Offline Materials Optimization with CliqueFlowmer

Jakub Grudzien Kuba* **Benjamin Kurt Miller†** **Sergey Levine*** **Pieter Abbeel***

Abstract

Recent advances in deep learning have inspired neural network-based approaches to *computational materials discovery* (CMD). A plethora of problems in this field involve finding materials that optimize a target property. Nevertheless, the increasingly popular generative modeling methods are ineffective at boldly exploring attractive regions of the materials space due to their maximum likelihood training. In this work, we offer an alternative CMD technique based on offline *model-based optimization* (MBO) that fuses direct optimization of a target material property into generation. To that end, we introduce a domain-specific model, *CliqueFlowmer*, that incorporates recent advances in clique-based MBO into transformer and flow generation. We validate this model's optimization abilities and show that materials it produces strongly outperform those from generative baselines. To support specialized materials discovery applications and broader interdisciplinary research, we release our **code**, **model weights**, and an **interactive demo**.

1 Introduction

Large neural network models have recently enabled solving challenging artificial intelligence (AI) tasks such as language modeling, automated coding, and image generation (Achiam et al., 2023; Team et al., 2024; Ho et al., 2020; Esser et al., 2024). Meanwhile, scientific problems that deal with the world of atoms, rather than the world of bits, are yet to benefit from this revolution (Thiel, 2025). Indeed, exploration of new physical breakthroughs continues to take place in physical wet labs and is driven by costly physical experimentation (Xiang et al., 1995; Freese et al., 2024; Shahzad et al., 2024). One such problem is *computational materials discovery* (CMD), wherein scientists strive to invent new chemical structures that display properties absent in known materials (Danielson et al., 1997; Jain et al., 2013; Xue et al., 2016; Lin et al., 2025). A systematic, sample-efficient method for CMD has the potential to deliver structures that serve as catalysts for key energy-conversion reactions and as functional components of advanced biomaterials. Such advances would greatly accelerate the development of clean-energy technologies and medical therapies, effectively extending the recent AI-driven progress from the world of bits to the world of atoms (Hutmacher, 2000; Nocera, 2009; Gokcekuyu et al., 2024; Han and Su, 2025).

Having realized the importance of this problem, AI researchers have been increasingly turning their attention to AI-driven CMD. Most commonly, the introduced methods have been utilizing the celebrated *diffusion* and *flow* models that had become very successful in the domain of image generation (Ho et al., 2020; Lipman et al., 2022; Miller et al., 2024; Inizan et al., 2025). While effective at harnessing the distribution of viable materials presented to them in the dataset, likelihood-based generative models do not actively explore the relation between the materials and their properties. This is a major impediment in settings where CMD is expected to optimize materials with respect to a specified metric (Yang et al., 2024; Havens et al., 2025). In the meantime, a new paradigm, known as *offline model-based optimization* (MBO) has made steps towards techniques that optimize *scientific designs* by bootstrapping models trained entirely on offline data (Kumar et al., 2021; Trabucco et al., 2022; Kuba et al., 2024). Nevertheless, MBO methods have been mainly deployed in standard

*BAIR, UC Berkeley, Berkeley, California, USA. Corresponding author: kuba@berkeley.edu
†FAIR, Meta, San Francisco, California, USA

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

benchmark tasks, free, for example, of the challenging intricacies of CMD in which data is inherently of the hybrid discrete-continuous nature, irregular shape, and subject to physical constraints.

To address the need for AI-driven CMD and the limitations of generative approaches, we introduce *CliqueFlowmer*—a model that renders materials data tractable by MBO and enables direct optimization of material structures. At its core is an auto-encoder that transforms multi-modal, irregular materials data into finite-dimensional vectors that can be optimized with *clique-based* MBO (Kuba et al., 2024) and decoded back into material form. Because CliqueFlowmer is built from general-purpose deep learning primitives, it makes CMD more accessible to the broader AI community while reducing reliance on specialized domain-specific software. Empirically, using data from Materials Project (Jain et al., 2013) together with M3GNet (Chen and Ong, 2022) and MEGNet (Chen et al., 2019) as property oracles, we show that materials optimized by CliqueFlowmer vastly outperform those sampled by generative baselines. While our experiments focus on offline materials optimization, the pretrained auto-encoder also provides a practical continuous representation space for materials.

2 Background

This section provides the necessary background on the key notions discussed by our work. First, we lay down foundations of material modeling that enables us to use machine learning methods in CMD. Then, we introduce the basic ideas of continuous normalizing flows (*i.e.,* flow matching). Lastly, we state the problem that we aim to solve (CMD) and formulate it through the lens of offline MBO.

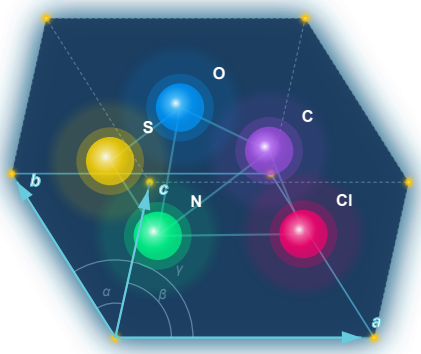


Figure 1: The unit cell of a hypothetical material. The cell has a shape of a parallelepiped determined by three axes, $\vec{\mathrm{a}}$, $\vec{\mathrm{b}}$, and $\vec{\mathrm{c}}$. The angles between the axes are $\mathrm{ang}(\vec{\mathrm{b}}, \vec{\mathrm{c}}) = \alpha$, $\mathrm{ang}(\vec{\mathrm{c}}, \vec{\mathrm{a}}) = \beta$, $\mathrm{ang}(\vec{\mathrm{a}}, \vec{\mathrm{b}}) = \gamma$. In this cell, there are five atoms, whose type sequence is $\mathbf{a} = [\mathrm{N}, \mathrm{Cl}, \mathrm{C}, \mathrm{O}, \mathrm{S}]$.

2.1 Modeling Materials

We characterize a material M by its *unit cell*—the material's unit amount. The cell is a parallelepiped with axis lengths $(a, b, c) \in \mathbb{R}^3_+$ in Å and interaxial angles $(\alpha, \beta, \gamma) \in (0, \pi)^3$ in radians. The content of the cell is a set of N_{atom} (which varies between materials) atoms that is represented by the sequence $\mathbf{a} \in \mathcal{A}^{N_{\text{atom}}}$ of their types, as well as their positions $\mathbf{X} \in [0, 1)^{N_{\text{atom}} \times 3}$ expressed in the basis induced by (a, b, c) and (α, β, γ). Thus, the space of materials M can be embedded in the product $\mathcal{M} = \mathbb{R}^3_+ \times (0, \pi)^3 \times (\mathcal{A} \times [0, 1)^3)^*$, and a single material can be characterized as a tuple $[\mathbf{a}, (a, b, c), (\alpha, \beta, \gamma), \mathbf{X}]$. [3] We denote the geometrical part of the material as $\mathbf{G} = [(a, b, c), (\alpha, \beta, \gamma), \mathbf{X}]$, and thus can write $[\mathbf{a}, \mathbf{G}]$ to represent material M.

2.2 Flow Matching

The goal of a continuous normalizing flow (Lipman et al., 2022) is to learn sampling from a data distribution $p_{\text{data}}(\mathbf{x})$. To that end, one sets the *target* distribution $p_1(\mathbf{x}) = p_{\text{data}}(\mathbf{x})$, as well as *source* distribution $p_0(\mathbf{x}) = p_{\text{source}}(\mathbf{x})$, such as the standard-normal. To learn to turn a sample from $p_{\text{source}}(\mathbf{x})$ into one from $p_{\text{data}}(\mathbf{x})$, one interpolates noise and data,

$$\mathbf{x}_t = (1 - t) \cdot \mathbf{x}_0 + t \cdot \mathbf{x}_1, \quad \text{where} \quad \mathbf{x}_0 \sim p_0(\mathbf{x}),\ \mathbf{x}_1 \sim p_1(\mathbf{x}) \quad \& \quad t \sim p_{\text{time}}(t),$$

and passes the mixture, together with timestep information, to a *velocity* neural network that minimizes

$$\mathbb{E}_{\mathbf{x}_0 \sim p_0, \mathbf{x}_1 \sim p_1, t \sim p_{\text{time}}} \left[\left(v_\theta(\mathbf{x}_t, t) - (\mathbf{x}_1 - \mathbf{x}_0) \right)^2 \right].$$

Once learned, the velocity network can be used to sample from the target distribution by solving the following ordinary differential equation (ODE) with initial conditions,

$$\mathrm{d}\mathbf{x}_t = v_\theta(\mathbf{x}_t, t)\mathrm{d}t, \qquad \mathbf{x}_0 \sim p_0(\mathbf{x}),$$

from $t = 0$ to $t = 1$, which results in a sample $\mathbf{x}_1 \sim p_1(\mathbf{x})$. Typically, the ODE is solved numerically, by discretizing the time interval $[0, 1]$ into N_{step} steps and using, *e.g.,* the Euler method. In this paper, we will use continuous normalizing flows to model the geometry of materials.

