AMGENC: Generating Charge Balanced Amorphous Materials

Yan Lin[1], **Jilin Hu**[2,*], **N. M. Anoop Krishnan**[3,4], **Morten M. Smedskjaer**[5]
[1] Department of Computer Science, Aalborg University, Denmark
[2] School of Data Science and Engineering, East China Normal University, China
[3] Department of Civil Engineering, Indian Institute of Technology Delhi, India
[4] Yardi School of Artificial Intelligence, Indian Institute of Technology Delhi, India
[5] Department of Chemistry and Bioscience, Aalborg University, Denmark
lyan@cs.aau.dk, jlhu@dase.ecnu.edu.cn, krishnan@iitd.ac.in, mos@bio.aau.dk

Abstract

Amorphous (disordered) materials require simulation cells with at least hundreds to thousands of atoms to accurately capture their diverse local atomic environments, compared to crystalline materials that can be described by unit cells containing a few to hundreds of atoms. To advance the design of amorphous materials with desired properties and facilitate the exploration of their vast design space, generative inverse design has emerged as a promising approach. It aims to directly output materials with properties closely aligned with the desired ones using probabilistic generative models conditioned on desired properties, which can be more resource efficient than the traditional trial-and-error approach. However, due to the inherent stochasticity of probabilistic generative models, when element assignments are unconstrained, a large portion of generated materials may be charge unbalanced, and existing methods either fix the composition in advance or discard charge unbalanced samples after generation. In this work, we propose AMGENC, a new generative inverse design method for amorphous materials that can guarantee the generation of formally charge balanced samples, with minimal additional computational overhead and without sacrificing inverse design accuracy. AMGENC achieves this through an element noise that gives the generation process a starting point centered around charge balance, and the combination of a per-step soft projection and a final discrete projection for steering the elements toward exact charge balance throughout the generation. We perform extensive experiments on two amorphous materials datasets. Experimental results provide evidence that AMGENC achieves its design goal.

1 Introduction

Glasses and other amorphous materials are solids that lack a periodic atomic arrangement or long-range atomic order, yet exhibit complex short- and medium-range order. They have shown great potential in domains including energy storage, thermal management, and advanced materials [27]. Formally, an amorphous material sample $\mathcal{M} = (\boldsymbol{L}, \boldsymbol{X}, \boldsymbol{E})$ consists of a lattice $\boldsymbol{L}$, atom positions $\boldsymbol{X}$, and element assignments $\boldsymbol{E}$ (Definition 1). While the periodic structure of crystalline materials allows them to be fully described by a small cell containing only a few to hundreds of atoms, amorphous materials require larger cells with at least hundreds or thousands of atoms to accurately capture their diverse local atomic environments.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

To advance the design of amorphous materials with desired properties, generative inverse design has emerged as a promising approach. Compared to traditional materials design that relies on a resource-intensive trial-and-error process, it starts with the target properties, typically represented as a vector $\boldsymbol{y} \in \mathbb{R}^{n_y}$ of n_y numerical properties, and directly outputs the sample $\mathcal{M}$ with properties closely matched to the target $\boldsymbol{y}$ using probabilistic generative models [19, 12, 15, 39, 26]. Essentially, it can be formulated as a conditional probabilistic model $p_\theta(\mathcal{M}|\boldsymbol{y})$, where θ represents the learnable parameters. This approach has shown success in designing crystalline materials and molecules [11, 34, 29, 5, 48, 53, 47, 16, 40, 51, 30]. Due to computational challenges in scaling generative models to hundreds or thousands of atoms, and the scarcity of diverse large-scale amorphous material datasets, generative inverse design remains relatively under-developed for amorphous materials [21, 49, 10]. Nevertheless, the vast and largely unexplored design space of amorphous materials makes generative inverse design particularly promising, as it can efficiently navigate regions that are intractable for the traditional trial-and-error approach.

One limitation of existing generative inverse design methods is their inability to enforce charge balance constraints on the elements $\boldsymbol{E}$ of generated samples without affecting the generative model's flexibility in adjusting the elements. Due to the stochastic nature of probabilistic generative models, there is a certain probability of them generating charge unbalanced samples when $\boldsymbol{E}$ is unconstrained. Here, charge balance refers to formal charge balance, i.e., zero total charge under a supplied formal charge assignment $\boldsymbol{c}$ (Definition 2). For ionic oxide glasses such as those studied in this work, formal charge balance is equivalent to a sample containing exactly the oxygen needed to form neutral oxide units with its cations (Appendix E.3). It is thus a necessary global condition on the composition of every sample, while local bonding, stability, and synthesizability require separate assessments. A concrete use case is property-conditioned screening of glasses with variable composition, such as lithium-containing oxide glasses for solid electrolytes [8, 18], where the target properties should steer both the composition and the structure. Some existing methods [21, 49] fix the composition, i.e., $\boldsymbol{E}$ is pre-defined and fixed during the generation, which limits the generation of materials with diverse compositions. Other existing methods [48, 53, 10] follow the post-hoc filtering approach, i.e., discarding all charge unbalanced samples after the generation. Post-hoc filtering is tolerable for crystalline materials, thanks to their small system scale, which usually results in less than 20% of the generated samples being discarded [40, 51, 30]. However, it can be problematic for amorphous materials, since each sample has hundreds to thousands of atoms, often resulting in more than 90% of the generated samples being discarded (Section 5.4). This defies the benefits of inverse design to some extent, since we would need to generate more samples to increase the chance of getting a charge balanced sample, effectively returning to the trial-and-error approach.

We identify that the main challenge in enforcing charge balance on $\boldsymbol{E}$ comes down to the fact that it is a non-differentiable constraint, since computing $\boldsymbol{E}$ involves $\arg\max$ for probabilistic generative models. While there are techniques for guiding the generation towards a non-differentiable preference [1, 17, 52], they improve rather than guarantee the satisfaction of the preference. Techniques for enforcing hard constraints either require the constraint to be differentiable [43] or restrict the action space of a sequential generator [32], and are not directly applicable to discrete element assignments that evolve jointly with atom positions in a single flow.

To this end, we propose *Amorphous Material Generation with Charge Balanced Constraint* (**AMGENC**), a new method for generative inverse design of amorphous materials that can guarantee the generation of charge balanced samples. Our contributions are as follows:

- We propose AMGENC, a flow-matching-based [26] generative inverse design method for amorphous materials that guarantees formally charge balanced generation with minimal computational overhead and without sacrificing inverse design accuracy.
- To achieve this, AMGENC introduces three components: a *data-centered conditional element noise* that gives the generation a starting point centered around charge balance, a *per-step soft linearized projection* that steers elements toward charge balance at each generation step by relaxing the non-differentiable constraint, and a *final discrete projection* that resolves any residual charge imbalance through minimum-cost element swaps via dynamic programming. The first two components keep the charge residual small, so that the final discrete projection changes less than one element assignment per sample on average.
- We conduct extensive experiments on two amorphous materials datasets across multiple conditioning configurations. Results show that AMGENC guarantees charge balanced generation while

matching or exceeding existing methods in inverse design accuracy, reducing the time to obtain the same amount of charge balanced samples by up to two orders of magnitude.

2 Related Works

Generative inverse design of materials. Traditional materials design [28, 3] involves resource-intensive trial-and-error, where many samples are created in laboratories or simulation environments and have their properties tested. Recent years have witnessed efforts on the generative inverse design approach aiming to be a more efficient alternative, where a conditional generative model directly generates materials with properties closely aligned with the target. Early methods were based on two classic generative models, variational auto-encoder (VAE) [19] and generative adversarial network (GAN) [12]. VAE-based methods [11, 34, 5] have limited effectiveness in generating complex atomic systems [6], while GAN-based methods [29] are hampered by the unstable training of GANs [24]. More recent works based on diffusion models [15, 39] or flow-matching [26] have shown promising results. Most of them focus on relatively small atomic systems, i.e., crystalline materials [48, 53, 40, 51, 30] and molecules [47, 16], while there are only a few studies on amorphous materials [49, 10]. For crystalline materials, several methods first generate or select a charge balanced composition and then generate a structure conditioned on it [9, 50, 4]. This is effective when the composition can be fixed before structure generation, and is complementary to our setting, where element assignments and atom positions evolve jointly in one property-conditioned generation process. Beyond such composition-first pipelines, *enforcing charge balance on generated samples* has not been extensively discussed in existing efforts for inverse design of materials. We believe this is mainly due to the technical challenge of enforcing such non-differentiable constraints, and existing techniques for guiding or constraining generative models are ineffective or not directly applicable in this context.

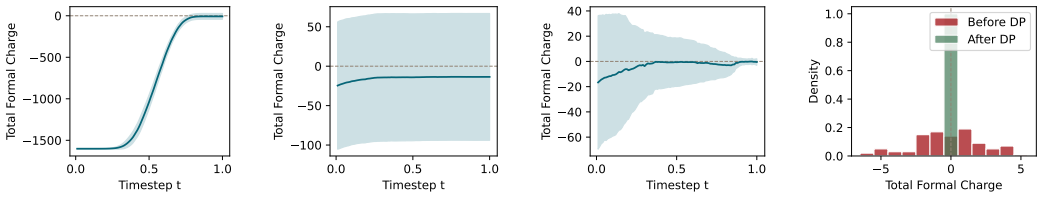
Guiding or constraining generative models. Classifier guidance [7] and classifier-free guidance [14] are two representative techniques for guiding generative models with a preference for the generated data. Classifier guidance requires the classifier to be differentiable. Alternatively, for non-differentiable classifiers or black-box preferences, gradient estimation [1] or stochastic search [17, 52] techniques can be used. Nonetheless, guidance techniques improve the satisfaction of a preference rather than guarantee an exact constraint on the final discrete output, such as the charge balance constraint in our context. Crystal-GFN [32] imposes hard physical and geometrical constraints by restricting the action space of a sequential crystal generator, a generation process different from the joint flow of positions and elements considered here. Physics-Constrained Flow Matching (PCFM) [43] introduces a solution for enforcing hard constraints on flow-matching. It can be applied to generation of continuous data, with a differentiable constraint expressed in an equality form. In our context, however, $\boldsymbol{E}$ is discrete, and the charge balance constraint is non-differentiable, making PCFM not directly applicable.

3 Preliminaries

Definition 1 (Amorphous Material Sample) *An amorphous material sample is represented as the positions and elements of atoms inside a unit cell, formally a tuple $\mathcal{M} = (\boldsymbol{L}, \boldsymbol{X}, \boldsymbol{E})$ of three matrices. $\boldsymbol{L} \in \mathbb{R}^{3\times 3}$ is the lattice of the unit cell, $\boldsymbol{X} \in \mathbb{R}^{n_a \times 3}$ is the positions of n_a atoms, and $\boldsymbol{E} \in \mathbb{R}^{n_a \times d_E}$ is the one-hot embeddings of atomic elements, where d_E is the number of element types under consideration. The set of n_y relevant properties is represented as $\boldsymbol{y} \in \mathbb{R}^{n_y}$, where each value denotes the magnitude of that property.*

Definition 2 (Charge) *Each element type $j \in \{1, \ldots, d_E\}$ has an associated formal charge value $c_j \in \mathbb{Z}$, collected into a vector $\boldsymbol{c} \in \mathbb{Z}^{d_E}$. The total charge of a sample $\mathcal{M}$ is defined as $Q(\boldsymbol{E}) = \mathbf{1}^\top \boldsymbol{E}\, \boldsymbol{c}$, where $\mathbf{1} \in \mathbb{R}^{n_a}$ is the all-ones vector. A sample is said to be charge balanced if $Q(\boldsymbol{E}) = 0$. This formal charge balance does not by itself imply electronic neutrality or local chemical validity.*

Definition 3 (Ghost Atoms) *Ghost atoms are introduced to allow a generative model to control the density of samples without modifying $\boldsymbol{L}$ or the number of atoms. Formally, they are a special atom type incorporated into each sample. In the datasets, ghost atoms are randomly positioned within the cell so that the total number of atoms is $n_a = \lfloor \rho \cdot \mathit{Vol}(\boldsymbol{L}) \rfloor$, where ρ is the maximum density. Ghost*



(a) Standard flow matching (b) Add element noise (c) Add soft projection (d) Add final projection

Figure 1: Illustration of how each component in AMGENC progressively reduces the charge residual during generation. (a)–(c): total charge $Q(\hat{\boldsymbol{E}}_1)$ over generation steps as components are added cumulatively. (d): distribution of total charge before and after the final discrete projection (DP).

atoms are treated like normal atoms by the model but are assigned a special chemical element class with formal charge set to 0. The model adjusts sample density by changing the fraction of ghost atoms, which are removed after generation.

