AMDEN: Amorphous Materials DEnoising Network

Jonas A. Finkler* Aalborg University jaf@bio.aau.dk

Yan Lin* Aalborg University lyan@cs.aau.dk

Tao Du The Hong Kong Polytechnic University tao.du@polyu.edu.hk

Jilin Hu East China Normal University hujilin@cs.aau.dk

Morten M. Smedskjær Aalborg University mos@bio.aau.dk

Abstract

Disordered (amorphous) materials, such as glasses, are emerging as promising candidates for applications within energy storage, nonlinear optics, and catalysis. Inverse design aims to directly predict the composition and structure of new materials with targeted properties using machine learning models. This avoids the time-consuming trial-and-error process of traditional materials design and has the potential to significantly accelerate the discovery of new materials.

In this work, we introduce AMDEN (Amorphous Material DEnoising Network), a diffusion model-based framework that generates structures of amorphous materials and can be conditioned on target properties. We demonstrate inherent challenges for diffusion models to generate relaxed structures. These low-energy configurations are typically obtained through a thermal motion-driven random search-like process that cannot be replicated by standard denoising procedures. We therefore introduce an energy-based AMDEN variant that implements Hamiltonian Monte Carlo refinement for generating these relaxed structures. We further introduce several amorphous material datasets with diverse properties and compositions to evaluate our framework and support future development.

1 Introduction

Amorphous materials are solids that lack a periodic atomic arrangement (i.e., long-range atomic order), yet exhibit complex short- and medium-range order. They have shown great potential in diverse domains including batteries, non-linear optics, and catalysis [1].

Traditional materials design relies on a trial-and-error approach, where candidate materials are synthesized or simulated to determine their properties. Inverse design aims to reverse this process using machine learning methods by directly predicting the composition and structure of new materials with targeted properties. This has the potential to significantly accelerate the materials discovery process [2]. One promising way to implement this approach is through probabilistic generative models [3, 4], in particular diffusion models [5], which generate atomic positions and elements conditioned on desired properties by transforming random noise to target material samples through a multi-step Markov process. Such models have shown success in generating crystalline materials [6, 7] and molecules [8–10], but remain under-developed for amorphous materials due to the lack of large-scale datasets and their unique atomic ordering characteristics.

There are a few existing efforts in generating atomic configurations of amorphous materials, based on variational autoencoders (VAEs) [3], generative adversarial networks (GANs) [4], or diffusion models.

*These authors contributed equally to this work

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI4Mat.

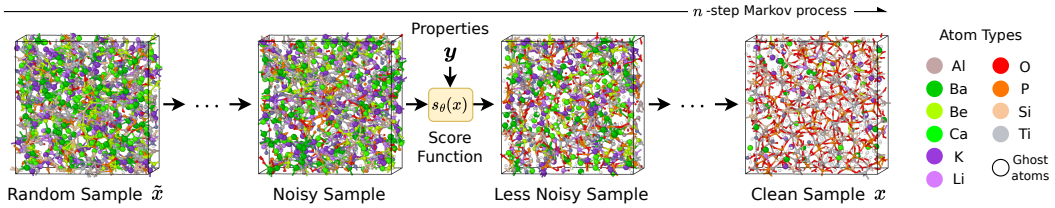

Figure 1: Pipeline of AMDEN model. A reverse-time stochastic differential equation is solved to transform an initially completely random sample into a valid materials sample.

However, the effectiveness of existing VAE-based methods [11–14] and GAN-based methods [15] is challenged by the limitations of the underlying VAE and GAN frameworks [16–20]. Meanwhile, existing diffusion model-based methods [21–23] have focused on relatively narrow types of amorphous materials and properties, leaving the generation of diverse amorphous materials largely unexplored.

In this work, we propose and validate an inverse design framework for amorphous materials. Our framework, named **AMDEN** (Amorphous Material DEnoising Network), is a diffusion model-based framework that generates structures of multi-element amorphous materials with desired properties. Figure 1 illustrates the pipeline of AMDEN. To effectively train and validate AMDEN and to support future development in inverse design of amorphous materials, we generated several amorphous material datasets with diverse properties and compositions. We first introduce three datasets of pure amorphous silicon with differing thermal histories to demonstrate the inherent challenges for diffusion models to generate relaxed structures. We further developed a multi-element glass dataset covering a wide range of compositions to test AMDEN's inverse design capabilities.

The standard implementation of AMDEN is not able to generate low-energy structures, which are obtained when the material is quenched at a low cooling rate. We therefore developed an energy-based variant of the score function, which incorporates Hamiltonian Monte Carlo refinement into the material diffusion process. This modified AMDEN implementation is able to generate samples that match the reference data closely in terms on energy and structure.

2 AMDEN Model

AMDEN generates amorphous materials samples $x = (\boldsymbol{C}, \boldsymbol{X}, \boldsymbol{E})$, consisting of cell vectors $\boldsymbol{C} \in \mathbb{R}^{3\times3}$, atomic positions $\boldsymbol{X} \in \mathbb{R}^{n\times3}$, and one-hot element embeddings $\boldsymbol{E} \in \mathbb{R}^{n\times d}$. The generation process can be conditioned on target properties $\boldsymbol{y}$, which are represented as an m-dimensional vector. We employ a stochastic differential equation (SDE) framework [24] with score-matching [25], where a learnable score function $s_\theta(x) \approx \nabla_x \ln p(x)$ guides the reverse diffusion from noise to valid samples following the target distribution $p(x)$. The stochastic nature provides flexibility for exploring optimal structures. To sample from $p(x)$, we start from a noisy sample $\tilde{x} = (\boldsymbol{C}, \tilde{\boldsymbol{X}}, \tilde{\boldsymbol{E}})$ with random positions and elements, then solve a reverse-time SDE through an n-step Markov process guided by $s_\theta(x)$. Note that the cell $\boldsymbol{C}$ remains intact throughout the process. We introduce *ghost atoms* that enable density control without changing the number of atoms in each sample (see Appendix C.4).

The score function uses an Equivariant Graph Neural Network (EGNN) [26] backbone to preserve translational, rotational and permutational invariances, processing a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where edges connect atoms within a cutoff radius. The nodes contain element embeddings $\boldsymbol{E}$, desired properties $\boldsymbol{y}$, and diffusion progress information, while edge features encode interatomic distances. Implementation details of AMDEN are provided in Appendix C.

3 Results

We first trained unconditional AMDEN models on datasets of amorphous Si samples obtained from melt-quench simulations. We developed three datasets which are identical in terms of composition and samples size but differ in the samples' thermal history. While samples of the *melt* variant are obtained from a melt at 2500 K, the *quench* variant is obtained after an almost instantaneous quench to 300 K and the *anneal* variant is obtained after quenching the melted structures to 300 K at 1 K/ps.

Details about the dataset generation are given in Appendix A.1. For inference, the unit cells $\boldsymbol{C}$ of the training samples were used and the number of atoms was kept fixed at 256.

