 1 V/Å. The DFT-calculated band gap at equilibrium is 1.97 eV, and we find most samples lie below this value under the applied strain and electric field (Fig. 2b). Most notably, the vast majority of cases result in a metal-insulator transition (MIT) to a band gap of zero, and this data imbalance can present significant challenges for generative models which we will see shortly. We also illustrate here the full design space in two dimensions using the Uniform Manifold Approximation and Projection (UMAP) method[37] (Fig. 2c), which shows no clear decision boundary between low E_g and high E_g states, especially in the 0–1 eV region where substantial intermixing of states can be observed. Therefore, the generative model used must also have a high degree of fidelity to avoid large errors in the target property of generated candidates.

Benchmarking model performance

To determine whether our approach succeeds for this type of challenging inverse problem, we first train and subsequently validate our model by comparing the band gap of generated samples with the surrogate model. We also compare our models with several other inverse-mapping methods in order to calibrate performance: (1) mixture density methods (MDN)[38] and (2) conditional variational autoencoders (cVAE)[39]. For our models we implemented the base (3) invertible neural networks (INN)[20] and (4) conditional invertible neural networks (cINN)[21] as well as the additional steps included in the MatDesINNe framework as (5) MatDesINNe-INN and (6) MatDesINNe- cINN. The results are compiled in Table 1, where 10,000 samples are obtained with each model and validated against the surrogate to reveal their relative performance. We find all models performed adequately well for the $E_g = 0$ case, an unsurprising result, as statistically most samples will have zero band gaps. However, for non-zero cases such as $E_g = 0.5$ and $E_g = 1.0$ the performance of the baseline models drops dramatically, far too low for use in any materials design situation. With invertible neural networks, cINN maintains an appreciable performance of ~0.2 eV for both non-zero gap cases, though the error still remains higher than ideal. Finally, with the MatDesINNe framework, the error is further reduced to near-zero (with respect to the surrogate model) for all target band gaps using MatDesINNe-cINN.

There are several potential reasons for the exceptional performance of MatDesINNe-cINN over the other models (MDN, cVAE, base INN and cINN). MDN is a classic model which relies on the Gaussian mixture model, which may not fit complex and strongly non-Gaussian problems such as our case here. cVAE is a model which is trained on a forward process utilizing the L2 loss to achieve a good reconstruction of the original input x, and the backward process to solve x which is decided from random samples drawn from latent space z conditioned on y. Here, the inference performance from decoding is limited by its invertible expressivity leading to inverse solutions with poor accuracy. This brings us the INN and cINN models. The cINN transforms x directly to a latent representation z given the observation y, which is done by providing y as an additional input to each affine coupling layer during both the forwards and backwards network passes, which yields a better performance for inference. These base models perform better than MDN and cVAE, though they still do not provide solutions with a sufficiently high accuracy. Consequently, the localization step in our framework plays an important role here to push these posterior samples (from INN and cINN) to the

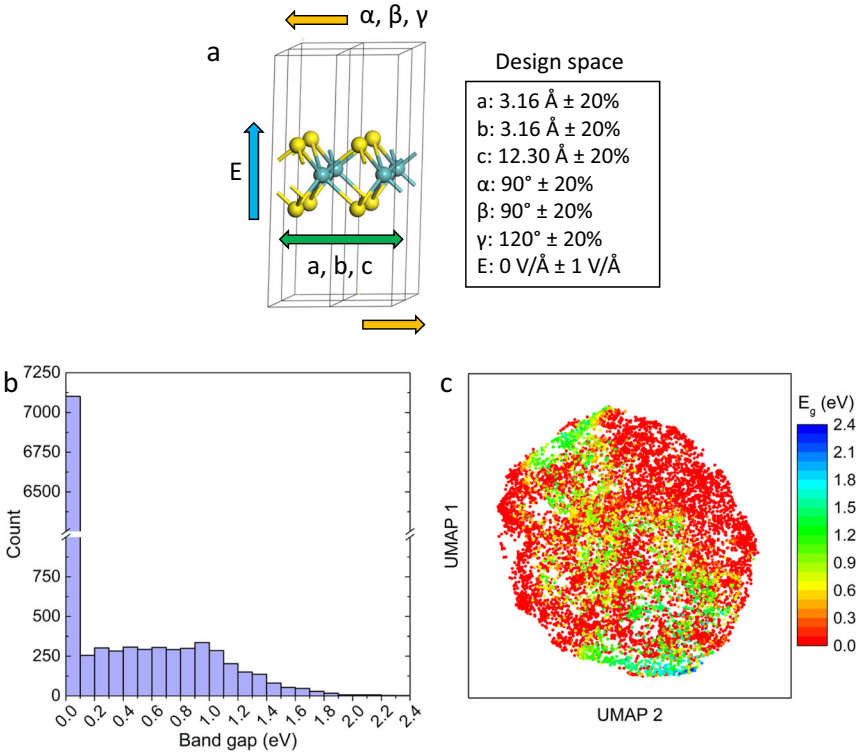

Fig. 2 Materials design space and data distribution. a MoS_2 model and design parameter space specification, which can be categorized as tensile/compressive (green arrow), shear (orange arrow) and electric field (blue arrow). **b** Distribution of DFT-computed band gaps within the sampled design space for monolayer MoS_2. **c** UMAP embedding of sampled design space with DFT-computed band gaps.

Table 1. Surrogate-validated performance and speed of tested models for 10000 samples given three targets: $E_g = 0$, 0.5 and 1 eV.

Method	$E_g = 0$ eV		$E_g = 0.5$ eV		$E_g = 1.0$ eV	
	MAE (eV)	Time (s)	MAE (eV)	Time (s)	MAE (eV)	Time (s)
MDN	0.184	5.255	0.421	5.288	0.840	4.934
cVAE	0.064	5.246	0.461	5.532	0.973	5.711
INN	0.038	5.912	0.527	5.598	0.835	5.891
cINN	0.063	5.833	0.219	6.026	0.193	5.591
MatDesINNe-INN	0.038	28.035	0.512	32.096	0.321	33.598
MatDesINNe-cINN	**0.020**	27.882	**0.013**	30.903	**0.015**	30.953
DFT	N/A	8.43E+05	N/A	9.60E+06	N/A	1.03E+07

optimal solutions via gradient descent. Prior to the localization, we also filtered out the out-of-distribution samples which cannot localize properly. As cINN provides better posterior samples than INN, MatDesINNe-cINN is also able to localize much better than MatDesINNe-INN as seen in Table 1.