Problem definition. *Inverse design of amorphous materials with charge balance constraint.* Given a set of desired properties $\boldsymbol{y}$, the goal is to learn a conditional probabilistic model $p_\theta(\mathcal{M}|\boldsymbol{y})$ that generates amorphous material samples $\mathcal{M}$ with properties closely aligned with $\boldsymbol{y}$, subject to the constraint that all generated samples are charge balanced, i.e., $Q(\boldsymbol{E}) = 0$.

4 AMGENC Architecture

AMGENC is built on conditional flow-matching [26] (Section 4.1). To enforce charge balance, it introduces three components acting at different stages of the generation pipeline: a *data-centered conditional element noise* (Section 4.2), a *per-step soft linearized projection* (Section 4.3), and a *final discrete projection* (Section 4.4).

4.1 Conditional Flow Matching

AMGENC uses flow-matching with linear interpolation paths. We define time $t \in [0, 1]$, where $t = 0$ corresponds to noise and $t = 1$ to clean data. During the flow, both positions $\boldsymbol{X}$ and element embeddings $\boldsymbol{E}$ evolve in continuous space, while the lattice $\boldsymbol{L}$ is fixed. The flow paths are:

$$\boldsymbol{X}_t = \boldsymbol{X}_0 + t \cdot \boldsymbol{v}_{\boldsymbol{X}} \tag{1}$$

$$\boldsymbol{E}_t = \boldsymbol{E}_0 + t \cdot \boldsymbol{v}_{\boldsymbol{E}} \tag{2}$$

where $\boldsymbol{X}_0$ is sampled uniformly in the unit cell, $\boldsymbol{E}_0$ is a source noise (discussed in Section 4.2), $\boldsymbol{v}_{\boldsymbol{X}}$ is the displacement from $\boldsymbol{X}_0$ to $\boldsymbol{X}_1$ computed under periodic boundary conditions, $\boldsymbol{v}_{\boldsymbol{E}} = \boldsymbol{E}_1 - \boldsymbol{E}_0$, and $\boldsymbol{X}_1$ and $\boldsymbol{E}_1$ are from the training data.

The velocity predictor $\boldsymbol{v}_\theta(\mathcal{M}_t, \boldsymbol{y}, t) = (\boldsymbol{v}_{\boldsymbol{X}}, \boldsymbol{v}_{\boldsymbol{E}})$ is parameterized by an E(n)-equivariant graph neural network (EGNN) [38]. Input node features are the concatenation of the scalar time t, the current element embedding, and a linear projection of the target property vector $\boldsymbol{y}$. Edges connect atom pairs within a cutoff radius r_{cut}, and the network outputs a position velocity $\boldsymbol{v}_{\boldsymbol{X}} \in \mathbb{R}^{n_a \times 3}$ and an element velocity $\boldsymbol{v}_{\boldsymbol{E}} \in \mathbb{R}^{n_a \times d_E}$. Details of the EGNN architecture are provided in Appendix B.

The training objective minimizes the mean squared error between predicted and target velocities:

$$\mathcal{L} = \mathbb{E}_{t,\mathcal{M}} \left[\|\boldsymbol{v}_{\boldsymbol{X},\theta} - \boldsymbol{v}_{\boldsymbol{X}}\|^2 + \|\boldsymbol{v}_{\boldsymbol{E},\theta} - \boldsymbol{v}_{\boldsymbol{E}}\|^2 \right] \tag{3}$$

where $t \sim \text{Uniform}(0, 1)$, $\mathcal{M} \sim p_{\text{data}}$.

During inference, generation proceeds by Euler integration from $t = 0$ to $t = 1$ with T steps of size $\Delta t = 1/T$. At each step, a clean extrapolation predicts the final element state:

$$\hat{\boldsymbol{E}}_1 = \boldsymbol{E}_t + (1 - t) \cdot \boldsymbol{v}_{\boldsymbol{E},\theta}(\mathcal{M}_t, \boldsymbol{y}, t) \tag{4}$$

where $\mathcal{M}_t = (\boldsymbol{L}, \boldsymbol{X}_t, \boldsymbol{E}_t)$ is the intermediate state. The discrete element assignments are obtained from $\hat{\boldsymbol{E}}_1$ via $\arg\max$ at $t = 1$.

In standard flow-matching, no mechanism exists to enforce constraints on the discrete element assignments. The following three subsections address this.

4.2 Data-Centered Conditional Element Noise

In standard flow-matching, element noise is sampled from an isotropic Gaussian $\boldsymbol{E}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. Through $\mathrm{softmax}$, this produces a near-uniform distribution over elements. However, in many material systems, the element distribution is highly imbalanced. For example, in oxide glasses, oxygens typically constitute over 60% of atoms, which becomes the fraction needed to offset the positive charges of other elements to achieve $Q = 0$. The uniform noise distribution reduces oxygen's prevalence from $\sim$60% to $\sim 1/d_E$, creating a large positive charge imbalance at $t = 0$ that the model must spend early trajectory capacity correcting (Figure 1a).

We address this with a *data-centered conditional element noise*. During training, the source noise is centered at the clean element matrix:

$$\boldsymbol{E}_0 = \boldsymbol{E}_1 + \sigma\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{5}$$

where $\sigma > 0$ is a noise scale hyperparameter. This is a conditional coupling where each noise sample is paired with the data point it is centered on, which reduces path crossings similar to optimal transport coupling in flow-matching [42], although no batch transport plan or transport cost is computed.

Over the training set, this induces a Gaussian mixture as the per-atom marginal noise distribution:

$$p(\boldsymbol{E}_{0,i}) = \sum_{k=1}^{d_E} f_k \cdot \mathcal{N}(\boldsymbol{E}_{0,i};\, \boldsymbol{e}_k,\, \sigma^2 \boldsymbol{I}) \tag{6}$$

where f_k is the fraction of atoms assigned to element k in the training set. At inference, since the true elements are unknown, we sample each row of $\boldsymbol{E}_0$ independently from this marginal by drawing $k_i \sim \mathrm{Cat}(f_1, \ldots, f_{d_E})$ and $\boldsymbol{E}_{0,i} \sim \mathcal{N}(\boldsymbol{e}_{k_i}, \sigma^2 \boldsymbol{I})$, matching the training distribution and keeping the source close to charge balance.

The above element noise provides a better starting point (Figure 1b) but does not guarantee charge balance on its own. The following two components actively steer and correct the elements during and after generation.

4.3 Per-Step Soft Linearized Projection

During inference, we apply a projection at each step to steer the predicted $\hat{\boldsymbol{E}}_1$ (Equation 4) toward charge balance. Inspired by the Gauss-Newton projection in PCFM [43], we move the predicted clean elements toward the constraint $Q = 0$ at each inference step with a minimum-norm correction for the linearized constraint. For a scalar constraint like ours, this correction has a simple closed form that finds the smallest change to the element logits needed to satisfy the linearized constraint, making it both efficient and minimally disruptive to the model's predictions (Section 5.5). It does, however, require the constraint to be differentiable. In our setting, charge balance involves discrete element assignments: $Q = \mathbf{1}^\top \mathrm{onehot}(\arg\max(\hat{\boldsymbol{E}}_1))\, \boldsymbol{c}$, which makes Q non-differentiable.

We make two design choices to handle this non-differentiability. First, we relax the $\arg\max$ with $\mathrm{softmax}(\hat{\boldsymbol{E}}_1/\tau)$ at temperature τ, yielding a differentiable soft charge whose gradient can be computed via backpropagation:

$$Q_{\text{soft}} = \mathbf{1}^\top \mathrm{softmax}(\hat{\boldsymbol{E}}_1/\tau)\, \boldsymbol{c} \tag{7}$$

Second, we decouple the charge value from its gradient, because no single τ serves both well: at high τ, Q_{soft} drifts from the true discrete charge Q; at low τ, Q_{soft} is accurate but its gradient vanishes due to softmax saturation, causing the projection to be unstable. We therefore use the exact discrete charge Q as the residual, and compute the gradient $\nabla_{\hat{\boldsymbol{E}}_1} Q_{\text{soft}}$ through $\mathrm{softmax}(\hat{\boldsymbol{E}}_1/\tau)$ at a moderate τ as the Jacobian of the relaxed constraint.

The soft linearized projection then corrects $\hat{\boldsymbol{E}}_1$:

$$\hat{\boldsymbol{E}}_1^{\text{proj}} = \hat{\boldsymbol{E}}_1 - \frac{Q}{\|\nabla_{\hat{\boldsymbol{E}}_1} Q_{\text{soft}}\|^2} \nabla_{\hat{\boldsymbol{E}}_1} Q_{\text{soft}} \tag{8}$$

Since the residual comes from the discrete charge Q and the Jacobian from its relaxation Q_{soft}, this is not an exact Gauss-Newton step, but a minimum-norm correction for the linearized relaxed

constraint. The gradient $\nabla_{\hat{\boldsymbol{E}}_1} Q_{\text{soft}}$ is largest at atoms where the model is least confident about the element assignment, so the correction preferentially adjusts these atoms while leaving confident ones unchanged.

After projection, we re-interpolate along the flow path in Equation 2 with the corrected target:

$$\bar{\boldsymbol{E}}_{t'} = (1 - t')\, \boldsymbol{E}_0 + t'\, \hat{\boldsymbol{E}}_1^{\text{proj}} \tag{9}$$

which gives the corrected starting state $\bar{\boldsymbol{E}}_{t'}$ for the next step at $t' = t + \Delta t$. This per-step correction progressively tightens the charge convergence over the generation trajectory (Figure 1c). The full inference procedure is given in Algorithm 1 in Appendix C.1.

4.4 Final Discrete Projection

After the flow reaches $t = 1$, the element assignments from $\arg\max$ may still have $Q \neq 0$, since both the softmax relaxation and the use of estimated $\hat{\boldsymbol{E}}_1$ at each step can introduce error. To guarantee exact charge balance, we apply a discrete projection as the last step of generation, finding the minimum-cost set of element swaps to achieve $Q = 0$ (Figure 1d).

Given the element logits $\hat{\boldsymbol{E}}_1$ at $t = 1$ and current element assignments $e_i = \arg\max_j \hat{\boldsymbol{E}}_{1,ij}$ of the i-th atom, we solve:

$$\min_{\{e'_i\}_{i=1}^{n_a}} \sum_{i=1}^{n_a} \left(\hat{\boldsymbol{E}}_{1,i,e_i} - \hat{\boldsymbol{E}}_{1,i,e'_i} \right) \quad \text{s.t.} \quad \sum_{i=1}^{n_a} c_{e'_i} = 0 \tag{10}$$

where c_j is the formal charge of element j. The cost $\hat{\boldsymbol{E}}_{1,i,e_i} - \hat{\boldsymbol{E}}_{1,i,e'_i}$ is the logit gap sacrificed by swapping atom i from e_i to e'_i, ensuring that atoms where the model is most confident are swapped last.

Since formal charges are integers, the charge residual Q lives in a bounded integer range, making it suitable as a state variable for dynamic programming. Our implementation of the algorithm (detailed in Algorithm 2 in Appendix C.2) processes atoms one by one, considering all possible element assignments at each atom while tracking the accumulated charge delta and total swap cost, to find the assignment that reaches $Q = 0$ with minimum cost.

The complexity is $O(n_a \cdot |Q_{\text{max}}| \cdot d_E)$, where $|Q_{\text{max}}|$ is the range of achievable charge values. In practice, after the soft linearized projection has already reduced the charge residual to a small range, the algorithm takes approximately 1% of total inference time while guaranteeing $Q = 0$ for every generated sample (Section 5.5).

5 Experiments

We evaluate the inverse design performance of AMGENC on two amorphous material datasets, against several existing generative inverse design methods for materials.

5.1 Datasets

The two datasets [10] are obtained using classical molecular dynamics simulation workflows based on LAMMPS [41] and ASE [22]. First, the **amorphous silica (a-SiO_2) dataset**, which contains silica (SiO_2) samples that vary in size (between 80 and 250 atoms) and whose properties depend on structures and densities, since the composition is fixed. Second, the **multi-element glass (MEG) dataset**, which consists of 9,027 glass samples, each containing approximately 800 atoms across 11 different elements. Initial structures were generated from varying compositions of the glass network formers SiO_2 and P_2O_5, and the network modifiers Al_2O_3, Li_2O, BeO, K_2O, CaO, TiO_2, BaO and ZnO. More details about the preparation of the datasets are provided in Appendix D.