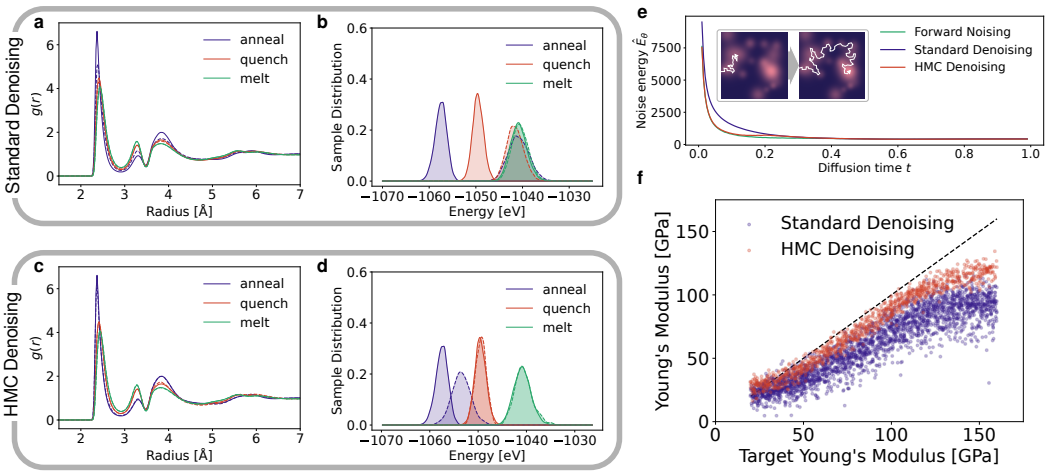

Figure 2: **a**, Radial distribution functions and **b** potential energy distributions after local geometry optimization for all three amorphous Si datasets (solid lines) and samples generated by AMDEN (dashed lines) using the standard denoising procedure. Panels **c** and **d** show the same analysis using HMC denoising. **e**, Noise energy $\hat{E}_\theta$ predicted by the model trained on the *anneal* Si dataset plotted against diffusion time t. **f**, Young's moduli of samples generated by AMDEN trained on the MEG dataset plotted against the target value used for conditioning.

To assess the quality of the generated samples, we computed radial distribution functions (RDFs), bond angle distributions and structure factors for the original data as well as after a local geometry optimization. Fig. 2a shows the RDFs after a local geometry optimization, while the other structural features and analysis of the unoptimized structures are provided in Appendix B. Local geometry optimizations were used to remove the influence of small residual noise and thermal motions and instead focus on the inherent structures. As expected, peaks in the RDF of Si become more intense and narrower as the extent of relaxation increases from *melt* to *quench* and *anneal* datasets. Interestingly, the discrepancy between the generated and reference samples follows the same trend. Considering the distribution of the potential energy of the geometry optimized samples shown in Fig. 2b, we observe that the energy of the training samples decreases with increasing extent of relaxation, while samples generated by AMDEN have an almost identical energy distribution for all three datasets. AMDEN thus appears unable to generate the low-energy structures that are reached through relaxation processes.

3.1 Relaxation Denoising

During relaxation the potential energy of a glassy system is gradually lowered, as the system explores configuration space, driven by thermal motion. A key feature of glassy potential energy landscapes (PEL) is the lack of a so-called funnel structure of the PEL, i.e., pathways to low energy minima through a sequence of catchment basins with continuously decreasing energy [27]. This PEL structure allows for incremental improvements in lowering the system's energy. In glassy systems, however, this is not possible, as many funnels are present with varying depth, separated by high barriers [28]. This is problematic for diffusion models, which rely on the assumption that samples can be generated by incrementally improving the previous state in the denoising process.

To address this, we propose a new variant of AMDEN, which instead of directly predicting the score function, predicts a so-called noise energy $\hat{E}_\theta(x)$, from which the score $s_\theta(x)$ is calculated as

$$s_\theta(x) = -\frac{1}{k_B T} \nabla_x \hat{E}_\theta(x). \tag{1}$$

Here, $k_\mathrm{B}T$ are normalization constants and set to 1 in our implementation. We note that $\hat{E}_\theta(x)$ is usually not a potential energy. Only in the case of an unconditional model, trained on Boltzmann

distributed samples, will $\hat{E}_\theta(x)$ recover the potential energy of the system, given that k_B and T are set to the Boltzmann constant and the equilibrium temperature of the training data, respectively.

Fig. 2e shows $\hat{E}_\theta$ as predicted by a model trained on the *anneal* Si dataset. The values are plotted against the diffusion coordinate t and obtained by averaging over ten forward noising trajectories, serving as ground truth, and ten denoising trajectories labeled as *forward* and *std denoising*, respectively. As seen in the plot, both curves diverge around $t = 0.4$ and the model ascribes a higher $\hat{E}_\theta$ and thus lower probability to the samples obtained from the standard denoising procedure. This indicates that the generated intermediate samples are not properly equilibrated on $\hat{E}_\theta(x)$ and thus are not representative of the target distribution $p(x)$. We therefore implement a modified variant of the denoising process, labeled *HMC denoising* in the figure, which incorporates Hamiltonian Monte Carlo [29] (HMC) steps on $\hat{E}_\theta$ to equilibrate the structures during denoising and sample from the true target distribution $p(x)$. As shown in Fig. 2e, the HMC denoising process is able to recover noise-energies matching the forward trajectory by overcoming barriers of higher $\hat{E}_\theta$ as illustrated in the figure inset. RDFs shown in Fig. 2c and bond angle distributions shown in Fig. 2d confirm that the samples obtained through HMC denoising are also structurally similar to those of the training data, while the standard denoising samples deviate significantly from the training data as shown in Figs. 2a and 2b. As seen in Fig. 2d, HMC denoising also lowers the potential energy of the generated *quench* and *anneal* samples, recovering the expected trend of lower potential energies with lower cooling rates. This is remarkable as neither the potential energy nor forces are seen by the model during training, underlining the high quality of the generated structures.

To test the inverse design capabilities of AMDEN, we created the multi-element glass dataset using classical MD simulations (see Appendix A.2 for details). It features a large variety of compositions, containing eleven different elements. We then trained AMDEN on the MEG dataset and conditioned the model on the Young's modulus E. Inference results using both the standard and our newly developed HMC denoising procedures are shown in Fig. 2f. The results show that including relaxation in the generation process is necessary, not only to reproduce the correct atomic structure of the sample but also macroscopic properties, which are the main target for inverse design. Without HMC denoising, the Young's moduli of the generated samples is consistently below the targeted value. With HMC denoising, this discrepancy is significantly reduced although a small offset still remains. The remaining discrepancy might be due to an insufficient number of HMC iterations or limitations of the model and will be subject to future investigations.

4 Discussion

We hypothesize that generating relaxed configurations beyond the training data is an inherent limitation of diffusion generative models, similar to their inability to sample spin glasses below a critical temperature [30] and not the result of an insufficiently expressive model as suggested by Comin and Lewis [31]. As observed by Lei et al. [22], diffusion models can also fail to generate crystalline structures by getting trapped in local minima when the crystalline order is removed from the structure, initializing the reverse diffusion process. Recently, it has been shown that a diffusion generative model was able to reproduce structural features and properties of amorphous SiO_2 across a range of cooling rates [23]. The authors report that "extra noise" is required during the denoising process to escape local minima in the learned score function and short MD simulations were used to further refine the final structures and remove outlier environments. Overall, these results match our observation that unmodified denoising procedures are insufficient to produce high quality amorphous structures that reproduce macroscopic properties.

While AMDEN is an important step towards the inverse design of amorphous materials, many challenges remain. Our proposed HMC denoising process is able to circumvent the limitation of the traditional approach in generating relaxed structures, but it comes with a significant computational cost. During training, backpropagation has to be performed twice, i.e., first to derive the score function from the noise energy, and second to update the weights of the denoiser network. Similarly, inference cost is increased as many evaluations of the model are required for each HMC update. Potential solutions include shrinking the model size through neural network quantization techniques [32] or employing shallower architectures, but these approaches may compromise model expressivity and generation performance. Another factor hindering computational efficiency is that the model is not guaranteed to generate charge-balanced samples due to the stochastic nature of generative

models. This may necessitate that multiple sampling rounds are performed to obtain perfectly charge-balanced samples. Guiding generative models with charge balance, a non-differentiable target, presents challenges, but we note that recent progress has been made in this area [33, 34].

References

[1] Yuanbin Liu, Ata Madanchi, Andy S Anker, Lena Simine, and Volker L Deringer. The amorphous state as a frontier in computational materials design. *Nature Reviews Materials*, pages 1–14, 2024.

[2] Alex Zunger. Inverse design in search of materials with target functionalities. *Nature Reviews Chemistry*, 2(4):0121, 2018.

[3] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, volume 33, pages 6840–6851, 2020.

[6] Tian Xie, Xiang Fu, Octavian-Eugen Ganea, Regina Barzilay, and Tommi S. Jaakkola. Crystal diffusion variational autoencoder for periodic material generation. In *International Conference on Learning Representations*, 2022.

