Generative Hierarchical Materials Search

Sherry Yang[*], **Simon Batzner**, **Ruiqi Gao**, **Muratahan Aykol**, **Alexander Gaunt**, **Brendan McMorrow**, **Danilo Rezende**, **Dale Schuurmans**, **Igor Mordatch**, **Ekin Dogus Cubuk**
Google DeepMind

Abstract

Generative models trained at scale can now produce text, video, and more recently, scientific data such as crystal structures. In applications of generative approaches to materials science, and in particular to crystal structures, the guidance from the domain expert in the form of high-level instructions can be essential for an automated system to output candidate crystals that are viable for downstream research. In this work, we formulate end-to-end language-to-structure generation as a multi-objective optimization problem, and propose Generative Hierarchical Materials Search (GenMS) for controllable generation of crystal structures. GenMS consists of (1) a language model that takes high-level natural language as input and generates intermediate textual information about a crystal (e.g., chemical formulae), and (2) a diffusion model that takes intermediate information as input and generates low-level continuous value crystal structures. GenMS additionally uses a graph neural network to predict properties (e.g., formation energy) from the generated crystal structures. During inference, GenMS leverages all three components to conduct a forward tree search over the space of possible structures. Experiments show that GenMS outperforms other alternatives of directly using language models to generate structures both in satisfying user request and in generating low-energy structures. We confirm that GenMS is able to generate common crystal structures such as double perovskites, or spinels, solely from natural language input, and hence can form the foundation for more complex structure generation in near future.

1 Introduction

Modern technologies increasingly rely on the development of materials, such as semiconductors [1], solar cells [2], and lithium batteries [3]. Large-scale generative models, trained on expansive internet data, exhibit intriguing generalization capabilities. For example, these models can synthesize a highly realistic image of "an astronaut riding a horse" by merging two distant concepts [4]. This raises a compelling question: can the generalization capabilities of large generative models, pretrained on existing materials science knowledge, be harnessed to combine knowledge from existing materials systems to propose candidate crystals?

Previous research has demonstrated that generative models can output crystal structures that are not in the the training data [5, 6, 7]. However, these works typically require either a vast number of unconditional samples to generate an unknown material [5, 8] or a chemical formula provided during inference [6, 9]. It is difficult for end users to come up with new chemical formulae, as it is hard to know which compositions will result in what material properties. Therefore, it is highly desirable to develop an interface that allows users to describe the desired characteristics of crystal structures — such as properties, compositions, space groups, and geometric characteristics — in natural language. For example, a user might specify "a stable chalcogenide with atom ratio 1:1:2 that is not on ICSD." Ideally, a model should automatically interpret these high-level language instructions to search for,

[*]Correspondence to sherryy@google.com and cubuk@google.com.
38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-1226

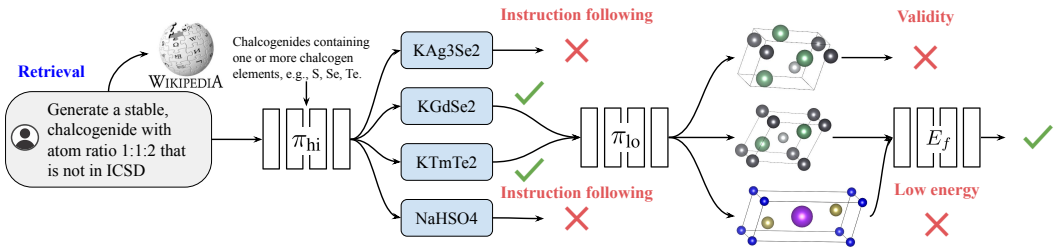

Figure 1: **Overview of GenMS.** GenMS takes a high-level language instruction as input, retrieves relevant information from the internet, and samples from a high-level LLM (π_{hi}) to generate candidate formulae that satisfy user requirement. GenMS then samples from a low-level diffusion model (π_{lo}) to generate structures conditioned on candidate formulae. Sampled structures then go through a property prediction module for selection.

generate, and validate a wide range of potential structures, ultimately producing one that best meets the user's specifications.

However, developing an end-to-end language-to-structure generative model presents several challenges, for which we make a few key observations. First, there are no existing labeled datasets that map language descriptions directly to crystal structures. Nevertheless, we observe that there is a wealth of language-to-formula data available online, including Wikipedia articles, research papers, and textbooks. This data can be complemented by formula-to-structure information from specialized materials databases such as the Materials Project [10], ICSD [11], OQMD [12], etc. Second, the task of converting language into structures is inherently multimodal, requiring the transformation of discrete linguistic inputs to continuous structural outputs. Nevertheless, it has been shown that semantic-level autoregressive models combined with low-level (pixel-level) diffusion models are effective for cross-modal generation, such as in text-to-video applications [13, 14]. Lastly, user descriptions of desired crystal structures can often be vague — users may not articulate all relevant details about the crystal they wish to generate. We observe that one can leverage generative models to infer missing information, and rely on additional search and selection mechanisms to identify structures that best satisfy a user's requirement.

Based on these observations, we propose Generative Hierarchical Materials Search (GenMS) for end-to-end language-to-structure generation. GenMS consists of (1) a large language model (LLM) pretrained on high-level materials science knowledge from the internet, (2) a diffusion model trained on low-level crystal structure databases, and (3) a graph neural network (GNN) for property prediction. To improve the efficiency of (2), GenMS proposes a compact representation of crystal structures for diffusion models. During inference, GenMS prompts the LLM to generate candidate chemical formulae according to user specification, samples structures from the diffusion model, and uses the GNN to predict the properties of the sampled structures. To sample structures that best satisfy user requirements during inference, we formulate language-to-structure as a multi-objective optimization problem, where user specifications are transformed into objectives that can be optimized at both the formula and structure level.

We first evaluate GenMS's ability to generate crystal structures from language instructions, and find that GenMS can successfully generate structures that satisfy user requests more than 80% of the time for three major families of structures, while proposing structures with low formation energies, as verified by DFT calculations. In contrast, using pretrained LLMs to directly generate crystal structures from user instructions in a zero-shot manner often results in close to a 0% success rate. Qualitative evaluations show that GenMS is able to generate complex structures, such as layered structures, double perovskites, and spinels, solely from natural language. We next study the effect of each individual component of GenMS. Here we find that language instructions have a significant impact on the structures generated, that the novel compact representation of crystals proposed by GenMS improves the DFT convergence rate of diffusion generated crystal structures by 50% over previous work, and that using a pretrained GNN to select samples leads to lower energy structures more than 80% of the time. Given such experimental evidence, we believe the development of language-to-structure models are promising for enabling users to find viable crystal structure candidates, complementing existing databases in utility

2 Generative Hierarchical Materials Search

We begin by formulating the problem of generating crystal structures from high-level language as a multi-objective optimization task. Given this formulation, we then propose a hierarchical, multi-modal tree search algorithm that leverages language models, diffusion models, and graph neural networks as submodules. Lastly, we discuss the specific design choices for each of the submodules.