In terms of computational cost, all machine learning models required less than a minute to generate 10,000 samples on a single standard CPU. We note this does not include data-generation or training time into the calculation. MatDesINNe requires slightly more time due to the additional localization step but provides substantially improved performance. The speeds provided for the models can be considered to be essentially on-the-fly for the generation, which can be further increased with using GPUs. Regardless of the method, the computation times are negligible compared to the DFT calculations: to generate 10,000 samples with a band gap of 1 ± 0.1 eV with DFT would require $1.03 * 10^7$ seconds or over five orders of magnitudes longer than MatDesINNe due to both the intrinsic DFT evaluation time and the low statistical probability for finding a target band gap. The speedup provided here with inverse learning can be considered as a lower bound for general problems, as more complex properties and higher levels of theory will necessitate much longer DFT evaluation times.

We then validate the best-performing model, MatDesINNe-cINN with DFT calculations to find the absolute performance relative to DFT. The DFT band gaps for each of the three targets are shown in Fig. 3 for 200 samples, showing an excellent performance with a MAE of 0.1 eV or lower. This accuracy is only slightly higher than usual experimental error bars for band gap measurements[40], which is a typical point of comparison when discussing chemical accuracy. While our model is trained on DFT band gaps which contain its own error bars, we expect a model expressive enough to fit well to the DFT energies to do similarly well for another, more accurate, quantum chemistry method. A low incidence of outliers is also observed; only 6 out of 200 samples, or 3%, have non-zero gaps for $E_g = 0$, and less than 2% of samples have zero band gaps for $E_g = 0.5$ and $E_g = 1$. The absolute performance here is limited by the accuracy of the surrogate model used for

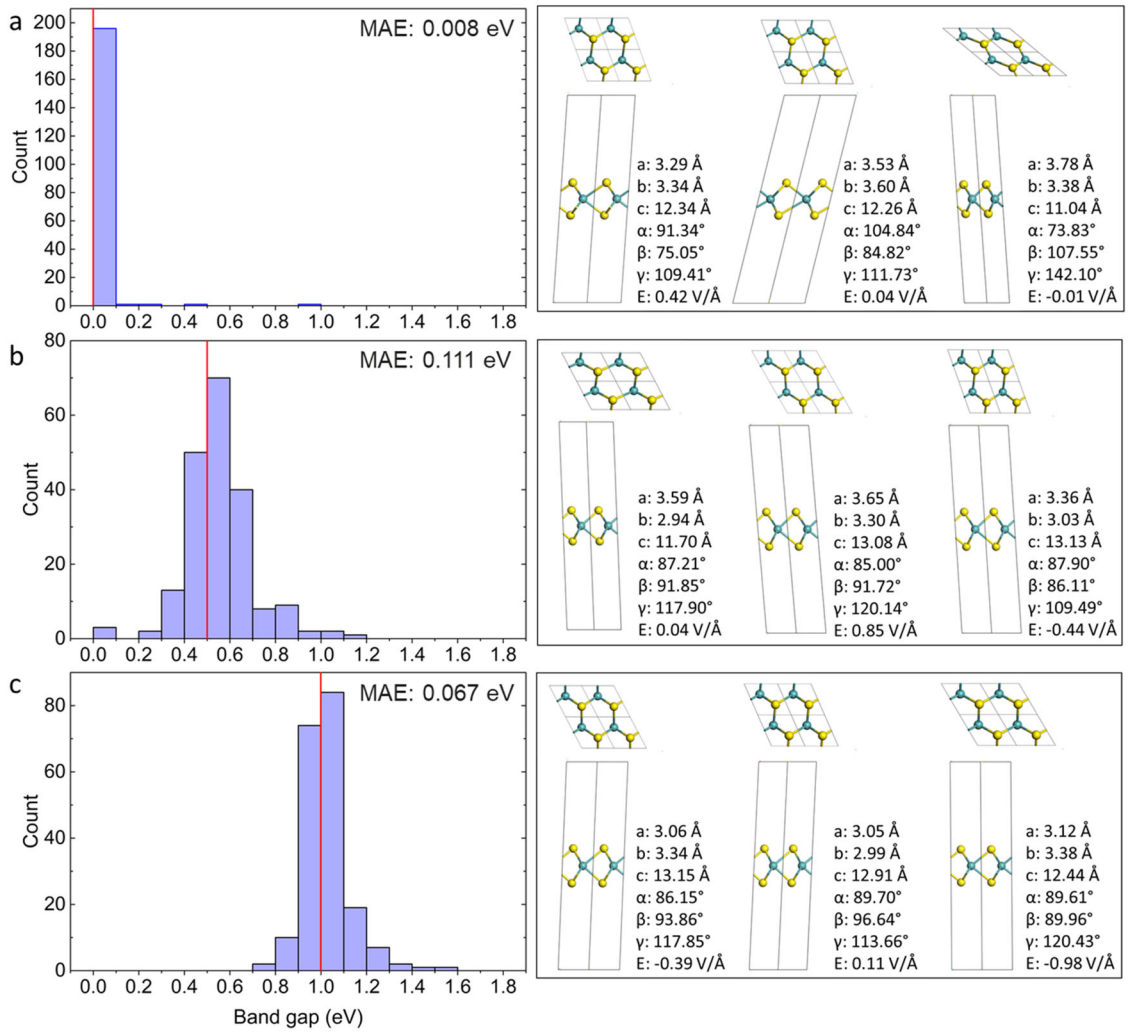

Fig. 3 Validation by DFT. DFT-validated performance of generated candidates for three targets: **a** $E_g = 0$, **b** 0.5 and **c** 1 eV (shown by the position of the red line). Example structures are shown for each target.

gradient-based optimization, hence the discrepancy between DFT-validated performance and earlier surrogate-validated performance in Table 1. By improving the accuracy of the surrogate model and thereby allowing the optimized samples to be closer to the true DFT value, we anticipate the DFT-validated performance can be further improved.