[7] Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, et al. A generative model for inorganic materials design. *Nature*, pages 1–3, 2025.

[8] Lemeng Wu, Chengyue Gong, Xingchao Liu, Mao Ye, and Qiang Liu. Diffusion-based molecule generation with informative prior bridges. *Advances in Neural Information Processing Systems*, 35:36533–36545, 2022.

[9] Minkai Xu, Alexander S Powers, Ron O Dror, Stefano Ermon, and Jure Leskovec. Geometric latent diffusion models for 3d molecule generation. In *International Conference on Machine Learning*, pages 38592–38610. PMLR, 2023.

[10] Yuxuan Song, Jingjing Gong, Minkai Xu, Ziyao Cao, Yanyan Lan, Stefano Ermon, Hao Zhou, and Wei-Ying Ma. Equivariant flow matching with hybrid probability transport for 3d molecule generation. *Advances in Neural Information Processing Systems*, 36:549–568, 2023.

[11] Niklas Gebauer, Michael Gastegger, and Kristof Schütt. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. *Advances in neural information processing systems*, 32, 2019.

[12] Jordan Hoffmann, Louis Maestrati, Yoshihide Sawada, Jian Tang, Jean Michel Sellier, and Yoshua Bengio. Data-driven approach to encoding and decoding 3-d crystal structures. *arXiv preprint arXiv:1909.00949*, 2019.

[13] Juhwan Noh, Jaehoon Kim, Helge S Stein, Benjamin Sanchez-Lengeling, John M Gregoire, Alan Aspuru-Guzik, and Yousung Jung. Inverse design of solid-state materials via a continuous representation. *Matter*, 1(5):1370–1384, 2019.

[14] Callum J Court, Batuhan Yildirim, Apoorv Jain, and Jacqueline M Cole. 3-d inorganic crystal structure generation and property prediction via representation learning. *Journal of Chemical Information and Modeling*, 60(10):4518–4535, 2020.

[15] Teng Long, Nuno M Fortunato, Ingo Opahle, Yixuan Zhang, Ilias Samathrakis, Chen Shen, Oliver Gutfleisch, and Hongbin Zhang. Constrained crystals deep convolutional generative adversarial network for the inverse design of crystal structures. *npj Computational Materials*, 7 (1):66, 2021.

[16] Jerry Li, Aleksander Madry, John Peebles, and Ludwig Schmidt. On the limitations of first-order approximation in gan dynamics. In *International Conference on Machine Learning*, pages 3005–3013. PMLR, 2018.

[17] Imant Daunhawer, Thomas M. Sutter, Kieran Chin-Cheong, Emanuele Palumbo, and Julia E. Vogt. On the limitations of multimodal vaes. In *International Conference on Learning Representations*, 2022.

[18] James Lucas, George Tucker, Roger B Grosse, and Mohammad Norouzi. Don't blame the elbo! a linear vae perspective on posterior collapse. *Advances in Neural Information Processing Systems*, 32, 2019.

[19] Kun Xu, Chongxuan Li, Jun Zhu, and Bo Zhang. Understanding and stabilizing gans' training dynamics using control theory. In *International conference on machine learning*, pages 10566–10575. PMLR, 2020.

[20] Evan Becker, Parthe Pandit, Sundeep Rangan, and Alyson K Fletcher. Instability and local minima in gan training with kernel discriminators. *Advances in Neural Information Processing Systems*, 35:20300–20312, 2022.

[21] Hyuna Kwon, Tim Hsu, Wenyu Sun, Wonseok Jeong, Fikret Aydin, James Chapman, Xiao Chen, Vincenzo Lordi, Matthew R Carbone, Deyu Lu, et al. Spectroscopy-guided discovery of three-dimensional structures of disordered materials with diffusion models. *Machine Learning: Science and Technology*, 5(4):045037, 2024.

[22] Bo Lei, Enze Chen, Hyuna Kwon, Tim Hsu, Babak Sadigh, Vincenzo Lordi, Timofey Frolov, and Fei Zhou. Grand canonical generative diffusion model for crystalline phases and grain boundaries. *arXiv preprint arXiv:2408.15601*, 2024.

[23] Kai Yang and Daniel Schwalbe-Koda. A generative diffusion model for amorphous materials. *arXiv preprint arXiv:2507.05024*, 2025.

[24] Peter E Kloeden, Eckhard Platen, Peter E Kloeden, and Eckhard Platen. *Stochastic differential equations*. Springer, 1992.

[25] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[26] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In *ICML*, volume 139, pages 9323–9332, 2021.

[27] Zamaan Raza, Björn Alling, and Igor A Abrikosov. Computer simulations of glasses: the potential energy landscape. *Journal of Physics: Condensed Matter*, 27(29):293201, 2015.

[28] SP Niblett, M Biedermann, DJ Wales, and VK De Souza. Pathways for diffusion in the potential energy landscape of the network glass former sio2. *The Journal of Chemical Physics*, 147(15), 2017.

[29] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[30] Davide Ghio, Yatin Dandi, Florent Krzakala, and Lenka Zdeborová. Sampling with flows, diffusion, and autoregressive neural networks from a spin-glass perspective. *Proceedings of the National Academy of Sciences*, 121(27):e2311810121, 2024.

[31] Massimiliano Comin and Laurent J Lewis. Deep-learning approach to the structure of amorphous silicon. *Physical Review B*, 100(9):094107, 2019.

[32] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew G. Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *CVPR*, pages 2704–2713, 2018.

[33] Yujia Huang, Adishree Ghatare, Yuanzhe Liu, Ziniu Hu, Qinsheng Zhang, Chandramouli Shama Sastry, Siddharth Gururani, Sageev Oore, and Yisong Yue. Symbolic music generation with non-differentiable rule guided diffusion. In *ICML*, 2024.

[34] Po-Hung Yeh, Kuang-Huei Lee, and Jun-Cheng Chen. Training-free diffusion model alignment with sampling demons. In *ICLR*, 2025.

[35] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comp. Phys. Comm.*, 271:108171, 2022. doi: 10.1016/j.cpc.2021.108171.

[36] Frank H Stillinger and Thomas A Weber. Computer simulation of local order in condensed phases of silicon. *Physical review B*, 31(8):5262, 1985.

[37] Marco Bertani, Maria Cristina Menziani, and Alfonso Pedone. Improved empirical force field for multicomponent oxide glasses and crystals. *Physical Review Materials*, 5(4):045602, 2021.

[38] Joel I. Gersten and Frederick W. Smith. *The Physics and Chemistry of Materials*. John Wiley & Sons, Inc., New York, Chichester, Weinheim, Brisbane, Singapore, Toronto, 2001. ISBN 0-471-05794-0. A Wiley-Interscience publication. Joel I. Gersten and Frederick W. Smith, The City College of the City University of New York.

[39] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, volume 32, pages 11895–11907, 2019.

[40] Daniel Levy, Sékou-Oumar Kaba, Carmelo Gonzales, Santiago Miret, and Siamak Ravanbakhsh. Using multiple vector channels improves e (n)-equivariant graph neural networks. *arXiv preprint arXiv:2309.03139*, 2023.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[42] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[43] Seyedmorteza Sadat, Manuel Kansy, Otmar Hilliges, and Romann M Weber. No training, no problem: Rethinking classifier-free guidance for diffusion models. *arXiv preprint arXiv:2407.02687*, 2024.

[44] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations*, 2015.

[45] Tim Salimans and Jonathan Ho. Should ebms model the energy or the score? In *Energy Based Models Workshop-ICLR 2021*, 2021.

A Data Set Generation

A.1 Amorphous Silicon Data Sets

We created three data sets (*melt*, *quench* and *anneal*) of amorphous silicon to study the effects of relaxation on the generation performance. All three data sets were created using LAMMPS [35] software with the Stillinger–Weber potential [36] and consist of $10\,000$ samples each. The simulations were initialized with a unit cell containing 256 atoms of crystalline silicon, but different thermal schedules were applied to obtain the final samples. All MD simulations were performed in the NPT ensemble at zero pressure.