5.2 Comparison Methods

We compare AMGENC with several generative methods for inverse design of materials. **CDVAE** [48] is a diffusion-based VAE that generates stable crystalline materials by denoising atomic coordinates

and types through Langevin dynamics. **CrystalFlow** [30] is a flow-based model that generates crystalline materials by learning continuous normalizing flows with conditional flow matching. **MatterGen** [53] is a diffusion model that generates crystalline materials across the periodic table and supports fine-tuning for property-guided generation. **Graphite** [21] is a diffusion model that generates amorphous materials using XANES spectroscopy as conditioning for inverse structure prediction. **AMDEN** [10] is a diffusion model that generates amorphous materials with optimization for low-energy structures by incorporating relaxation into the diffusion generation process. All baselines are retrained on the same datasets and conditioned on the same target properties as AMGENC, with the adaptation of each baseline detailed in Appendix I.

5.3 Experimental Settings

We evaluate the models' inverse design performance by performing generation conditioned on specific target properties, and compare the actual properties of generated samples with the target. We also evaluate the charge balance of generated samples under each configuration, as well as the computational efficiency implication. Models are trained with the Adam optimizer (learning rate 10^{-3}) for 200 epochs with batch size 8. All methods use 100 sampling steps during inference for a fair comparison. Each experiment is repeated 5 times, and the mean and standard deviation of each metric are reported. Details of the computational environment are provided in Appendix J.

Settings for the a-SiO_2 dataset. For the a-SiO_2 dataset, we are interested in the following two properties of samples. Shear modulus G characterizes a material's resistance to elastic deformation under shear stress, representing mechanical stiffness. Ring size distribution (RSD) quantifies the medium-range order in amorphous silica by measuring the average number of Si atoms in rings formed by the Si-O network. We perform generation conditioned on either one of them, denoted as a-SiO_2 | G and a-SiO_2 | RSD respectively, with target shear modulus linearly interpolated between 10 and 50 [GPa] and target RSD linearly interpolated between 4 and 6 atoms, both across 2,000 samples with cubic cells with edge length of 18 Å, and $\rho = 0.11$ atoms/Å^3. The formal charge values are $c_{\mathrm{Si}} = +4$ and $c_{\mathrm{O}} = -2$, with ghost atoms assigned $c = 0$, so $Q = 0$ is equivalent to the exact SiO_2 stoichiometry $N_{\mathrm{O}} = 2N_{\mathrm{Si}}$ after removing ghost atoms. Although a-SiO_2 has a fixed composition, the two target properties are closely related to sample density, which is controlled via ghost atoms. The element configuration thus remains unfixed, and charge balance enforcement is still necessary.

Settings for the MEG dataset. For the MEG dataset, we are interested in the following three properties of the samples. In addition to shear modulus G, Young's modulus E can be used to characterize elastic properties. The lithium molar concentration C_{Li} quantifies the composition ratio, which is important for applications related to, e.g., glassy solid electrolytes for batteries [8, 18]. We perform generation conditioned on the combination of Young's modulus and lithium molar concentration, or on the shear modulus, denoted as MEG | E+C_{Li} and MEG | G respectively. The target Young's modulus is linearly interpolated between 20 and 160 GPa, the target lithium molar concentration is fixed at 15%, and the target shear modulus is linearly interpolated between 10 and 70 GPa, all across 2,000 samples with cubic cells with edge length of 23 Å, and $\rho = 0.11$ atoms/Å^3. The formal charge values for the 11 element types are derived from the BMP potential [2]. The full assignment is provided in Appendix E.1. Under these formal charge values, $Q = 0$ is equivalent to a sample containing exactly the oxygen needed to form neutral oxide units with its cations, while the cation ratios remain free for the property-conditioned generation to steer, e.g., toward the target C_{Li} (Appendix E.3).

For both datasets, all training samples are charge balanced (Appendix D.3), and the target ranges extend beyond the training data distributions, requiring the model to extrapolate (Appendix F.3).

Evaluation metrics. We evaluate the inverse design performance by comparing target properties with properties of generated samples using three regression metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). Implementation details are provided in Appendix F.2. Detailed procedures for calculating the properties from generated samples are provided in Appendix F.1. For charge balance, we report three metrics over generated samples: the probability $P(Q{=}0)$ of charge balance, measured as the percentage of samples satisfying $Q(\boldsymbol{E}) = 0$, and the mean absolute charge $|\bar{Q}|$ and standard deviation σ_Q of the charge value $Q(\boldsymbol{E})$ (definitions in Appendix E.2). For computational efficiency, we report the wall-clock

Table 1: Inverse design performance comparison across conditioning configurations. Lower is better for all metrics. **Bold** and underline indicate the best and second best results, respectively.

Config	Metric	CDVAE	CrystalFlow	MatterGen	Graphite	AMDEN	AMGENC
a-SiO_2 | G	MAE ↓	4.25±0.14	3.96±0.11	3.89±0.12	3.58±0.08	3.43±0.11	**3.32±0.10**
	RMSE ↓	5.18±0.16	4.81±0.15	4.72±0.14	4.35±0.10	4.21±0.12	**4.19±0.13**
	MAPE ↓	15.8±0.7%	15.1±0.6%	14.9±0.5%	14.3±0.3%	15.2±0.6%	**14.1±0.4%**
a-SiO_2 | RSD	MAE ↓	0.22±0.01	0.20±0.02	0.19±0.02	0.17±0.01	0.17±0.01	**0.16±0.01**
	RMSE ↓	0.27±0.02	0.25±0.01	0.24±0.02	0.21±0.01	0.21±0.01	**0.20±0.01**
	MAPE ↓	4.5±0.3%	4.0±0.2%	3.9±0.3%	3.6±0.1%	3.5±0.2%	**3.4±0.2%**
MEG | E+C_{Li}	E MAE ↓	13.02±0.38	11.65±0.29	11.32±0.33	10.15±0.21	**9.85±0.27**	9.92±0.26
	E RMSE ↓	15.88±0.45	14.23±0.32	13.82±0.40	12.38±0.26	12.02±0.34	**11.82±0.29**
	E MAPE ↓	26.3±0.9%	23.6±0.6%	22.9±0.8%	20.5±0.5%	**19.9±0.6%**	20.1±0.5%
	C_{Li} MAE ↓	2.02±0.05	1.80±0.04	1.75±0.06	1.56±0.03	1.52±0.04	**1.49±0.04**
	C_{Li} RMSE ↓	2.45±0.06	2.19±0.04	2.12±0.07	1.90±0.04	1.85±0.05	**1.82±0.04**
	C_{Li} MAPE ↓	13.3±0.4%	11.9±0.3%	11.6±0.5%	10.4±0.2%	10.1±0.3%	**9.8±0.3%**
MEG | G	MAE ↓	7.08±0.21	6.34±0.15	6.15±0.18	5.52±0.12	5.35±0.15	**5.32±0.14**
	RMSE ↓	8.75±0.25	7.85±0.19	7.62±0.22	6.82±0.15	6.62±0.19	**6.48±0.17**
	MAPE ↓	17.8±0.7%	16.0±0.4%	15.5±0.5%	13.9±0.3%	13.5±0.5%	**13.3±0.4%**

time $T_{1\mathrm{k}}$ to generate 1,000 samples, and the time $T_{1\mathrm{k}}^{Q=0}$ to generate 1,000 charge-balanced samples, which accounts for the need to discard and regenerate samples that violate charge balance.

5.4 Comparison of Methods

Inverse design metrics comparison. Table 1 reports the inverse design performance across all conditioning configurations. AMGENC achieves the best or second best result on every metric in all four configurations, ranking first on all metrics on the a-SiO_2 dataset and trading the top position with AMDEN on the MEG dataset. Since none of the baselines enforce charge balance, the comparable inverse design performance of AMGENC suggests that its components for enforcing charge balance do not sacrifice inverse design accuracy. We further compare several structural features of the generated samples with those of the training samples in Appendix G.

Charge balance comparison. Table 2 reports the charge balance metrics. AMGENC guarantees the generation of charge balanced samples across all configurations. All baseline methods, where element assignments are unconstrained, produce charge balanced samples at low rates that decrease as the number of atoms and element types grows. On the a-SiO_2 dataset with two element types and 80 to 250 atoms, the best baseline reaches at most 6.8% $P(Q{=}0)$. On the MEG dataset with 11 element types and approximately 800 atoms, this drops to at most 2.0%. This trend suggests that for larger amorphous material systems, post-hoc filtering alone would discard more generated samples. As none of the baselines enforce charge balance, this comparison quantifies how the cost of leaving the constraint unenforced grows with system scale.

Computation efficiency comparison. Table 3 reports the computation efficiency. While the generation time $T_{1\mathrm{k}}$ is comparable across methods, the time to generate 1,000 charge balanced samples ($T_{1\mathrm{k}}^{Q=0}$) reveals a large gap. Baselines that rely on post-hoc filtering must generate and discard far more samples due to their low $P(Q{=}0)$. On the MEG dataset, where baselines discard over 98% of generated samples, the best baseline requires approximately 39 to 49 hours, while AMGENC needs approximately 47 minutes. This gap would widen for larger systems where $P(Q{=}0)$ is expected to be lower.

5.5 Model Analysis

We provide insights into the effectiveness of design choices and hyperparameter settings in AMGENC with the following model analysis experiments.

Table 2: Charge balance comparison across conditioning configurations. Higher $P(Q{=}0)$ and lower $|\bar{Q}|$, σ_Q are better. **Bold** and underline indicate the best and second best results, respectively.

Config	Metric	CDVAE	CrystalFlow	MatterGen	Graphite	AMDEN	AMGENC
a-SiO_2 | G	$P(Q{=}0)\uparrow$	3.6±0.5%	3.1±0.4%	4.2±0.6%	5.2±0.8%	5.5±0.8%	**100.0±0.0%**
	$\lvert\bar{Q}\rvert\downarrow$	2.93±0.41	3.28±0.48	2.26±0.38	0.98±0.17	1.15±0.22	**0.00±0.00**
	$\sigma_Q\downarrow$	10.47±1.23	11.35±1.42	9.03±0.95	7.52±0.84	7.31±0.76	**0.00±0.00**
a-SiO_2 | RSD	$P(Q{=}0)\uparrow$	4.1±0.5%	3.5±0.6%	4.8±0.6%	6.8±0.9%	6.4±0.7%	**100.0±0.0%**
	$\lvert\bar{Q}\rvert\downarrow$	6.24±0.73	6.92±0.78	5.38±0.64	3.92±0.48	3.73±0.45	**0.00±0.00**
	$\sigma_Q\downarrow$	8.15±0.92	8.83±1.08	6.92±0.81	4.93±0.58	4.67±0.53	**0.00±0.00**
MEG | E+C_{Li}	$P(Q{=}0)\uparrow$	1.0±0.2%	0.8±0.1%	1.1±0.3%	1.4±0.3%	1.6±0.4%	**100.0±0.0%**
	$\lvert\bar{Q}\rvert\downarrow$	26.58±3.84	38.15±5.45	24.73±3.12	21.06±2.85	18.84±2.51	**0.00±0.00**
	$\sigma_Q\downarrow$	78.23±6.53	82.47±5.81	66.48±5.82	49.18±4.15	50.95±4.38	**0.00±0.00**
MEG | G	$P(Q{=}0)\uparrow$	1.3±0.3%	1.0±0.2%	1.4±0.3%	1.8±0.4%	2.0±0.4%	**100.0±0.0%**
	$\lvert\bar{Q}\rvert\downarrow$	16.47±2.83	17.93±2.42	16.82±2.25	11.37±1.68	11.91±1.76	**0.00±0.00**
	$\sigma_Q\downarrow$	67.14±5.72	72.38±5.13	57.86±5.04	46.53±3.91	45.22±3.85	**0.00±0.00**

Table 3: Computation efficiency comparison across conditioning configurations. Lower is better for all metrics. **Bold** and underline indicate the best and second best results, respectively.