In addition to the high fidelity of the generated candidates with this approach, we find they adequately cover the distribution of the original training data and while maintaining a high degree of diversity in the design parameter space. To demonstrate this, distributions of the training data and generated data are compared side-by-side for $E_g = 0$, 1 in Fig. 4, and plotted with respect to average in-plane (a, b, γ) and out-of-plane (c, α, β) strain for ease of visualization. The distributions for both $E_g = 0$ and $E_g = 1$ are shown to match quite well with the training data and not localized to a specific region in the design space. This example illustrates another strength of this approach compared to methods such as conditional GANs which often struggle with maintaining a high sample diversity[21]. Furthermore, we investigated the uniqueness of the generated samples by comparing their similarity to the original training data. Samples whose parameters which fall outside a tolerance of an average or maximum of 0.01 or 1% of the existing data are considered unique. The results are compiled in Table 2 which shows in general, the majority of generated samples were found to be unique according to this criterion.

Materials generation and applications

With a sufficiently accurate and expressive generative model as demonstrated here, the problem of generating specific design parameters with a target band gap becomes trivial. We next expand upon these capabilities by investigating the overall parameter space as mapped out by the model. For example, we can group the strain parameters into two categories: tensile/compressive (a, b, c) and shear (α, β, γ) and plot their overall distributions in Fig. 5. Here, we find the distribution in average shear strain is fairly normal, while it is significantly more skewed for tensile/compressive strain. It is readily apparent E_g is far more sensitive to shear strain than tensile/compressive strain, where few samples exist for $E_g = 1$ when average shear strain is over 5% in either direction. When viewed with respect to the absolute strain (Fig. 5c, d), zero gap and non-zero gap samples can be better distinguished, with a rough correlation where higher absolute strain leads to greater probability for finding $E_g = 0$ materials, especially for the shear strain case. Meanwhile, $E_g = 0.5$ and 1 share much of the same strain space, with the exception of the tensile strain region (positive values) where $E_g = 1$ drops off quickly but $E_g = 0.5$ persist up to approximately

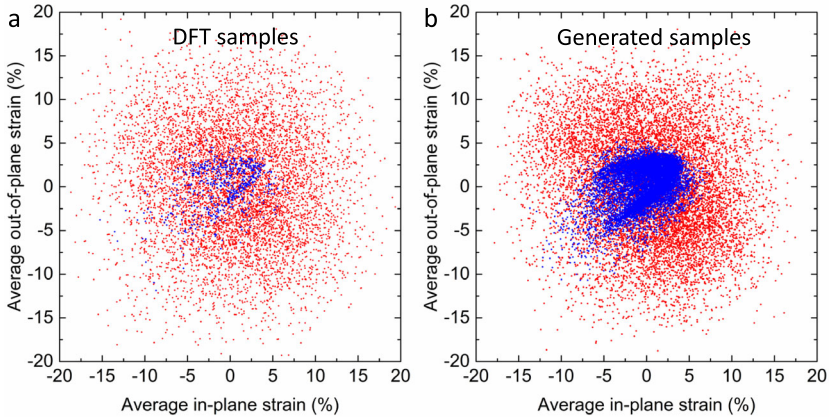

Fig. 4 Generated data distributions compared to DFT. Distributions of average in-plane strain vs. out-of-plane strain for **a** DFT training data and **b** generated data cases. For each case, red denotes $E_g = 0$ eV samples, while blue denotes the $E_g = 1 \pm 0.1$ eV band gap samples.

Table 2. Percentage of unique samples for a given target band gap.

Tolerance (samples outside of range)	$E_g = 0.0$ eV	$E_g = 0.5$ eV	$E_g = 1.0$ eV
average > 0.01	99.96%	98.61%	90.82%
max > 0.01	99.98%	98.34%	85.00%

17% in tensile strain. These analyses can help provide design principles and provide guidance into regions of the strain space for further sampling.

For a more targeted application, we propose to use the generated samples to probe the metal-insulator transition of MoS_2, a key property in neuromorphic devices. One useful insight gained by generative modeling is revealing regions in the design parameter space where MIT occurs with minor perturbations[41,42]. To illustrate this, we use UMAP to reduce the 7-dimensional space into two dimensions for two situations, the MIT between $E_g = 0$ and 0.5 (Fig. 6a) and the MIT between $E_g = 0$ and 1 (Fig. 6b). In terms of global structure, we see two relatively distinct regions in the reduced dimensional space for both cases, though the $E_g = 0$ and 1 case shows a much clearer decision boundary. This is an expected behavior consistent with the observation suggested previously in Fig. 5 in which a high E_g correlates with low shear strain and vice versa with little overlap. Meanwhile, locally, there are many regions in the reduced dimensional space where zero and nonzero band gaps coexist, even for the case of $E_g = 0$ and 1. As UMAP tends to locally group entities which are similar in the full input dimension, an overlap in points with zero and nonzero gaps suggest a transition between these two states only requires a minor change in the applied strain with two examples shown in the zoomed-in insets for Fig. 6. In the example for the $E_g = 0$ and 0.5 case, MIT occurs from a 7% tensile strain in the y-axis and a 0.82 V/Å change in electric field. For the $E_g = 0$ and 1 case, a 6% tensile strain in the y-axis and 7% compressive strain in the z-axis is needed. This can highlight potentially useful regions in strain space where MIT can be easily induced, allowing for fast and energy-efficient switching. Alternatively, if the goal is to prevent the occurrence of MIT, it is then desirable to select regions in the strain space where no $E_g = 0$ cases can be found in the vicinity.

