DEVELOPING A CRYSTAL STRUCTURE PREDICTION ALGORITHM WITH AGENT-GUIDED SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Language-model agents show promise for machine learning research, but evidence that agent-guided research can advance the state of the art on established tasks remains limited. We investigate this possibility in crystal structure prediction (CSP), a central task in materials discovery that combines affordable experimentation with a comparatively underexplored generative modeling landscape. An agent-guided search explores eighteen generative framework–architecture pairs and identifies masked generative modeling as the strongest approach under the initial training budget. Subsequent experimentation, combined with four targeted human interventions, produces MaskGXT, a masked generative crystal transformer that jointly predicts crystallographic symmetry, lattice parameters, and atomic coordinates. MaskGXT incorporates circular ordinal label smoothing, sub-bin coordinate refinement, and space-group-stratified sampling to address periodic geometry, coordinate precision, and polymorph coverage. Our experiments show that MaskGXT achieves the highest match rates compared to an extensive set of CSP baselines on MP-20 and MPTS-52 benchmarks.

1 INTRODUCTION

Language-model agents are increasingly capable of conducting iterative machine learning research, from proposing ideas and implementing algorithms to running experiments and analyzing results (Lu et al., 2024; Yamada et al., 2025; Jiang et al., 2025). This progress has motivated efforts toward both end-to-end research automation and autonomous experimentation with targeted human guidance (Liu et al., 2026). While these systems show promise, evidence that agent-guided research can produce algorithms that advance the state of the art on established tasks remains limited. Demonstrating such advances would provide concrete evidence of the practical value of agent-guided research.

Crystal structure prediction (CSP) is a central problem in computational materials discovery which seeks stable atomic arrangements and lattice geometries given a chemical composition (Butler et al., 2018; Oganov et al., 2019). Despite recent efforts (Xie et al., 2022; Jiao et al., 2023; Miller et al., 2024), its generative modeling landscape remains comparatively underexplored, partly due to CSP lying at the narrow intersection of crystallography, materials science, and generative modeling. This creates an opportunity to combine agents' exploration of modeling alternatives with targeted scientific expertise. Affordable training and established structure-matching metrics further make CSP suitable for iterative algorithm development.

In this work, we explore the design space of generative models for CSP through an agent-guided research loop that proposes candidate algorithms, implements and evaluates them, and expands promising directions. The search considered eighteen generative framework–architecture pairs and identified masked generative modeling (Devlin et al., 2019; Chang et al., 2022) as the strongest approach under the initial training budget, measured by validation METRe (Martirossyan et al., 2025), a polymorph-aware match rate. Subsequent development combined agent-led experimentation with four targeted human interventions concerning crystal symmetry, symmetry-aware augmentation, coordinate precision, and polymorph coverage.

The resulting model, MaskGXT, casts CSP as masked token generation. It represents the space group, Wyckoff positions, lattice parameters, and atomic coordinates as a joint sequence and predicts masked tokens in parallel. MaskGXT adapts this framework to crystal structures through circular ordinal label smoothing for periodic coordinates, a sub-bin offset head for geometric precision, and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

space-group-stratified sampling for polymorph coverage. These components address the geometric and structural requirements that arise when applying discrete generative modeling to crystals.

MaskGXT achieves the highest match rates among the evaluated methods on MP-20 and MPTS-52, reaching 79.75% METRe on the MP-20 polymorph split, compared with 73.64% for the strongest evaluated baseline. Ablations examine how coordinate modeling and symmetry information contribute to geometric accuracy and polymorph recovery. Together, these results establish a competitive CSP algorithm and demonstrate a practical route to scientific algorithm development through agent-guided exploration with targeted expert input.

In summary, our contributions are twofold:

- **A demonstration of scientific algorithm development through human–agent collaboration.** We show how agent-guided exploration with four targeted human interventions produces a competitive CSP algorithm. The documented search trajectory distinguishes agent-originated proposals from expert guidance, providing a concrete example of how researchers can delegate implementation and experimentation while steering selected mechanisms and objectives.
- **State-of-the-art generative model for CSP.** We introduce MaskGXT, a masked generative model that considers crystallographic symmetry and geometry and improves match rates on established CSP benchmarks. Our component analyses identify adaptations that make masked modeling effective for periodic coordinates, geometric precision, and polymorph coverage, informing future crystal generative models.

2 Related Work

Crystal structure prediction models. Learning-based CSP has largely followed two modeling routes. The first treats lattice parameters and fractional coordinates as continuous variables: CDVAE (Xie et al., 2022) introduced diffusion-based crystal generation, and later work improved the geometric formulation through equivariant diffusion, Riemannian flow matching, and stochastic interpolants (Jiao et al., 2023; Miller et al., 2024; Zeni et al., 2025; Luo et al., 2025; Hoellmer et al., 2025), at the cost of stochastic or iterative integration at sampling time. The second route incorporates crystallographic symmetry explicitly: template-conditioned models take a space group or Wyckoff template as input (Jiao et al., 2024; Levy et al., 2025), template-generating models produce the symmetry description and leave the coordinates to a downstream step (Zhu et al., 2024; Kelvinius et al., 2025; Kazeev et al., 2025), and joint models such as CrystalFormer emit space group, Wyckoff letters, and coordinates autoregressively in one sequence (Cao et al., 2025). MaskGXT belongs to the joint family, but decodes all positions in parallel instead of autoregressively, enabling space group stratified decoding to generate multiple candidate structures.