Config	Metric	CDVAE	CrystalFlow	MatterGen	Graphite	AMDEN	AMGENC
a-SiO_2 | G	$T_{1\mathrm{k}}\downarrow$	26.37±0.45m	28.42±0.47m	31.89±0.58m	23.74±0.31m	**22.93±0.38m**	23.68±0.53m
	$T_{1\mathrm{k}}^{Q=0}\downarrow$	12.21±1.71h	15.28±2.63h	12.65±1.82h	7.61±1.17h	6.95±1.02h	**23.68±0.53m**
a-SiO_2 | RSD	$T_{1\mathrm{k}}\downarrow$	26.50±0.39m	28.65±0.54m	32.27±0.61m	23.76±0.32m	**23.25±0.40m**	24.08±0.65m
	$T_{1\mathrm{k}}^{Q=0}\downarrow$	10.77±1.32h	13.64±2.08h	11.20±1.42h	5.82±0.77h	6.05±0.67h	**24.08±0.65m**
MEG | E+C_{Li}	$T_{1\mathrm{k}}\downarrow$	53.10±0.91m	57.38±0.82m	64.23±1.12m	47.82±0.65m	**46.82±0.78m**	47.44±0.88m
	$T_{1\mathrm{k}}^{Q=0}\downarrow$	88.50±17.70h	119.54±29.89h	97.32±26.60h	56.93±12.22h	48.77±12.22h	**47.44±0.88m**
MEG | G	$T_{1\mathrm{k}}\downarrow$	52.87±0.87m	57.15±0.78m	63.96±1.08m	47.62±0.63m	**46.62±0.74m**	47.13±0.86m
	$T_{1\mathrm{k}}^{Q=0}\downarrow$	67.78±15.64h	95.25±19.05h	76.14±16.37h	44.09±9.82h	38.85±7.79h	**47.13±0.86m**

Effectiveness of components. We evaluate the contribution of each component by cumulatively adding them to a base flow matching model on the MEG | E+C_{Li} configuration. Starting from *Base*, a standard flow matching model without any charge balance components (Section 4.1), we progressively add data-centered conditional element noise (Section 4.2) as *+Noise*, then per-step soft linearized projection (Section 4.3) as *+Proj.*, and finally the discrete projection (Section 4.4) to arrive at full AMGENC. Table 4 reports the results. Figure 1 further visualizes the effect of each component, where (a) Base, (b) +Noise, and (c) +Proj. correspond to the cumulative addition of components. Adding the data-centered conditional element noise (+Noise) reduces $|\bar{Q}|$ substantially by giving the generation a starting point closer to charge balance, but without active correction during generation, the charge spread across samples remains large. The soft linearized projection (+Proj.) addresses this by steering elements toward charge balance at each generation step, as visible in the progressively tighter convergence in Figures 1a–1c. Notably, +Proj. also improves inverse design metrics, suggesting that steering element assignments toward charge balance also aligns them more closely with the training data distribution, where all samples are charge balanced. The final discrete projection brings $P(Q{=}0)$ to 100% with negligible change in inverse design metrics, confirming that the residual charge after +Proj. is small enough to be resolved with minimal element swaps (Figure 1d).

We further study the sensitivity of two critical hyperparameters in these components, the element noise scale σ and softmax temperature τ, in Appendix A.

Contribution of components beyond the final discrete projection. Since the final discrete projection (DP) alone guarantees $P(Q{=}0) = 100\%$, we also apply it directly to Base and +Noise, denoted *Base+DP* and *+Noise+DP* (Table 5). Base+DP changes 14.73 element assignments per sample and degrades the inverse design and structural metrics relative to Base. The data-centered

Table 4: Ablation study on the MEG | E+C_{Li} configuration. **Bold** and underline indicate the best and second best results, respectively.

Category	Metric	Base	+Noise	+Proj.	AMGENC
Inverse Design	E MAE ↓	10.68±0.30	10.42±0.31	9.95±0.27	**9.92±0.26**
	E RMSE ↓	13.02±0.35	12.72±0.36	**11.80±0.30**	11.82±0.29
	E MAPE ↓	21.6±0.6%	21.1±0.6%	20.2±0.5%	**20.1±0.5%**
	C_{Li} MAE ↓	1.60±0.04	1.56±0.05	1.51±0.04	**1.49±0.04**
	C_{Li} RMSE ↓	1.95±0.05	1.91±0.06	1.83±0.05	**1.82±0.04**
	C_{Li} MAPE ↓	10.7±0.4%	10.4±0.4%	9.9±0.3%	**9.8±0.3%**
Charge Balance	$P(Q{=}0)$ ↑	1.4±0.4%	2.1±0.6%	16.2±2.5%	**100.0±0.0%**
	$\lvert\bar{Q}\rvert$ ↓	34.62±5.15	13.57±2.08	0.57±0.09	**0.00±0.00**
	σ_Q ↓	61.62±5.84	80.40±7.23	2.41±0.38	**0.00±0.00**
Cost	$T_{1\text{k}}$ ↓	**47.02±0.67m**	**47.02±0.67m**	47.19±0.68m	47.44±0.88m
	$T_{1\text{k}}^{Q=0}$ ↓	55.98±16.01h	37.32±10.68h	4.85±0.75h	**47.44±0.88m**

Table 5: Contribution of components beyond the final discrete projection (DP) on MEG | E+C_{Li}. n_{DP}: mean number of element assignments changed by DP per sample. JSD: Jensen-Shannon divergence of partial radial distribution functions. CN MAE: MAE of first-shell coordination numbers (Appendix H). **Bold** and underline indicate the best and second best results.

Metric	Base	Base+DP	+Noise+DP	+Proj.	AMGENC
E MAPE ↓	21.6%	23.9%	22.5%	20.2%	**20.1%**
C_{Li} MAPE ↓	10.7%	11.7%	11.0%	9.9%	**9.8%**
$P(Q{=}0)$ ↑	1.4%	**100.0%**	**100.0%**	16.2%	**100.0%**
n_{DP} ↓	–	14.73	9.21	–	**0.81**
JSD ↓	0.01475	0.01578	0.01527	**0.01415**	0.01418
CN MAE ↓	0.04594	0.04899	0.04769	**0.04471**	0.04482

conditional element noise and the soft linearized projection reduce this number to 9.21 and 0.81, while AMGENC matches the metrics of +Proj. Thus, DP guarantees exact charge balance, while the preceding components reduce the charge residual enough for DP to preserve these metrics.

Effect of the final discrete projection. Across the four configurations, DP changes 0.59 to 0.81 element assignments per sample on average, and at most four in any sample (Appendix H). With atom positions unchanged, DP changes the structural metrics by at most 0.25%, and on MEG | E+C_{Li} the mean atomic percentage of each element by at most 0.01 percentage points.

Computational overhead of components. As shown in Table 4, the data-centered conditional element noise adds no generation time since it only affects the initial noise sampling, and the soft linearized projection and DP together add less than 1% to $T_{1\text{k}}$. Meanwhile, $T_{1\text{k}}^{Q=0}$ drops from 55.98 hours to 47.44 minutes as each component improves $P(Q{=}0)$.

6 Conclusion

This work introduces AMGENC, a generative inverse design method for amorphous materials that guarantees the generation of formally charge balanced samples. Built on flow matching, AMGENC combines a data-centered conditional element noise that starts the generation closer to charge balance, a per-step soft linearized projection that steers element assignments toward charge balance via a relaxed constraint, and a final discrete projection that guarantees exact charge balance through minimum-cost element swaps. Experiments on two amorphous material datasets provide evidence that AMGENC achieves its design goals without compromising inverse design performance or adding significant computational overhead. Broader impacts and limitations are discussed in Appendix K.

Acknowledgments and Disclosure of Funding

We thank Jonas A. Finkler (Aalborg University) and Tao Du (The Hong Kong Polytechnic University) for their contributions to datasets and implementation code in our prior joint works [10, 25] that are partially reused in this work, and, together with Suresh Bishnoi (Aalborg University), for their involvement in the preliminary discussions leading to this work.

This work was supported by a research grant (VIL57373) from VILLUM FONDEN. The authors declare no competing interests.

References

[1] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *CVPR*, pages 843–852, 2023.

[2] Marco Bertani, Maria Cristina Menziani, and Alfonso Pedone. Improved empirical force field for multicomponent oxide glasses and crystals. *Phys. Rev. Mater.*, 5(4):045602, 2021.

[3] Jiazhen Cai, Xuan Chu, Kun Xu, Hongbo Li, and Jing Wei. Machine learning-driven new material discovery. *Nanoscale Adv.*, 2(8):3115–3130, 2020.

[4] Mouyang Cheng, Weiliang Luo, Hao Tang, Bowen Yu, Yongqiang Cheng, Weiwei Xie, Ju Li, Heather J. Kulik, and Mingda Li. Enhancing materials discovery with valence-constrained design in generative modeling. *Nat. Comput. Sci.*, 6(10):1114–1123, 2026.

[5] Callum J Court, Batuhan Yildirim, Apoorv Jain, and Jacqueline M Cole. 3-d inorganic crystal structure generation and property prediction via representation learning. *J. Chem. Inf. Model.*, 60(10):4518–4535, 2020.

[6] Imant Daunhawer, Thomas M. Sutter, Kieran Chin-Cheong, Emanuele Palumbo, and Julia E. Vogt. On the limitations of multimodal vaes. In *ICLR*, 2022.

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 34:8780–8794, 2021.

[8] Junwei Ding, Dongfang Ji, Yuanzheng Yue, and Morten M Smedskjaer. Amorphous materials for lithium-ion and post-lithium-ion batteries. *Small*, 20(5):2304270, 2024.

[9] Rongzhi Dong, Nihang Fu, Edirisuriya M. D. Siriwardane, and Jianjun Hu. Generative design of inorganic compounds using deep diffusion language models. *J. Phys. Chem. A*, 128(29): 5980–5989, 2024.

[10] Jonas A. Finkler, Yan Lin, Tao Du, Jilin Hu, and Morten M. Smedskjaer. Inverse design of amorphous materials with targeted properties. *arXiv preprint arXiv:2509.13916*, 2025.

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

[12] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[13] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

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

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

[16] Emiel Hoogeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *ICML*, volume 162, pages 8867–8887, 2022.

[17] Yujia Huang, Adishree Ghatare, Yuanzhe Liu, Ziniu Hu, Qinsheng Zhang, Chandramouli S Sastry, Siddharth Gururani, Sageev Oore, and Yisong Yue. Symbolic music generation with non-differentiable rule guided diffusion. *arXiv preprint arXiv:2402.14285*, 2024.

[18] Sergiy Kalnaus, Nancy J Dudney, Andrew S Westover, Erik Herbert, and Steve Hackney. Solid-state batteries: The critical role of mechanics. *Science*, 381(6664):eabg5998, 2023.

[19] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[20] Gregory M Kurtzer, Vanessa Sochat, and Michael W Bauer. Singularity: Scientific containers for mobility of compute. *PLoS ONE*, 12(5):e0177459, 2017.

[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. *Mach. Learn.: Sci. Technol.*, 5(4):045037, 2024.

[22] Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, et al. The atomic simulation environment—a python library for working with atoms. *J. Phys.: Condens. Matter*, 29(27):273002, 2017.

[23] 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.

[24] Jerry Li, Aleksander Madry, John Peebles, and Ludwig Schmidt. On the limitations of first-order approximation in gan dynamics. In *ICML*, pages 3005–3013, 2018.

[25] Yan Lin, Jonas A Finkler, Tao Du, Jilin Hu, and Morten M Smedskjaer. Amshortcut: An inference-and training-efficient inverse design model for amorphous materials. *arXiv preprint arXiv:2603.29812*, 2026.

[26] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

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

[28] Yue Liu, Tianlu Zhao, Wangwei Ju, and Siqi Shi. Materials discovery and design using machine learning. *J. Materiomics*, 3(3):159–177, 2017.

[29] 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 Comput. Mater.*, 7(1):66, 2021.

[30] Xiaoshan Luo, Zhenyu Wang, Qingchang Wang, Xuechen Shao, Jian Lv, Lei Wang, Yanchao Wang, and Yanming Ma. Crystalflow: A flow-based generative model for crystalline materials. *Nat. Commun.*, 16(1):9267, 2025.

[31] RL McGreevy and L Pusztai. Reverse monte carlo simulation: A new technique for the determination of disordered structures. *Mol. Simul.*, 1(6):359–367, 1988.

[32] Mila AI4Science, Alex Hernandez-Garcia, Alexandre Duval, Alexandra Volokhova, Yoshua Bengio, Divya Sharma, Pierre Luc Carrier, Yasmine Benabed, Michał Koziarski, Victor Schmidt, Gian-Marco Rignanese, Pierre-Paul De Breuck, and Paulette Clancy. Crystal-gfn: Sampling crystals with desirable properties and constraints. *arXiv preprint arXiv:2310.04925*, 2023.