2.1 Language to structure as a multi-objective optimization

Given some high-level language description $g \in \mathcal{G}$ of desired structures, we want to learn a conditional crystal structure generator $\pi(\cdot|g) : \mathcal{G} \mapsto \Delta(\mathcal{X})$[2] that can be used to sample crystal structures $x \in \mathcal{X}$ conditioned on language. One option is to parametrize π with a pretrained LLM. However, pretrained LLMs alone are not able to predict sufficiently accurate crystal structures, due to the lack of low-level structural information about crystals (e.g., 3D atom coordinates) in the pretraining data.

If we had access to a paired language-to-structure dataset, $\mathcal{D} = \{g_i, x_i\}_{i=1}^N$, π could be trained using a maximum likelihood objective. However, materials data naturally exist at different levels of abstraction and are segregated into different sources: high-level symbolic knowledge is documented in sources like Wikipedia articles, research papers, and textbooks, whereas detailed low-level crystal information, including continuous-valued atom positions, is stored in specialized crystal databases like the Materials Project [10] and ICSD [11]. Even though a direct language-to-structure dataset $\mathcal{D}$ remains unavailable, the pretraining data for LLMs, including Wikipedia articles, research papers, and textbooks, can be viewed as a high-level symbolic dataset $\mathcal{D}_{\text{hi}} = \{g_i, z_i\}_{i=1}^m$, where $z \in \mathcal{Z}$ denotes symbolic textual information such as chemical formulae. Meanwhile, many crystal databases already feature paired data, $\mathcal{D}_{\text{lo}} = \{z_i, x_i\}_{i=1}^n$, linking chemical formulae to detailed crystal structures.

Given this observation, we propose to factorize the crystal generator as $\pi = \pi_{\text{hi}} \circ \pi_{\text{lo}}$, where $\pi_{\text{hi}} : \mathcal{G} \mapsto \Delta(\mathcal{Z})$ and $\pi_{\text{lo}} : \mathcal{Z} \mapsto \Delta(\mathcal{X})$, so that π_{hi} and π_{lo} can be trained using different datasets $\mathcal{D}_{\text{hi}}$ and $\mathcal{D}_{\text{lo}}$. Furthermore, we consider two heuristic functions, $R_{\text{hi}}(g, z) : \mathcal{G} \times \mathcal{Z} \mapsto \mathbb{R}$ and $R_{\text{lo}}(z, x) : \mathcal{Z} \times \mathcal{X} \mapsto \mathbb{R}$, where the high-level heuristic function R_{hi} can be used to select formulae that satisfy the language input at a high level, and the low-level heuristic function R_{lo} can be used to select structures that are both valid and exhibit desirable properties such as low formation energy. To this end, we propose to search for crystal structure given language input by finding a chemical formula / space group z with a corresponding crystal structure x that jointly optimize

$$z^*, x^* = \arg \max_{z, x \sim \pi_{\text{hi}}, \pi_{\text{lo}}} \mathbb{E}_{z \sim \pi_{\text{hi}}, x \sim \pi_{\text{lo}}(z)} [\lambda_{\text{hi}} \cdot R_{\text{hi}}(g, z) + \lambda_{\text{lo}} \cdot R_{\text{lo}}(z, x)], \tag{1}$$

where λ_{hi} and λ_{lo} are hyperparameters to control how much weight to put on high and low-level heuristics. Note that R_{hi} and R_{lo} can also be combinations of multiple objectives. For instance, R_{hi} can be a weighted sum of instruction following and simplicity, where R_{lo} can be a weighted sum of properties such as band gap, conductivity, and formation energy.

2.2 Searching through language and structure

Given the objective in Equation 1, it is clear that a pretrained LLM (even with finetuning) is insufficient to optimize for the best structure x^*. Instead, we propose to first sample a set of intermediate chemical formulae from a pretrained LLM $\pi_{\text{hi}}(g)$ conditioned on language input g. We then use the high-level heuristic function R_{hi} to prune and rank the intermediate formulae. In practice, R_{hi} is a combination of (i) a regular expression checker (to ensure sampled formulae are valid chemical formulae), (ii) a uniqueness checker against formulae from existing crystal datasets such as Materials Project and ICSD, and (iii) a formula compliance checker to ensure the sampled formulae are compatible with user request (e.g., atom ratio 113 for perovskites, 227 for pyrochlore, and 124 for spinel). For formulae that pass these checks, we prompt a pretrained LLM as R_{hi} to rank the formulae by how likely they are to comply with the user request g. We then select the top W ranked formulae to generate L crystal structures each using π_{lo} parametrized by a diffusion model, and use a graph neural network R_{lo} to rank the $W \times L$ structures by their predicted formation energy. Note that additional checkers can be integrated in R_{lo}, such as structural and compositional validity defined in [5]. We illustrate the overall search procedure in Algorithm 1.

[2]We use $\Delta(\cdot)$ to denote a probability simplex function.

https://doi.org/10.52202/079017-1226

Algorithm 1 Generative Hierarchical Materials Search

Input: Language input g
Functions: High-level language policy $\pi_{\text{hi}}(z|g)$, high-level heuristic function $R_{\text{hi}}(g, z)$, low-level diffusion policy $\pi_{\text{lo}}(x|z)$, low-level heuristic function $R_{\text{lo}}(z, x)$.
Hyperparameters: High-level language branching factor H, low-level structure branching factor L, max width for formulae W.
plans $\leftarrow [[g] \ \forall \ i \in \{1 \dots H\}]$ # Initialize H different plans starting with language input.
for $h = 1 \dots H$ **do**
    $g \leftarrow$ plans$[h][-1]$ # Get the high-level language specification from the tree.
    $\{z_i\}_{i=1}^{H} \leftarrow \pi_{\text{hi}}(g)$ # Generate H different intermediate formulae.
    z^* = argmax$(\{g, z_i\}_{i=1}^{H}, R_{\text{hi}})$
    plans[h].append(z^*) # Add formula with the best heuristic value to plan.
end for
plans $\leftarrow$ sort(plans, R_{hi}) # Sort formulae based on heuristic.
for $w = 1 \dots W$ **do**
    $z \leftarrow$ plans$[w][-1]$ # Get the best intermediate formula from the tree.
    $\{x_i\}_{i=1}^{L} \leftarrow \pi_{\text{lo}}(z)$ # Generate L low-level structures.
    x^* = argmax$(\{z, x_i\}_{i=1}^{H}, R_{\text{lo}})$
    plans[w].append(x^*) # Add structure with the best heuristic value to plan.
end for
return plans[0][0] # Return the best structure.