Discrete and masked generative modeling. Discrete generative models corrupt a token sequence and learn to reverse the corruption: discrete-diffusion models such as D3PM denoise a categorical process over many steps (Austin et al., 2021), while masked generative models corrupt by masking and reconstruct all positions in parallel (Devlin et al., 2019; Chang et al., 2022). Discrete representations have entered crystal generation along two lines. Symmetry-first models discretize only the symmetry description, keeping the coordinates continuous or absent: WyckoffDiff runs categorical diffusion over space group and Wyckoff tokens (Kelvinius et al., 2025), and SymmCD discretizes the site symmetry and atom type (Levy et al., 2025). Text-first models discretize the geometry itself, writing the crystal as a CIF string and generating it left to right with a language model (Flam-Shepherd & Aspuru-Guzik, 2023; Gruver et al., 2024; Antunes et al., 2024; Sriram et al., 2024), decoding a coordinate token as an unrelated symbol. MaskGXT instead builds the periodicity and ordinality of the fractional coordinate bins with circular ordinal smoothing and a sub-bin offset head. It decodes in parallel from a single bidirectional transformer, whose categorical posterior makes confidence-ranked greedy decoding and space group stratified sampling possible.

Agent-guided search. A growing body of work studies language-model agents that automate parts of the scientific research loop (Wei et al., 2025). The AI Scientist and its successor generate ideas, write code, and run experiments with limited human intervention (Lu et al., 2024; Yamada et al., 2025), FunSearch and AlphaEvolve use program search to discover algorithms (Romera-Paredes et al., 2024; Novikov et al., 2025), and multi-agent systems explore collaboration and long-horizon

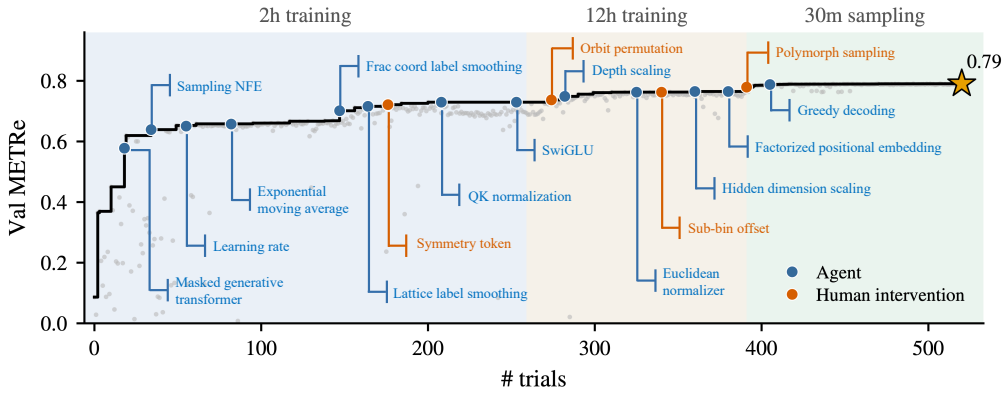


Figure 1: **Search trajectory that produced MaskGXT.** Validation METRe against the number of trials. Each gray point is one trained, scored candidate; the black step line is the running best. Blue markers are components the agent introduced; orange markers are the four human interventions. Shaded regions give the per-candidate budget: 2h, then 12h training, then 30m sampling.

research trajectories (Liu et al., 2026; Gao et al., 2026). AutoResearch instead repeatedly modifies training code and retains empirically improved variants (Karpathy, 2026). AI engineering agentic benchmarks take a related approach, evaluating whether such agents can solve software- and machine-learning-development tasks by editing code and running experiments (Wijk et al., 2024; Grosnit et al., 2024; Yang et al., 2024; Jiang et al., 2025; Toledo et al., 2025; Chan et al., 2025; Nathani et al., 2025).

3 Agent-Guided Search for Crystal Structure Prediction

This section describes the search that produced MaskGXT (Sections 4 and 5) and the sparse human steering along the way; Figures 1 and 2 trace the run and the tree it grew.

Agent-guided search. We search with an agent built on AIDE's (Jiang et al., 2025) tree-search loop for machine learning engineering. We run it on the task of maximizing validation METRe (Martirossyan et al., 2025) on the MP-20 polymorph split. A single orchestrator, driven by Claude Opus 4.7 (Anthropic, 2026), expands a tree of candidate training programs using AIDE's core loop: *draft* implements a candidate, *improve* makes one change to it, and *debug* repairs a failed one. An *idea* step, dispatched ahead of each *draft*, proposes a (generative framework, architecture) pair from outside materials science. The orchestrator is also the channel for the sparse human steering below. The search spends more compute per candidate as its scope narrows, in three stages that each fix what the previous one found. *Discovery* trains each candidate for two hours and picks the framework, *scale-up* improves the model under a 12-hour budget, and *sampling* tunes the decoder under a 30-minute budget, since a decoder change needs no retraining. Appendix A details the operators.