[33] Shinji Munetoh, Teruaki Motooka, Koji Moriguchi, and Akira Shintani. Interatomic potential for si–o systems using tersoff parameterization. *Comput. Mater. Sci.*, 39(2):334–339, 2007.

[34] 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.

[35] Shyue Ping Ong, William Davidson Richards, Anubhav Jain, Geoffroy Hautier, Michael Kocher, Shreyas Cholia, Dan Gunter, Vincent L Chevrier, Kristin A Persson, and Gerbrand Ceder. Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. *Comput. Mater. Sci.*, 68:314–319, 2013.

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019.

[37] Tanguy Rouxel. Elastic properties and short-to medium-range order in glasses. *J. Am. Ceram. Soc.*, 90(10):3019–3039, 2007.

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

[39] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[40] Anuroop Sriram, Benjamin K Miller, Ricky T Chen, and Brandon M Wood. Flowllm: Flow matching for material generation with large language models as base distributions. *NeurIPS*, 37:46025–46046, 2024.

[41] 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. *Comput. Phys. Commun.*, 271:108171, 2022.

[42] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Trans. Mach. Learn. Res.*, 2024.

[43] Utkarsh Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gomez-Bombarelli, and Christopher Vincent Rackauckas. Physics-constrained flow matching: Sampling generative models with hard constraints. In *NeurIPS*, 2025.

[44] 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.

[45] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nat. Methods*, 17(3):261–272, 2020.

[46] Adrian C. Wright. The comparison of molecular dynamics simulations with diffraction experiments. *J. Non-Cryst. Solids*, 159(3):264–268, 1993.

[47] Lemeng Wu, Chengyue Gong, Xingchao Liu, Mao Ye, and Qiang Liu. Diffusion-based molecule generation with informative prior bridges. In *NeurIPS*, 2022.

[48] Tian Xie, Xiang Fu, Octavian-Eugen Ganea, Regina Barzilay, and Tommi S. Jaakkola. Crystal diffusion variational autoencoder for periodic material generation. In *ICLR*, 2022.

[49] Kai Yang and Daniel Schwalbe-Koda. A generative diffusion model for amorphous materials. *npj Comput. Mater.*, 2025.

[50] Sherry Yang, Simon Batzner, Ruiqi Gao, Muratahan Aykol, Alexander L. Gaunt, Brendan McMorrow, Danilo J. Rezende, Dale Schuurmans, Igor Mordatch, and Ekin D. Cubuk. Generative hierarchical materials search. In *NeurIPS*, volume 37, pages 38799–38819, 2024.

[51] Cai-Yuan Ye, Hong-Ming Weng, and Quan-Sheng Wu. Con-cdvae: A method for the conditional generation of crystal structures. *Comput. Mater. Today*, 1:100003, 2024.

[52] Po-Hung Yeh, Kuang-Huei Lee, and Jun-Cheng Chen. Training-free diffusion model alignment with sampling demons. *arXiv preprint arXiv:2410.05760*, 2024.

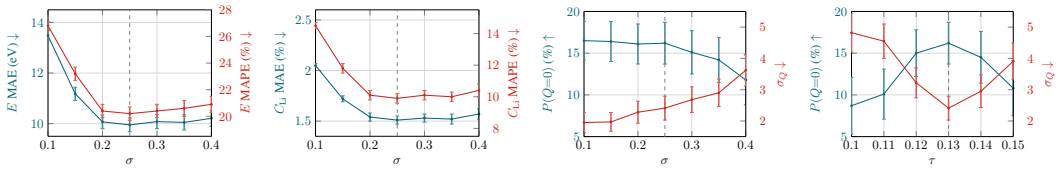
[53] Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Zilong Wang, Aliaksandra Shysheya, Jonathan Crabbé, Shoko Ueda, et al. A generative model for inorganic materials design. *Nature*, pages 1–3, 2025.

A Charge Balance Component Hyperparameter Sensitivity

We study the sensitivity of two critical hyperparameters introduced by the charge balance components on the MEG | E+C_{Li} configuration: the element noise scale σ of the data-centered conditional element noise (Section 4.2) and the softmax temperature τ of the soft linearized projection (Section 4.3). We report $P(Q{=}0)$ and σ_Q on the +Proj. variant, since with the final discrete projection, full AMGENC always achieves $P(Q{=}0) = 100\%$, which would mask the effect of these hyperparameters. Figure 2 summarizes the results.

For σ (Figures 2a–2c), when it is too small, the element noise (Equation 5) concentrates tightly around the one-hot vectors, reducing the diversity of flow paths and degrading inverse design MAE and MAPE. Increasing σ to 0.25 steadily improves both E and C_{Li} MAE and MAPE. Beyond 0.25, the noise overwhelms the one-hot centering and approaches the isotropic Gaussian of standard flow-matching, causing $P(Q{=}0)$ to drop and σ_Q to rise, while degrading MAE and MAPE slightly.

For τ (Figure 2d), when it is too small, $\mathrm{softmax}(\hat{\boldsymbol{E}}_1/\tau)$ saturates and its gradient vanishes, making the soft linearized projection (Equation 8) unstable; when it is too large, Q_{soft} deviates from the discrete charge Q, reducing projection accuracy. The optimum $\tau{=}0.13$ balances both effects.



(a) E MAE, MAPE vs. σ (b) C_{Li} MAE, MAPE vs. σ (c) $P(Q{=}0)$, σ_Q vs. σ (d) $P(Q{=}0)$, σ_Q vs. τ

Figure 2: Hyperparameter sensitivity on MEG | E+C_{Li}.

B EGNN Architecture

Given an input sample $\mathcal{M} = (\boldsymbol{L}, \boldsymbol{X}, \boldsymbol{E})$, the input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to the EGNN is composed of atoms in the sample as nodes in the node set $\mathcal{V}$, and each edge in the edge set $\mathcal{E}$ connects a pair of atoms with distances less than a cutoff radius r_{cut}. The distances are computed with periodic boundary conditions.

The EGNN is composed of L equivariant layers. The l-th layer takes as input: 1) node features $\boldsymbol{H}^{(l)} \in \mathbb{R}^{n_a \times d_h}$ containing information about the corresponding atoms at the l-th layer; 2) positional coordinates $\boldsymbol{X}^{(l)} \in \mathbb{R}^{n_a \times k \times 3}$ of the atoms, where k is the number of vector channels [23]; and 3) edge set $\mathcal{E}$ of the graph $\mathcal{G}$, with edge attributes $\boldsymbol{e}_{ij}$ assigned to each edge.

For the initial layer, the node features $\boldsymbol{H}_i^{(0)}$ are assembled by concatenating: 1) the time step t; 2) the element embedding $\boldsymbol{E}_i \in \mathbb{R}^{d_E}$; and 3) a linear projection of the target property vector $\boldsymbol{y}$. The positions $\boldsymbol{X}^{(0)}$ are the original positions $\boldsymbol{X}$ replicated across k channels. The edge attributes $\boldsymbol{e}_{ij}$ are 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 updates the node features and positional coordinates, incorporating self-attention [44]:

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

where $N(i)$ represents the neighbors of atom i derived from the edge set $\mathcal{E}$, σ is the sigmoid activation function for self-attention, and n_{norm} is a normalization factor to ensure numerical stability. $\phi_e^{(l)}$ and $\phi_H^{(l)}$ are multi-layer perceptrons (MLPs) with SiLU activation functions and layer normalization, structured as:

$$
\begin{aligned}
\phi_e^{(l)}(\boldsymbol{H}_i, \boldsymbol{H}_j, \boldsymbol{e}_{ij}) &= \text{MLP}_{\text{edge}}([\boldsymbol{H}_i, \boldsymbol{H}_j, \boldsymbol{e}_{ij}]) \\
\phi_H^{(l)}(\boldsymbol{H}_i, \mathbf{m}_{\text{agg}}) &= \text{MLP}_{\text{node}}([\boldsymbol{H}_i, \mathbf{m}_{\text{agg}}])
\end{aligned}
\tag{13}
$$

$\boldsymbol{\Phi}_{ij}^{(l)}$ is a learned transformation matrix that maps between the k vector channels.

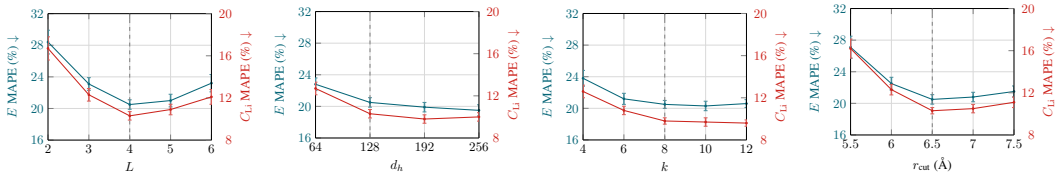
A smooth cutoff function is used to prevent discontinuities when atoms leave or enter the cutoff radius:

$$
f_{\text{cut}}(r) = 2 \tanh \left(1 - \frac{\min(r, r_{\text{cut}})}{r_{\text{cut}}} \right)^2 \tag{14}
$$

At the last layer, the EGNN outputs $\boldsymbol{H}^{(L)}$ and $\boldsymbol{X}^{(L)}$ as the final node features and positional coordinates, respectively. The element velocity $\boldsymbol{v_E}$ is taken directly from $\boldsymbol{H}^{(L)}$, and the position velocity $\boldsymbol{v_X}$ is computed as the deviation between the original positions and the first channel of the output positions, scaled by a learnable scalar parameter.

Parameter choices. We use $L = 4$ EGNN layers with hidden dimension $d_h = 128$, $k = 8$ vector channels, cutoff radius $r_{\text{cut}} = 6.5$ Å, and normalization factor $n_{\text{norm}} = 40$. The self-attention hidden dimension is 128. Among these, L, d_h, and k all govern model capacity. Increasing any of them beyond the optimum yields marginal accuracy improvement while reducing computational efficiency. L additionally controls the receptive depth over the material graph. Too few layers limit long-range information propagation, while too many risk over-smoothing. The cutoff radius $r_{\text{cut}} = 6.5$ Å is chosen to cover typical nearest- and second-nearest-neighbor shells in the amorphous structures, and $n_{\text{norm}} = 40$ is set to the average number of neighbors within r_{cut} to stabilize the aggregation magnitudes. Figure 3 empirically confirms these choices.

Parameter sensitivity. Figure 3 shows the effect of each architectural hyperparameter on E and C_{Li} MAPE, evaluated on the MEG | E+C_{Li} configuration. L exhibits a steep improvement from 2 to 4 layers, followed by degradation at deeper settings consistent with over-smoothing. d_h shows a large jump from 64 to 128 with marginal returns thereafter, thus we select d_h=128 as the most efficient operating point; k follows a similar pattern, where we select k=8 as the optimum. r_{cut} displays a clear U-shape centered at 6.5 Å: too small a radius misses relevant interactions, while too large a radius introduces noisy long-range edges.



(a) MAPE vs. L (b) MAPE vs. d_h (c) MAPE vs. k (d) MAPE vs. r_{cut}

Figure 3: EGNN architectural hyperparameter sensitivity on the MEG | E+C_{Li} configuration. Dashed lines mark the chosen values.

C Algorithms

C.1 The Full Inference Procedure

Algorithm 1 presents the full inference procedure of AMGENC, combining the data-centered conditional element noise (Section 4.2), per-step soft linearized projection (Section 4.3), and final discrete projection (Section 4.4). At each integration step, the velocity network predicts position and element velocities. Positions are advanced by a standard Euler step, while elements undergo a clean extrapolation to t=1 followed by the soft linearized projection that reduces the charge residual. The projected clean state is then re-interpolated along the flow path to obtain the element state for the next step. After the final step, element assignments are discretized via $\arg\max$ and corrected to exact charge balance by the final discrete projection (Algorithm 2).

Algorithm 1 AMGENC inference procedure.