[3] $\mathcal{S}^*$ denotes the countable union of the products of the set $\mathcal{S}$ with itself. That is, $\mathcal{S}^* = \cup_{n=1}^{\infty} \mathcal{S}^n$.

2.3 Model-Based Optimization in Computational Materials Discovery

We consider a function $f(\mathsf{M}) \in \mathbb{R}$, often referred to as *target property*, and assume that we are given a finite dataset $\mathcal{D} = \{\mathsf{M}^i, \mathrm{y}^i = f(\mathsf{M}^i)\}_{i=1}^{N_{\text{data}}}$ of N_{data} examples of materials and their values of the property. Given this dataset, our goal is to discover materials that optimize the property,

$$\min_{\mathsf{M} \in \mathcal{M}} f(\mathsf{M}) \tag{1}$$

without the necessity to evaluate consecutive candidates in a wet lab[4] (Danielson et al., 1997). That is, we want to do this *fully offline*. The standard approach to this problem proceeds in two stages: *1)* train a generative model $p_\theta(\mathsf{M})$ to sample candidate materials, and *2)* select the most promising candidate among N_{prop} proposals (Gómez-Bombarelli et al., 2018; Noh et al., 2019; Park et al., 2024; Zeni et al., 2025). However, this approach is quite inefficient for optimization, since sampling from $p_\theta(\mathsf{M})$ only explores the regions of the materials space supported by the training data; methods that more effectively reach promising regions are therefore desirable (more background in Appendix D).

In this work, we propose to solve the optimization problem in Equation (1) directly. Specifically, we use tools from offline *model-based optimization* (MBO) to model the target property $f(\mathsf{M})$ and search for candidate minimizers. Since the main obstacle is the irregular structure of the materials space, we adopt the *clique*-based MBO paradigm, which enables optimization over irregular designs by encoding them into clique-structured latent representations and optimizing those latents (Kuba et al., 2024). To make this approach applicable to CMD, for the first time, we introduce *CliqueFlowmer*, a neural-network model that maps materials into representations tractable by MBO.

3 CliqueFlowmer

This section introduces the architecture and the training algorithm of our MBO model for CMD. The neural network consists of several carefully designed components, built from standard deep learning primitives, that enable it to handle materials data. In Section 3.1 we introduce an encoder that maps a material M to a continuous latent representation $\mathbf{z}$. Then, Section 3.2 describes a decomposable model that predicts the property of interest from $\mathbf{z}$. Section 3.3 delineates a decoder that reconstructs the material—both the atom types and the geometry, from the representation. As we demonstrate later, such a setup allows to parameterize materials as continuous vectors which can be optimized for properties with gradient-based techniques (see Figure (2)).

3.1 Encoder

Our encoder has to combine four distinct pieces of information: lattice lengths (a, b, c), lattice angles (α, β, γ), atom positions $\mathbf{X}$, and atom types $\mathbf{a}$ (the latter two being irregular), and produce a fixed-dimensional continuous vector $\mathbf{z} \in \mathbb{R}^{d_{\mathbf{z}}}$. To do this, first, we map the continuous inputs to vectors of size d_{model} with their corresponding MLPs,

$$\mathbf{h}^{\text{len}} = \text{MLP}_\theta^{\text{len}}(a, b, c),\ \mathbf{h}^{\text{ang}} = \text{MLP}_\theta^{\text{ang}}(\alpha, \beta, \gamma) \in \mathbb{R}^{d_{\text{model}}},\ \&\ \mathbf{H}^{\text{pos}} = \text{MLP}_\theta^{\text{pos}}(\mathbf{X}) \in \mathbb{R}^{N_{\text{atom}} \times d_{\text{model}}}$$

and we map the atom types to continuous d_{model}-dimensional embeddings $\mathbf{H}^{\text{atom}} \in \mathbb{R}^{N_{\text{atom}} \times d_{\text{model}}}$. The MLP-produced hidden states are then concatenated, $\mathbf{H}^{\text{in}} = [\mathbf{h}^{\text{len}}, \mathbf{h}^{\text{ang}}, \mathbf{H}^{\text{pos}}]$, and passed into a transformer, where they are conditioned by $\mathbf{H}^{\text{atom}}$ via adaptive layer-norm (Peebles and Xie, 2023, AdaLN) in place of the standard layer-norm (Lei Ba et al., 2016; Vaswani et al., 2017),

$$\mathbf{H}^{\text{out}} = T_\theta^{\text{enc}}(\mathbf{H}^{\text{in}}, \mathbf{H}^{\text{atom}}).$$

For a single instance of N_{atom} atoms in the unit cell, the output of the transformer is a tensor of shape $(2 + N_{\text{atom}}) \times d_{\text{model}}$, thus depending on the cell size. To produce a fixed-dimensional representation $\mathbf{z}$, we pool the tensor along the first dimension, using attention with a learnable query vector $\mathbf{q} \in \mathbb{R}^{d_{\text{model}}}$, yielding a hidden state,

$$\mathbf{h}^{\text{pool}} = \text{Att}(\mathbf{q}, \mathbf{H}^{\text{out}}\mathbf{W}^K, \mathbf{H}^{\text{out}}\mathbf{W}^V) \in \mathbb{R}^{d_{\text{model}}}, \tag{2}$$

which is further post-processed by GELU (Hendrycks and Gimpel, 2016) and layer-norm, yielding $\mathbf{h}^{\text{post}}$. It is then fed to a linear layer that produces the mean $\mu_{\mathbf{z}}$ and log-standard deviation $\log \sigma_{\mathbf{z}}$ parameters of the normal distribution of the latent representation. We highlight that the pooling layer from Equation (2) is what enables us to navigate the transdimensional materials space $\mathcal{M}$ with a fixed-dimensional vector space, paving the road for MBO.

[4] Maximization problems can be represented analogously and solved using the same techniques.

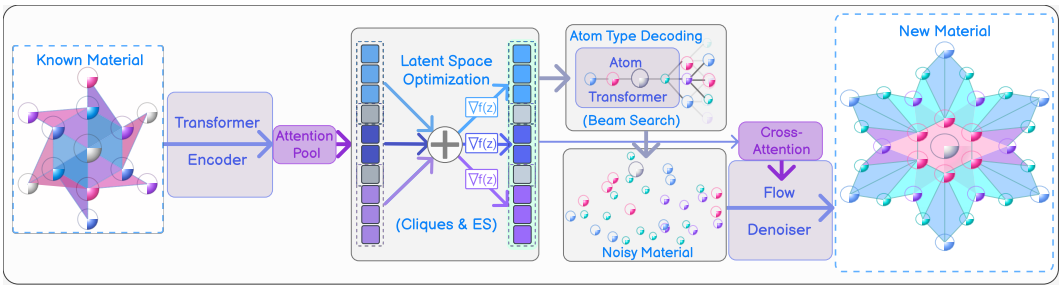


Figure 2: Computational materials discovery through MBO with CliqueFlowmer. Known materials are encoded by a transformer, and an attention-based pooling layer, into a fixed-dimensional latent space. The latent variable admits a clique decomposition, with respect to the target property, where each clique contributes additively to the target property. The representations are optimized with evolution strategies. Atom types are then decoded with an autoregressive transformer and beam search. The flow model reconstructs the material's geometry from the atom sequence and the latent representation by which it is conditioned via cross-attention.