DISCUSSION

INNs exhibit the attractive property of intrinsic invertibility and serves as a promising approach for solving inverse design problems. In our materials design framework MatDesINNe, we train INNs and use then to generate samples given a target property. To ensure samples have target properties within chemical accuracy, we further incorporate down-selection and localization via optimization on the generated samples. We apply the framework to engineering band gaps in monolayer MoS_2 via applied strain and electric field. We find our approach outperforms other baseline methods and succeeds at the conditional generative task with a target property MAE of around 0.1 eV or lower. This approach is then used to provide high-fidelity and diverse candidates at several orders of magnitude lower computational cost than direct screening with DFT. Using this model, we then generate large quantities of candidate materials to explore the design space and obtain useful design principles or insights for further sampling. Finally, we show how our model can help tackle complex problems such as the MIT in MoS_2 by densely populating the design space for two target band gap states and identifying potential regions where fast switching of states can be achieved. Moreover, our approach is materials and applications agnostic and can be applied for general materials design tasks for arbitrary structures and compositions, provided well-defined design parameters and target properties(s) are chosen and available for training. We have made this code available in an open-source repository for this purpose.

In the current work, we focused on inverse learning models and investigated their ability to deal with imbalanced datasets and accurately map complex underlying physics for materials discovery. We showed this is a challenging problem even for a fairly small parameter space, where current methods such as MDN, VAEs and base INNs will fail. An alternative existing approach for this problem is by modeling only the forward process with machine learning and using high-throughput screening to find the candidates with random or grid-based sampling. Using a deep neural network with two hidden layers, we obtained a forward prediction MAE of 0.084 eV on the same dataset which is on par with the performance our inverse learning model. However, the forward modeling approach quickly becomes intractable for high-dimensional materials design spaces, a problem which can be tackled by inverse learning models like INNs, as previously shown for image generation[21]. One potential concern is that the affine coupling layers used in this work may have limitations in its expressivity for capturing complex distributions[43]. In the future, other invertible architectures may be considered, such as neural spline flows, which have better expressivity[44].

We have also not incorporated additional atomic or compositional dimensions as design parameters, which have been included in recent studies using variational autoencoders (VAEs) and generative adversarial networks (GANs)[45–48]. One challenge here is that generative methods involving atomic structure remain hampered by a lack of a suitable invertible crystal representation

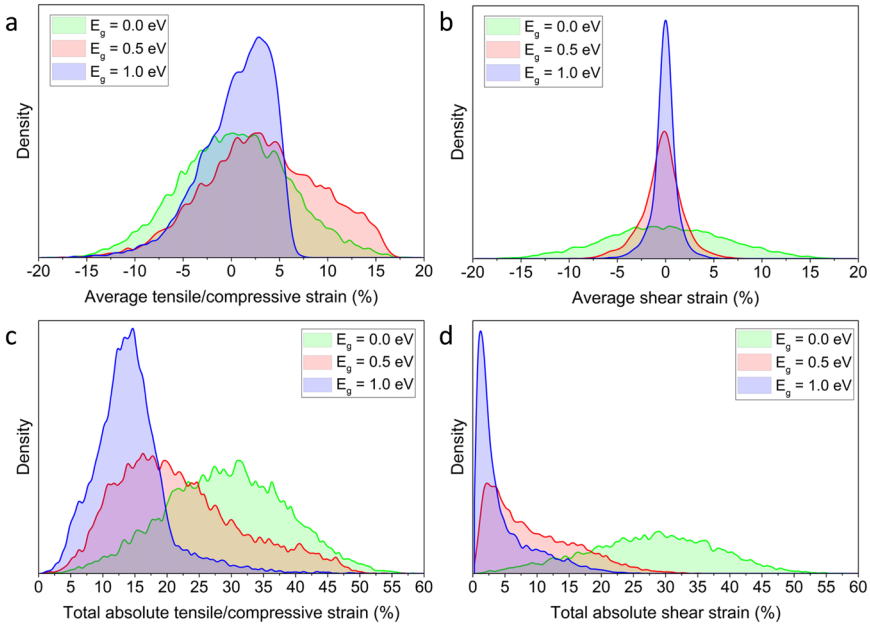

Fig. 5 Generated data distributions by tensile/compressive and shear strain. Distributions of generated data by **a** average tensile/compressive strain, **b** average shear strain, **c** average absolute tensile/compressive strain and **d** average absolute shear strain for three targets: $E_g = 0$, 0.5 and 1 eV.

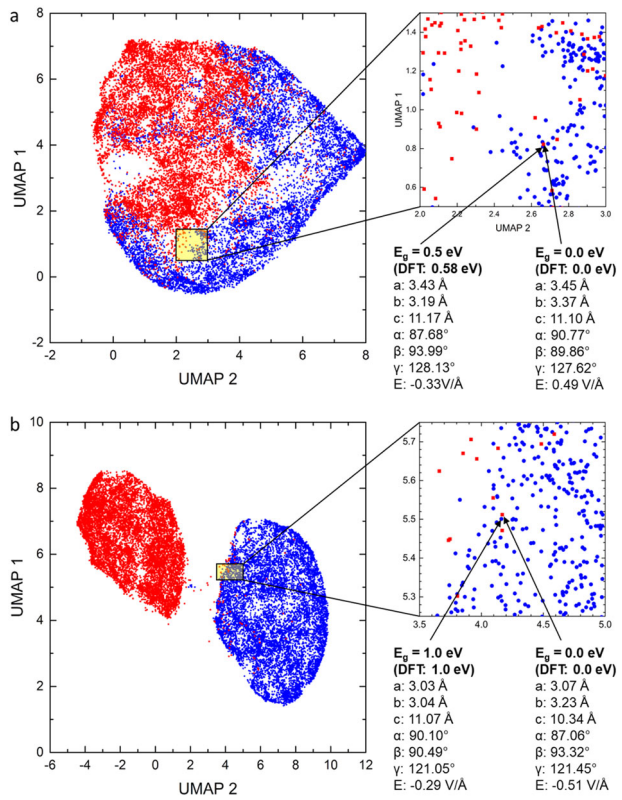

Fig. 6 Metal-insulator transition in embedding space. UMAP embedding of strain parameters for **a** $E_g = 0.5$ and **b** $E_g = 1$ eV with $E_g = 0$ eV samples. Red denotes $E_g = 0$ eV samples, while blue denotes the nonzero band gap samples. Selected regions (in yellow) are zoomed in to highlight specific examples where MIT occurs. Samples were validated with DFT and the true band gaps are shown in parentheses.

which is ideally rotationally and translationally invariant[11]. We leave the problem of a generative model for atomic structures for the future when such a representation is developed.