Search trajectory. Table 1 gives the best three of the 18 candidates *idea* proposed. Under the two-hour discovery budget, node 18's masked generative modeling (Chang et al., 2022) lineage reached 72.96% validation METRe against 45.03% for the next best framework, and the rest of the search builds on it. Figure 1 traces what followed: the agent carried out most of the remaining engineering on its own, and the four human interventions, sparse and targeted, supplied mechanisms the agent's own search had not found. Appendix B reproduces the transcripts.

Table 1: **Best three of the 18 frameworks tried.** Highest validation METRe reached under the two-hour discovery budget.

Generative framework	Architecture	METRe (%)
Masked discrete diffusion	**Bidirectional Transformer**	**72.96**
Consistency model	Diffusion Transformer (DiT)	45.03
Bayesian Flow Network	Perceiver-IO set transformer	42.24

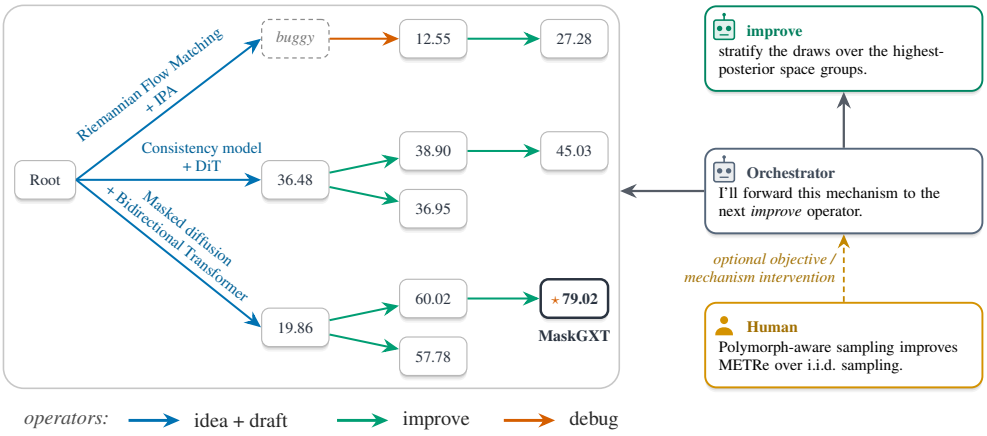


Figure 2: **Search tree.** Each node is a candidate CSP model and each edge a search operator. The three lineages drawn are the Riemannian flow matching, consistency model, and masked diffusion drafts, with their draft and final validation METRe; the intermediate nodes are a schematic of a search this size, not a literal log. The masked diffusion lineage becomes MaskGXT, and human interventions enter through the orchestrator.

4 MaskGXT: Masked Generative Crystal Transformer

We now introduce MaskGXT, the crystal structure prediction model that our search of Section 3 produced. Rather than iteratively evolving continuous atomic coordinates through a diffusion or flow process, MaskGXT instead casts the crystal as a discrete, partially masked token sequence and learns to fully reconstruct all of its missing tokens together.

Masked generative modeling of a crystal has to meet two demands that its image and text counterparts do not (Figure 3). Fractional coordinates are periodic and ordered, so their bins carry a metric that a plain categorical target discards, and quantizing them caps geometric precision. MaskGXT answers both with circular ordinal targets over a discrete representation on the fractional-coordinate torus and a dedicated sub-bin offset head, as well as symmetry augmentations that further expose the model to the many equivalent descriptions of one crystal.

4.1 Discrete Crystal Tokenization

Crystal tokenization. In CSP, the input is a set of atom types $\mathbf{A} = (a_1, \ldots, a_N)$, and the goal is to predict the lattice $\mathbf{L}$ and fractional coordinates $\mathbf{F} = (\mathbf{f}_1, \ldots, \mathbf{f}_N) \in [0, 1)^{3\times N}$. MaskGXT tokenizes a crystal into $1{+}6{+}5N$ tokens: one space group token, six lattice tokens for the quantized lengths and angles, and five tokens per site, namely three quantized coordinate tokens, a Wyckoff token, and an atom-type token (Figure 3). Atom types $\mathbf{A}$ are given as a condition, leaving the space group g, lattice, coordinate, and Wyckoff tokens $\mathbf{W}_g$ to predict. Symmetry is thus predicted jointly with the structure, rather than imposed through a separate module. Because the symmetry, lattice, and coordinate tokens are each decoded independently, this consistency is instead encouraged through training rather than being enforced as a hard constraint (Appendix C).

Coordinate discretization. To model crystal geometry with a masked transformer, MaskGXT represents continuous quantities as discrete tokens. It maps each fractional coordinate $f \in [0, 1)$ to a single discrete bin

$$q(f) = \lfloor (f \bmod 1)\, K \rfloor \in \{0, \ldots, K-1\}, \tag{1}$$

and dequantizes it when generating a structure at the bin center, refined by a learned sub-bin offset. The modulo in Equation (1) places each coordinate on a circle of K discrete bins, so that a site near $f = 1$ always sits just one step away from a site near $f = 0$.