3.2 Predictor

Once sampled, in addition to the standard flat form, the latent vector attains a form of a chain of cliques $\texttt{chain}(\mathbf{z}, d_{\text{clique}}, d_{\text{knot}})$ of clique size d_{clique} and knot size d_{knot}. That is, it is a $N_{\text{clique}} \times d_{\text{clique}}$ matrix $\mathbf{Z}$ with entry values from $\mathbf{z}$ such that $\mathbf{Z}_{c,i} = \mathbf{z}_k$, where $k = (c-1) \cdot (d_{\text{clique}} - d_{\text{knot}}) + i$, *i.e.*, the last d_{knot} entries of row c and the first d_{knot} entries of row $c+1$, for $c = 1, \ldots, N_{\text{clique}} - 1$, are equal. We write $\mathbf{Z}_c$ to denote its cth row. The matrix is then fed to an MLP predictor of property $\mathbf{y} = f(\mathsf{M})$ that decomposes its prediction over the cliques of $\mathbf{Z}$,

$$f(\mathsf{M}) \approx f_\theta(\mathbf{z}) = \sum_{c=1}^{N_{\text{cliques}}} f_\theta(\mathbf{Z}_c, c).$$

Imposing such a decomposable structure on the latent space is known to improve the effectiveness of MBO (Kuba et al., 2024, & Appendix D.4). Intuitively (see Figure 2), it enables composing optimal in-distribution examples of each clique to form a competitive in-distribution solution—a property also known as *stitching* (Fu et al., 2020; Shin et al., 2025). See Appendix B.3 for ablation studies.

3.3 Decoder

The decoder consists of two modules: the atom type decoder and the geometry decoder. The former infers the atom types given the latent representation, $p_\theta(\mathbf{a}|\mathbf{z})$, and the latter infers the unit cell geometry given the latent representation and the atom types, $p_\theta(\mathbf{G}|\mathbf{z}, \mathbf{a})$.

Atom types. The atom type decoder is quite straightforward—we utilize the standard causal transformer with next-token prediction which we condition on the latent representation via AdaLN. To that end, we map the $d_{\mathbf{z}}$-dimensional $\mathbf{z}$ to the d_{model}-dimensional space and then use it in combination with adaptive layer-norm inside of the causal transformer to produce a hidden state

$$\mathbf{z}^{\text{mod}} = \text{LayerNorm}(\text{GELU}(\text{Lin}_\theta(\mathbf{z}))), \quad \& \quad \mathbf{h}_{k+1} = T^{\text{dec}}_\theta(\mathbf{a}_{0:k}, \mathbf{z}^{\text{mod}})_{k+1}, \tag{3}$$

for $k = 0, \ldots, N_{\text{atom}}$, where a_0 and $\mathrm{a}_{N_{\text{atom}}+1}$ are $\langle\text{Start}\rangle$ and $\langle\text{Stop}\rangle$ tokens. We found the layer-norm very helpful at ensuring equitable training progress of all modalities. That hidden state is then refined into log-likelihoods with an MLP and a log-softmax layer that predicts the next (atom type) token.

Geometry. We decode the shape of the unit cell and the atom positions with a continuous normalizing flow, conditioned on $\mathbf{z}$, for which we build another specialized transformer. We use the flow-matching framework, wherein we decode the material geometry $\mathbf{G} = [(a, b, c),\ (\alpha, \beta, \gamma),\ \mathbf{X}]$ by, first, initializing at a sample from a prior distribution and, then, solving an ODE,

$$\text{Sample} \quad \mathbf{G}_0 = [(a, b, c),\ (\alpha, \beta, \gamma),\ \mathbf{X}]_0 \sim p_0(\mathbf{G}|\mathbf{a}), \quad \text{Solve} \quad \mathrm{d}\mathbf{G}_t = V_\theta(\mathbf{G}_t, t|\mathbf{a}, \mathbf{z})\mathrm{d}t,$$

from $t = 0$ to $t = 1$, where $V_\theta(\mathbf{G}_t, t|\mathbf{a}, \mathbf{z})$ is a transformer denoiser. The prior distribution $p_0(\mathbf{G}|\mathbf{a}) = p_0^{\text{len}}(a, b, c|\mathbf{a}) \cdot p_0^{\text{ang}}(\alpha, \beta, \gamma|\mathbf{a}) \cdot p_0^{\text{pos}}(\mathbf{X}|\mathbf{a})$ is modeled meticulously—first, the angles are modeled as random uniform in interval $(\pi/3, 2\pi/3)$, and positions are random uniform on $(0, 1)$. The prior

over the lengths is chosen so that the density of the unit cell $\mathrm{den}(\mathsf{M}) = N_{\text{atom}}/\,\mathrm{Vol}(\mathsf{M})$ is invariant to the number of atoms (more details in Appendix C.2). Similarly to the encoder, given a noisy unit cell shape and atom positions $\mathbf{G}_t = [(a, b, c), (\alpha, \beta, \gamma), \mathbf{X}]_t$ at time t, we embed them in the d_{model}-dimensional space with MLPs. This is then fed to a (non-causal) transformer that is conditioned on the embeddings of the decoded atom types $\mathbf{H}^{\text{atom}}$ via AdaLN. Crucially, each block of the transformer contains a cross-attention layer (Vaswani et al., 2017) between the hidden states and the latent representations in the form $\mathbf{Z} = \texttt{chain}(\mathbf{z}, d_{\text{clique}}, d_{\text{knot}})$. More specifically, cliques of $\mathbf{Z}$ get first mapped to the d_{model}-dimensional space and a cross-attention layer is added between the self-attention and feed-forward layers of the transformer block

$$\mathbf{H}^{\mathbf{z}} = \mathrm{LayerNorm}(\mathrm{GELU}(\mathrm{MLP}^{\text{lat}}_{\theta}(\mathbf{Z}))) \quad \& \quad \mathbf{H}^{\text{cross}}_t = \mathbf{H}^{\text{self}}_t + \mathrm{CrossAtt}(\mathbf{H}^{\text{self}}_t, \mathbf{H}^{\mathbf{z}})$$

(more information in Appendix C.1). The output hidden state $\mathbf{O}_t$ of the transformer is then decomposed into the lengths, angles, and position parts, and each of them is used to predict the corresponding modality with an MLP,

$$\hat{\mathbf{v}}^{\text{len}}_t = \mathrm{MLP}^{\text{len}}_{\theta}(\mathbf{o}^{\text{len}}_t), \quad \hat{\mathbf{v}}^{\text{ang}}_t = \mathrm{MLP}^{\text{ang}}_{\theta}(\mathbf{o}^{\text{ang}}_t), \quad \hat{\mathbf{V}}^{\text{pos}}_t = \mathrm{MLP}^{\text{pos}}_{\theta}(\mathbf{O}^{\text{pos}}_t)$$

which together form the prediction from our denoiser, $V_\theta(\mathbf{G}_t, t|\mathbf{a}, \mathbf{z}) = [\hat{\mathbf{v}}^{\text{len}}_t, \hat{\mathbf{v}}^{\text{ang}}_t, \hat{\mathbf{V}}^{\text{pos}}_t]$. Lastly, to sample the denoising timestep for training, we draw a sample from, our novel, *lifted logit-normal* distribution, which is a Bernoulli mixture of the logit-normal distribution and the uniform distribution (consult Appendix C.4 for more information).

3.4 Training

The goal of training CliqueFlowmer is to learn a map $p_\theta(\mathbf{z}|\mathbf{a}, \mathbf{G})$ of a material $\mathsf{M} = [\mathbf{a}, \mathbf{G}]$ to a fixed-dimensional continuous vector $\mathbf{z}$, a decomposable approximator $f_\theta(\mathbf{z})$ of the target property, and the inverse $p_\theta(\mathbf{a}, \mathbf{G}|\mathbf{z}) = p_\theta(\mathbf{a}|\mathbf{z})p_\theta(\mathbf{G}|\mathbf{a}, \mathbf{z})$ of the encoding map. Thus, to train CliqueFlowmer, we must combine training of its four integrated submodules. At the nucleus of this fusion there is sampling the latent representation $\mathbf{z} \sim p_\theta(\mathbf{z}|\mathbf{a}, \mathbf{G})$—given this vector, the atom type decoder learns to maximize the conditional next-token log-likelihood

$$-\mathcal{L}_{\text{atom}}(\mathbf{a}, \mathbf{z}) = \sum_{i=0}^{N_{\text{atom}}} \log p_\theta(\mathrm{a}_{i+1}|\mathbf{a}_{0:i}, \mathbf{z}),$$