The *melt* data set was generated by heating the crystalline silicon from 2500 K, to 3000 K over 200 ps, equilibrating the melt for 300 ps, and then cooling it down again to 2500 K at a rate of 10^{12} K/s. The final samples were taken after equilibrating for another 300 ps at 2500 K.

The *anneal* and *quench* data sets were both initialized at 300 K, heated to 2500 K over 200 ps, and equilibrated for 300 ps. Samples for the *anneal* data set were then cooled down at a rate of 10^{12} K/s to 300 K and equilibrated for another 300 ps, while the cooling step was omitted for samples for the *quench* data set. The structures of the *anneal* data set were thus allowed to relax during the cooling period, while the *quench* samples were obtained from an almost instantaneous quenching procedure. However, a small amount of relaxation is still expected during the time period taken by the thermostat to adjust the temperature of the system to the lower target value.

A.2 Multi Element Glass Data Set

We created the multi element glass (MEG) data set to test our model's performance on data including a larger variety of elements. The data set consists of 9,027 samples, containing 11 different elements. Initial structures were generated from varying compositions of the glass formers SiO_2 and P_2O_5, and the modifiers Al_2O_3 Li_2O, BeO, K_2O, CaO, TiO_2, BaO and ZnO.

Structural samples and corresponding properties of the MEG data set were obtained using the workflow described below. Simulations were performed using LAMMPS [35] software and the Bertain–Menziani–Pedone (BMP)-shrm potential [37].

1. Elemental compositions were generated to include different ratios of the three glass formers, up to four different modifiers with total concentration of $40\,\%$ relative to the glass former concentration.
2. Initial structures of the generated compositions, containing roughly 800 atoms, were created by randomly placing the atoms in a simulation cell with a volume $V = 3\sum_i \frac{4}{3}\pi r_i^3$, with r_i being the covalent radius of atom i. The atoms positions were then adjusted to ensure that no two atoms were closer than the sum of their respective covalent radii. Finally, a local geometry optimization was performed to optimize the atomic positions and cell dimensions.
3. To ensure proper melting while avoiding evaporation, an initial temperature for the melt-quench procedure needed to be determined for each composition. For this task, the initial cells were doubled in size along one dimension to form a vacuum region. A short molecular dynamics (MD) simulation was then performed in the NVT ensemble during which the temperature was increased up to 8000 K for a duration of 100 ps. The evaporation temperature T_{evap} was then identified at the onset of pressure increase during the dynamics simulation.
4. Structural samples were obtained from a melt-quench simulation in the NPT ensemble, initialized at $T_{\text{init}} = \frac{3}{4}T_{\text{evap}}$. The samples were first melted for 400 ps, then quenched to 300 K at 5 K/ps and finally equilibrated for 300 ps. Out of 9,240 compositions, 213 samples were identified that did not melt property during the initial phase of the simulation and were excluded from the final data set.
5. Melt-quenched samples were then equilibrated at 50 K for 100 ps and subsequently heated to 500 K over 500 ps to extract heat capacities and thermal expansion coefficients.
6. Samples were also relaxed to compute the elasticity tensor using finite differences of the stress tensor.

Due to the finite number of atoms, some amount of uncertainty in the computed properties is expected. To assess these, we performed two independent runs of the workflow for a random subset of compositions, resulting in two sets of structural samples for which properties were calculated. For one set of samples, the final heating simulation of the workflow was then repeated with the same initial structure but using a different random seed to assess the variability introduced by the heating simulation. Correlation plots of all properties and corresponding Pearson correlation coefficients are shown in Fig. 3. The elastic constants, which were deterministically computed from the structural samples, correlate well between the independent runs of the workflow, indicating a strong dependence on the composition. Similarly, the evaporation temperature shows a strong correlation between the independent runs. The thermal expansion coefficient and the molar heat capacity show a weaker correlation between the independent runs but a good agreement between the two heating simulations. Overall, this indicates that the simulation workflows to obtain glass properties work reliably, with variability being attributed to differences between the structures of the samples.

Heat capacities were obtained as the gradient of a linear fit to the total energy versus temperature of the heating simulation in step 5 of the workflow. Similarly, the thermal expansion coefficient at room

temperature was obtained from a linear fit $V(T)$ to the volume versus temperature of the heating simulation and calculated as

$$\alpha = \frac{1}{V(T)} \left. \frac{\partial V(T)}{\partial T} \right|_{T=300\,\mathrm{K}} . \tag{2}$$

Elastic constants were obtained as described in Section A.3.

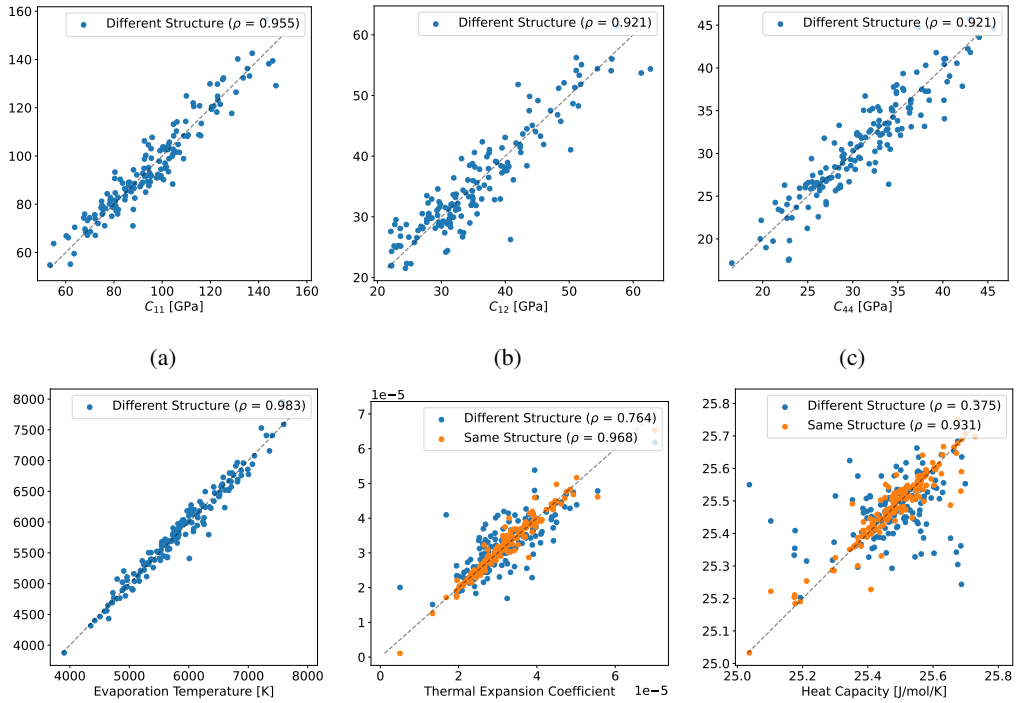

Figure 3: Correlation plots of the independent elastic constants C_{11} (a), C_{12} (b) and C_{44} (c), the evaporation temperature T_{evap} (d), the thermal expansion coefficient (e), and the molar heat capacity (f) for a random subset of samples of the MEG data set. The properties were computed from two independent runs of the workflow, initialized with the same compositions. Properties obtained from the heating simulation of the workflow were computed a third time by re-running the heating simulation initialized with the same structural sample but using a different random seed. Pearson correlation coefficients ρ are shown in the figure legends.

A.3 Mechanical properties

Young's and shear moduli were computed using the strain tensor C_{ijkl}. First, a local geometry optimization was performed on the structural samples to obtain the relaxed atomic positions and lattice vectors. The strain tensor was calculated as the derivative of the stress tensor σ_{ij} with respect to the strain ε_{kl}, i.e.,

$$C_{ijkl} = \left. \frac{\partial \sigma_{ij}}{\partial \varepsilon_{kl}} \right|_{\varepsilon=0} . \tag{3}$$

Finite differences were used to calculate the derivatives and atomic positions were relaxed after straining the unit cell before the stress tensor was computed.