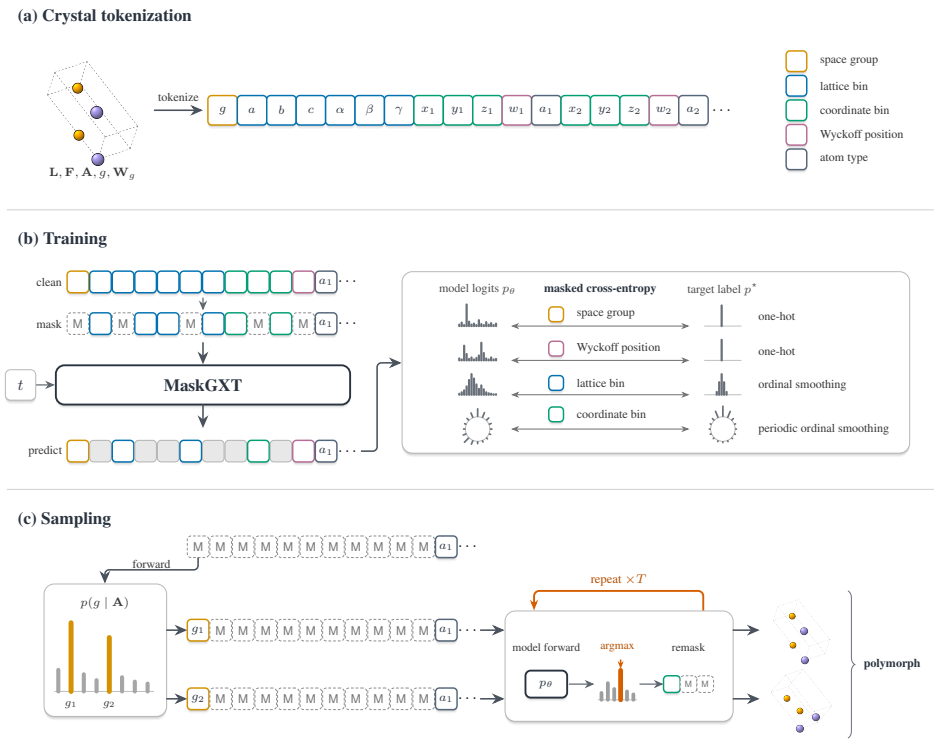


Figure 3: **MaskGXT tokenization, training, and sampling.** (a) A crystal $(\mathbf{L}, \mathbf{F}, \mathbf{A}, g, \mathbf{W}_g)$ is tokenized into $1{+}6{+}5N$ tokens: one space group token, six lattice tokens (lengths and angles), and, per site, three coordinate tokens, one Wyckoff token, and one atom-type token, with atom types $\mathbf{A}$ given as side information. (b) A random fraction of tokens is masked and the transformer reconstructs them, trained with a masked cross-entropy loss. (c) From a fully masked sequence with only $\mathbf{A}$ known, the space group posterior seeds distinct polymorph branches, and greedy decoding then unmasks the remaining tokens over T steps.

4.2 Masked Learning Objective

Ordinal, circular label smoothing. Treating the bins as unordered categories would discard the metric structure of coordinates and the adjacency of bin 0 and bin $K{-}1$ on this circle. MaskGXT therefore replaces the one-hot target by an *ordinal, circular* soft target (Szegedy et al., 2016; Müller et al., 2019): for a target bin b, the supervision over bin b' is

$$p_b^\star(b') \;\propto\; \exp\!\left(-\frac{d(b,b')^2}{2\sigma^2}\right), \qquad d(b,b') = \min\big(|b-b'|,\; K-|b-b'|\big), \tag{2}$$

where σ controls the spread. The soft target gives most of its probability to the true bin and a little to nearby bins, so predicting a neighboring bin incurs less loss than a far one. Lattice tokens use the same construction with the ordinary, non-circular distance, since lattice parameters do not wrap.

Masked cross-entropy. Following masked generative modeling (Austin et al., 2021; Chang et al., 2022), training corrupts a crystal by independently replacing each token with a mask symbol with probability $t \sim \mathcal{U}(0,1)$, and trains a transformer f_θ to predict the original tokens at the masked positions. The loss is a masked cross-entropy against the per-token targets, summed over the four streams with weights w_m:

$$\mathcal{L}(\theta) \;=\; \mathbb{E}_{t,\,\mathcal{C},\,\mathcal{M}} \sum_{m\in\{\text{lat},\text{coord},\text{sg},\text{wyk}\}} w_m \,\frac{1}{|\mathcal{M}_m|} \sum_{i\in\mathcal{M}_m} \mathrm{CE}\big(f_\theta(\mathcal{C}_{\setminus\mathcal{M}})_i,\; p^\star_{y_i}\big)\,, \tag{3}$$

where $\mathcal{C}$ is the tokenized crystal, $\mathcal{M}$ the set of masked positions, $\mathcal{C}_{\setminus\mathcal{M}}$ the sequence with those positions replaced by the special mask symbol, $\mathcal{M}_m$ the masked positions within stream m, and y_i

the clean target value at position i; the weights w_m balance the four streams, and the space group and Wyckoff streams instead use simple one-hot targets.