We anticipate a materials design framework such as MatDesINNe will provide both theoretical insights as shown in this work as well as offer a means of integrating computation with experiments. The traditional approach of linear discovery via model, make, and measure remains severely limited by the large number of variables, complex coupled/competing underlying phenomena, and slow process times. Methods which can navigate the design space in a rational fashion to select experiments, as well as learn the outcome of these experiments can therefore greatly improve efficiency and introduce autonomous guidance. By generating high fidelity and diverse samples on-the-fly with chemical accuracy, MatDesINNe can satisfy many of these requirements and be incorporated into the autonomous experimentation process.

METHODS

Density functional theory

Lattice constants and angles a, b, c, α, β, γ were sampled within a range of 20% deviation from the equilibrium crystal parameters: a = b = 3.16 Å, c = 12.30 Å, α = β = 90°, γ = 120°. An external electric field is also applied in the z-direction from −1 to 1 V/Å. A total of 10799 structures were generated and band gaps obtained using density functional theory (DFT). The DFT calculations were performed with the Vienna Ab Initio Simulation Package (VASP)[49,50]. The Perdew-Burke-Ernzerhof (PBE)[51] functional within the generalized-gradient approximation (GGA) was used for electron exchange and correlation energies. The projector-augmented wave method was used to describe the electron-core interaction[49,52]. A kinetic energy cutoff of 500 eV was used. All calculations were performed with spin polarization. The Brillouin zone was sampled using a Monkhorst-Pack scheme with a 9x9x1 grid[53]. We note that PBE, and more generally GGA functionals, often struggle with calculating accurate band gaps, especially for strongly correlated systems. In the present case, our computed PBE gap of an unperturbed monolayer MoS2 (1.97 eV) was found to be close to the experimental gap (1.90 eV)[54] and should serve an adequate method for demonstration purposes.

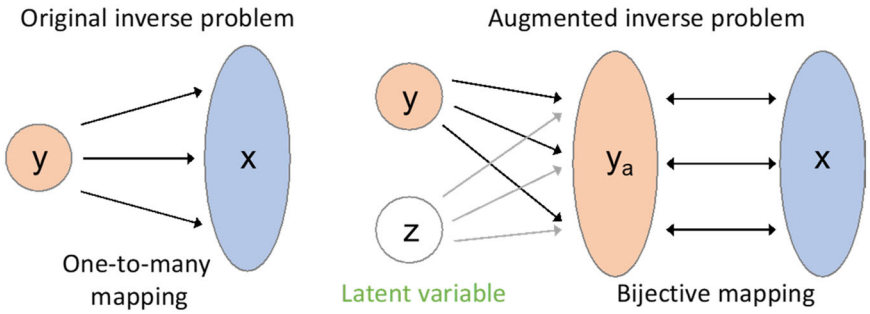

Fig. 7 Inverse problem specification. The original inverse problem is often ill-posed due to the one-to-many mapping. An augmented inverse problem is formulated based on bijective mapping with introducing additional latent random variable *z*.

Invertible neural network model

Inverse problem specification. Typically, a mathematical or physical model is developed to describe how measured observations $y \in \mathbb{R}^M$ arise from the hidden parameters $x \in \mathbb{R}^D$ to yield such a mapping $y \in \Omega(x)$. To completely capture all possible inverse solutions given observed measurements, a proper inverse model should enable the estimation of the full posterior distribution $p(x|y)$ of hidden parameters x conditioned on an observation y.

Invertible neural networks. A recent study[21] showed that the invertible neural networks (INNs) can first be trained in the forward pass and then used in the reverse mode to sample from $p(x|y)$ for any specific y. This is achieved by adding a latent variable $z \in \mathbb{R}^K, K = D - M$, which encodes the inherent information loss in the forward process. In other words, the latent variable z drawn from a Gaussian distribution $p(z) = N(0, I_K)$ is able to encode the intrinsic information about x that is not contained in y. To this end, an augmented inverse problem is formulated based on such a *bijective* mapping, as shown in Fig. 7:

$$x = h(y_a; \phi) = h(y, z; \phi), z \sim p(z) \tag{1}$$

where h is a deterministic function of y and z, parametrized on an INN with parameter Φ. Forward training optimizers the mapping $x \to y_a = [y, z]$ and implicitly determines the inverse mapping $x = h(y, z)$. In the context of INNs, the posterior distribution $p(x|y)$ is represented by the deterministic function $x = h(y, z)$ that transforms the known probability distribution $p(z)$ to parameter x-space, conditional on measurements y. Thus, given a chosen observation y^* with the learned h, we can obtain the posterior samples x_k which follows the posterior distribution $p(x|y^*)$ via a transformation $x_k = h(y^*, z_k)$ with prior samples drawn from $z_k \sim p(z)$.

The invertible architecture allows us to simultaneously learn the model $h(y, z; \phi)$ of the inverse process jointly with a model $f(x; \phi)$ which approximates the true forward process $\Omega(\mathrm{x})$:

$$[y, z] = f(x; \phi) = [f_x(x; \phi), f_z(x; \phi)] = h^{-1}(x; \phi) \tag{2}$$

Where $f_y(x; \phi) \approx \Omega(\mathrm{x})$, model f and h share the same parameters Φ in a single invertible neural network. Therefore, our approximated posterior model $\hat{p}(x|y)$ is built into the invertible neural network representation as

$$\hat{p}(x = h(y, z; \phi)|y) = \frac{p(z)}{|J_x|}, \quad J_x = \det\left(\frac{\partial h(y, z; \phi)}{\partial [y, z]}\right) \tag{3}$$

Where J_x is the Jacobian determinant that can be efficiently computed by using affine coupling blocks[55].