where $\mathrm{a}_0 = \langle\mathrm{Start}\rangle$ and $\mathrm{a}_{N_{\text{atom}}+1} = \langle\mathrm{Stop}\rangle$. Meanwhile, the flow-matching model that uses cross-attention to condition on the clique form $\mathbf{Z}$ of the latent, samples a noise geometry $\mathbf{G}_0$ for a cell with N_{atom} atoms, interpolates it with the ground-truth geometry at a random timestep, $\mathbf{G}_t = (1-t)\cdot\mathbf{G}_0 + t\cdot\mathbf{G}$, and minimizes the difference between the velocity network's prediction and the path between geometries,

$$\begin{aligned}
\mathcal{L}_{\text{len}}(\mathbf{a}, \mathbf{z}, \mathbf{G}, \mathbf{G}_0) &= \big((a-a_0, b-b_0, c-c_0) - \hat{\mathbf{v}}^{\text{len}}_t\big)^2 \\
\mathcal{L}_{\text{ang}}(\mathbf{a}, \mathbf{z}, \mathbf{G}, \mathbf{G}_0) &= \big((\alpha-\alpha_0, \beta-\beta_0, \gamma-\gamma_0) - \hat{\mathbf{v}}^{\text{ang}}_t\big)^2 \\
\mathcal{L}_{\text{pos}}(\mathbf{a}, \mathbf{z}, \mathbf{G}, \mathbf{G}_0) &= \sum_{i=1}^{N_{\text{atom}}} \big((\mathbf{x}^i - \mathbf{x}^i_0) - \hat{\mathbf{v}}^{\text{pos},i}_t\big)^2.
\end{aligned}$$

While training the flow denoiser, we mask the latent $\mathbf{z}$ and replace it with a random normal noise, $\epsilon_{\mathbf{z}} \sim \mathcal{N}(\mathbf{0}_{d_{\mathbf{z}}}, \mathbf{I}_{d_{\mathbf{z}}})$ with probability $p_{\text{lat}} = 0.1$. We decompose the flow loss into distinct components to ensure that the unit cell (lengths and angles) receives equal weight for every material in the batch. For clarity, we define the sum of flow losses as

$$\mathcal{L}_{\text{flow}}(\mathbf{a}, \mathbf{z}, \mathbf{G}, \mathbf{G}_0) = \mathcal{L}_{\text{len}}(\mathbf{a}, \mathbf{z}, \mathbf{G}, \mathbf{G}_0) + \mathcal{L}_{\text{ang}}(\mathbf{a}, \mathbf{z}, \mathbf{G}, \mathbf{G}_0) + \tau_{\text{pos}}\cdot\mathcal{L}_{\text{pos}}(\mathbf{a}, \mathbf{z}, \mathbf{G}, \mathbf{G}_0),$$

where τ_{pos} is a positive scalar. Further, the latent representation, in its clique form, is also tasked with minimizing the prediction error of the target property and the clique-based KL-divergence is computed for the parameters of the normal distribution of $\mathbf{z}$,

$$\mathcal{L}_{\text{pred}}(\mathsf{M}, \mathbf{z}) = \big(f_\theta(\mathbf{z}) - f(\mathsf{M})\big)^2 \quad \& \quad \mathcal{L}_{\text{lat}}(\mu_{\mathbf{z}}, \sigma_{\mathbf{z}}) = \mathrm{KL}\big(\mathcal{N}(\mu_{\mathbf{z},c}, \sigma^2_{\mathbf{z},c}),\ \mathcal{N}(\mathbf{0}_{d_{\text{clique}}}, \mathbf{I}_{d_{\text{clique}}})\big), \quad (4)$$

where the clique c is drawn from the uniform distribution $\mathcal{U}[N_{\text{clique}}]$ (Kuba et al., 2024). To summarize, the total expected loss takes form

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathsf{M}\sim\mathcal{D}, \mathbf{z}\sim p_\theta, \mathbf{G}_0\sim p_0}\big[\mathcal{L}_{\text{atom}}(\mathbf{a}, \mathbf{z}) + \mathcal{L}_{\text{flow}}(\mathbf{a}, \mathbf{z}, \mathbf{G}, \mathbf{G}_0) + \tau_{\text{pred}}\mathcal{L}_{\text{pred}}(\mathsf{M}, \mathbf{z}) + \beta\mathcal{L}_{\text{lat}}(\mu_{\mathbf{z}}, \sigma_{\mathbf{z}})\big],$$

where β and τ_{pred} are positive scalars which we warm up linearly one after another. While this loss consists of multiple terms, the architectural building blocks introduced in the previous subsections allow for their minimization and, as a result, learning representations that enable MBO for CMD.

Metric	CrystalFormer	DiffCSP	DiffCSP++	MatterGen	**CliqueFlowmer**	**-Top**
E_{form} (↓)	0.71	0.59	0.65	0.60	−0.81	**−0.99**
Stable (↑)	14.1	19.2	**19.8**	18.5	14.5	13.6
S.U.N. (↑)	12.8	**18.6**	18.5	17.6	14.4	13.4
E_{form} / S.U.N. (↓)	1.06	1.02	1.10	0.98	−0.65	**−1.06**
Band Gap (↓)	0.52	0.63	0.48	0.57	**0.03**	0.07
Stable (↑)	14.1	19.2	19.8	18.5	61.3	**69.4**
S.U.N. (↑)	12.8	18.6	18.5	17.6	61.3	**69.4**
Band Gap / S.U.N. (↓)	0.37	0.55	0.23	0.40	**0.02**	0.05

Table 1: Target properties and stability metrics across generative baselines and CliqueFlowmer. Formation energies are in eV/atom, band gaps in eV, and Stable and S.U.N. rates in %. "Property / S.U.N." denotes the mean property among S.U.N. structures, retaining the property's units. Row-wise best results are **bold teal**; second-best are underlined blue. Among other results, **CliqueFlowmer** and **CliqueFlowmer-Top** reduce average formation energy from 0.46 in the initial samples to -0.81 and -0.99, respectively, and reduce band gap from 0.57 to 0.03 and 0.07. More metrics in Appendix A.1.

3.5 Computational Materials Discovery with CliqueFlowmer

To optimize new materials with our model, we conduct a form of gradient-based search in the representation space. We initialize this process by encoding a sample of existing materials with the encoder, $\mathbf{z} = \text{Enc}_\theta(\mathbf{a}, \mathbf{G})$. Then, we optimize the prediction of the target property,

$$\mathbf{z}_\star = \arg\min_{\mathbf{z}^{\text{new}}} f_\theta(\mathbf{z}^{\text{new}}). \tag{5}$$

The choice of the minimization algorithm plays a significant role. While we have found that exact back-propagation of Equation (5) is prone to adversarial exploitation of the model, back-propagation-free evolution strategies (Salimans et al., 2017, ES) were very effective (see Section 4.2). That is, at every iteration, we draw N_{pert} noise vectors ϵ from the standard normal distribution, $\epsilon \sim \mathcal{N}(\mathbf{0}_{d_\mathbf{z}}, \mathbf{I}_{d_\mathbf{z}})$, and compute the predicted values $f_\theta(\mathbf{z} + \sigma\epsilon)$ of the perturbations of $\mathbf{z}$ at scale σ. Then, we rank these perturbations, $\epsilon^1, \ldots, \epsilon^{N_{\text{pert}}}$, based on the predicted values of f (so that the smallest value gets the lowest rank). We standardize the ranks to have zero mean and unit variance and compute the ES gradient as a linear combination of the perturbations weighted by the ranks. Additionally, we use antithetic sampling (see Appendix B.1 for more information about ES). Then, we take a gradient step with AdamW optimizer (Loshchilov et al., 2017). The weight decay step of the optimizer, $\mathbf{z} \mapsto (1 - \eta\lambda)\mathbf{z}$, where η is the learning rate, is crucial since it brings the optimized latent variable closer to the origin. In effect, this step increases the likelihood of the latent under the Gaussian prior that CliqueFlowmer was trained with (see Equation (4)), which mitigates the problem of going out of distribution with $\mathbf{z}$.