Since the investigated samples are largely isotropic, we can reduce C_{ijkl} to C_{ij} using Voigt notation, averaging redundant entries in the full tensor. The Young's and shear moduli are then computed as

$$E = \frac{(C_{11} - C_{12}) \cdot (C_{11} + 2\,C_{12})}{C_{11} + C_{12}} \tag{4}$$

and

$$G = C_{44} \tag{5}$$

respectively [38].

B Structural features of the amorphous Si data set

To analyze the quality of the structures generated by AMDEN, we computed radial distribution functions, bond angle distributions, structure factors and the potential energy distribution of the generated and the training samples. All features were computed before and after performing a local geometry optimization of the structures. Features obtained from the standard denoising procedure are shown in Fig. 4, while those obtained from the Hamiltonian Monte Carlo (HMC) denoising procedure are shown in Fig. 5. Figures shown in the main text are included here again for completeness.

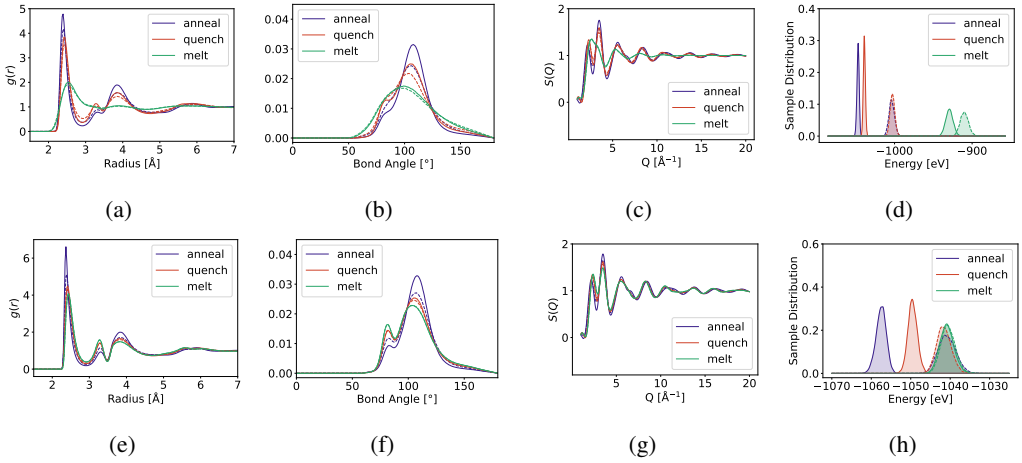

Figure 4: Radial distribution function (a), bond angle distribution (b), structure factor (c) and energies (d) of the generated structures compared to the training data. Panels (e), (f), (g) and (h) show the features in the same order after performing a local geometry optimization using the Tersoff potential used for generating the training data. RDF and energy distributions of the training data are shown by solid lines, while dashed lines are obtained from the AMDEN-generated samples.

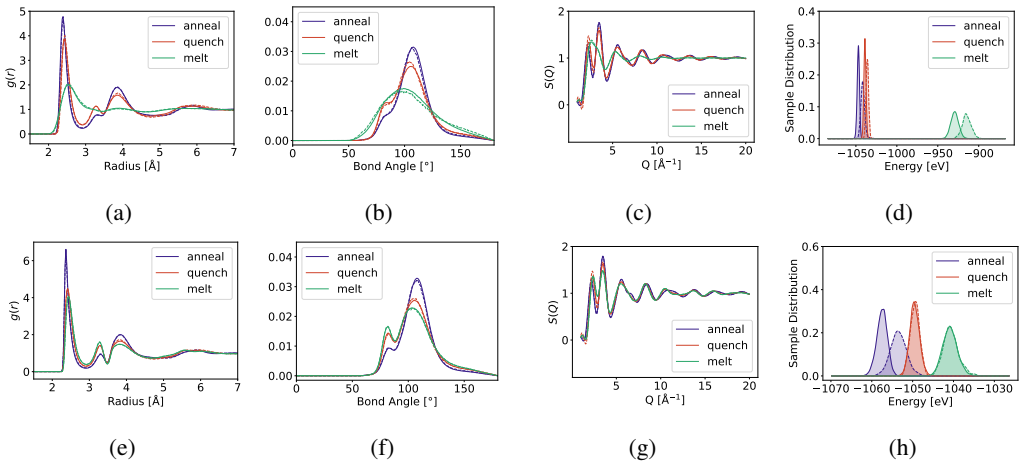

Figure 5: Radial distribution function (a), bond angle distribution (b), structure factor (c) and energies (d) of the structures generated using Hamiltonian Monte Carlo (HMC) denoising compared to the training data. Panels (e), (f), (g) and (h) show the features in the same order after performing a local geometry optimization using the Tersoff potential used for generating the training data. RDF and energy distributions of the training data are shown by solid lines, while dashed lines are obtained from the AMDEN-generated samples.

C Implementation Details of AMDEN

C.1 Materials diffusion process

Following the SDE framework [24, 25], our materials diffusion process is composed of two subprocesses: a forward diffusion process and a reverse denoising diffusion process. The forward diffusion process provides ground truth for noisy samples at intermediate steps x_t and noise components to supervise the score function s_θ.

The forward diffusion process gradually transformed a clean material sample x into random noise $\tilde{x}$ through a sequence of time step $t \in [t_{\text{min}}, t_{\text{max}}]$. Formally, given a clean sample $x = (\boldsymbol{C}, \boldsymbol{X}, \boldsymbol{E})$, the positions and element embeddings of the noisy sample x_t at time step t were calculated as,

$$\begin{aligned} \boldsymbol{X}_t &= \alpha_{\boldsymbol{X}}(t) \cdot \boldsymbol{X} + \sigma_{\boldsymbol{X}}(t) \cdot \boldsymbol{\epsilon}_{\boldsymbol{X}}, \\ \boldsymbol{E}_t &= \alpha_{\boldsymbol{E}}(t) \cdot \boldsymbol{E} + \sigma_{\boldsymbol{E}}(t) \cdot \boldsymbol{\epsilon}_{\boldsymbol{E}}, \end{aligned} \tag{6}$$

where $\boldsymbol{\epsilon}_{\boldsymbol{X}}$ and $\boldsymbol{\epsilon}_{\boldsymbol{E}}$ are noise components sampled from $\mathcal{N}(0, \mathbf{I})$, and α and σ are time-dependent scaling factors specific to each noise schedule.

For atomic positions, we employed a variance exploding (VE) schedule [39],

$$\alpha_{\boldsymbol{X}}(t) = 1, \quad \sigma_{\boldsymbol{X}}(t) = \sigma_{\text{max}}^{\boldsymbol{X}} \cdot t, \tag{7}$$

where $\sigma_{\text{max}}^{\boldsymbol{X}}$ was set to 1.7 Å for models trained on the MEG dataset and 1.5 Å for all other models. For element embeddings, we implemented a variance preserving (VP) schedule [39] with a cosine progression,

$$\alpha_{\boldsymbol{E}}(t) = \cos\left(\frac{\pi}{2}t\right), \quad \sigma_{\boldsymbol{E}}(t) = \sin\left(\frac{\pi}{2}t\right) \cdot \sigma_{\text{max}}^{\boldsymbol{E}}, \tag{8}$$

where $\sigma_{\text{max}}^{\boldsymbol{E}}$ was set to 1.5. Combined with the periodic boundary condition, the two schedules determine the positions $\boldsymbol{X}_{t_{\text{max}}}$ and element embeddings $\boldsymbol{E}_{t_{\text{max}}}$ of the sample $x_{t_{\text{max}}}$. At the last step of the forward diffusion process the atomic position and element embeddings follow a uniform distribution within the cell $\boldsymbol{C}$ and standard Gaussian distribution $\mathcal{N}(0, \mathbf{I})$, respectively. This created a starting point for the reverse denoising diffusion process that can be sampled trivially.