Alternative search strategies. The search algorithm described above, Algorithm 1, follows the best-first search strategy, i.e., intermediate formulae and final structures are sorted and searched over based on the preference of a heuristic function. Alternative search strategies such as breadth-first or depth-first can also be employed. The most suitable search strategy depends on the downstream application and computational resources available. For instance, if large-scale density function theory (DFT) calculations are available downstream, we can employ breadth-first search to devise more diverse composition.

Prevent heuristic exploitation. One concern of using a heuristic GNN to select structures with the lowest formation energy is that the GNN might exploit irregularities in the predicted structures, especially when a predicted structure lies outside of the training manifold of the energy GNN. To mitigate this issue, we use the GNN pretrained by [15] on DFT energies and forces of unrelaxed structures (hence the GNN has seen more irregular structures prior to relaxation.) Furthermore, we discard sampled structures from π_{lo} if they result in energy predictions from R_{lo} that lie outside of a threshold range.

2.3 Choices of parametrization for the submodules

Since controllable crystal structure generation from language input is multimodal by nature, there are various design choices for the parametrization of the submodules in Equation 1, namely the generators $\pi_{\text{hi}}, \pi_{\text{lo}}$ and the heuristic functions $R_{\text{hi}}, R_{\text{lo}}$. In this section, we discuss the parametrization choices we have found to be the most effective.

Retrieval augmentation and long-context deduplication. One important recent advance in LLMs is increased context length [16]. The factorization $\pi = \pi_{\text{hi}} \circ \pi_{\text{lo}}$ provides a natural way to integrate additional context in π_{hi} via long-conext generation. Specifically, we further factorize π_{hi} into $\pi_{\text{hi}} = \pi_{\text{hi}}^{\text{retrival}} \circ \pi_{\text{hi}}^{\text{RAG}}$, where $\pi_{\text{hi}}^{\text{retrival}}(\cdot|g)$ is a deterministic retrieval function that uses the Wikipedia API to retrieve textual information related to language input g, while $\pi_{\text{hi}}^{\text{RAG}}$ is a retrieval augmented generative (RAG) model that proposes chemical formulae and space groups conditioned on the information retrieved from the internet. Another use case for long-context LLMs is to further encourage the generation of *new* compositions by providing the formulae for all known crystals in the context, then asking π_{hi} to produce a formula that is not in the context. As we will see in Section 3.2, this drastically improves the efficiency of the search, as a large subset of the search space with known crystals can be eliminated.

Compact crystal representation. In order to support efficient tree search at inference time, we need to ensure that sampling from both π_{hi} and π_{lo} are efficient. Previous work on diffusion models

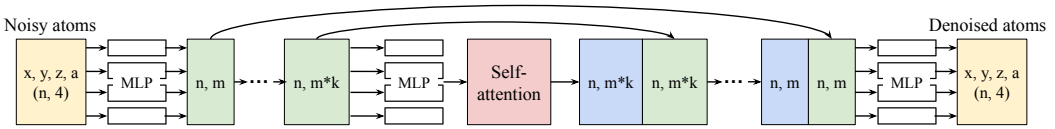

Figure 2: **Diffusion architecture with compact crystal representation.** The diffusion model in GenMS represents crystal structures by the x, y, z location of each atom plus the atom number a represented as a continuous value. Each atom undergoes blocks consisting of multi-layer perceptrons followed by order-invariant self-attention. The MLP and self-attention blocks are repeated k times where each repetition increases the dimension of the hidden units. The concatenation of skip connections are employed as in other U-Net architectures.

for crystal structure generation has leveraged sparse data structures, such as voxel images [17, 18, 19], graphs [5], and periodic table shaped tensors [6]. These existing representations of crystals incur computational overhead due to sparsity (voxel images, padded tensors) or quadratic complexity as the number of atoms in the system increases (graphs). Instead, we propose a new compact representation of crystal structures, where each crystal $x \in \mathcal{X} \subset \mathbb{R}^{A\times 4}$ is represented by a $A \times 4$ tensor, with A being the number of atoms in the crystal, and the inner 4 dimensions representing the x, y, z location of an atom along with its atom number. Here we directly represent the atom number as a continuous value normalized to the range of the input in the diffusion model to further improve inference speed, as opposed to representing the atom number using a one-hot vector. In addition, we use another 2×3 vector to represent the lattice structure (i.e., angles and lengths of the unit cell). Figure 2 illustrates the architecture for the diffusion model with compact crystal representations, where each atom undergoes multi-layer perceptron (MLP) followed by order-invariant self-attention (without positional encoding) across atoms. Different from typical U-Net architecture for image generation, there is no downsampling or upsampling passes that change the input resolution. Nevertheless, we follow the concatenation of skip connections commonly used in U-Net architectures [20]. Additional details and hyperparameters for the diffusion model can be found in Appendix A.3.

3 Experimental Evaluation

We now evaluate the ability of GenMS to generate crystal structures from high-level language descriptions. First, we evaluate the success of end-to-end generation in Section 3.1. We then investigate the individual components of GenMS in Section 3.2. See details of experimental setups in Appendix A.