Invertible neural network training. In this work use an invertible architecture with affine coupling layers[20]. A forward L2 loss is defined where y_t is the true output:

$$L_y = \|y - y_t\|_2^2 \tag{4}$$

Then a backward loss $L_z = \mathrm{MMD}(z)$ is used to fit the probability distribution of latent variable $p(z)$ to a standard Gaussian distribution. MMD refers to the Maximum Mean Discrepancy[56]. The total loss is then defined with weighting factors λ_y and λ_z:

$$L(y, z) = \lambda_y L_y + \lambda_z L_z \tag{5}$$

It is also possible to train the invertible neural network with a maximum likelihood loss by integrating the forward L2 loss and unsupervised backward loss[20]:

$$L(y, z) = \frac{1}{2}\left(\frac{1}{\sigma^2}(y - y_t)^2 + z^2\right) - \log\left|\det J_{x \mapsto [y,z]}\right| \tag{6}$$

Conditional invertible neural networks (cINN). Instead of training an invertible neural network to predict y and x with additional latent variable z, the conditional invertible neural network[21] transforms x directly to a latent representation z conditional on the observation y. This is achieved by using y as an additional input to each affine coupling layer in both forward and backward processes. The model is then trained with a maximum likelihood loss:

$$L(z) = \frac{1}{2}z^2 - \log\left|\det J_{x \mapsto z}\right| \tag{7}$$

Localization from posterior samples

After finishing the invertible neural network training, a set of posterior samples can be drawn from the approximate posterior distribution $\hat{p}(x|y)$. Compared with the prior distribution $p(x)$, which is typical a uniform random guess, these posterior samples drawn from $\hat{p}(x|y)$ significantly narrow the search space from a global search to a local search. The small gap between the approximate posterior and true posterior is due to the residual training loss but can be bridged by gradient descent with few steps. Conventional global search methods via optimization are very computationally intensive when the design space is high. However, the invertible model helps us narrow the search through generative posterior samples, which can be interpreted as intelligent prior information. Incorporating a local search via gradient descent, we then obtain more accurate inverse solutions. This procedure typically consists of three steps:

Step 1: Prior exploration. Given a specific target property $\hat{y}$, we first draw samples from the latent space $z_i \sim p(z)_{i=1}^m$ and then transform these samples to the design space $\hat{x}_i \sim \hat{p}(x|\hat{y})_{i=1}^m$ via an invertible bijective mapping. These posterior samples serving as good initialization shorten the distance to the exact inverse solution.

Step 2: Gradient estimation. We save the surrogate model with minimized L2 loss in INN training and evaluate the learned model by changing the input x to the network. The gradient at the current $\hat{x}_i$ can be efficiently computed by automatic differentiation:

$$g_i = \left.\frac{\partial L\left(\hat{f}_y(\hat{x}_i; \phi^*), \hat{y}\right)}{\partial x}\right|_{x=\hat{x}_i} \quad \hat{x}_i \sim \hat{p}(x|\hat{y}), i = 1, \ldots, m \tag{8}$$

Step 3: Localization via gradient decent. We precisely localize the posterior samples drawn from $\hat{p}(x|\hat{y})$ to exact inverse solutions via gradient descent $\hat{x}_{i+1} = \hat{x}_i - \gamma g_i$, where γ is the learning rate. We use the Adam optimizer to adaptively update the solution. This local search with intelligent priors is much more efficient compared with a generic random search in the entire design space.

MatDesINNe framework

The MatDesINNE framework consists of (1) training, (2) inference, (3) down-selection, and (4) localization steps. In step 1, for the INN, weight coefficients for each loss are initialized and the total loss is minimized via bi-directional training with stochastic gradient descent. The forward model with minimal l2 loss is saved as a forward surrogate. For the cINN, the loss is minimized via maximum likelihood training with stochastic gradient descent. In step 2, random samples are generated from the latent space $p(z)$, and the corresponding posterior samples conditioned on the prior sample z and a given observation y through an invertible transformation are computed. In step 3, down-selection is performed by removing far outlier samples based on the surrogate predictions, as well as samples with parameters outside the training range. In the case of cINN, the majority of samples still remain after down-selection, while the ones which are removed are unlikely to localize to good solutions and. In step 4, localization is performed on the remaining samples by computing the gradient at the current x using the saved forward surrogate with automatic differentiation and optimized via gradient decent.

Model implementation

Our implementation is based on PyTorch and FrEIA (https://github.com/VLL-HD/FrEIA) which is used for building the invertible neural network blocks. For INN and cINN model, we use 6 invertible blocks, and 2 fully connected (fc) layers for each block. There are 256 neurons in each fc layer and the ReLU activation is used in each fc. We use Adam algorithm as the optimizer for invertible training and localization training with a learning rate 1E-3 and a weight decay rate 1E-5. The with weighting factors λ_y and λ_z in INN total loss are equal to 1. To stabilize the invertible training, we add small perturbations with Gaussian noise where σ_y is 5E-3 and σ_z is 2E-3. We use a total number of 11000 DFT samples where 80% DFT data are used for training and 20% DFT data are used for testing. The batch size for invertible training is 500 and the number of training epochs is 1000. The experiments are performed on Linux workstation (Ubuntu 18.04) with 16 Xeon CPUs and a NVIDIA Quadro RTX 6000 GPU. The average training time per model is about 10–15 minutes using a single GPU.

Baseline methods

Conditional variational autoencoders (cVAE). The conditional variational autoencoder[39] uses the evidence lower bound and encodes the x into a Gaussian distributed random latent variable z conditioned on y. The forward training process utilizes the L2 loss to achieve a good reconstruction of the original input x while the backward process solves x which is decoded from random samples that are drawn from z conditioned on y. The loss function is defined as:

$$L = \alpha(x - \hat{x})^2 - \frac{1}{2}\beta\left(1 + \log\sigma_z - \mu_z^2 - \sigma_z\right) \tag{9}$$

Our implementation of cVAE is based on the literature[57]. The encoder and decoder models are constructed by three layers with 128 hidden nodes for each layer. The latent dimension is 3. The optimizer is Adam with a learning rate 1E-3 and batch size 500. The entire training is done by 1000 epochs.