The reverse denoising process gradually transforms random noise into a material sample through iterative denoising manipulated by the learnable score function s_θ. To progress the process at t by Δt, the Euler-Maruyama method was applied to solve the reverse SDE [24],

$$\begin{aligned} \boldsymbol{X}_{t-\Delta t} &= \boldsymbol{X}_t - \left(f_{\boldsymbol{X}}(t, \boldsymbol{X}_t) \cdot \Delta t + g_{\boldsymbol{X}}(t) \cdot \mathbf{z}_{\boldsymbol{X}} \cdot \sqrt{\Delta t}\right), \\ \boldsymbol{E}_{t-\Delta t} &= \boldsymbol{E}_t - \left(f_{\boldsymbol{E}}(t, \boldsymbol{E}_t) \cdot \Delta t + g_{\boldsymbol{E}}(t) \cdot \mathbf{z}_{\boldsymbol{E}} \cdot \sqrt{\Delta t}\right), \end{aligned} \tag{9}$$

where $\mathbf{z}_{\boldsymbol{X}}$ and $\mathbf{z}_{\boldsymbol{E}}$ are noises sampled independently from a standard normal distribution, f and g are coefficients defined as,

$$\begin{aligned} f_{\boldsymbol{X}}(t, \boldsymbol{X}_t) &= -g_{\boldsymbol{X}}^2(t) \cdot s_\theta^{\boldsymbol{X}}, \\ g_{\boldsymbol{X}}(t) &= \sqrt{2{\sigma_{\text{max}}^{\boldsymbol{X}}}^2 \cdot t}, \\ f_{\boldsymbol{E}}(t, \boldsymbol{E}_t) &= -\frac{\pi}{2}\tan\left(\frac{\pi}{2}t\right) \cdot \boldsymbol{E}_t - g_{\boldsymbol{E}}^2(t) \cdot s_\theta^{\boldsymbol{E}}, \\ g_{\boldsymbol{E}}(t) &= \sqrt{\pi \cdot \alpha_{\boldsymbol{E}}(t) \cdot \sigma_{\boldsymbol{E}}(t) \cdot \sigma_{\text{max}} + \pi \cdot \tan\left(\frac{\pi}{2}t\right) \cdot \sigma_{\boldsymbol{E}}(t)^2}. \end{aligned} \tag{10}$$

In these equations, $s_\theta^{\boldsymbol{X}}$ and $s_\theta^{\boldsymbol{E}}$ are position and element components of the score function, respectively.

The complete reverse denoising process started from a noise sample $\tilde{x} = (\boldsymbol{C}, \boldsymbol{X}_{t_{\text{max}}}, \boldsymbol{E}_{t_{\text{max}}})$ where $\boldsymbol{X}_{t_{\text{max}}}$ was sampled from uniform random distributed in $\boldsymbol{C}$ and $\boldsymbol{E}_{t_{\text{max}}}$ was sampled from $\mathcal{N}(0, \mathbf{I})$, aligning with the sample $x_{t_{\text{max}}}$ produced at the last step of the forward diffusion process. The reverse process progressed sequentially from $t_{\text{max}} = 0.99$ to $t_{\text{min}} = 0.01$ with $n = 200$ evenly spaced intervals and a final step to $t = 0$.

C.2 Equivariant graph neural network

The EGNN serves as the equivariant backbone of the score function. The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ provided to EGNN is a graph of the atoms in each sample x, calculated with a cutoff radius $r_{\text{cut}} = 6.5$ Å. The cutoff radius was chosen to ensures that all bonded and strongly interacting atoms share a direct edge connection while still keeping the graph sufficiently sparse. Our EGNN implementation was composed of $L = 4$ EGNN layers. The l-th layer takes as input: 1) Node features $\boldsymbol{H}^{(l)} \in \mathbb{R}^{n \times d_h}$ containing information of the corresponding atoms; 2) Positional coordinates $\boldsymbol{X}^{(l)} \in \mathbb{R}^{n \times k \times 3}$ of the atoms, where $k = 8$ is the number of vector channels [40]; and 3) Edge set $\mathcal{E}$ of the graph $\mathcal{G}$.

For the initial layer, the positions $\boldsymbol{X}^{(0)}$ were replicated original positions $\boldsymbol{X}$ for k channels. The edge attributes $\boldsymbol{e}_{ij}$ were derived from the distance embedding,

$$\boldsymbol{e}_{ij} = \tanh\left(\frac{\|\boldsymbol{X}_i - \boldsymbol{X}_j - \mathbf{o}_{ij}\|^2}{r_{\text{cut}}^2}\right) \cdot 2 - 1 \tag{11}$$

where $\mathbf{o}_{ij}$ is the offset vector accounting for periodic boundary conditions.

Each layer updated the node features and positional coordinates, incorporating self-attention [41] with a hidden dimension of 128 as,

$$\begin{aligned}
\mathbf{m}_{ij}^{(l)} &= \phi_e^{(l)}(\boldsymbol{H}_i^{(l-1)}, \boldsymbol{H}_j^{(l-1)}, \boldsymbol{e}_{ij}), \\
\alpha_{ij}^{(l)} &= \sigma(\text{MLP}_{\text{att}}(\mathbf{m}_{ij}^{(l)})), \\
\hat{\mathbf{m}}_{ij}^{(l)} &= \alpha_{ij}^{(l)} \cdot \mathbf{m}_{ij}^{(l)}, \\
\boldsymbol{H}_i^{(l)} &= \boldsymbol{H}_i^{(l-1)} + \phi_H^{(l)}\left(\boldsymbol{H}_i^{(l-1)}, \sum_{j \in N(i)} \frac{f_{\text{cut}}(d_{ik}^{(0)}) \cdot \hat{\mathbf{m}}_{ij}^{(l)}}{n_{\text{norm}}}\right), \\
\boldsymbol{\Phi}_{ij}^{(l)} &= \text{MLP}_{\text{coord}}([\boldsymbol{H}_i^{(l)}, \boldsymbol{H}_j^{(l)}, \boldsymbol{e}_{ij}]) \in \mathbb{R}^{k \times k}, \\
\mathbf{d}_{ij}^{(l)} &= \boldsymbol{X}_i^{(l-1)} - \boldsymbol{X}_j^{(l-1)} - \mathbf{o}_{ij}, \\
\boldsymbol{X}_i^{(l)\prime} &= \sum_{j \in N(i)} \frac{1}{n_{\text{norm}}} \cdot \boldsymbol{\Phi}_{ij}^{(l)} \cdot \mathbf{d}_{ij}^{(l)}, \\
\boldsymbol{X}_i^{(l)} &= \boldsymbol{X}_i^{(l-1)} + \boldsymbol{X}_i^{(l)\prime},
\end{aligned} \tag{12}$$

where $N(i)$ represents the neighbors of atom i, derived from the edge set $\mathcal{E}$ and σ is the sigmoid activation function for self-attention. n_{norm} is a normalization factor (typically proportional to the average number of neighbors) to ensure numerical stability, which we set to 40. $\phi_e^{(l)}$, $\phi_H^{(l)}$ are implemented as multi-layer perceptrons (MLPs) with SiLU activation functions and layer normalization. $\boldsymbol{\Phi}_{ij}^{(l)}$ is a learned transformation matrix that maps between the k vector channels. A smooth cutoff function is used to prevent discontinuities when atoms leave or enter the cutoff radius, which is defined as follows.

$$f_{\text{cut}}(r) = 2\tanh\left(1 - \frac{\min(r, r_{\text{cut}})}{r_{\text{cut}}}\right)^2. \tag{13}$$

In our implementation, the learnable functions were structured as follows,

$$\begin{aligned}
\phi_e^{(l)}(\boldsymbol{H}_i, \boldsymbol{H}_j, \boldsymbol{e}_{ij}) &= \text{MLP}_{\text{edge}}([\boldsymbol{H}_i, \boldsymbol{H}_j, \boldsymbol{e}_{ij}]), \\
\phi_H^{(l)}(\boldsymbol{H}_i, \mathbf{m}_{\text{agg}}) &= \text{MLP}_{\text{node}}([\boldsymbol{H}_i, \mathbf{m}_{\text{agg}}]).
\end{aligned} \tag{14}$$

At the last layer, EGNN outputs $\boldsymbol{H}^{(L)}$ and $\boldsymbol{X}^{(L)}$ as the final node features and positional coordinates, respectively.