Sub-bin coordinate refinement. While discretization provides a stable categorical representation, it limits geometric precision to half a bin width. To recover sub-bin accuracy, we attach a lightweight regression head that reads the trunk feature $\mathbf{h}$ at each lattice and coordinate position and predicts a bounded offset $\delta \in (-\frac{1}{2}, \frac{1}{2})$. The offset is supervised by the true residual relative to the corresponding bin center using a smooth-L_1 loss, which is added to Equation (3). Specifically,

$$\delta = \tfrac{1}{2}\tanh\big(\mathbf{w}_{\text{off}}^{\top}\mathbf{h}\big)\,, \quad \delta^{\star} = Kf - \big(q(f) + \tfrac{1}{2}\big), \quad \mathcal{L}_{\text{off}} = \mathbb{E}\,\rho(\delta - \delta^{\star})\,, \tag{4}$$

where f denotes the ground-truth fractional or lattice value, $\delta^{\star}$ is its residual from the center of bin $q(f)$, and ρ is the smooth-L_1 penalty, averaged over unmasked lattice and coordinate positions. During decoding, MaskGXT reconstructs each continuous value by shifting the predicted bin center by this offset, $\hat{f} = (q(f) + \frac{1}{2} + \delta)/K$ (Table 3a). The $\frac{1}{2}\tanh$ bounds the offset to half a bin width, so the refined value always remains correctly within its bin.

Architecture. The network has three parts: token embeddings, a bidirectional Transformer trunk, and per-stream heads (Figure 5). A per-site linear projection fuses the five token embeddings of each site into one atom-wise feature, so the trunk processes a length-$1{+}6{+}N$ sequence of one space group, six lattice, and N atom-wise features, each carrying a learned positional encoding over the canonical site order and a mask-ratio embedding. The trunk is a Pre-LN Transformer (Xiong et al., 2020) with QK-normalized self-attention (Dehghani et al., 2023) and SwiGLU feedforward blocks (Shazeer, 2020), and linear heads read the categorical logits of the four token streams and the sub-bin offsets of Equation (4) directly from its final output. Appendix D gives the full forward pass and all of the block-by-block equations in detail.

4.3 Symmetry Augmentation

The same crystal admits many equivalent descriptions, since its atom ordering and unit-cell origin are arbitrary. MaskGXT trains the model to assign the same space group and Wyckoff tokens to all of them through two online augmentations, applied per sample before tokenization (Algorithm 1).

Canonicalization. For the atom ordering, we follow the canonical ordering of MCFlow (Seong et al., 2026), which groups sites into symmetry orbits and sorts them. The order among orbits that share an element and Wyckoff position, and of atoms within an orbit, remains arbitrary, so during training we shuffle whole orbits and the atoms within each orbit with the probability in Table 7. Permuting only within the orbit structure, rather than over all $N!$ atom orders, augments only symmetry-equivalent descriptions and outperforms full random permutation.

Euclidean-normalizer shift. For the origin, shifting the unit cell changes both the coordinates and the Wyckoff site of each atom, so we move them together. We draw a continuous shift $\mathbf{r} \sim \mathcal{U}[0,1)^3$, translate every fractional coordinate by $\mathbf{r}$ modulo one, and assign each atom the Wyckoff letter of the Euclidean-normalizer translation coset whose representative is nearest to $\mathbf{r}$. Unlike Kazeev et al. (2025), who enumerate the discrete coset representatives of the normalizer, our shift is continuous. We precompute the normalizer translation cosets and their Wyckoff letters for all 230 space groups, so relabeling is a table lookup at train time (Algorithm 1).

4.4 Sampling

Space group stratified sampling. A composition may form several polymorphs, so MaskGXT samples a diverse set of candidates by branching over space groups. For a requested budget of S candidates per composition, it computes the space group posterior from the fully masked sequence $\mathcal{C}_{\text{mask}}$ with only the composition observed, $p_\theta(g \mid A) = \text{softmax}\big(f_\theta(\mathcal{C}_{\text{mask}})_{\text{sg}}\big)$, selects the S highest-posterior space groups, and creates one sampling branch for each selected g. In each branch, the space group token is fixed to g and the remaining lattice, coordinate, and Wyckoff tokens are decoded by the same greedy procedure below (Algorithm 3).

Table 2: **Crystal structure prediction across the three test sets.** MP-20 and MPTS-52 report match rate (MR) and matched-pair RMSE, each as unfiltered/filtered values; the MP-20 polymorph split reports METRe and cRMSE. MaskGXT uses greedy decoding on MP-20 and MPTS-52 and adds space group stratified sampling on the polymorph split. Crystalite's filtered values are from Veljković et al. (2026); METRe values are from Martirossyan et al. (2025), and from Seong et al. (2026) for MCFlow. Crystalite's unfiltered values are measured here from its released CSP checkpoints, and its whole polymorph-split row from a retrain in the configuration of its released MP-20 CSP checkpoint, as MaskGXT also uses one training configuration for both splits. Parameter counts are measured from each model's released code or weights, except MCFlow's, which it reports; Appendix H compares training compute. Our rows are the mean over three seeds; Table 6 gives their standard deviations. **Bold** marks the best value in each column.