Given an optimized latent representation $\mathbf{z}_\star$, we modulate it like in Equation (3), and pass to the atom-type decoder, from which we sample the atom-type sequence using beam search (see Appendix D.3) with beam width $N_{\text{beam}} = 10$,

$$\mathbf{a}_\star \sim \text{BeamSearch}(T_\theta^{\text{dec}}, \mathbf{z}_\star^{\text{mod}}).$$

Next, we consider the clique form of the latent representation, $\mathbf{Z}_\star = \texttt{chain}(\mathbf{z}_\star, d_{\text{clique}}, d_{\text{knot}})$, and pass it to our continuous normalizing flow model, from which we decode the material geometry. For this end, we use the *Euler method* with *classifier-free guidance* (Ho and Salimans, 2022, CFG), in which the effective velocity used in the ODE is

$$V_\theta^\omega(\mathbf{G}_t, t|\mathbf{a}_\star, \mathbf{z}_\star) = (1 + \omega) \cdot V_\theta(\mathbf{G}_t, t|\mathbf{a}_\star, \mathbf{z}_\star) - \omega \cdot V_\theta(\mathbf{G}_t, t|\mathbf{a}_\star, \epsilon_\mathbf{z}),$$

where $\epsilon_\mathbf{z}$ is standard normal noise. In our experiments, we used $\omega = 2$ (see Appendix C.3 for ablations). We provide a discussion on the time complexity of this procedure in Appendix A.4.

4 Experiments

We tested CliqueFlowmer in two tasks for which we trained two separate models. Both of them were trained on the MP-20 dataset, with the standard 60-20-20% train-val-test split, that contains 45K example materials altogether (Xie et al., 2021) for a total of 7×10^5 gradient steps with batch size of 1024. In the first task, for the role of the target property, we chose the predicted formation energy per

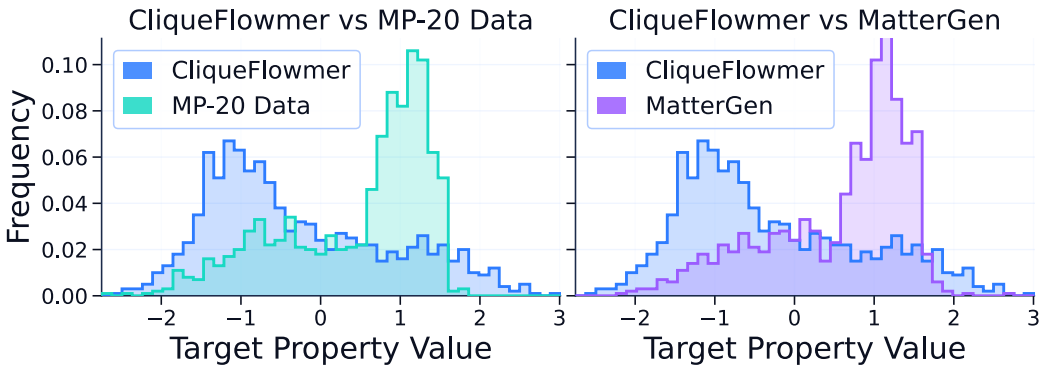
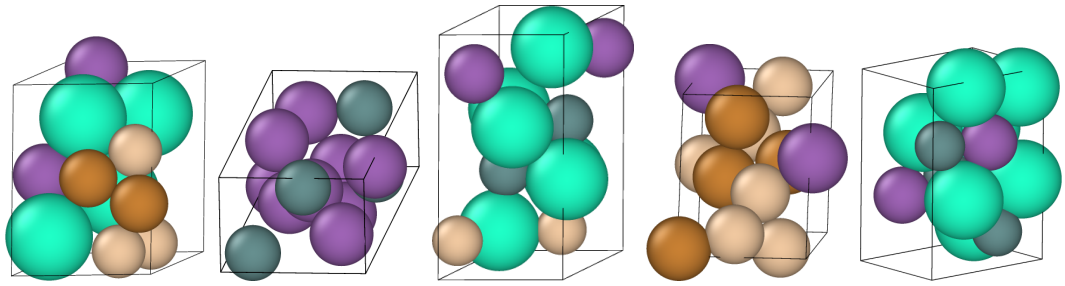


Figure 3: The distribution of formation energy per atom (horizontal axis, eV/atom) among discovered materials. We compare CliqueFlowmer (blue) to materials from MP-20 dataset (green, *left*) and to those generated by MatterGen (purple, *right*). CliqueFlowmer significantly shifts the distribution of the target property to its lower values. For more histograms, see Appendix A.3.



(a) Dy_2CuSi_2Sb 0.06 eV (b) Ge_4Sb_9 0.01 eV (c) $Dy_3SiGeSb$ 0.14 eV (d) Cu_2Si_3Sb 0.02 eV (e) $Dy_7(Ge_2Sb)_2$ 0.03 eV

Figure 4: Examples of crystals optimized by CliqueFlowmer for band-gap minimization. Each panel shows a unit cell, its composition, and its reported band gap in eV.

atom $E_{\text{form}}(\mathsf{M})$ (eV/atom) from M3GNet (Chen and Ong, 2022), and for the second one, the band gap $\Delta_{\text{band}}(\mathsf{M})$ (eV) predicted by MEGNet (Chen et al., 2019). Formation energy provides an accessible benchmark for property optimization, although minimizing it alone does not ensure thermodynamic stability. The band-gap optimization objective includes a formation-energy penalty (Appendix E).

4.1 MBO With CliqueFlowmer

To test CliqueFlowmer's ability to conduct CMD through MBO, we use the following protocol. We sample N existing materials and encode them with the CliqueFlowmer encoder. Then, we optimize the materials' representations, using $T = 2000$ ES-gradient steps. The converged representations are then decoded with the CliqueFlowmer decoder, thus rendering N new materials (see Figure 4 for examples). Additionally, one can filter out the most promising candidates prior to their structure-level oracle evaluation. Namely, once optimization converges, one can predict the target property with the prediction head, $f_\theta(\mathbf{z})$, and select top-$k\%$ solutions with the best values. The attractiveness of this approach stems from the computational savings it offers—having run the cheap optimization stage in the latent space, one can reject the majority of suboptimal latents before passing the most promising ones to the expensive denoising and relaxation stages (Appendix A.4 discusses the wall-clock time complexity of our method). We refer to this approach *CliqueFlowmer-Top*, set $k = 10$ in our experiments, and consider it the final method of our paper.

We measure the target property value of the $N = 10^4$ original materials and the value of the newly-proposed ones. Additionally, to examine if our method produces both interesting and stable materials, we calculate the S.U.N. rate (stable, unique, novel) (Zeni et al., 2023) among the proposed solutions. Then, to study the optimization performance among the interesting materials, the target property is re-evaluated for examples that are classified as S.U.N. We compare our results to CrystalFormer (Cao et al., 2025), DiffCSP (Jiao et al., 2023), DiffCSP++ (Jiao et al., 2024), and MatterGen (Zeni et al., 2023), whose generated structures we obtained from Kazeev et al. (2025). Due to computational

limitations, we relax all the materials (including those from the baselines) approximately with M3GNet (Chen and Ong, 2022) and determine stability with energy above local hull[5]. To compensate for the bias induced by these approximations, we use the *strict* criterion for stability (see Appendix D.1). We also evaluate selected candidates with VASP using Materials Project relaxation inputs and energy corrections; Appendix A.2 details this DFT protocol and its implications for stability estimates.

Results in Table 1 and Figure (3) show that CliqueFlowmer drastically reduces the value of the target property (M3GNet formation energy & MEGNet band gap), significantly outperforming materials generated by the baselines. In the formation energy task, this quality is further amplified by CliqueFlowmer-Top, which reduces the objective value even further. We suspect that, in the band gap task, we do not observe this effect because the band gap is lower-bounded by zero, and thus the ordering of the top materials in the latent space may be driven by approximation errors. Furthermore, while the purpose of CliqueFlowmer is property optimization, the structures proposed by the model outperform the baselines in terms of the S.U.N. rate in the band gap optimization task. This may be surprising given that it is formation energy that is more tightly related to stability. However, it is known in materials science that simply minimizing the formation energy may lead to low-energy regions of the materials space with many competing phases, resulting in fewer stable materials (Riebesell et al., 2025). Furthermore, in both tasks, the S.U.N. materials from our method outperformed those from the baselines, in terms of the target property, regardless of the S.U.N. rate. Ultimately, the strong optimization performance (overall and /S.U.N) and high S.U.N. rates exceeded our expectations. See Appendix B.2 & E for more information.