3.1 End-to-end evaluation

Baselines and metrics. We aim to evaluate GenMS's ability to generate unique, valid, and potentially stable crystal structures from well-known crystal families that satisfy high-level language specifications. We consider few-shot prompting of LLMs to generate crystal information files (CIF) as a baseline. Specifically, we give the Gemini long context model [16] a number of CIF files from a particular crystal family, as specified by language input as prompt, with the number of CIF files ranging from 1, 5, 25 to as many as can fit in the context. We ask the LLM to generate 100 samples given each language instruction. See additional details of baselines in Appendix A.2. We do not compare to finetuning LLMs to generate CIF files in this section, as there are no high-level language to low-level crystal structure datasets available for finetuning such an instruction following LLM. Nevertheless, we will compare the diffusion model in GenMS to formula-conditioned structure generation using finetuned LLM in Section 3.2. We consider language input that directs the model to generate unique and stable crystals from a particular crystal family (perovskite, pyrochlore, and spinel). We consider the following metrics for evaluation: (i) CIF validity, which measures whether the generated CIF file can be properly parsed by pymatgen parser [21]. (ii) Structural and composition validity, which verify atom distances and charge balances using SMACT [22], following [5]. (iii) Formation energy per atom (E_f) in the unit of eV/atom, which measures the stability of predicted structures using a pretrained GNN. We further conduct DFT calculations to compute E_f (see details in Appendix A.4) for structures predicted by GenMS. (iv) Uniqueness, which measures the percentage of generated formulae that do not exist in Materials Project [10] or ICSD [11]. Finally, (v) the match rate, which measures the percentage of generated structures that can be matched (according to the pymatgen

https://doi.org/10.52202/079017-1226

Family	Metric	Prompting CIF				GenMS
		1 shot	5 shot	25 shot	Max	
Perovskites	CIF validity ↑	0.94	1.00	0.98	0.88	**1.00**
	Structural validity ↑	0.04	0.28	0.66	0.22	**1.00**
	Composition validity ↑	0.07	0.17	0.45	0.00	**0.85**
	E_f (GNN/DFT) ↓	N/A	-0.19	0.28	0.53	**-0.47/-1.32**
	Uniqueness ↑	0.07	0.29	0.68	0.16	**0.90**
	Match rate ↑	0.00	0.09	0.36	0.19	**0.93**
Pyrochlore	CIF validity↑	0.40	0.60	0.64	0.88	**1.00**
	Structural validity↑	0.36	0.36	0.28	0.23	**0.95**
	Composition validity↑	0.00	0.22	0.18	0.00	**0.89**
	E_f (GNN/DFT)↓	1.28	1.19	0.63	-1.22	**-1.37/-2.56**
	Uniqueness↑	0.36	0.38	0.45	0.08	**0.49**
	Match rate ↑	0.00	0.00	0.04	0.00	**0.86**
Spinel	CIF validity	0.73	0.96	0.97	0.96	**1.00**
	Structural validity↑	0.47	0.61	0.71	1.00	**1.00**
	Composition validity↑	0.29	0.92	0.92	1.00	**1.00**
	E_f (GNN/DFT)↓	1.09	-0.85	-0.97	-1.37	**-1.38/-1.77**
	Uniqueness↑	0.48	0.13	**0.51**	0.08	0.44
	Match rate ↑	0.00	0.12	0.18	0.08	**0.89**

Table 1: **End-to-end evaluation** of generating crystal structure from natural language. GenMS significantly outperforms LLM prompting baselines in producing unique and low-energy (predicted by GNN) structures that satisfy user request. We further conduct DFT calculation to compute E_f (formation energy in eV/atom) averaged across structures generated by GenMS. Values before "/" in the row "(GNN/DFT)" represent GNN predicted E_f, and after "/" represent DFT computed E_f. We report E_f from GNN prior to relaxation, and E_f from DFT post relaxation. DFT calculations for baselines are eliminated as many structures from the baselines do not follow user instruction. N/A represents E_f predicted by GNN falling outside of the reasonable range.

structure matcher) to one of the structures of the corresponding family in Materials Project. More details of these metrics can be found in Appendix A.1.

Results on specifying crystal family. The evaluation of GenMS and baselines are shown in Table 1. Since GenMS does not rely on an LLM to directly generate CIF files, the compact crystal representation (described in Section 2.3) always results in structures that can be parsed by pymatgen (100% CIF validity). In addition, structures generated by GenMS have a much higher validity and match rate compared to those generated by the baselines. GenMS struggles slightly with uniqueness, as less than half of the generated formulae for pyrochlore and spinel are unique with respect to MP and ICSD. Structures produced by GenMS have lower average E_f. Increasing the number of CIF files in the context generally improves the performance of the baselines (1, 5, and 25-shot), but including too many files in the context can hurt performance (Prompting CIF Max).

Qualitative evaluation. In addition to the three families of structures evaluated above, we qualitatively evaluated GenMS's ability to generate structures that satisfy ad hoc user requests, such as "a layered material", "an elpasolite", and so on. GenMS can consistently produce structures that satisfy user request as shown in Figure 3, and have plausible initial geometries. Interestingly, we observe that GenMS can understand semantic-level request, suggesting more "fluoride" like chemistries when asked for "elpasolite", which is reasonable as elpasolite is associated with the mineral K2NaAlF6.

Effect of search. Next, we aimed to understand the effect of search in GenMS, especially in producing low-energy structures. For each of the family of crystals in Table 1, we analyzed the effect of the language and structure branching factors (H and L in Algorithm 1). Only crystals that match input specification were considered for energy computation. We found that increasing the branching factor of both language and structure enables GenMS to generate structures with lower formation energies (at a higher inference cost).

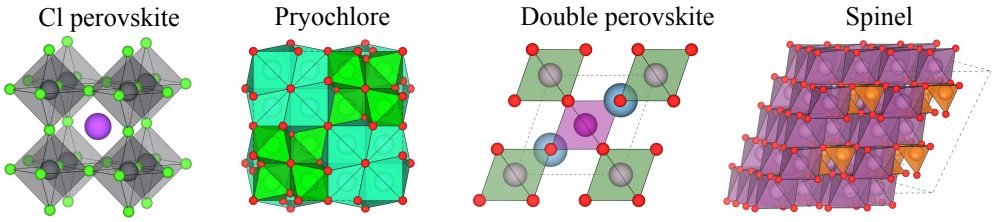

Figure 3: **Qualitative evaluation.** We test GenMS on a set of ad hoc language inputs to generate plausible examples from well-known crystal families. GenMS is able to search for the corresponding structures that satisfy user requests and have plausible initial geometries. Visualization provided by VESTA [23].