Model	Params	MP-20		MPTS-52		MP-20 polymorph split	
		MR (%) ↑	RMSE ↓	MR (%) ↑	RMSE ↓	METRe (%) ↑	cRMSE ↓
DiffCSP	12M	57.82 / 52.51	0.0627 / 0.0600	15.79 / 14.29	0.1533 / 0.1489	53.14	0.279
FlowMM	12M	66.22 / 59.98	0.0661 / 0.0629	22.29 / 20.28	0.1541 / 0.1486	65.18	0.226
OMatG	12M	69.83 / 63.75	0.0741 / 0.0720	27.38 / 25.15	0.1970 / 0.1931	70.50	0.187
MCFlow	134M	69.23 / 63.14	0.0663 / 0.0611	28.77 / 26.46	0.1610 / 0.1577	70.70	0.195
Crystalite	270M	72.10 / 66.05	**0.0325 / 0.0329**	33.99 / 31.49	**0.0734 / 0.0701**	73.64	0.157
MaskGXT	248M	**73.97 / 67.36**	0.0342 / 0.0331	**37.21 / 33.97**	0.0974 / 0.0930	**79.75**	**0.128**

Confidence-ranked greedy decoding. Within each branch, MaskGXT returns the most probable structure by replacing stochastic token sampling with deterministic token selection. Starting from the fully masked sequence with **A** and the branch space group given, it iteratively predicts posterior distributions over all of the masked token streams, fills the highest-confidence positions using token-wise *argmax*, and re-masks the remaining positions according to the decoding schedule (Algorithm 2). Because the output space is discretized, this greedy rule is well defined over a finite vocabulary for lattice, coordinate, space group, and Wyckoff tokens. It therefore returns a high-confidence deterministic structure, in contrast to diffusion- and flow-based samplers whose final outputs are obtained by integrating or sampling a continuous generative process.

5 Experiments

The experiments answer two questions: (1) whether the discrete masked formulation outperforms continuous diffusion- and flow-based CSP models, under both the standard one-to-one protocol and polymorph-aware evaluation, and (2) whether MaskGXT's components improve geometric accuracy and polymorph coverage.

5.1 Setup

Datasets. We evaluate on two standard inorganic-crystal CSP benchmarks, *MP-20* (Xie et al., 2022) and *MPTS-52* (Baird et al., 2024), and on the MP-20 polymorph split of Martirossyan et al. (2025), which repartitions MP-20 by composition rather than by structure so that all polymorphs of a composition are retained together, giving a direct test of polymorph coverage under METRe (Appendix F gives split sizes and construction). This polymorph-split test set is disjoint from the validation split used during the search of Section 3; MP-20 and MPTS-52 are evaluated on their standard test splits.

Evaluation metrics. For standard one-to-one evaluation we report match rate (MR), the fraction of matched test references, and matched-pair RMSE, each in an *unfiltered* and a *validity-filtered* form. For polymorph-aware evaluation we report METRe (Martirossyan et al., 2025) and the corrected RMSE (cRMSE); Appendix F gives the full `StructureMatcher` protocol and metric definitions.

5.2 Baselines and MaskGXT Protocol

We compare MaskGXT against DiffCSP (Jiao et al., 2023), FlowMM (Miller et al., 2024), OMatG (Hoellmer et al., 2025), Crystalite (Veljković et al., 2026), and MCFlow (Seong et al., 2026).

For OMatG, we use the linear-ODE variant; for MCFlow, we use the base variant. METRe values are reproduced from Martirossyan et al. (2025), and from Seong et al. (2026) for MCFlow, except Crystalite's, which we measure after retraining it on the polymorph split, the only split it releases no checkpoint for, in the configuration of its released CSP checkpoint for MP-20.

MaskGXT is evaluated as a composition-conditioned CSP model: given only the test composition, it generates crystal structures by confidence-ranked masked parallel decoding over the token sequence described in Section 4, using T=150 decoding steps, beyond which coverage saturates (Figure 6). Unless otherwise specified, we follow the one-to-one protocol used by the baselines and generate one structure per test reference for MR and RMSE evaluation. For METRe, we use the space group stratified sampling protocol in Section 4, which spreads generations over high-posterior space groups to improve coverage of multiple polymorphs of the same composition. The reported MaskGXT model is a 248M-parameter transformer whose architecture and hyperparameters are given in Table 7, with the coordinate bin count reduced to 32 on MPTS-52, trained for up to 3000 epochs with early stopping on validation METRe. It is smaller than Crystalite, the strongest baseline, and reaches the checkpoint we evaluate on five to eight times less training compute (Appendix H).

5.3 One-to-One and Polymorph-Aware Evaluation

Table 2 reports standard one-to-one CSP performance on MP-20 and MPTS-52. MaskGXT achieves the highest match rate on both datasets, under both the unfiltered and validity-filtered definitions, with the largest margin on the harder MPTS-52 split. Crystalite attains the lowest matched-pair RMSE, so the two methods trade coverage against the precision of the structures they match. This suggests that sub-bin offset regression keeps discretized coordinates fully competitive with existing diffusion-based models, and that the discrete masked formulation generalizes well beyond the original MP-20 development and training setting.