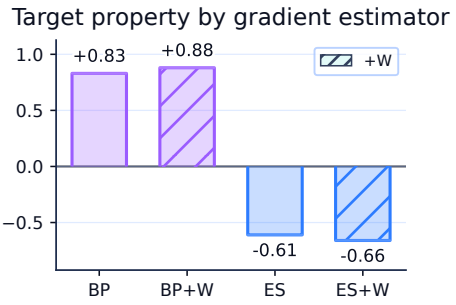


Figure 5: Comparison of gradient estimators—back-propagation (BP) vs. evolution strategies (ES), and weight decay. We plot the average change in formation energy per atom (eV/atom) due to optimization, over 100 materials.

4.2 Optimization Algorithm

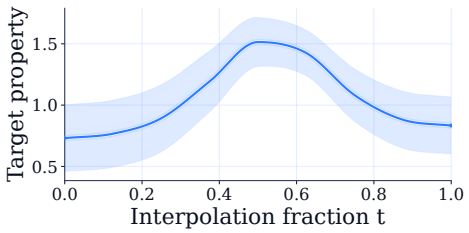


Figure 6: Average $f(\mathsf{M})$ (formation energy per atom, eV/atom) with standard error of the mean (SEM) over the course of linear interpolation. The curve was smoothed out with the Gaussian kernel. The value tends to be higher along the interpolation trajectory.

As described in Section 3.5, having trained CliqueFlowmer, we discover new materials by solving the optimization problem from Equation (5). In this paper, we utilized the back-propagation-free ES algorithm (Salimans et al., 2017). This section provides an empirical justification of this counterintuitive choice. We compare the performance of the gradient derived by back-propagation (BP) to ES by optimizing $N = 100$ structures, in CliqueFlowmer's latent space, for $T = 1000$ steps. In our experiments, both gradients were applied by Adam optimization algorithm (Kingma and Ba, 2014), as well as by its decoupled weight decay variant (Loshchilov et al., 2017, BP+W, ES+W). Figure (5) shows that, instead of decreasing the minimized property, BP and BP+W have drastically increased it. Meanwhile, ES has successfully progressed towards the property minimization, and ES+W has done that most effectively. Thus, our main experiments use ES+W. More information in Appendix B.2.

4.3 Material Representations

Our model learns representations that are meant to navigate the transdimensional space of materials smoothly. To demonstrate their ability to do so, we interpolate two materials, $\mathsf{M}^{(0)}$ and $\mathsf{M}^{(1)}$, by linearly mixing their representations $\mathbf{z}^{(0)}$ and $\mathbf{z}^{(1)}$. Figure 7 shows decoded interpolations for five pairs of MP-20 materials. We use Pymatgen's 2D plots for better clarity (Ong et al., 2013). The resulting materials gradually evolve from $\mathsf{M}^{(0)}$ to $\mathsf{M}^{(1)}$ by altering their unit cell shape, atom positions,

[5]As a result, our Stable and S.U.N. rates overestimate the true numbers but they do so for all methods.

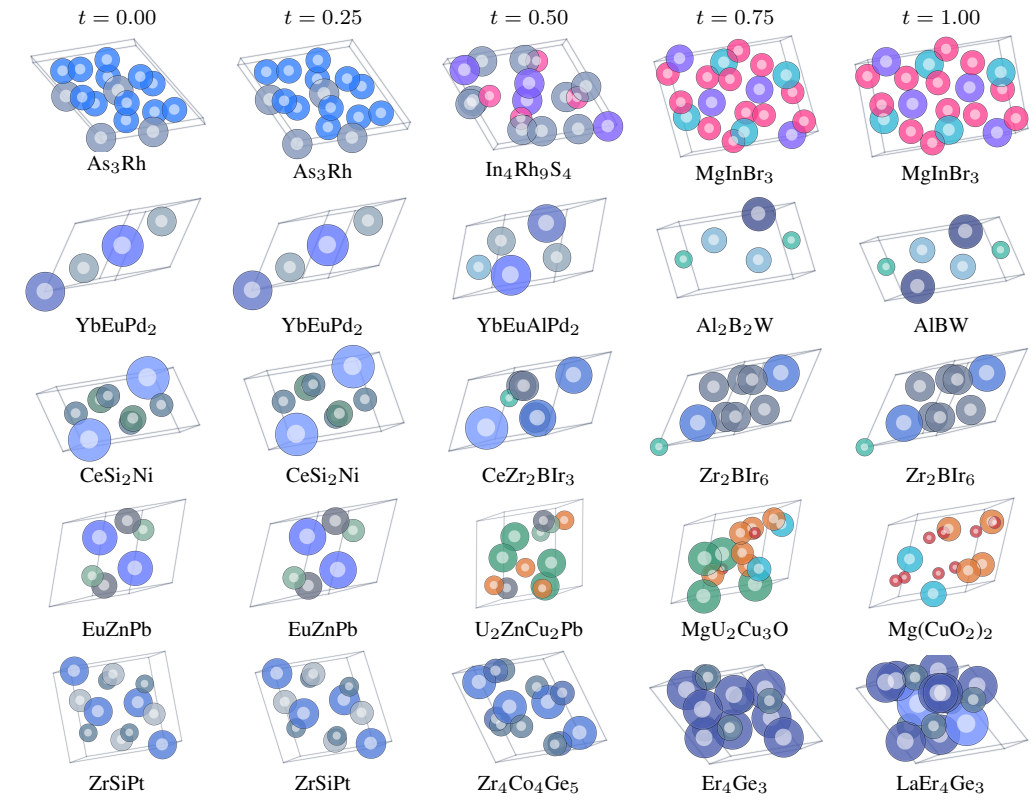


Figure 7: **Latent interpolation of materials.** Each row decodes $\mathbf{z}^{(t)} = (1-t)\mathbf{z}^{(0)} + t\mathbf{z}^{(1)}$ between a pair of MP-20 materials. Columns show $t = 0, 0.25, 0.50, 0.75, 1$. Cell shapes, atom positions, and compositions vary along the paths; most composition changes occur at intermediate values of t.

and the atomic composition (which happens mainly between the inner timesteps), indicating that the latents smoothly navigate the materials space. This result motivates examining if one can obtain well-performing (in terms of $f(\mathsf{M})$) materials by mixing existing structures, similarly to a study in Gómez-Bombarelli et al. (2018). To quantify how promising that is, we sample $N_{\text{pair}} = 10$ pairs of existing materials and conduct an 8-step linear interpolation, at the same time evaluating $f(\mathsf{M})$ of the decoded, mixed structures. The results in Figure (6) show that the target property, unfortunately, tends to be higher along the interpolation trajectory. In particular, it reaches its apex in the middle of interpolation. Nevertheless, the failure of this naïve approach strengthens the motivation for more sophisticated tools, like MBO.

Intervening on individual cliques. We also interpolate a single clique between the latent encodings $\mathbf{z}^{(0)}$ and $\mathbf{z}^{(1)}$ of As_3Rh and $MgInBr_3$. Let I_c denote the coordinate indices belonging to clique c (Section 3.2). For $t \in [0, 1]$, we construct

$$z_i^{(t)} = \begin{cases} (1-t)z_i^{(0)} + tz_i^{(1)}, & i \in I_c, \\ z_i^{(0)}, & i \notin I_c. \end{cases} \tag{6}$$

Coordinates outside the selected clique remain fixed; shared coordinates also affect neighboring cliques. We decode the resulting latents for $c \in \{1, 6\}$ and $t \in \{0, 0.25, 0.50, 0.75, 1\}$. Figure 8 shows that clique 1 changes atomic positions and atom counts while retaining As and Rh, whereas clique 6 introduces V. Note that, since only part of the latent changes, interpolations at $t = 1$ need not decode to $MgInBr_3$. These examples suggest distinct effects of clique interventions, without establishing a unique physical interpretation for each clique.