Perovskite			Pyrochlore			Spinel		
Language branch (H)	Structure branch (L)	E_f (DFT)	Language branch (H)	Structure branch (L)	E_f (DFT)	Language branch (H)	Structure branch (L)	E_f (DFT)
1	1	-0.55	1	1	-2.40	1	1	-1.67
1	100	-0.79	1	100	-2.51	1	100	-1.74
25	1	-2.76	25	1	-3.02	25	1	-1.82
25	100	**-2.91**	25	100	**-3.24**	25	100	**-1.95**

Table 2: E_f **(computed by DFT) vs. branching factor.** GenMS can generate structures with lower formation energy (computed by DFT) at the cost of slower inference when language and structure branching factors are increased.

3.2 Evaluating individual components of GenMS

Next, we evaluate the individual component of GenMS, including the effect of using language to narrow down the search space, the choice of the compact representation of crystal structures, and finally the best-of-N sampling strategy for choosing the crystal structures with low formation energy.

Effect of language. We want to understand whether GenMS can provide effective control over formulae proposed by the LLM at the semantic-level through natural language. In Table 3, we first show that requesting a particular element to be in the formula always results in formulas with that particular element being proposed by the pretrained LLM π_{hi}. We then show that when a user requests for metal, the model is 4 times more likely to generate formulae for metal. The model also respects a user's request for the generated formulae to be unique (with respect to either a user provided list of known formulae in the context of the LLM, or the name of some crystal database).

Next, we study the effect of retrieval augmented generation (RAG). We use GenMS with and without RAG to propose 25 formulae for each of the three major crystal families from Section 3.1 and generates 4 structures per formula using the diffusion model. We report the rate of valid formulae proposed by the LLM and the structures that can be matched with existing structures from the corresponding family in Table 4. RAG improves both the rate of valid formulae and matched structures.

	Element constraint	Metal only	Unique (custom list)	Unique (Materials Project)
Not asking	N/A	0.25	0.24	0.16
Asking	**1.00**	**1.00**	**0.88**	**0.96**

Table 3: **Effect of language.** Asking for a specific element from the periodic table results in formulae that always contain that element. Asking for metal and formulae unique with respect to some existing formula sets result in formulae that are more likely to satisfy user requests.

	Valid formula	Match rate
Without RAG	0.97	0.72
With RAG	**1.00**	**0.89**

Table 4: **Effect of RAG.** Using retrieval augmented generation improves the percentage of valid formulae and matched structures. See details for the structure matcher used in Appendix A.1.

https://doi.org/10.52202/079017-1226

	UniMat [6]	GenMS
DFT converge	0.62	**0.93**
Mean E_f	-0.40 $\pm$ 0.06	**-0.49** $\pm$ 0.03

Table 5: **DFT evaluation of GenMS vs UniMat.** Structures proposed by GenMS result in much high DFT convergence rate and lower average E_f than structures proposed by UniMat. Error bars reflect standard error.

	Small LLM [9]	Large LLM	GenMS
Success	0.86	0.87	**0.93**
Match (unseen)	0.26	0.37	**0.48**

Table 6: **Comparison to finetuned LLMs.** GenMS's diffusion model with compact representations achieves high success rate of generating valid crystals, as well as a high matching rate to holdout structures compared to CrystalLM [9].

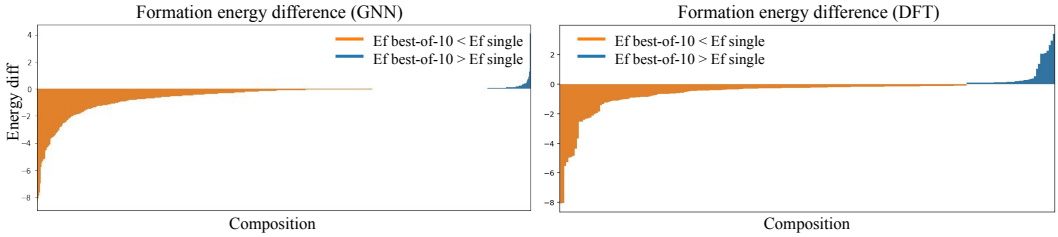

Figure 4: **Formation energy between Best-of-N and a single sample.** Both according to energy predicted by GNN and calculated by DFT, best-of-N with N = 10 leads to improvements in energy compared to single samples for 80% of 1,000 compositions considered.

Compact crystal representation. We now evaluate the diffusion model π_{lo} trained using the compact representation of crystals structures described in Section 2.3. We compare diffusion model with compact crystal representation against two prior work for generating crystal structures conditioned on composition. UniMat [6] proposed a periodic table representation of crystals which requires a large amount of paddings to handle atoms that do not exist in the structure. CrystalLM [9] proposes to finetune an LLM to directly generate CIF files from input compositions. In Table 5, we report the DFT convergence rate and DFT calculated E_f on a set of holdout structures following [6]. We observe that the compact crystal representation results in both higher convergence rate and lower E_f than the sparse representation in [6]. To compare GenMS's diffusion model against finetuning LLMs to generate CIF files directly, we follow the experimental setting of CrystaLLM where we train a composition conditioned diffusion model on a combination of Materials Project [10], OQMD [24], and NOMAD [25], and test the success rate of generating matching structures for unseen compositions following [9]. In Table 6, we see that GenMS has significantly higher rate in producing a valid crystal and a crystal that can be matched to the test set in [9].

Best-of-N structure sampling. To better understand the effect of high structure branching factor in Algorithm 1 across different compositions, we measure the difference in the formation energy, using a holdout test set of 1,000 compositions, between using the energy prediction GNN to select the best of 10 samples compared to only predicting a single structure. The energy difference with and without best-of-N sampling is shown in Figure 4. Using best-of-N with $N = 10$ results in improved energy for over 80% of structures (as also verified by DFT calculations). We found the energy prediction GNN to be a good indicator of the true energy of the crystal structures, i.e., the GNN predicted energy difference (left) and the DFT calculated energy difference (right) are very similar in Figure 4.

4 Related work

Hierarchical and latent image and video generation. Image and video generative models have exhibited an impressive ability to synthesize photorealistic images or videos when given text description as input. Many of the state-of-the-art models adopt a hierarchical modeling approach that inspired the design of with GenMS. For example, latent diffusion models [26, 27] contains (1) a language model that converts text to high-level text embeddings, (2) a diffusion model takes the text embeddings as input and output latents in a compressed latent space, and (3) a feed forward decoder network [26] or a diffusion decoder [28, 14] that given the generated latents generates full-resolution signals in the pixel space. Cascaded diffusion models [29, 30, 31] instead proposed to generate signals at the lowest resolution with a standard diffusion model, followed by a few super-resolution

models that successively upsample signals and add high-resolution details. Similar to GenMS, by breaking down complicated image or video generation into a hierarchy of less challenging problems, these models can generate high quality samples more efficiently and effectively.