The polymorph-split columns of Table 2 evaluate polymorph coverage with METRe on a split that partitions MP-20 by composition rather than by structure, so that each test composition is unseen during training and keeps all of its reference polymorphs. MaskGXT obtains the best METRe and cRMSE. This pattern matches the design of the sampler: because MaskGXT predicts the space group as an explicit token, space group stratified sampling allocates different generations to distinct high-posterior space groups, improving coverage of compositions with multiple polymorphs. MaskGXT's match rate also scales smoothly with parameter count, and even a 30M-parameter model already outperforms Crystalite's match rate on MP-20 (Appendix I).

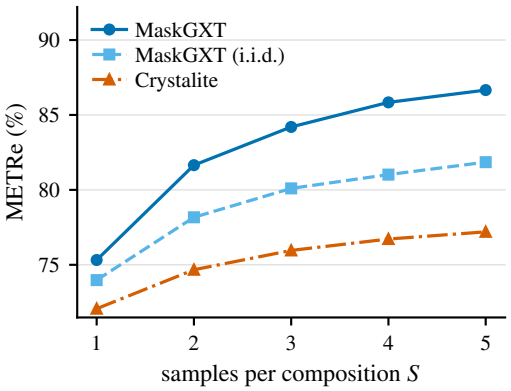


Figure 4: **Samples per composition.** METRe on the MP-20 polymorph split when every unique composition receives the same budget of S generated structures, measured against the full reference set. Every curve is measured here, on one trained checkpoint per model.

Uniform sampling. The METRe protocol in Table 2 gives each composition a budget equal to its number of reference structures, which is unavailable at deployment time. We therefore give every unique composition the same budget of S structures, sweeping S from one to five (Figure 4). MaskGXT leads Crystalite at every budget, and its lead grows with the budget since space group stratification focus on sampling on a distinct high-posterior space group, while independent draws increasingly repeat only the dominant one. MaskGXT also attains the lowest measured cRMSE at every sampling budget.

5.4 Ablations

Table 3a removes one training-time component of MaskGXT at a time, on the MP-20 polymorph split. Label smoothing and symmetry tokens both improve METRe, showing that periodic ordinal supervision and explicit crystallographic tokens contribute to coverage. Removing symmetry tokens costs the most, because space group stratified sampling is no longer available; removing the

Table 3: **Architecture and decoder ablation, on the MP-20 polymorph split.** (a) Each row removes one training-time component from MaskGXT and is retrained from scratch; a row without symmetry tokens cannot use space group stratified sampling, so its METRe comes from the plain stochastic decoder instead. (b) Each row removes one inference-time decoding component, resampling the same three main-results checkpoints, so no model is retrained; "– both" is masked generative modeling's original decoder. **MaskGXT** is the shared full-model baseline. Architecture rows are the mean over three seeds; decoder rows are the mean over the three checkpoints; sample standard deviation in parentheses.

(a) Architecture (train-time)

Model	METRe (%) ↑	cRMSE ↓
MaskGXT	79.75 (0.21)	0.1282 (0.0006)
– label smoothing	78.17 (0.26)	0.1381 (0.0018)
– symmetry tokens	75.50 (0.58)	0.1684 (0.0016)
– offset regression	79.62 (0.17)	0.1361 (0.0005)
– canonical ordering	77.92 (0.25)	0.1402 (0.0005)
– normalizer relabel	79.01 (0.31)	0.1320 (0.0009)

(b) Decoder (inference-time)

Model	METRe (%) ↑	cRMSE ↓
MaskGXT	79.75 (0.21)	0.1282 (0.0006)
– SG stratification	75.43 (0.10)	0.1468 (0.0004)
– greedy decoding	78.81 (0.37)	0.1497 (0.0012)
– both	76.52 (0.28)	0.1595 (0.0006)

canonical ordering or the Euclidean-normalizer relabel costs less but still hurts, so the symmetry machinery matters beyond the tokens themselves. Holding the decoder fixed at the plain stochastic one separates the two effects: 76.52% with symmetry tokens against 75.50% without. Most of the drop therefore comes from the stratified decoder that the tokens enable, not from the representation itself. Removing offset regression has little effect on METRe but substantially increases cRMSE, clearly confirming that the offset head primarily improves overall geometric precision.

Table 3b ablates the decoding strategy on the same checkpoints. Removing space group stratification causes the largest METRe drop, confirming that spreading generations across high-posterior space groups is the main mechanism behind MaskGXT's improved polymorph coverage. Removing greedy decoding lowers METRe and worsens cRMSE, indicating that confidence-ranked argmax decoding produces more accurate high-posterior structures than stochastic decoding. Removing both recovers a plain stochastic decoder, which scores *above* the stratification-only row: without stratification, argmax drives every generation toward the same structure, so greedy decoding pays off only when stratification spreads them much further apart.

6 Conclusion

We present MaskGXT, which represents the lattice, coordinates, space group, and Wyckoff positions as one token sequence and decodes it with confidence-ranked greedy decoding and space group stratified sampling. It attains the best match rate on MP-20 and MPTS-52 among the models we evaluate and substantially improves METRe on the MP-20 polymorph split; symmetry tokens with stratified decoding account for the coverage gain and the offset head for precision. An agent carefully selected masked generative modeling and implemented it, with a human separately supplying four crystallographic mechanisms (Appendix J).