5 Related Work

Automating computational materials discovery has motivated growing interest in deep generative modeling for materials (Jain et al., 2013). Representative approaches include diffusion-based models

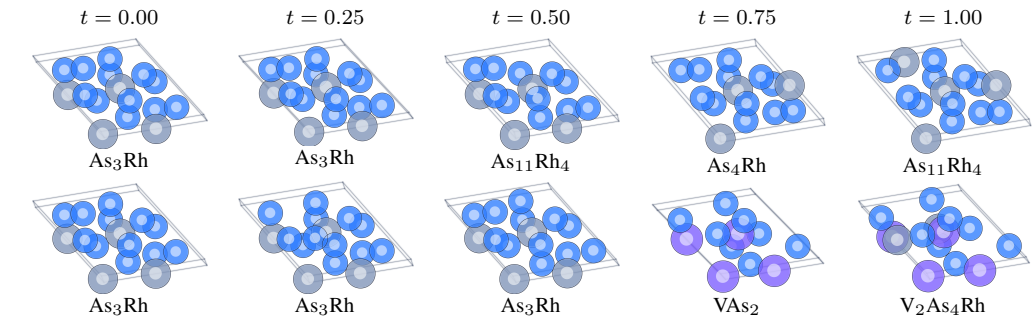


Figure 8: **Single-clique interpolation.** Coordinates of clique 1 (top) or clique 6 (bottom) are interpolated from As_3Rh toward $MgInBr_3$, while coordinates outside the selected clique remain fixed. Clique 1 retains As and Rh while changing geometry and atom count; clique 6 introduces V at later timesteps. Only a subset of coordinates is interpolated, so $t = 1$ need not decode to $MgInBr_3$.

such as CDVAE (Xie et al., 2021) and MatterGen (Zeni et al., 2023), flow-based generators such as FlowMM (Miller et al., 2024), and autoregressive models such as CrystalFormer (Cao et al., 2025). These methods primarily focus on likelihood-based generation or property-conditioned sampling, whereas our focus is direct property optimization. Unlike prior materials generators, CliqueFlowmer compresses transdimensional crystal structures into fixed-dimensional latent representations explicitly designed for optimization. It combines transformers and latent-variable modeling similarly in spirit to Joshi et al. (2025), but, crucially, supports compression and gradient-based search in materials space while avoiding expensive equivariant graph architectures (Li et al., 2025).

A parallel line of work steers diffusion samplers toward property-favorable Boltzmann distributions (Li et al., 2024; Uehara et al., 2024; Tan et al., 2025; Liu et al., 2025). While these approaches tilt generative sampling toward desirable regions, they do not directly solve optimization over materials. More closely related are offline model-based optimization methods such as MINs (Kumar and Levine, 2020), COMs (Trabucco et al., 2021), MatchOpt (Hoang et al., 2025), and especially Cliqueformer (Kuba et al., 2024), which introduced clique-based MBO in transformers. However, prior MBO methods target simplified benchmark domains and are not compatible with crystalline materials. CliqueFlowmer bridges this gap by integrating structured MBO with a domain-specific generative architecture for direct optimization in computational materials discovery.

6 Limitations & Conclusion

In this work, we proposed addressing the problem of computational materials discovery (CMD) with the tools from offline model-based optimization (MBO). We introduced CliqueFlowmer—a model that represents materials as continuous vectors and optimizes them with clique-based MBO. Our experimental results with CMD apparatus, such as MP-20 data and target property oracles, show that CliqueFlowmer enables efficient optimization of materials offline. The major limitation of our work is the impact of optimized properties—while the most easily measurable with oracles, they are not of the greatest interest of industrial labs that design materials for specialized applications. Together with increased access to oracles of more properties, we expect CliqueFlowmer to solve more impactful problems. Appropriate regulatory oversight is essential to ensure that discovered materials are deployed safely and for societal benefit. Another limitation is the accuracy of our evaluation. For example, some works estimate properties of generated materials with software conducting (high-cost) density functional theory calculations which are known to be very accurate. Meanwhile we employed machine learning-based oracles in the majority of our evaluations. These can induce errors, as confirmed by our limited DFT checks. Overall, the most strongly validated conclusion of our study is that CliqueFlowmer is highly effective at offline property optimization; by contrast, absolute stability-based metrics such as S.U.N. are more sensitive to the surrogate-oracle stack and should be interpreted with greater caution. Furthermore, while we focused on property optimization in our research, the continuous representations derived from CliqueFlowmer can be employed by researchers in any manner desired. Lastly, we hope that this work will inspire the CMD community to add MBO to its toolkit.

Acknowledgments and Disclosure of Funding

Kuba Grudzien's research is supported by a grant from NSF AI4OPT AI Institute for Advances in Optimization (Award 2112533). Pieter Abbeel holds concurrent appointments as a Professor at UC Berkeley and as an Amazon Scholar. This paper describes work performed at UC Berkeley and is not associated with Amazon.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Zhendong Cao, Xiaoshan Luo, Jian Lv, and Lei Wang. Space group informed transformer for crystalline materials generation. *Science Bulletin*, 2025.

Chi Chen and Shyue Ping Ong. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science*, 2(11):718–728, 2022.

Chi Chen, Weike Ye, Yunxing Zuo, Chen Zheng, and Shyue Ping Ong. Graph networks as a universal machine learning framework for molecules and crystals. *Chemistry of Materials*, 31(9):3564–3572, 2019.

Earl Danielson, Josh H Golden, Eric W McFarland, Casper M Reaves, W Henry Weinberg, and Xin Di Wu. A combinatorial approach to the discovery and optimization of luminescent materials. *Nature*, 389(6654):944–948, 1997.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Thomas Freese, Nils Elzinga, Matthias Heinemann, Michael M Lerch, and Ben L Feringa. The relevance of sustainable laboratory practices. *Rsc Sustainability*, 2(5):1300–1336, 2024.

Nikolaus Frohner, Jan Gmys, Nouredine Melab, Günther Raidl, and El-Ghazali Talbi. Parallel beam search for combinatorial optimization. In *Workshop Proceedings of the 51st International Conference on Parallel Processing*, pages 1–8, 2022.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer, 2004.

Yasemin Gokcekuyu, Fatih Ekinci, Mehmet Serdar Guzel, Koray Acici, Sahin Aydin, and Tunc Asuroglu. Artificial intelligence in biomaterials: a comprehensive review. *Applied Sciences*, 14 (15):6590, 2024.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Kuba Grudzien, Masatoshi Uehara, Sergey Levine, and Pieter Abbeel. Functional graphical models: Structure enables offline data-driven optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2449–2457. PMLR, 2024.

Ning Han and Bao-Lian Su. Ai-driven material discovery for energy, catalysis and sustainability. *National Science Review*, 12(5):nwaf110, 2025.

Aaron Havens, Benjamin Kurt Miller, Bing Yan, Carles Domingo-Enrich, Anuroop Sriram, Brandon Wood, Daniel Levine, Bin Hu, Brandon Amos, Brian Karrer, et al. Adjoint sampling: Highly scalable diffusion samplers via adjoint matching. *arXiv preprint arXiv:2504.11713*, 2025.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Jochen Heyd, Gustavo E. Scuseria, and Matthias Ernzerhof. Hybrid functionals based on a screened coulomb potential. *The Journal of Chemical Physics*, 118(18):8207–8215, 2003. doi: 10.1063/1.1564060.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Minh Hoang, Azza Fadhel, Aryan Deshwal, Janardhan Rao Doppa, and Trong Nghia Hoang. Learning surrogates for offline black-box optimization via gradient matching. *arXiv preprint arXiv:2503.01883*, 2025.

C. A. R. Hoare. Quicksort. *The Computer Journal*, 5(1):10–16, 1961.

Dietmar W Hutmacher. Scaffolds in tissue engineering bone and cartilage. *Biomaterials*, 21(24): 2529–2543, 2000.

Theo Jaffrelot Inizan, Aaron Kaplan, Sherry Yang, Yen-hsu Lin, Mona Abdelgaid, Jian Yin, Zhiling Zheng, Saber Mirzaei, Ali H Alawadhi, Ekin D Cubuk, et al. Generative model for enhancing reticular material discovery. In *AI4X 2025*, 2025. URL https://openreview.net/pdf?id=m8pGrT9WDa.

Anubhav Jain, Geoffroy Hautier, Shyue Ping Ong, Charles J. Moore, Christopher C. Fischer, Kristin A. Persson, and Gerbrand Ceder. Formation enthalpies by mixing GGA and GGA+U calculations. *Physical Review B*, 84(4):045115, 2011. doi: 10.1103/PhysRevB.84.045115.

Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL materials*, 1(1), 2013.

Rui Jiao, Wenbing Huang, Peijia Lin, Jiaqi Han, Pin Chen, Yutong Lu, and Yang Liu. Crystal structure prediction by joint equivariant diffusion. *Advances in Neural Information Processing Systems*, 36:17464–17497, 2023.