Generative models for crystal structures. A number of works [9, 8, 32] have proposed to train or fine-tune language models to generate output files containing crystal information or low-level atom positions. However, it remains expensive and challenging to train and generate detailed structural information with LLMs. On the other hand, diffusion models, as a powerful class of generative model in vision, have been applied to generate crystal structures [5, 7, 6]. However these methods either reply on training with a large set of unconditional samples and brute-force sampling for new materials not in the training set, or necessitate predetermined compositions as conditioning information during inference. Handling of candidate structure generation requires a model capable of independent reasoning about chemical compositions based on high-level user specifications and structure optimization, as done in GenMS.

Hierarchical search and planning. The problem of learning to generate low-level continuous output from high-level language instructions, while employing intermediate search and planning steps, has been studied in other domains such as continuous control [33], self-driving [34], and robotics [35]. While some works have focused on purely using LLMs to search and plan through complex output spaces [36, 37], other research has shown that solely relying on LLMs to search and plan can fail short due to the lack of low-level information (e.g., locations, precise motions) captured in the model [38]. Recently, video generation models have been applied to provide additional details about the physical world so that low-level control actions can be extracted more accurately [39, 40, 41, 42]. GenMS follows a similar approach but focuses on generating crstyal structures, using diffusion models on top of LLMs to provide additional details about crystal structure, enabling high-level plans (i.e., symbolic chemical formulae) to be verified at a low-level (i.e., crystal structures with precise atom locations).

Large language models for science. Recently, there has been a surge of interest in applying large langauge models in domains of science, such as physics [43], biology [44], chemistry [45, 46], and materials science [47]. In these settings, LLMs generally serve as a conversational [44] or educational [48] tool, where LLMs output natural language to be consumed by human users (e.g., an answer to a scientic question asking about the property of some existing crystal structure). On the other hand, we are interested in the ability of a pretrained LLM to propose intermediate textual information such as chemical formulae for interesting crystal structures. Closest to our work are [49, 50] which leverage an LLM to generate SMILES or other chemical strings for molecular design. Nevertheless, we are interested in generating not just the formulae, but the actual crystal structures with continuous-valued atom locations, as many materials property can only be calculated and verified once the full structure available.

5 Conclusion and future work

We have introduced GenMS, an initial attempt at enabling end-to-end generation of candidate crystal structures that look physically viable and satisfy instructions expressed in natural language. GenMS can generate examples from families such as pyrochlores and spinels purely from natural language prompts. We hope the design principles of GenMS will initiate broad interest in exploiting language as a natural interface for flexible design and generation of crystal structures that meet user-specified criteria, and enable the domain experts to work more efficiently. GenMS has a few limitations that call for future work:

- **Generating complex structures.** While GenMS is able to generate simple structures such as those shown in Figure 3, we found that GenMS is less effective in generating complex structures such as Mxenes and Kagome lattices. Controllable generation of highly complex crystal structures is an interesting area of future work.
- **Impact on experimental exploration.** While we have shown that GenMS is effective in generating crystal structures that are not in public databases and that satisfy user requirements, its effectiveness in suggesting specific materials with target properties (e.g., battery electrodes or electrolytes, semiconductors, superconductors etc.) requires further experimental verification.

https://doi.org/10.52202/079017-1226

- **Synthesizability.** While the goal of GenMS is to provide an end-to-end generative framework from natural language instructions to realistic crystal structures, synthesizability of the generated crystals is not currently part of the pipeline. We foresee development in multimodal models and integration of other computational tools from materials science to allow predicted structures to be assessed for synthesizability.
- **Extension to other chemical systems.** We have shown that GenMS can effectively generate crystal structures from natural language. We note that GenMS can also potentially be extended to generating molecules and protein structures from natural language (e.g. "generate a protein with an alpha-helix"). We leave these explorations for future work.

References

[1] Lev I Berger. *Semiconductor materials*. CRC press, 2020. 1

[2] Martin A Green, Anita Ho-Baillie, and Henry J Snaith. The emergence of perovskite solar cells. *Nature photonics*, 8(7):506–514, 2014. 1

[3] KJPC Mizushima, PC Jones, PJ Wiseman, and John B Goodenough. Lixcoo2 (0< x<-1): A new cathode material for batteries of high energy density. *Materials Research Bulletin*, 15(6):783–789, 1980. 1

[4] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021. 1

[5] 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. 1, 3, 5, 9, 14

[6] Mengjiao Yang, KwangHwan Cho, Amil Merchant, Pieter Abbeel, Dale Schuurmans, Igor Mordatch, and Ekin Dogus Cubuk. Scalable diffusion for materials generation. *arXiv preprint arXiv:2311.09235*, 2023. 1, 5, 8, 9

[7] 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. 1, 9

[8] Daniel Flam-Shepherd and Alán Aspuru-Guzik. Language models can generate molecules, materials, and protein binding sites directly in three dimensions as xyz, cif, and pdb files. *arXiv preprint arXiv:2305.05708*, 2023. 1, 9

[9] Luis M Antunes, Keith T Butler, and Ricardo Grau-Crespo. Crystal structure generation with autoregressive large language modeling. *arXiv preprint arXiv:2307.04340*, 2023. 1, 8, 9, 14

[10] 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. 2, 3, 5, 8, 14

[11] Mariette Hellenbrandt. The inorganic crystal structure database (icsd)—present and future. *Crystallography Reviews*, 10(1):17–22, 2004. 2, 3, 5, 14

[12] 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. 2

[13] 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. 2

[14] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 2, 8

[15] Amil Merchant, Simon Batzner, Samuel S Schoenholz, Muratahan Aykol, Gowoon Cheon, and Ekin Dogus Cubuk. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023. 4

[16] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 4, 5

[17] Jordan Hoffmann, Louis Maestrati, Yoshihide Sawada, Jian Tang, Jean Michel Sellier, and Yoshua Bengio. Data-driven approach to encoding and decoding 3-d crystal structures. *arXiv preprint arXiv:1909.00949*, 2019. 5

[18] 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. 5

[19] Callum J Court, Batuhan Yildirim, Apoorv Jain, and Jacqueline M Cole. 3-d inorganic crystal structure generation and property prediction via representation learning. *Journal of Chemical Information and Modeling*, 60(10):4518–4535, 2020. 5, 14