C.3 Score function

The score function extended the EGNN framework to predict noise components in both positions and element embeddings during the reverse denoising diffusion process, optionally conditioned on desired

properties. The score function s_θ mapped a noisy material sample $x_t = (\boldsymbol{C}, \boldsymbol{X}_t, \boldsymbol{E}_t)$, diffusion step t, and desired properties $\boldsymbol{y}$ to position and element components,

$$s_\theta : (\boldsymbol{C}, \boldsymbol{X}_t, \boldsymbol{E}_t, t, \boldsymbol{y}) \mapsto (s_\theta^{\boldsymbol{X}}, s_\theta^{\boldsymbol{E}}), \tag{15}$$

which we implemented with the following three main components.

First, the property embedding layer transformed material properties into a continuous embedding space. For each property y_i in the property set $\mathbb{Y}$, we implemented,

$$\mathbf{h}_{y_i} = \text{LayerNorm}(\text{Linear}(y_i)), \tag{16}$$

where Linear is a fully-connected linear layer that maps the property y_i into a $d_{\text{emb}} = 8$-dimensional embedding vector.

Second, the feature assembly layer combined time embeddings, element representations, and property embeddings. The initial node features provided to the EGNN backbone were assembled by concatenating, 1) diffusion time step t; 2) element embeddings: $\mathbf{e}_i \in \mathbb{R}^{d_e}$; and 3) property embeddings: $\mathbf{h}_{y_1}, \mathbf{h}_{y_2}, \ldots, \mathbf{h}_{y_m}$ (for each property in $\mathbb{Y}$). Formally, the initial node features $\boldsymbol{H}_i^{(0)}$ for atom i were given by,

$$\boldsymbol{H}_i^{(0)} = [t, \mathbf{e}_i, \mathbf{h}_{y_1}, \mathbf{h}_{y_2}, \ldots, \mathbf{h}_{y_m}]. \tag{17}$$

Third, the EGNN backbone processed the assembled features while preserving geometric equivariance. The assembled node features, along with the raw noisy material sample x_t, were processed through the EGNN backbone. We calculated the predicted noise components for positions and elements as,

$$\epsilon_\theta^{\boldsymbol{X}} = \boldsymbol{X} - \boldsymbol{X}^{(L,0)}, \quad \epsilon_\theta^{\boldsymbol{E}} = \boldsymbol{H}^{(L)}, \tag{18}$$

where for positions we used the deviation between the original positions and the first channel of output positions, and for elements we directly used the output node features. The score was then calculated as $s_\theta = -\epsilon_\theta / \sigma(t)$ for both positions and elements.

We further incorporate classifier-free guidance (CFG) [42] to enhance the calculation of conditioned scores. Denoting the unconditioned noise as $\tilde{\epsilon}_\theta^{\boldsymbol{X}}$ and $\tilde{\epsilon}_\theta^{\boldsymbol{E}}$, the final noise under CFG is calculated as,

$$\begin{aligned} {\epsilon'_\theta}^{\boldsymbol{X}} &= (1 + w_{\text{cond}}){\epsilon_\theta}^{\boldsymbol{X}} - w_{\text{cond}}\tilde{\epsilon}_\theta^{\boldsymbol{X}}, \\ {\epsilon'_\theta}^{\boldsymbol{E}} &= (1 + w_{\text{cond}}){\epsilon_\theta}^{\boldsymbol{E}} - w_{\text{cond}}\tilde{\epsilon}_\theta^{\boldsymbol{E}}, \end{aligned} \tag{19}$$

where w_{cond} is the condition weight. The unconditioned noise can be calculated by using a dedicated unconditioned model trained without the properties $\boldsymbol{y}$, or by leveraging the independent condition guidance (ICG) [43]–feeding the model with random properties sampled from the distribution of the training set. For the MEG dataset, we use the ICG technique and set $w_{\text{cond}} = 0.25$.

The training of the score function was done by supervising the reverse denoising process at randomly sampled time steps. Specifically, for each training sample, we: 1) randomly generated a diffusion time step $t \sim \sqcap(t_{\text{min}}, t_{\text{max}})$; 2) applied the forward diffusion process to obtain the noisy sample and ground truth noise components; 3) fed the noisy sample into the score function to predict the noise components; and 4) computed the loss as the mean squared error between the prediction and ground truth.

Formally, the training objective was,

$$\mathcal{L}(\theta) = \mathbb{E}_{x,t}\Big[\|\boldsymbol{\epsilon}_{\boldsymbol{X}} - \epsilon_\theta^{\boldsymbol{X}}\|^2 + \lambda\|\boldsymbol{\epsilon}_{\boldsymbol{E}} - \epsilon_\theta^{\boldsymbol{E}}\|^2\Big], \tag{20}$$

where $\lambda = 0.5$ is a weighting factor balancing the importance of position and element noise prediction. The training was performed with the Adam optimizer [44] for 400, 500, and 1000 epochs with batch sizes of 2 and 8, and learning rates of 10^{-3} and 10^{-3} for the MEG and Si datasets, respectively. Virtual machines each with one NVIDIA A40 or V100 GPUs, 8 CPU cores, and 48 GB of RAM are used for both training and inference.

C.4 Density control via ghost atoms

A unique feature of AMDEN is its ability to control material density during inference through a *ghost atom* mechanism. This approach enables generating materials with specific density targets,

addressing the fundamental limitation that diffusion models cannot alter the number of atoms in generated structures. In our context, diffusion models manipulate samples by adding or removing noise from atomic positions and element embeddings, but cannot add or remove atoms to change sample densities. Existing solutions include voxel-based approaches [22] that enable diffusion models to decide whether an atom exists in each small cell. However, voxel-based approaches cannot preserve the equivariance of amorphous materials. In contrast, the ghost atom approach maintains a fixed total number of atoms and preserves AMDEN's equivariant architecture. It introduces a special type of atom that is removed from the final generated samples. Given a sample x and a target maximum density ρ_{target}, we first calculate the desired number of atoms based on ρ_{target} and cell volume as $n_{\text{target}} = \lfloor \rho_{\text{target}} \cdot \text{Volume}(\boldsymbol{C}) \rfloor$. If n_{target} exceeds the number of actual atoms in the generated structure, we supplement with ghost atoms. A number of ghost atoms, $n_{\text{ghost}} = \max(0, n_{\text{target}} - n_{\text{actual}})$, are then randomly positioned within the cell. During training and denoising, ghost atoms are treated like normal atoms but are assigned a special chemical element class. The model can thus adjust the density of the sample by increasing or decreasing the fraction of atoms that are assigned the ghost atom type. As a final step after denoising, ghost atoms are removed from the sample.

For the MEG dataset, we set $\rho_{\text{target}} = 0.11$. Since the Si datasets are designed specifically for evaluating the structural accuracy of generation, ghost atoms are not used there.

C.5 Hamiltonian Monte Carlo refinement

The energy-based score function was introduced to support the prediction of noise energy utilized in Section 3.1. The key difference between the standard score function and the energy-based variant lies in how noise components are predicted. While the standard score function directly predicted position and element noise components, the energy-based variant reformulated the noise prediction problem using an energy function $\hat{E}_\theta$ and then computed noise as derivatives of this energy. It should be noted that $\hat{E}_\theta$ does not necessarily relate to any physical energy, such as the potential energy. Only in the special case, when the model was trained on Boltzmann distributed samples and $t = 0$, the model would learn the potential energy of the system, up to an additive constant.