Mixture density networks (MDN). Mixture density networks[38] directly model the inverse problem, with $p(x|y)$ as the input and predicts the parameters μ_x, Σ_x^{-1} of a Gaussian mixture model $p(x|y)$ as output. The model was and trained by maximizing the likelihood of the training data with the loss function:

$$L = \frac{1}{2}\left(x\mu_x^T \cdot \Sigma_x^{-1} \cdot x\mu_x\right) - \log\left|\Sigma_x^{-1}\right|^{1/2} \tag{10}$$

The mixture density network consists of categorical network and mixture diagonal normal network. Both network models are constructed by three-layer neural networks. The hidden dimension is equal to the input dimension. For this case, we use 3 components for MDN model. Similarly, we use Adam optimizer with a learning rate 1E-3. The batch size is 500 and the number of training epochs is 1000.

DATA AVAILABILITY

The DFT dataset used for training is provided at https://github.com/jxzhangjhu/MatDesINNe

CODE AVAILABILITY

The machine learning code used in this work is available at https://github.com/jxzhangjhu/MatDesINNe

Received: 9 July 2021; Accepted: 6 November 2021;
Published online: 09 December 2021

REFERENCES

1. Mannodi-Kanakkithodi, A. & Chan, M. K. Y. Computational data-driven materials discovery. *Trends Chem.* **3**, 79–82 (2021).
2. Alberi, K. et al. The 2019 materials by design roadmap. *J. Phys. D: Appl. Phys.* **52**, 013001 (2018).
3. de Pablo, J. J. et al. New frontiers for the materials genome initiative. *npj Comput. Mater.* **5**, 41 (2019).
4. Pollice, R. et al. Data-driven strategies for accelerated materials design. *Acc. Chem. Res.* **54**, 849–860 (2021).
5. Chen, C. et al. A critical review of machine learning of energy materials. *Adv. Energy Mater.* **10**, 1903242 (2020).
6. Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O. & Walsh, A. Machine learning for molecular and materials science. *Nature* **559**, 547–555 (2018).
7. Zhong, M. et al. Accelerated discovery of CO_2 electrocatalysts using active machine learning. *Nature* **581**, 178–183 (2020).
8. Batra, R., Song, L. & Ramprasad, R. Emerging materials intelligence ecosystems propelled by machine learning. *Nat. Rev. Mater.* **6**, 655–678 (2021).
9. Jiang, J., Chen, M. & Fan, J. A. Deep neural networks for the evaluation and design of photonic devices. *Nat. Rev. Mater.* **6**, 679–700 (2021).
10. Gómez-Bombarelli, R. et al. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent. Sci.* **4**, 268–276 (2018).
11. Noh, J., Gu, G. H., Kim, S. & Jung, Y. Machine-enabled inverse design of inorganic solid materials: promises and challenges. *Chem. Sci.* **11**, 4871–4881 (2020).
12. Batra, R. et al. Polymers for extreme conditions designed using syntax-directed variational autoencoders. *Chem. Mater.* **32**, 10489–10500 (2020).
13. Yao, Z. et al. Inverse design of nanoporous crystalline reticular materials with deep generative models. *Nat. Mach. Intell.* **3**, 76–86 (2021).
14. Kim, B., Lee, S. & Kim, J. Inverse design of porous materials using artificial neural networks. *Sci. Adv.* **6**, eaax9324 (2020).
15. Zunger, A. Inverse design in search of materials with target functionalities. *Nat. Rev. Chem.* **2**, 0121 (2018).
16. Sanchez-Lengeling, B. & Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **361**, 360–365 (2018).
17. Terayama, K., Sumita, M., Tamura, R. & Tsuda, K. Black-box optimization for automated discovery. *Acc. Chem. Res.* **54**, 1334–1346 (2021).
18. Willcox, K. E., Ghattas, O. & Heimbach, P. The imperative of physics-based modeling and inverse theory in computational science. *Nat. Comput. Sci.* **1**, 166–168 (2021).
19. Kingma, D. P. & Welling, M. Auto-encoding variational Bayes. Preprint at http://arxiv.org/abs/1312.6114 (2014).
20. Ardizzone, L., Kruse, J., Rother, C. & Köthe, U. Analyzing Inverse Problems with Invertible Neural Networks. *ICML* (2018).
21. Ardizzone, L., Lüth, C., Kruse, J., Rother, C. & Köthe, U. Guided image generation with conditional invertible neural networks. Preprint at https://arxiv.org/abs/1907.02392 (2019).
22. Kruse, J., Ardizzone, L., Rother, C. & Köthe, U. Benchmarking invertible architectures on inverse problems. Preprint at https://arxiv.org/abs/2101.10763 (2019).
23. Asim, M., Daniels, M., Leong O., Ahmed, A. & Hand, P. Invertible generative models for inverse problems: mitigating representation error and dataset bias. *ICML*. **119**, 399–409 (2020).
24. Goodfellow, I. J. et al. Generative adversarial nets. *NeurIPS*. **27**, (2014).
25. Baydin, A. G., Pearlmutter, B. A., Radul, A. A. & Siskind, J. M. Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.* **18**, 5595–5637 (2018).
26. Chaves, A. et al. Bandgap engineering of two-dimensional semiconductor materials. *npj 2D Mater. Appl.* **4**, 29 (2020).
27. López-Suárez, M., Neri, I. & Rurali, R. Band gap engineering of MoS_2 upon compression. *J. Appl. Phys.* **119**, 165105 (2016).
28. Castellanos-Gomez, A. et al. Elastic properties of freely suspended MoS_2 nanosheets. *Adv. Mater.* **24**, 772–775 (2012).
29. Dai, Z., Liu, L. & Zhang, Z. Strain engineering of 2D materials: issues and opportunities at the interface. *Adv. Mater.* **31**, 1805417 (2019).
30. Peng, Z., Chen, X., Fan, Y., Srolovitz, D. J. & Lei, D. Strain engineering of 2D semiconductors and graphene: from strain fields to band-structure tuning and photonic applications. *Light Sci. Appl.* **9**, 190 (2020).