Require: Lattice $\boldsymbol{L}$, number of atoms n_a, target properties $\boldsymbol{y}$, number of steps T, element frequencies $\{f_k\}_{k=1}^{d_E}$, formal charges $\boldsymbol{c} \in \mathbb{Z}^{d_E}$, noise scale σ, softmax temperature τ
Ensure: Generated material $\mathcal{M} = (\boldsymbol{L}, \boldsymbol{X}_1, \boldsymbol{E})$ with $Q = 0$

Sample positions $\boldsymbol{X}_0$ uniformly in the unit cell
Sample element noise: for each atom i, draw $k_i \sim \mathrm{Cat}(f_1, \ldots, f_{d_E})$, $\boldsymbol{E}_{0,i} \sim \mathcal{N}(\boldsymbol{e}_{k_i}, \sigma^2 \boldsymbol{I})$ {Equation 6}
$\Delta t \leftarrow 1/T$
for $i = 0$ **to** $T - 1$ **do**
  $t \leftarrow i \cdot \Delta t$
  $(\boldsymbol{v}_{\boldsymbol{X}}, \boldsymbol{v}_{\boldsymbol{E}}) \leftarrow \boldsymbol{v}_\theta(\mathcal{M}_t, \boldsymbol{y}, t)$ {Velocity prediction}
  $\boldsymbol{X}_{t+\Delta t} \leftarrow \boldsymbol{X}_t + \Delta t \cdot \boldsymbol{v}_{\boldsymbol{X}}$ {Euler step for positions}
  $\hat{\boldsymbol{E}}_1 \leftarrow \boldsymbol{E}_t + (1 - t) \cdot \boldsymbol{v}_{\boldsymbol{E}}$ {Clean extrapolation (Equation 4)}
  $Q \leftarrow \mathbf{1}^\top \mathrm{onehot}(\arg\max(\hat{\boldsymbol{E}}_1))\, \boldsymbol{c}$ {Discrete charge}
  if $Q \neq 0$ **then**
    $Q_{\text{soft}} \leftarrow \mathbf{1}^\top \mathrm{softmax}(\hat{\boldsymbol{E}}_1/\tau)\, \boldsymbol{c}$
    Compute $\nabla_{\hat{\boldsymbol{E}}_1} Q_{\text{soft}}$ via backpropagation
    $\hat{\boldsymbol{E}}_1^{\text{proj}} \leftarrow \hat{\boldsymbol{E}}_1 - \dfrac{Q}{\|\nabla_{\hat{\boldsymbol{E}}_1} Q_{\text{soft}}\|^2} \nabla_{\hat{\boldsymbol{E}}_1} Q_{\text{soft}}$ {Equation 8}
  else
    $\hat{\boldsymbol{E}}_1^{\text{proj}} \leftarrow \hat{\boldsymbol{E}}_1$
  end if
  $\boldsymbol{E}_{t+\Delta t} \leftarrow (1 - t - \Delta t)\, \boldsymbol{E}_0 + (t + \Delta t)\, \hat{\boldsymbol{E}}_1^{\text{proj}}$ {Equation 9}
end for
$e_i \leftarrow \arg\max_j \hat{\boldsymbol{E}}_{1,ij}$ for all i {Discretize elements}
$\{e_i'\} \leftarrow$ Final-Discrete-Projection$(\hat{\boldsymbol{E}}_1, \boldsymbol{c})$ {Algorithm 2}
return $\mathcal{M} = (\boldsymbol{L}, \boldsymbol{X}_1, \mathrm{onehot}(\{e_i'\}))$

C.2 Final Discrete Projection

Algorithm 2 solves the optimization problem in Equation 10 via dynamic programming. Since formal charges c_j are integers, the total charge delta from the original assignments can serve as a discrete state variable for the dynamic programming. The algorithm processes atoms one at a time: at each atom, it considers every possible element assignment and extends the cost table by accounting for the resulting change in charge and the logit gap sacrificed by the swap. After all n_a atoms are processed, the optimal corrected assignments $\{e'_i\}$ satisfying $\sum_i c_{e'_i} = 0$ are recovered by tracing back through the cost table from the target state. The overall complexity is $O(n_a \cdot |Q_{\max}| \cdot d_E)$, where $|Q_{\max}|$ is the range of reachable charge states. In our implementation, the inner loop over elements $j \in \{1, \ldots, d_E\}$ is vectorized into matrix operations, computing all swap costs and charge deltas for the d_E candidates simultaneously at each atom i to reduce wall-clock time.

Algorithm 2 Minimum-cost discrete projection via dynamic programming.

Require: Element logits $\hat{\boldsymbol{E}}_1 \in \mathbb{R}^{n_a \times d_E}$ at $t = 1$, formal charges $\boldsymbol{c} \in \mathbb{Z}^{d_E}$
Ensure: Corrected assignments $\{e'_i\}_{i=1}^{n_a}$ with $\sum_i c_{e'_i} = 0$

$e_i \leftarrow \arg\max_j \hat{\boldsymbol{E}}_{1,ij}$ for all i {Current assignments}
$Q \leftarrow \sum_{i=1}^{n_a} c_{e_i}$ {Current total charge}
if $Q = 0$ **then**
  return $\{e_i\}$ {Already balanced}
end if
Initialize cost table: $\text{cost}[0, 0] \leftarrow 0$, all others $\leftarrow \infty$
for $i = 1$ **to** n_a **do**
  for each element $j \in \{1, \ldots, d_E\}$ **do**
    $s \leftarrow \hat{\boldsymbol{E}}_{1,i,e_i} - \hat{\boldsymbol{E}}_{1,i,j}$ {Swap cost}
    $\delta \leftarrow c_j - c_{e_i}$ {Charge delta}
    for each reachable state q **do**
      $\text{cost}[i, q + \delta] \leftarrow \min(\text{cost}[i, q + \delta],\ \text{cost}[i-1, q] + s)$
    end for
  end for
end for
Backtrack from state $q = -Q$ to recover optimal $\{e'_i\}$
return $\{e'_i\}$

D Dataset Preparation

D.1 Amorphous Silica (a-SiO_2) Dataset

The a-SiO_2 dataset contains $6,000$ samples that share the composition of pure silica, SiO_2. To maximize the variation of properties between the samples generated with the same simulation workflow, relatively small unit cells are chosen with the number of atoms uniformly selected in the range of 80 to 250. Atoms are initially placed in a unit cell with a volume $V = 4 \sum_i \frac{4}{3} \pi r_i^3$, with r_i being the covalent radius of the i-th atom, avoiding unphysical overlap between neighboring atoms. A local structure relaxation is performed on the initial configuration followed by an MD simulation in the NPT ensemble at 3500 K for 2000 ps. To limit the effects of relaxation, which we observe for our other datasets, we use an instantaneous quenching procedure by performing a local structure optimization and a subsequent equilibration at 300 K for 10 ps. All simulations are performed using LAMMPS [41] software and the Tersoff potential parameterized by [33].

D.2 Multi-Element Glass (MEG) Dataset

The MEG dataset contains 9,027 multi-component glass samples with 11 different elements (Si, P, Al, Li, Be, K, Ca, Ti, Ba, Zn, O). Compositions are generated from varying ratios of glass formers (SiO_2, P_2O_5) and modifiers (Al_2O_3, Li_2O, BeO, K_2O, CaO, TiO_2, BaO, ZnO), with up to four modifiers at 40% total concentration relative to the glass formers. Initial structures contain approximately 800 atoms placed randomly in a simulation cell with volume $V = 3 \sum_i \frac{4}{3} \pi r_i^3$, where r_i is the covalent

radius of atom i. Samples are prepared using a melt-quench procedure: melting at $\frac{3}{4}T_{\text{evap}}$ for 400 ps, quenching to 300 K at 5 K/ps, and equilibrating for 300 ps. All simulations use LAMMPS [41] with the Bertani–Menziani–Pedone (BMP)-shrm potential [2].

D.3 Charge Balance of the Datasets

Under the formal charge values in Appendix E.1, all 6,000 a-SiO_2 samples and all 9,027 MEG samples have total charge $Q = 0$. This follows from the construction of the datasets: the a-SiO_2 samples have the fixed SiO_2 stoichiometry, and the MEG compositions are assembled from the neutral oxide units listed above.

E Charge Balance

E.1 Charge Value Assignment

The formal charge values c_j in Definition 2 are the integer oxidation states from the Bertani–Menziani–Pedone (BMP) potential [2]. For the a-SiO_2 dataset, the two element types have formal charges $c_{\text{Si}} = +4$ and $c_{\text{O}} = -2$. For the MEG dataset, the 11 element types have the following formal charges: Si $(+4)$, O (-2), P $(+5)$, Al $(+3)$, Li $(+1)$, Be $(+2)$, K $(+1)$, Ca $(+2)$, Ti $(+4)$, Ba $(+2)$, and Zn $(+2)$. Ghost atoms are assigned $c = 0$.

E.2 Charge Balance Metrics

Given N generated samples with charge values $Q(\boldsymbol{E}_i)$ for $i = 1, \ldots, N$, we report the following metrics.

Probability of charge balance. The percentage of generated samples satisfying $Q(\boldsymbol{E}) = 0$.

Mean absolute charge value.

$$|\bar{Q}| = \left| \frac{1}{N} \sum_{i=1}^{N} Q(\boldsymbol{E}_i) \right| \tag{15}$$

Standard deviation of charge value.

$$\sigma_Q = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (Q(\boldsymbol{E}_i) - \bar{Q})^2} \tag{16}$$

E.3 Charge Balance and Stoichiometry

In the training samples, zero total charge follows from how the compositions were constructed (Appendix D.3). During generation, however, every element assignment is predicted, so unconstrained generation can produce element counts with nonzero total charge, and AMGENC enforces $Q = 0$ on each generated sample. Let N_j denote the number of atoms of element j in a sample after removing ghost atoms.

For the a-SiO_2 dataset, a sample contains only Si and O, and

$$Q = 4N_{\text{Si}} - 2N_{\text{O}} = 0 \quad \Longleftrightarrow \quad N_{\text{O}} = 2N_{\text{Si}} \tag{17}$$

so formal charge balance is equivalent to the exact SiO_2 stoichiometry.

For the MEG dataset, there is no single target formula, since the compositions vary over mixtures of the ten constituent oxides. For any generated cation counts, these oxide units require

$$N_{\text{O}}^{\text{oxide}} = 2N_{\text{Si}} + \tfrac{5}{2}N_{\text{P}} + \tfrac{3}{2}N_{\text{Al}} + \tfrac{1}{2}N_{\text{Li}} + N_{\text{Be}} + \tfrac{1}{2}N_{\text{K}} + N_{\text{Ca}} + 2N_{\text{Ti}} + N_{\text{Ba}} + N_{\text{Zn}} \tag{18}$$

oxygen atoms, and under the formal charge values in Appendix E.1, $Q = 2(N_{\text{O}}^{\text{oxide}} - N_{\text{O}})$. Formal charge balance is thus equivalent to a sample containing exactly the oxygen needed to form neutral

oxide units with its cations. The constraint leaves the cation ratios free, so the property-conditioned generation can select cation ratios beyond the compositions of the training samples.

Table 6 reports two rates over the samples generated by AMGENC: the stoichiometric validity rate, i.e., the percentage of samples satisfying the above criterion, and the training composition match rate, i.e., the percentage of samples whose simplest integer formula, obtained by dividing all element counts by their greatest common divisor, is identical to that of at least one training sample. On the MEG dataset, the low match rates show that AMGENC usually generates new cation ratios instead of repeating the compositions of training samples. This is expected for property-conditioned generation, especially for the target $C_{\mathrm{Li}} = 15\%$, which lies outside the property distribution of the training samples (Figure 4b).

Table 6: Stoichiometric validity and training composition match rates of samples generated by AMGENC.

Config	Stoichiometric criterion	Stoichiometric validity rate	Training composition match rate
a-SiO_2 | G	$N_{\mathrm{O}} = 2N_{\mathrm{Si}}$	100%	100%
a-SiO_2 | RSD	$N_{\mathrm{O}} = 2N_{\mathrm{Si}}$	100%	100%
MEG | E+C_{Li}	$N_{\mathrm{O}} = N_{\mathrm{O}}^{\mathrm{oxide}}$	100%	0.00%
MEG | G	$N_{\mathrm{O}} = N_{\mathrm{O}}^{\mathrm{oxide}}$	100%	0.15%

F Inverse Design Evaluation

F.1 Property Calculation

We evaluate inverse design performance using material properties relevant to the dataset. For a-SiO_2 samples, we focus on shear modulus and ring size distribution. For MEG samples, we focus on Young's modulus, shear modulus, and lithium molar concentration.

a-SiO_2 dataset properties. For a-SiO_2 samples, properties are computed directly from atomic structures. Shear modulus is calculated using finite differences of the stress tensor: structures are relaxed with the Tersoff potential (force tolerance 0.05 eV/Å), then subjected to small strains ($\delta = 0.02$) to compute elastic constants C_{44}, C_{55}, and C_{66} from stress responses. Shear modulus is their average: $G = \frac{1}{3}(C_{44} + C_{55} + C_{66})$. Ring size distribution is computed by identifying closed rings in the Si-O network using a depth-first search algorithm. Atoms are considered bonded if their distance is below 1.3 times the sum of their covalent radii (Si: 1.11 Å, O: 0.66 Å). The algorithm searches for the shortest path between bonded atom pairs while excluding the direct bond, ensuring proper ring closure with periodic boundary conditions. Ring sizes are reported as the number of Si atoms per ring, averaged across all identified rings.