The standard score function s_θ was defined through the targeted distribution $p(x)$ as,

$$s_\theta(x) \approx \nabla_x \ln p(x). \tag{21}$$

In the energy-based variant, $p(x)$ was predicted through $\hat{E}_\theta$ as,

$$p(x) = \frac{1}{Z} \exp\left(\frac{-\hat{E}_\theta(x)}{k_{\text{B}} T}\right), \tag{22}$$

where Z is the unknown partition function normalizing $p(x)$, k_{B} is the Boltzmann constant, and T is the temperature. Since $\hat{E}_\theta$ does not match the potential energy k_{B} and T are arbitrary normalization constants and chosen to be $k_{\text{B}}T = 1$, we thus obtain

$$s_\theta(x) = -\frac{1}{k_{\text{B}} T} \nabla_x \hat{E}_\theta(x). \tag{23}$$

Specifically, the score function was reconfigured to output a scalar atomic energy value for each atom, instead of directly outputting noise vectors. For each atom i, the noise energy was constructed as a combination of: 1) position-based energy [45]: $e_i^{\boldsymbol{X}} = \frac{1}{2}\|\boldsymbol{X}_i - \boldsymbol{X}_i^{(L)}\|^2$, derived from squared distances between original positions and EGNN-output positions; and 2) atomic energy: $e_i^{\text{atom}} = \boldsymbol{H}_i^{(L)}$, EGNN-output node features.

The total system noise energy was given by,

$$\hat{E}_\theta(x) = \frac{1}{\sigma(t)} \sum_{i=1}^{n} \left(e_i^{\boldsymbol{X}} + \gamma \cdot e_i^{\text{atom}}\right), \tag{24}$$

where γ is a learnable scale factor balancing the contribution of atomic energies.

Then, the position and element components of the score function were computed through automatic differentiation,

$$\begin{aligned} s_\theta^{\boldsymbol{X}} &= -\frac{1}{k_{\mathrm{B}}T}\nabla_{\boldsymbol{X}}\hat{E}_\theta(x), \\ s_\theta^{\boldsymbol{E}} &= -\frac{1}{k_{\mathrm{B}}T}\nabla_{\boldsymbol{E}}\hat{E}_\theta(x). \end{aligned} \tag{25}$$

We note that the energy-based score function was computationally heavier compared to the standard variant, thus its training was performed with smaller batch sizes: 1 and 4 for MEG and Si datasets, respectively. For the same reason it is trained for fewer (i.e., 150 epochs) on the MEG dataset, and utilized virtual machines each with one NVIDIA A100 GPU for training. Other hardware configures were kept the same.

Hamiltonian Monte Carlo denoising. As discussed in previous sections, the standard denoising procedure outlined in Section C.1 was unable to produce samples of relaxed structures. This was because the model failed to steer the denoising trajectory towards relaxed samples in the early stages of denoising. However, analysis of energy-based score function shows that the model is able to assign a higher probability to relaxed samples at later stages of the denoising process. This allowed us to refine the generated samples by equilibrating on the predicted distribution $p(x)$ during the denoising process using Hamiltonian Monte Carlo (HMC).

A series of HMC iterations were therefore performed between the traditional denoising iterations. The application of HMC iterations can be limited to a range of diffusion time steps t within a specified range $[t_{\text{min}}^{\text{HMC}}, t_{\text{max}}^{\text{HMC}}]$ to reduce the computational cost. We set $t_{\text{min}}^{\text{HMC}} = 0.0$ and $t_{\text{max}}^{\text{HMC}} = 0.5$, respectively. Exactly one HMC iteration is performed before each diffusion denoising step within the range.

In each HMC iteration, atomic coordinates were updated using the following steps:

1. Initializing momenta from a Maxwell–Boltzmann distribution: $\mathbf{p} \sim \mathcal{N}(0, k_{\mathrm{B}}T\mathbf{M})$, where $\mathbf{M}$ is the diagonal mass matrix chosen as $\mathbf{M} = \mathbf{I}$;
2. Computing the initial energy $\hat{E}_\theta$ and total energy $E_{\text{tot}} = \hat{E}_\theta + \frac{1}{2}\|\mathbf{p}\|^2$;
3. Evolving the system using velocity Verlet integration with 15 steps,

$$\begin{aligned} \boldsymbol{X} &\leftarrow \boldsymbol{X} + \mathbf{p}\cdot dt + \frac{1}{2}\boldsymbol{F}\cdot dt^2, \\ \boldsymbol{F} &\leftarrow -\nabla_{\boldsymbol{X}}\hat{E}_\theta(\boldsymbol{X}), \\ \mathbf{p} &\leftarrow \mathbf{p} + \frac{1}{2}(\boldsymbol{F} + \boldsymbol{F}_{\text{last}})\cdot dt; \end{aligned} \tag{26}$$

4. Computing the final energy and accepting or rejecting the final structure according to the Metropolis–Hastings criterion with probability,

$$\alpha = \min\left[1, \exp\left(\frac{-(E_{\text{tot}}^{\text{final}} - E_{\text{tot}}^{\text{initial}})}{k_{\mathrm{B}}T}\right)\right]. \tag{27}$$

To ensure a consistent acceptance rate of roughly 0.5, the timestep dt was adjusted using $dt = \sigma_t dt^0$, where dt^0 is a constant. This accounted for the fact that $\hat{E}_\theta$ becomes less smooth for lower σ_t. We set $dt^0 = 0.2$ for the MEG dataset and $dt^0 = 0.4$ for Si datasets. We set the number of diffusion steps to $n = 2000$ when performing HMC refinement to ensure there are enough HMC iterations for finding lower energy structures. We also set $w_{\text{cond}} = 0$, i.e., we do not use CFG when performing HMC refinement to avoid adding further computational burden.

This physics-inspired sampling approach enabled the generation of more stable and realistic material structures by allowing the system to explore the energy landscape at each diffusion step, effectively annealing the structure as the noise level decreased. On the other hand, since the energy was provided by the denoiser network conditioned on properties , incorporation of HMC sampling preserved the property-conditioned generation process.

D Broader Impacts

Positive Societal Impacts: AMDEN contributes to accelerating the discovery of amorphous materials for applications in batteries, non-linear optics, and catalysis, as highlighted in our introduction. By

enabling computational screening of material compositions and properties before synthesis, our approach reduces the time and resources required for materials development compared to traditional trial-and-error methods. This computational approach also provides an alternative pathway for materials exploration when experimental facilities are limited.

Potential Negative Impacts: While AMDEN is designed for beneficial applications, we acknowledge several potential risks. The ability to rapidly generate novel material structures could potentially be misused to design materials with harmful properties, although the significant barrier of actual synthesis provides some protection against malicious use. The high computational requirements of our method, particularly for HMC denoising, may create access barriers for researchers at under-resourced institutions, potentially exacerbating existing inequalities in scientific research.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The major claims of the paper are to introduce a diffusion model-based framework that generates structures of amorphous materials and can be conditioned on target properties, and to demonstrate the inherent challenges for diffusion models to generate relaxed structures. All claims accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of this work in Section 4.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not introduce theoretical proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Implementation, data preparation, and experiment details are given in the paper and appendix, which provide sufficient information to reproduce the results presented in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: This work presents ongoing research submitted to a NeurIPS workshop. The implementation is currently under active development. Code and datasets will be made available upon completion of the ongoing research efforts.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The detailed experimental settings are given in the Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Our analysis uses structural features (RDF and bond angle distributions) and energy distributions. These are obtained from 10 000 samples and thus converged to an accuracy of the magnitude of the line width shown in the figures. Since our analysis is based on the qualitative discrepancy and agreement of these features, error bars would not enhance the understanding of the results. For the reported Young's moduli of generated MEG samples, target and resulting moduli of each sample are shown in the scatter plot and no averaging was performed.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The detailed experimental hardware configuration is given in Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We ensured that the study complied with the NeurIPS Code of Ethics in all respects.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both potential positive and negative societal impacts in the Broader Impacts section (Section D) of the Appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not suffer from this risk.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.