31. Ramasubramaniam, A., Naveh, D. & Towe, E. Tunable band gaps in bilayer transition-metal dichalcogenides. *Phys. Rev. B* **84**, 205325 (2011).
32. Ryou, J., Kim, Y.-S., Kc, S. & Cho, K. Monolayer MoS_2 bandgap modulation by dielectric environments and tunable bandgap transistors. *Sci. Rep.* **6**, 29184 (2016).
33. Shao, Z., Cao, X., Luo, H. & Jin, P. Recent progress in the phase-transition mechanism and modulation of vanadium dioxide materials. *NPG Asia Mater.* **10**, 581–605 (2018).
34. Zhu, X., Li, D., Liang, X. & Lu, W. D. Ionic modulation and ionic coupling effects in MoS_2 devices for neuromorphic computing. *Nat. Mater.* **18**, 141–148 (2019).
35. Feng, J., Qian, X., Huang, C.-W. & Li, J. Strain-engineered artificial atom as a broad-spectrum solar energy funnel. *Nat. Photonics* **6**, 866–872 (2012).
36. Bao, X. et al. Band structure engineering in 2D materials for optoelectronic applications. *Adv. Mater. Technol.* **3**, 1800072 (2018).
37. McInnes, L., Healy, J. & Melville, J. Umap: uniform manifold approximation and projection for dimension reduction. Preprint at https://arxiv.org/abs/1802.03426 (2018).
38. Bishop, C. M. *Mixture density networks*. (Aston University, 1994).
39. Sohn, K., Lee, H. & Yan, X. Learning structured output representation using deep conditional generative models. *NeurIPS* **28**, 3483–3491 (2015).
40. Zanatta, A. R. Revisiting the optical bandgap of semiconductors and the proposal of a unified methodology to its determination. *Sci. Rep.* **9**, 11225 (2019).
41. Ganesh, P. et al. Doping a bad metal: Origin of suppression of the metal-insulator transition in nonstoichiometric VO_2. *Phys. Rev. B* **101**, 155129 (2020).
42. Lu, Q. et al. Metal-insulator transition tuned by oxygen vacancy migration across TiO_2/VO_2 interface. *Sci. Rep.* **10**, 18554 (2020).
43. Kobyzev I., Prince S., Brubaker M. Normalizing flows: an introduction and review of current methods. *IEEE PAMI* 1-1 (2020).
44. Durkan, C., Bekasov, A., Murray, I. & Papamakarios, G. Neural spline flows. *NeurIPS* **32**, 7511–7522 (2019).
45. Noh, J. et al. Inverse design of solid-state materials via a continuous representation. *Matter* **1**, 1370–1384 (2019).
46. Kim, S., Noh, J., Gu, G. H., Aspuru-Guzik, A. & Jung, Y. Generative adversarial networks for crystal structure prediction. *ACS Cent. Sci.* **6**, 1412–1420 (2020).
47. Dong, Y. et al. Inverse design of two-dimensional graphene/h-BN hybrids by a regressional and conditional GAN. *Carbon* **169**, 9–16 (2020).
48. Long, T. et al. Constrained crystals deep convolutional generative adversarial network for the inverse design of crystal structures. *npj Comput. Mater.* **7**, 66 (2021).
49. Kresse, G. & Furthmuller, J. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Comput. Mater. Sci.* **6**, 15–50 (1996).
50. Kresse, G. & Furthmuller, J. Efficient iterative schemes for ab Initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B* **54**, 11169–11186 (1996).
51. Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized gradient approximation made simple. *Phys. Rev. Lett.* **77**, 3865–3868 (1996).
52. Blöchl, P. E. Projector augmented-wave method. *Phys. Rev. B* **50**, 17953–17979 (1994).
53. Monkhorst, H. J. & Pack, J. D. Special points for Brillouin-zone integrations. *Phys. Rev. B* **13**, 5188–5192 (1976).
54. Mak, K. F., Lee, C., Hone, J., Shan, J. & Heinz, T. F. Atomically thin MoS_2: a new direct-gap semiconductor. *Phys. Rev. Lett.* **105**, 136805 (2010).
55. Dinh L., Sohl-Dickstein J., Bengio S. Density estimation using real nvp. Preprint at https://arxiv.org/abs/1605.08803 (2016).
56. Dziugaite G. K., Roy D. M., Ghahramani Z. Training generative neural networks via maximum mean discrepancy optimization. *UAI*, 258–267 (2015).
57. Ma, W., Cheng, F., Xu, Y., Wen, Q. & Liu, Y. Probabilistic rrepresentation and inverse design of metamaterials based on a deep generative model with semi-supervised learning strategy. *Adv. Mater.* **31**, 1901111 (2019).

ACKNOWLEDGEMENTS

This work was performed at the Center for Nanophase Materials Sciences, which is a US Department of Energy Office of Science User Facility. Support was provided by the Center for Understanding and Control of Acid Gas-Induced Evolution of Materials for Energy (UNCAGE-ME), an Energy Frontier Research Center funded by U.S. Department of Energy, Office of Science, Basic Energy Sciences. VF was also supported by a Eugene P. Wigner Fellowship at Oak Ridge National Laboratory. JZ was supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics Program; and by the Artificial Intelligence Initiative at the Oak Ridge National Laboratory (ORNL). ORNL is operated by UT-Battelle, LLC., for the U.S. Department of Energy under Contract DEAC05-00OR22725. This research used resources of the National Energy Research Scientific Computing Center, supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

AUTHOR CONTRIBUTIONS

V.F. conceived the project, V.F. and J.Z., performed the calculations and analysis, and all authors contributed to discussion of results and the writing of the manuscript.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Correspondence and requests for materials should be addressed to Victor Fung or Jiaxin Zhang.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.