MEG Dataset properties. For MEG samples, elastic properties are computed using the BMP-shrm potential [2]. Structures are first relaxed, then elastic constants C_{ij} are calculated from finite differences of the stress tensor using Voigt notation. Young's modulus is computed as $E = \frac{(C_{11}-C_{12})(C_{11}+2C_{12})}{C_{11}+C_{12}}$ and shear modulus as $G = C_{44}$. Lithium molar concentration C_{Li} is computed directly from the elemental composition as the ratio of lithium atoms to total atoms.

Implementation details. Ghost atoms are excluded from all calculations. Structural relaxations use quasi-Newton optimization with variable cell shapes, maximum 1500 steps. Ring finding incorporates periodic boundary conditions for rings crossing cell boundaries.

F.2 Inverse Design Metrics

For inverse design evaluation, we compare target properties y_{target} with computed or predicted properties $y_{\mathrm{generated}}$ of generated samples using three standard regression metrics:

Mean Absolute Error (MAE).

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_{\text{generated},i} - y_{\text{target},i}| \tag{19}$$

Root Mean Square Error (RMSE).

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{\text{generated},i} - y_{\text{target},i})^2} \tag{20}$$

Mean Absolute Percentage Error (MAPE).

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_{\text{generated},i} - y_{\text{target},i}}{y_{\text{target},i}}\right| \tag{21}$$

These metrics are computed separately for each property (shear modulus, RSD, Young's modulus, Li ratio) across all generated samples. MAE provides interpretable error magnitude in original units, RMSE penalizes large deviations more heavily, and MAPE enables comparison across properties with different scales and units.

F.3 Training Data and Target Property Distributions

Figure 4 compares the property distributions of training data and samples generated by AMGENC for each inverse design configuration. The AMGENC distributions extend beyond the training data range, demonstrating that the model is capable of extrapolating to property values not seen during training.

On the other hand, it is shown that the generated samples extrapolate more readily toward lower property values than toward higher ones. This reflects physical constraints of the material systems: on the MEG dataset, network modifiers such as Li reduce network connectivity and thus stiffness [37], making very high elastic moduli difficult to achieve at the targeted Li content ($C_{\text{Li}} = 0.15$). Similarly, for a-SiO_2, the achievable range of shear modulus and ring size distribution is bounded by the structural constraints of the silica network.

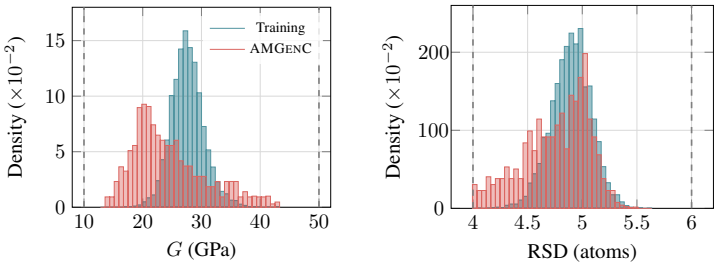
G Structural Features of Generated Samples

Although structural accuracy is not the primary focus of this work, we compare key structural features of samples generated by AMGENC with those of the training samples. Figures 5 and 6 compare the partial radial distribution functions $g(r)$ of generated and training samples. Overall, the generated samples reproduce the correct first-shell peak positions across all element pairs. In some cases the peaks are lower and broader than those of the training data. This can be attributed to two factors: the generation is conditioned on target properties that extend beyond the training distribution (Appendix F.3), leading to compositional shifts between generated and training samples; and diffusion and flow matching models have inherent limitations in fully reproducing the well-relaxed local structures of amorphous materials [10]. Note that for the a-SiO_2 dataset, the training curves may not appear perfectly smooth due to the relatively small simulation cells in the training data. For the MEG dataset, some plots appear noisier than others due to the low abundance of certain network modifiers in the glass compositions.

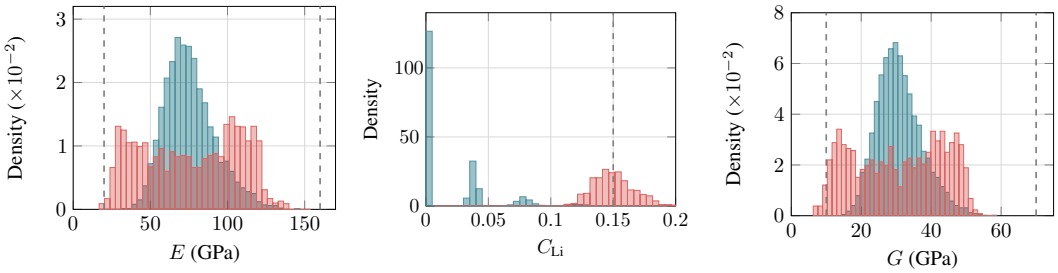
Figure 7 compares the cumulative coordination numbers $n(r)$ of oxygen neighbors around each cation type in the MEG samples. The first-shell plateau values confirm that the generated structures reproduce the expected coordination numbers. The agreement is strongest in the first coordination shell. Beyond the first shell, minor deviations appear, consistent with the compositional differences between the conditioned generation and the training distribution.

H Effect of the Final Discrete Projection

The final discrete projection (DP, Section 4.4) keeps every atom position fixed and changes only element assignments, so comparing the same generated samples immediately before and after DP isolates its effect.



(a) a-SiO_2 | G and a-SiO_2 | RSD configurations



(b) MEG | E+C_{Li} and MEG | G configurations

Figure 4: Property distributions of training data and samples generated by AMGENC for each inverse design configuration. Dashed lines indicate the target generation range (or target value for C_{Li}).

Changed element assignments. Table 7 reports the number of element assignments changed by DP in samples generated by AMGENC. The atom fraction is computed over all atoms of a sample, including ghost atoms. Across the four configurations, DP changes less than one element assignment per sample on average and at most four in any sample.

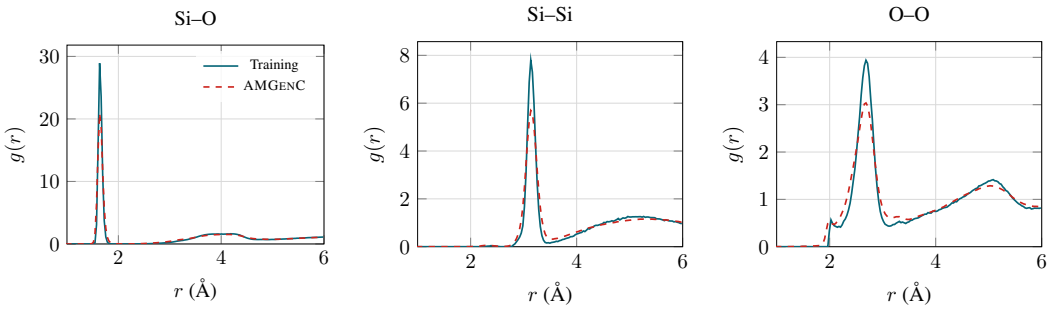
Table 7: Element assignments changed by DP in samples generated by AMGENC.

Config	Mean changed	Maximum changed	Mean atom fraction changed
a-SiO_2 | G	0.63	2	0.098%
a-SiO_2 | RSD	0.59	2	0.092%
MEG | E+C_{Li}	0.81	4	0.061%
MEG | G	0.77	3	0.058%

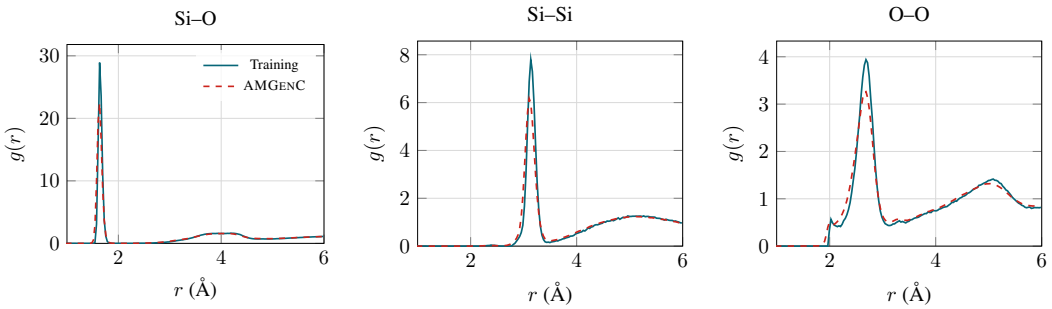
Structural metrics. Each structural metric compares the distribution over generated samples with that over training samples. Partial radial distribution functions (RDFs) use 200 bins over 0 to 6 Å, and bond angle distributions use 180 bins over 0 to 180°. The first-shell cutoff is the first local minimum after the main peak of the partial RDF of training samples, which also defines the neighbors for bond angles. We report the Jensen-Shannon divergence (JSD, base 2) and the Wright factor [46] of the partial RDFs, the MAE of the cumulative coordination number $n(r)$ from zero distance to the first-shell cutoff (CN MAE), and the JSD of the bond angle distributions. For MEG, all metrics are computed for each cation type $X \in \{\mathrm{Si}, \mathrm{P}, \mathrm{Al}, \mathrm{Li}, \mathrm{Be}, \mathrm{K}, \mathrm{Ca}, \mathrm{Ti}, \mathrm{Ba}, \mathrm{Zn}\}$ paired with oxygen, and averaged over X. All comparisons use the same training samples and cutoffs. The same metrics are used for Table 5.

Table 8 reports the structural metrics before and after DP. Every metric changes by at most 0.12% on a-SiO_2 | RSD and 0.25% on MEG | E+C_{Li} relative to its value before DP, showing that DP preserves the measured partial RDF, coordination number, and bond angle distributions.

Composition and inverse design metrics. On MEG | E+C_{Li}, DP changes the mean atomic percentage of every element by at most 0.01 percentage points, and its standard deviation across samples by at most 0.02 percentage points. For example, the Li atomic percentage changes only from



(a) a-SiO_2 | G



(b) a-SiO_2 | RSD

Figure 5: Partial radial distribution functions $g(r)$ for a-SiO_2 samples. Solid: training samples. Dashed: samples generated by AMGENC under the (a) a-SiO_2 | G configuration and (b) a-SiO_2 | RSD configuration.

Table 8: Structural metrics of samples generated by AMGENC before and after the final discrete projection (DP), compared with training samples. Lower is better for all metrics.

Config	Metric	Before DP	After DP
a-SiO_2 | RSD	Mean JSD of $g_{\mathrm{SiO}}(r)$, $g_{\mathrm{SiSi}}(r)$, $g_{\mathrm{OO}}(r)$	0.01192	0.01193
	Mean Wright factor of $g_{\mathrm{SiO}}(r)$, $g_{\mathrm{SiSi}}(r)$, $g_{\mathrm{OO}}(r)$	0.22106	0.22118
	CN MAE of $n_{\mathrm{SiO}}(r)$	0.03759	0.03762
	Mean JSD of $\angle$OSiO, $\angle$SiOSi distributions	0.00838	0.00839
MEG | E+C_{Li}	Mean JSD of $g_{XO}(r)$	0.01415	0.01418
	Mean Wright factor of $g_{XO}(r)$	0.19952	0.19989
	Mean CN MAE of $n_{XO}(r)$	0.04471	0.04482
	Mean JSD of $\angle OXO$ distributions	0.01829	0.01832

14.44% to 14.45%, after the property-conditioned generation has already shifted it from the training mean of 1.98% toward the 15% target. The inverse design MAPE changes from 20.2% to 20.1% for E and from 9.9% to 9.8% for C_{Li} (Table 4). The property-conditioned generation thus determines the composition, while DP has a negligible effect on it.

I Baseline Adaptation

All baselines are retrained on the same datasets and conditioned on the same target properties as AMGENC, with the training and sampling settings in Section 5.3. The adaptation of each baseline is as follows.