Rui Jiao, Wenbing Huang, Yu Liu, Deli Zhao, and Yang Liu. Space group constrained crystal generation. *arXiv preprint arXiv:2402.03992*, 2024.

Chaitanya K Joshi, Xiang Fu, Yi-Lun Liao, Vahe Gharakhanyan, Benjamin Kurt Miller, Anuroop Sriram, and Zachary W Ulissi. All-atom diffusion transformers: Unified generative modelling of molecules and materials. *arXiv preprint arXiv:2503.03965*, 2025.

Nikita Kazeev, Daniel Levy, Rui Jiao, Andrey Okhotin, Ruiming Zhu, Wei Nong, et al. Generated crystals for wyformer, diffcsp, diffcsp++, wycryst, symmcd, crystalformer, miad & mattergen. figshare Dataset, 2025. URL https://doi.org/10.6084/m9.figshare.29145101.v4.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Scott Kirklin, James E Saal, Bryce Meredig, Alex Thompson, Jeff W Doak, Muratahan Aykol, Stephan Rühl, and Chris Wolverton. The open quantum materials database (oqmd): assessing the accuracy of dft formation energies. *npj Computational Materials*, 1(1):1–15, 2015.

Walter Kohn and Lu Jeu Sham. Self-consistent equations including exchange and correlation effects. *Physical review*, 140(4A):A1133, 1965.

G. Kresse and J. Furthmüller. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Computational Materials Science*, 6(1):15–50, 1996a. doi: 10.1016/0927-0256(96)00008-0.

G. Kresse and J. Furthmüller. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Physical Review B*, 54(16):11169–11186, 1996b. doi: 10.1103/PhysRevB .54.11169.

Jakub Grudzien Kuba, Pieter Abbeel, and Sergey Levine. Cliqueformer: Model-based optimization with structured transformers. *arXiv preprint arXiv:2410.13106*, 2024.

Aviral Kumar and Sergey Levine. Model inversion networks for model-based optimization. *Advances in Neural Information Processing Systems*, 33:5126–5137, 2020.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Aviral Kumar, Amir Yazdanbakhsh, Milad Hashemi, Kevin Swersky, and Sergey Levine. Data-driven offline optimization for architecting hardware accelerators. *arXiv preprint arXiv:2110.11346*, 2021.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *ArXiv e-prints*, pages arXiv–1607, 2016.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024. URL https://arxiv.org/abs/2408.08252.

Yunyang Li, Lin Huang, Zhihao Ding, Xinran Wei, Chu Wang, Han Yang, Zun Wang, Chang Liu, Yu Shi, Peiran Jin, et al. E2former: An efficient and equivariant transformer with linear-scaling tensor products. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Dian-Zhao Lin, Kai-Jui Pan, Yuyin Li, Charles B Musgrave III, Lingyu Zhang, Krish N Jayarapu, Tianchen Li, Jasmine Vy Tran, William A Goddard III, Zhengtang Luo, et al. A high-throughput experimentation platform for data-driven discovery in electrochemistry. *Science Advances*, 11(14): eadu4391, 2025.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Guan-Horng Liu, Jaemoo Choi, Yongxin Chen, Benjamin Kurt Miller, and Ricky TQ Chen. Adjoint schr\" odinger bridge sampler. *arXiv preprint arXiv:2506.22565*, 2025.

Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5, 2017.

Benjamin Kurt Miller, Ricky TQ Chen, Anuroop Sriram, and Brandon M Wood. Flowmm: Generating materials with riemannian flow matching. *arXiv preprint arXiv:2406.04713*, 2024.

Daniel G Nocera. Chemistry of personalized solar energy. *Inorganic chemistry*, 48(21):10001–10017, 2009.

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.

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. *Computational Materials Science*, 68:314–319, 2013.

Hyunsoo Park, Zhenzhu Li, and Aron Walsh. Has generative artificial intelligence solved inverse materials design? *Matter*, 7(7):2355–2367, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 2002.

Janosh Riebesell, Rhys EA Goodall, Philipp Benner, Yuan Chiang, Bowen Deng, Gerbrand Ceder, Mark Asta, Alpha A Lee, Anubhav Jain, and Kristin A Persson. A framework to evaluate machine learning crystal stability predictions. *nature machine intelligence*, 7(6):836–847, 2025.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Khurram Shahzad, Andrei Ionut Mardare, and Achim Walter Hassel. Accelerating materials discovery: combinatorial synthesis, high-throughput characterization, and computational advances. *Science and Technology of Advanced Materials: Methods*, 4(1):2292486, 2024.

Dong-Hee Shin, Deok-Joong Lee, Young-Han Son, and Tae-Eui Kam. Treatment stitching with schr\" odinger bridge for enhancing offline reinforcement learning in adaptive treatment strategies. *arXiv preprint arXiv:2511.12075*, 2025.

Wenhao Sun, Stephen T Dacek, Shyue Ping Ong, Geoffroy Hautier, Anubhav Jain, William D Richards, Anthony C Gamst, Kristin A Persson, and Gerbrand Ceder. The thermodynamic scale of inorganic crystalline metastability. *Science advances*, 2(11):e1600225, 2016.

Charlie B Tan, Avishek Joey Bose, Chen Lin, Leon Klein, Michael M Bronstein, and Alexander Tong. Scalable equilibrium sampling with sequential boltzmann generators. *arXiv preprint arXiv:2502.18462*, 2025.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Peter Thiel. Paypal co-founder peter thiel warns of tech stagnation: 'without ai, there's just nothing going on'. *Economic Times*, 2025. URL https://economictimes.indiatimes.com/magazines/panache/paypal-co-founder-peter-thiel-warns-of-tech-stagnation-without-ai-theres-just-nothing-going-on/articleshow/122141331.cms.

Brandon Trabucco, Aviral Kumar, Xinyang Geng, and Sergey Levine. Conservative objective models for effective offline model-based optimization. In *International Conference on Machine Learning*, pages 10358–10368. PMLR, 2021.

Brandon Trabucco, Xinyang Geng, Aviral Kumar, and Sergey Levine. Design-bench: Benchmarks for data-driven offline model-based optimization. In *International Conference on Machine Learning*, pages 21658–21676. PMLR, 2022.

Masatoshi Uehara, Yulai Zhao, Ehsan Hajiramezanali, Gabriele Scalia, Gökcen Eraslan, Avantika Lal, Sergey Levine, and Tommaso Biancalani. Bridging model-based optimization and generative modeling via conservative fine-tuning of diffusion models. *arXiv preprint arXiv:2405.19673*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Amanda Wang, Ryan Kingsbury, Matthew McDermott, Matthew Horton, Anubhav Jain, Shyue Ping Ong, Shyam Dwaraknath, and Kristin A. Persson. A framework for quantifying uncertainty in DFT energy corrections. *Scientific Reports*, 11(1):15496, 2021. doi: 10.1038/s41598-021-94550-5.

Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

X-D Xiang, Xiaodong Sun, Gabriel Briceno, Yulin Lou, Kai-An Wang, Hauyee Chang, William G Wallace-Freedman, Sung-Wei Chen, and Peter G Schultz. A combinatorial approach to materials discovery. *Science*, 268(5218):1738–1740, 1995.

Tian Xie, Xiang Fu, Octavian-Eugen Ganea, Regina Barzilay, and Tommi Jaakkola. Crystal diffusion variational autoencoder for periodic material generation. *arXiv preprint arXiv:2110.06197*, 2021.

Dezhen Xue, Prasanna V Balachandran, John Hogden, James Theiler, Deqing Xue, and Turab Lookman. Accelerated search for materials with targeted properties by adaptive design. *Nature communications*, 7(1):11241, 2016.

Yu Yan, Fei Hu, Jiusheng Chen, Nikhil Bhendawade, Ting Ye, Yeyun Gong, Nan Duan, Desheng Cui, Bingyu Chi, and Ruofei Zhang. Fastseq: Make sequence generation faster. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 218–226, 2021.

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. *arXiv preprint arXiv:2409.06762*, 2024.

Claudio Zeni, Robert Pinsler, Daniel Zügner, Andrew Fowler, Matthew Horton, Xiang Fu, Sasha Shysheya, Jonathan Crabbé, Lixin Sun, Jake Smith, et al. Mattergen: a generative model for inorganic materials design. *arXiv preprint arXiv:2312.03687*, 2023.

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*, 639(8055):624–632, 2025.