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 5

[21] 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. 5, 15

[22] Daniel W Davies, Keith T Butler, Adam J Jackson, Jonathan M Skelton, Kazuki Morita, and Aron Walsh. Smact: Semiconducting materials by analogy and chemical theory. *Journal of Open Source Software*, 4(38):1361, 2019. 5, 14

[23] Koichi Momma and Fujio Izumi. Vesta 3 for three-dimensional visualization of crystal, volumetric and morphology data. *Journal of applied crystallography*, 44(6):1272–1276, 2011. 7

[24] James E Saal, Scott Kirklin, Muratahan Aykol, Bryce Meredig, and Christopher Wolverton. Materials design and discovery with high-throughput density functional theory: the open quantum materials database (oqmd). *Jom*, 65:1501–1509, 2013. 8

[25] Claudia Draxl and Matthias Scheffler. The nomad laboratory: from data sharing to artificial intelligence. *Journal of Physics: Materials*, 2(3):036001, 2019. 8

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 8

[27] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in neural information processing systems*, 34:11287–11302, 2021. 8

[28] Aditya Ramesh et al. Hierarchical text-conditional image generation with clip latents, 2022. 8

[29] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022. 8

[30] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 8

 https://doi.org/10.52202/079017-1226

[31] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 8

[32] Nate Gruver, Anuroop Sriram, Andrea Madotto, Andrew Gordon Wilson, C Lawrence Zitnick, and Zachary Ulissi. Fine-tuned language models generate stable inorganic materials as text. *arXiv preprint arXiv:2402.04379*, 2024. 9

[33] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023. 9

[34] Xingcheng Zhou, Mingyu Liu, Bare Luka Zagar, Ekim Yurtsever, and Alois C Knoll. Vision language models in autonomous driving and intelligent transportation systems. *arXiv preprint arXiv:2310.14414*, 2023. 9

[35] Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou, Kaizhao Liang, Jintai Chen, Juanwu Lu, Zichong Yang, Kuei-Da Liao, et al. A survey on multimodal large language models for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 958–979, 2024. 9

[36] Yaqi Xie, Chen Yu, Tongyao Zhu, Jinbin Bai, Ze Gong, and Harold Soh. Translating natural language to planning goals with large-language models. *arXiv preprint arXiv:2302.05128*, 2023. 9

[37] Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models-a critical investigation. *Advances in Neural Information Processing Systems*, 36:75993–76005, 2023. 9

[38] Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Large language models still can't plan (a benchmark for llms on planning and reasoning about change). *arXiv preprint arXiv:2206.10498*, 2022. 9

[39] Mengjiao Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023. 9

[40] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems*, 36, 2024. 9

[41] Yilun Du, Mengjiao Yang, Pete Florence, Fei Xia, Ayzaan Wahid, Brian Ichter, Pierre Sermanet, Tianhe Yu, Pieter Abbeel, Joshua B Tenenbaum, et al. Video language planning. *arXiv preprint arXiv:2310.10625*, 2023. 9

[42] Anurag Ajay, Seungwook Han, Yilun Du, Shuang Li, Abhi Gupta, Tommi Jaakkola, Josh Tenenbaum, Leslie Kaelbling, Akash Srivastava, and Pulkit Agrawal. Compositional foundation models for hierarchical planning. *Advances in Neural Information Processing Systems*, 36, 2024. 9

[43] Jason Holmes, Zhengliang Liu, Lian Zhang, Yuzhen Ding, Terence T Sio, Lisa A McGee, Jonathan B Ashman, Xiang Li, Tianming Liu, Jiajian Shen, et al. Evaluating large language models on a highly-specialized topic, radiation oncology physics. *Frontiers in Oncology*, 13, 2023. 9

[44] Rachel K Luu and Markus J Buehler. Bioinspiredllm: Conversational large language model for the mechanics of biological and bio-inspired materials. *Advanced Science*, 11(10):2306724, 2024. 9

[45] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, pages 1–11, 2024. 9

[46] Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Dongzhan Zhou, et al. Chemllm: A chemical large language model. *arXiv preprint arXiv:2402.06852*, 2024. 9

[47] Ge Lei, Ronan Docherty, and Samuel J Cooper. Materials science in the era of large language models: a perspective. *arXiv preprint arXiv:2403.06949*, 2024. 9

[48] Liangtai Sun, Yang Han, Zihan Zhao, Da Ma, Zhennan Shen, Baocai Chen, Lu Chen, and Kai Yu. Scieval: A multi-level large language model evaluation benchmark for scientific research. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19053–19061, 2024. 9

[49] Hisaki Ikebata, Kenta Hongo, Tetsu Isomura, Ryo Maezono, and Ryo Yoshida. Bayesian molecular design with a chemical language model. *Journal of computer-aided molecular design*, 31:379–391, 2017. 9

[50] Michael Moret, Irene Pachon Angona, Leandro Cotos, Shen Yan, Kenneth Atz, Cyrill Brunner, Martin Baumgartner, Francesca Grisoni, and Gisbert Schneider. Leveraging molecular structure and bioactivity with chemical language models for de novo drug design. *Nature Communications*, 14(1):114, 2023. 9

[51] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*, pages 424–432. Springer, 2016. 14

[52] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022. 14

[53] Georg Kresse and Jürgen Furthmüller. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Physical review B*, 54(16):11169, 1996. 15

[54] Georg Kresse and Jürgen 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, 1996. 15

[55] John P Perdew, Matthias Ernzerhof, and Kieron Burke. Rationale for mixing exact exchange with density functional approximations. *The Journal of chemical physics*, 105(22):9982–9985, 1996. 15

[56] Peter E Blöchl. Projector augmented-wave method. *Physical review B*, 50(24):17953, 1994. 15

[57] Georg Kresse and Daniel Joubert. From ultrasoft pseudopotentials to the projector augmented-wave method. *Physical review b*, 59(3):1758, 1999. 15

[58] Kiran Mathew, Joseph H Montoya, Alireza Faghaninia, Shyam Dwarakanath, Muratahan Aykol, Hanmei Tang, Iek-heng Chu, Tess Smidt, Brandon Bocklund, Matthew Horton, et al. Atomate: A high-level interface to generate, execute, and analyze computational materials science workflows. *Computational Materials Science*, 139:140–152, 2017. 15

 https://doi.org/10.52202/079017-1226

Appendix

A Experiment details

In this section, we provide additional experimental details, including metrics used for evaluation, baselines, architecture and training of the diffusion model with the compact crystal representation, and details of the setup for the DFT calculations.