CDVAE. Its original workflow first chooses the composition, lattice, and number of atoms, then generates atom positions and elements. We retain this workflow, retrain it on the amorphous datasets,

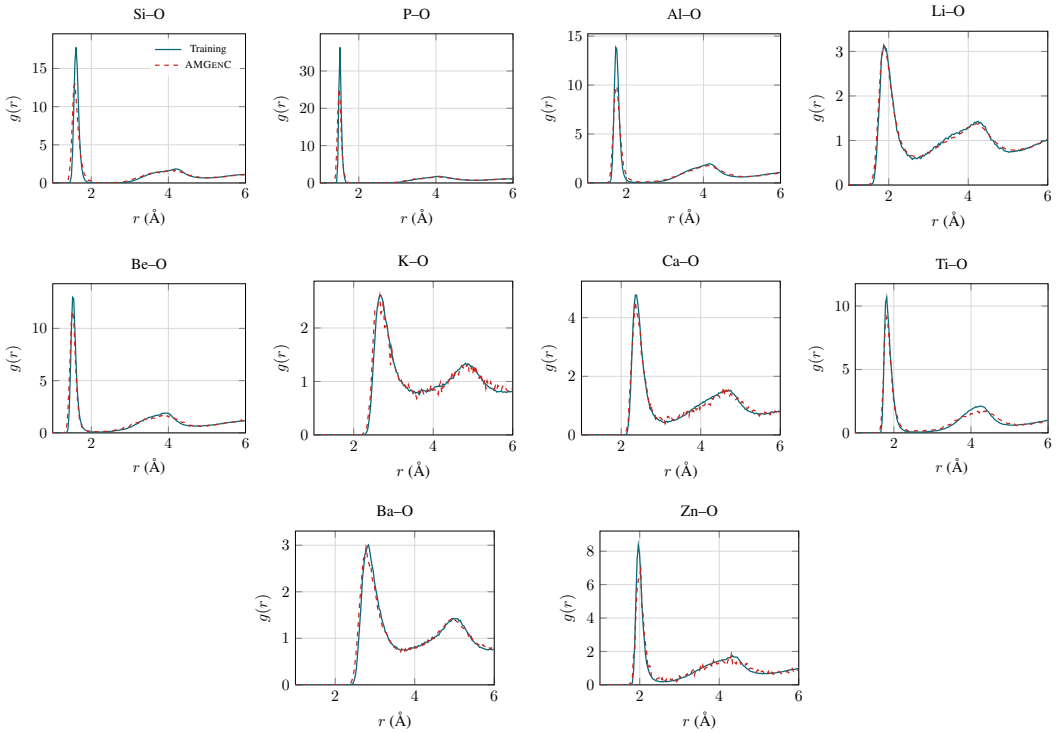


Figure 6: Partial radial distribution functions $g(r)$ for element–oxygen pairs in the MEG samples. Solid: training samples. Dashed: samples generated by AMGENC under the MEG | E+C_{Li} configuration.

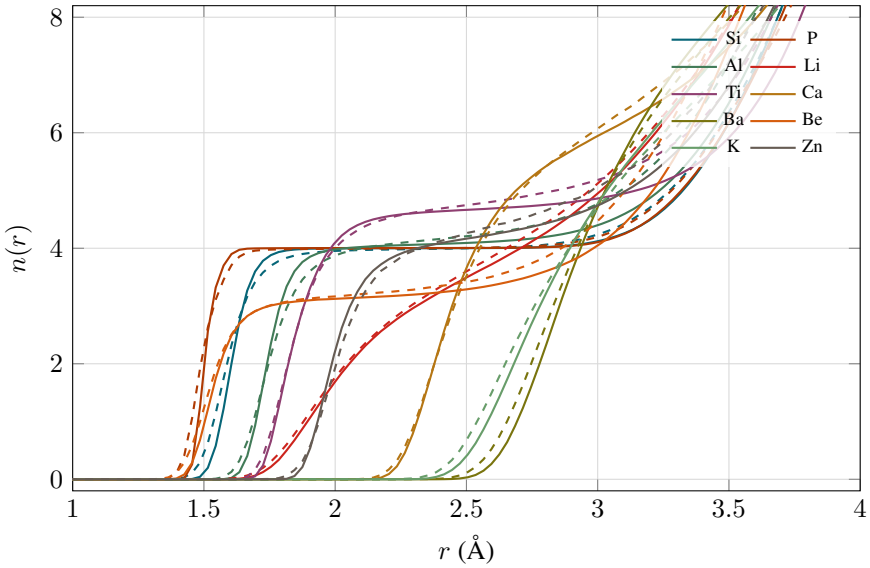


Figure 7: Cumulative coordination number $n(r)$ of oxygen neighbors around each cation type in the MEG samples. Solid: training samples. Dashed: samples generated by AMGENC under the MEG | E+C_{Li} configuration.

and replace its optimization of a single property toward an extreme value with matching all target property values.

CrystalFlow. Its original workflow generates the lattice, atom positions, and elements jointly while conditioning on numerical properties. We retain this workflow and retrain it on the amorphous datasets with all target properties as conditioning.

MatterGen. Its original workflow first learns material generation from unlabeled structures, then adapts to labeled property targets. We follow this workflow with unlabeled and labeled amorphous samples, and retain its joint generation of the lattice, atom positions, and elements.

Graphite. Its original workflow generates amorphous structures using the error of a XANES spectrum prediction to guide sampling. We retrain it for multiple elements and change the prediction target from per-atom XANES spectra to the target properties.

AMDEN. Its original task and workflow already match our setting, so we retrain it for each dataset and property set. It generates atom positions and elements jointly inside a fixed cell, using removable empty sites, similar to ghost atoms in this work, to vary the number of atoms and the density.

J Experimental Environment

Hardware. All experiments are conducted on an internal GPU cluster managed by the SLURM workload scheduler.[2] Each compute node is equipped with two AMD EPYC 7302 16-core processors (32 cores, 64 threads total, base 1.5 GHz, boost 3.0 GHz), 256 GB of system memory, and three NVIDIA A40 GPUs (48 GB memory each). Each experiment uses a single A40 GPU, 4 CPU cores, and 64 GB of system memory.

Software. The implementation uses Python 3.12 and PyTorch 2.9.1 [36] with CUDA 12.4. Key dependencies include NumPy 2.4.1 [13], SciPy 1.17.0 [45], ASE 3.27.0 [22], and pymatgen 2025.10.7 [35]. All experiments run inside Singularity [20] containers built from the `nvidia/cuda:12.4.1-devel-ubuntu22.04` base image to ensure reproducibility across nodes. The implementation code and sample datasets are available at `https://github.com/Logan-Lin/AMGenC-code`.

K Discussion

K.1 Broader Impacts

Accelerating materials discovery. By guaranteeing charge balanced outputs, AMGENC makes generative inverse design of amorphous materials more practical, reducing wasted computation from discarding charge unbalanced samples. As shown in Table 3, post-hoc filtering can require orders of magnitude more generation time to obtain charge balanced samples, whereas AMGENC eliminates this overhead entirely. This can accelerate the discovery of amorphous materials for applications such as energy storage, thermal management, and advanced materials, where efficient exploration of the vast design space is important.

Potential for broader applications. Beyond amorphous materials, the proposed framework for enforcing non-differentiable hard constraints on discrete outputs, through the combination of data-centered conditional element noise, soft linearized projection, and final discrete projection, may generalize to other generative modeling settings where hard constraints on discrete data must be satisfied, such as point cloud generation with categorical label constraints or graph generation with degree-sum invariants.

K.2 Limitations

Gap between generation and real-world synthesis. Generated samples represent computationally predicted atomic configurations, but current experimental techniques cannot directly fabricate amorphous materials atom by atom. Nevertheless, generated samples are highly informative as candidates that narrow the search space for composition and processing choices, much as traditional

[2] `https://slurm.schedmd.com/overview.html`

computational methods like reverse Monte Carlo [31] have long been used to guide materials design. Additionally, these candidates still require validation through simulation (e.g., molecular dynamics relaxation) or experimental synthesis before real-world deployment.

Scope of the formal charge assignment. The guarantee of AMGENC is defined relative to the supplied formal charge assignment. Formal charge balance is a necessary global condition on the composition of the evaluated oxide glasses (Appendix E.3), but it does not imply electronic neutrality or local chemical validity, which require separate assessments such as bond valence and stability analysis. For the evaluated datasets, the formal charge assignment is unambiguous, since the samples are constructed from neutral oxide units with fixed oxidation states. Materials with ambiguous or composition-dependent oxidation states require extensions. Fixed rational charges can be multiplied by a common denominator and passed to the same discrete projection. Known mixed-valence states, such as Fe^{2+} and Fe^{3+}, can be represented as separate element classes when the allowed states are specified in advance. Composition-dependent charges would require a differentiable charge predictor for the soft linearized projection and an exactly testable discrete charge representation for the final discrete projection. Evaluating these extensions remains future work.

Structural accuracy. This work focuses on enforcing charge balance on element compositions and is not aimed at improving the structural accuracy of generated amorphous materials. As shown in Appendix G, minor structural discrepancies remain in the partial radial distribution functions of generated samples, where peaks can be lower and broader than those of the training data. This is consistent with known limitations of diffusion and flow matching models in fully reproducing the well-relaxed local structures of amorphous materials [10]. Improving structural accuracy is an orthogonal direction that could complement the charge balance guarantees provided by AMGENC.

Generalizability. While the proposed framework for enforcing hard constraints on discrete outputs may apply to broader generative modeling problems, we evaluate it only in the context of amorphous materials generation across two datasets. Assessing how well the approach transfers to other domains with different constraint structures remains future work.

K.3 LLM Usage

LLM usage in this work is limited to proofreading the drafted paper and assisting with part of the implementation code, after the methodology and algorithms were designed and specified. LLMs were not used as a component of the proposed method.

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 abstract and introduction state three claims: guaranteed formal charge balance under the supplied formal charge assignment, minimal computational overhead, and no sacrifice in inverse design accuracy. These are directly supported by experimental results in Section 5.4 (Tables 1, 2, 3) and Section 5.5 (Tables 4, 5).

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: Appendix K.2 discusses four limitations: the gap between generated samples and real-world synthesis, the scope of the formal charge assignment, the focus on charge balance rather than structural accuracy, and the lack of evaluation on the generalizability of the framework beyond amorphous materials.

   Guidelines:

   - The answer [N/A] 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: [N/A]

   Justification: The paper does not include formal theoretical results (theorems, lemmas, or proofs). The contributions are methodological and empirical.

   Guidelines:

   - The answer [N/A] 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: On top of the method details in Section 4, the appendices provide full algorithmic pseudocode (Appendix C), network architecture and hyperparameters (Appendix B), dataset preparation procedures (Appendix D), and computational environment (Appendix J).

   Guidelines:

   - The answer [N/A] 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: [Yes]

   Justification: The implementation code and sample datasets are publicly available on GitHub, linked in Appendix J. Both datasets used in this work are the ones introduced and published with AMDEN [10].

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5.3 specifies the optimizer, learning rate, batch size, number of epochs, sampling steps, and evaluation protocol. Appendices provide further details on architecture hyperparameters (Appendix B), dataset preparation (Appendix D), charge assignments (Appendix E.1), and property calculation procedures (Appendix F.1).

   Guidelines:

   - The answer [N/A] 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: [Yes]

Justification: Each experiment is repeated 5 times. All tables report the mean and standard deviation across runs, and all hyperparameter sensitivity plots include error bars (Section 5.3 and Section 5.5).

Guidelines:

- The answer [N/A] 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: Appendix J reports the GPU type, memory, CPU allocation, software versions, and container setup used for all experiments.

   Guidelines:

   - The answer [N/A] 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: The research involves computational methods for materials science. It does not involve human subjects, personal data, or applications with foreseeable negative societal impact.

   Guidelines:

   - The answer [N/A] 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: Appendix K.1 discusses positive impacts (accelerating materials discovery) and potential for broader applications beyond materials, while Appendix K.2 discusses concerns such as the gap between generation and real-world synthesis.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The released model and data are specific to amorphous materials generation and do not pose risks for misuse.

    Guidelines:

    - The answer [N/A] 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: [Yes]

Justification: All existing datasets and software used (LAMMPS, ASE, PyTorch, etc.) are properly cited.

Guidelines:

- The answer [N/A] 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: [Yes]

Justification: The implementation code and sample datasets are released in a public GitHub repository with documentation (Appendix J).

Guidelines:

- The answer [N/A] 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: [N/A]

Justification: This work does not involve crowdsourcing or human subjects.

Guidelines:

- The answer [N/A] 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: [N/A]

Justification: This work does not involve human subjects.

Guidelines:

- The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were used only for proofreading and assisting with implementation code, not as a component of the proposed method (Appendix K.3).

Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.