A.1 Details of evaluation metrics

Structure and composition validity. The structure and composition validity metrics follow [5]. The structure validity determins that a structure is valid as long as the shortest distance between any pair of atoms is larger than 0.5 Å [19]. The composition is valid if the overall charge is neutral as computed by SMACT [22].

Uniqueness. We determine a generated formula is unique if the reduced form of the formula does not exist in either Materials Project [10] or ICSD [11]. For instance, if ICSD contains formula in the form of AB2, we consider A2B4 generated by the model as a duplicate (thus not unique) structure.

Match rate. To compute the match rate, we use the `StructureMatcher` module from pymatgen's analysis package. We set the hyperparameters of the matcher following [9], specifically with $\texttt{stol} = 0.5, \texttt{ltol} = 0.3, \texttt{angle\_tol} = 10$. For each family of crystals in perovskite, pyrochlore, and spinel, we first curate the reference set by downloading CIF files from Materials Project [10] that is likely to belong to each family based on formula and space group. We then use `fit_anonymous` method of the matcher to compare each generated structure to the structures in the reference set. A generated structure is considered matched if `fit_anonymous` returns true for at least one reference structure of the corresponding family. Note that this approach might result in false positive matches. For example, when we selected the reference set for pyrochlore, we downloaded CIF files Material Project that have composition A2B2O7. However, not all A2B2O7 are pyrochlore, so generated structures may still not be a pyrochlore despite being matched to one of the reference structures.

A.2 Details of baselines

We use the following prompts in Table 7 to generate the CIF files for the end-to-end prompting baseline or to generate the chemical formulae for GenMS.

Method	Prompt
Prompt CIF (baseline)	"I want you to generate another crystal information (CIF) files for a stable and potentially realistic material that belongs to {category}. [(Optional) Here are some information about {category} from Wikipedia.] Below are some examples of CIF files from this category: {example1, example2, ...} Please generate one more file for a crystal that is not in existing materials databases like Materials Project and ICSD. Please make sure the CIF file is valid. Just generate the file and do not say anything else."
Prompt formula (GenMS)	[(Optional) Here are some information about {category} from Wikipedia.] Please give me a list of chemical formulae for a hypothetical material for {category}. I want the formula to be stable, and potentially realistic and do not exist in dataset like Materials Project or ICSD. Please just give the formula and do not say anything else."

Table 7: **LLM prompts for baseline and GenMS.**

A.3 Compute, architecture, and training

We repurpose the 3D U-Net architecture [51, 52] into modeling atoms within a crystal structure by their x, y, z locations concatenated with atom number (number of protons) a. As a result, we can

represent each crystal structure using an $Ax4$ matrix where A is the total number of atoms in the structure, and the dimension with size 4 represents the x, y, z location and atom number of each atom. We repurpose the spatial downsampling and upsampling passes from typical U-Net for images or videos, and keep the resolution (number of points) the same, but still employ residual network with concatenating skip connections (see Figure 2 from the main text). Below we show the architecture and hyperparameters used in the diffusion model for crystals with compact representation.

Hyperparameter	**Value**
Learning rate	5e-5
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.99$)
Base hidden dimension	256
Hidden dimension multipliers	1, 2, 4
Number of mlp and self-attention blocks	9
Batch size	512
EMA	0.9999
Weight decay	0.0
Prediction target	ϵ
Attention head dimension	64
Dropout	0.1
Training hardware	64 TPU-v4 chips
Diffusion noise schedule	cosine
Noise schedule log SNR range	[-20, 20]
Training steps	200000
Sampling timesteps	256
Sampling log-variance interpolation	$\gamma = 0.1$

Table 8: **Hyperparameters for training** the diffusion model in GenMS.

A.4 Details of DFT calculations

In all our density functional theory (DFT) calculations, we employ the Vienna ab initio simulation package (VASP) [53, 54] with the Perdew-Burke-Ernzerhof (PBE) [55] functional and projector-augmented wave (PAW) potentials [56, 57]. Our computational settings align with those used in the Materials Project, as implemented in pymatgen [21] and atomate [58]. These settings include the application of the Hubbard U parameter to selected transition metals in DFT+U calculations, a plane-wave basis cutoff of 520 eV, specific magnetization settings, and the use of PBE pseudopotentials. However, we opt for updated versions of potentials for Li, Na, Mg, Ge, and Ga, maintaining the same valence electron count. For structural optimization, our protocol involves a two-stage relaxation of all geometric parameters, followed by a final static computation. We utilize the custodian package [21] to manage any issues with VASP and to make necessary adjustments to the simulations. Additionally, we generate gamma-centered k-points for hexagonal cells, deviating from the conventional Monkhorst-Pack scheme. We initialize our simulations with ferromagnetic spin, observing that attempts to explore alternative spin configurations were computationally too demanding. In our ab initio molecular dynamics (AIMD) simulations, we disable spin polarization and employ the NVT ensemble with a 2 fs timestep. For systems containing hydrogen, we reduce the timestep to 0.5 fs to ensure accuracy.

https://doi.org/10.52202/079017-1226

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: We provided thorough experiments and explanations of the algorithms to jusify the claim.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discussed that in the conclusion section and we pointed out four major limitations.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

Answer: [NA]

Justification: We do not introduce new theoretical result.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: We use public LLMs, and the prompts are all included in the appendix. For our work we have provided enough details for reimplementation, and we will work on open sourcing as well.

   Guidelines:

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

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

https://doi.org/10.52202/079017-1226

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

Answer: [No]

Justification: The data and prompts are all public and we have provided the information. We are working on open sourcing the code after going through the internal approval process.

Guidelines:

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

6. **Experimental Setting/Details**

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

Answer: [Yes]

Justification: We have provided the information in the appendix and main paper.

Guidelines:

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

7. **Experiment Statistical Significance**

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

Answer: [Yes]

Justification: We provided the variance of the results in almost all tables when applicable.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: We report this in our experiment section.

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: We have checked.

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [NA]

    Justification: This work is exploratory and is at a too early of a stage to have broad impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

https://doi.org/10.52202/079017-1226

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: No risk as far as we can see.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We have properly cited the models (Gemini) we used in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

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

    Answer: [NA]

    Justification: No new assets.

    Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [NA]

    Justification: Not involved.

    Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [NA]

    Justification: no crowdsourcing nor research

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-1226