AI Use Statement

MaskGXT was developed with an agent built on AIDE's (Jiang et al., 2025) tree-search loop, which wrote the model code and ran the search described in Section 3, under the human interventions marked in Figure 1 and documented in Section 3 and Appendix B. Beyond that, the authors used large language model assistance for implementation of the evaluation harness, experiment orchestration, and manuscript editing. All reported numbers were produced by runs the authors launched and verified, and the authors take full and sole responsibility for the content of this paper.

Reproducibility Statement

The code is available, anonymised for review, at https://anonymous.4open.science/r/haco_maskgxt_iclr2027-8CB7/, including the model's training, sampling, and evaluation scripts and the agent and search record that produced it. The release includes the complete search record, with browser viewers for the search tree and the sampled crystal structures. Section 5.1 gives the dataset splits, the evaluation protocol, and the metric definitions; Table 7 lists the full model and training configuration; Section 4 and Appendix E specify the tokenization, the training objective, and the sampler. Training uses fixed seeds, and each reported MaskGXT number is the mean over independently seeded runs with the spread reported alongside it.

References

Anthropic. Claude opus 4.7. https://www.anthropic.com/news/claude-opus-4-7, April 2026. Accessed: 2026-06-14.

Luis M. Antunes, Keith T. Butler, and Ricardo Grau-Crespo. Crystal structure generation with autoregressive large language modeling. *Nature Communications*, 15:10570, 2024.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Sterling G. Baird, Hasan Muhammad Sayeed, Joseph Montoya, and Taylor D. Sparks. sparks-baird/matbench-genmetrics. https://github.com/sparks-baird/matbench-genmetrics, 2024. [Accessed 03-05-2024].

Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018.

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

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Madry. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *International Conference on Learning Representations (ICLR)*, 2025.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11315–11325, 2022.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning (ICML)*, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pp. 4171–4186, 2019.

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.

Shanghua Gao, Ada Fang, and Marinka Zitnik. Autoscientists: Self-organizing agent teams for long-running scientific experimentation. *arXiv preprint arXiv:2605.28655*, 2026.

Antoine Grosnit, Alexandre Maraval, James Doran, Giuseppe Paolo, Albert Thomas, Refinath Shahul Hameed Nabeezath Beevi, Jonas Gonzalez, Khyati Khandelwal, Ignacio Iacobacci, Abdelhakim Benechehab, Hamza Cherkaoui, Youssef Attia El-Hili, Kun Shao, Jianye Hao, Jun Yao, Balázs Kégl, Haitham Bou-Ammar, and Jun Wang. Large language models orchestrating structured reasoning achieve kaggle grandmaster level. *arXiv preprint arXiv:2411.03562*, 2024.

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. In *International Conference on Learning Representations (ICLR)*, 2024.

Philipp Hoellmer, Thomas Egg, Maya M. Martirossyan, Eric Fuemmeler, Zeren Shui, Amit Gupta, Pawan Prakash, Adrian Roitberg, Mingjie Liu, George Karypis, Mark Transtrum, Richard G. Hennig, Ellad B. Tadmor, and Stefano Martiniani. Open materials generation with stochastic interpolants. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2502.02582.

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):011002, 2013.

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. AIDE: AI-driven exploration in the space of code. *arXiv preprint arXiv:2502.13138*, 2025.

Rui Jiao, Wenbing Huang, Peijia Lin, Jiaqi Han, Pin Chen, Yutong Lu, and Yang Liu. Crystal structure prediction by joint equivariant diffusion. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2309.04475.

Rui Jiao, Wenbing Huang, Yu Liu, Deli Zhao, and Yang Liu. Space group constrained crystal generation. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2402.03992.

Andrej Karpathy. AutoResearch: Ai agents running research on single-gpu nanochat training automatically. https://github.com/karpathy/autoresearch, 2026. GitHub repository. Accessed: 2026-06-15.

Nikita Kazeev, Wei Nong, Ignat Romanov, Ruiming Zhu, Andrey Ustyuzhanin, Shuya Yamazaki, and Kedar Hippalgaonkar. Wyckoff transformer: Generation of symmetric crystals. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2503.02407.

Filip Ekström Kelvinius, Oskar B Andersson, Abhijith S Parackal, Dong Qian, Rickard Armiento, and Fredrik Lindsten. WyckoffDiff–a generative diffusion model for crystal symmetry. In *Forty-second International Conference on Machine Learning*, 2025.

Minae Kwon, Sang Michael Xie, Kalesha Bullard, and Dorsa Sadigh. Reward design with language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Daniel Levy, Siba Smarak Panigrahi, Sékou-Oumar Kaba, Qiang Zhu, Kin Long Kelvin Lee, Mikhail Galkin, Santiago Miret, and Siamak Ravanbakhsh. Symmcd: Symmetry-preserving crystal generation with diffusion models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2502.03638.

Jiaqi Liu, Shi Qiu, Mairui Li, Bingzhou Li, Haonian Ji, Siwei Han, Xinyu Ye, Peng Xia, Zihan Dong, Congyu Zhang, et al. Autoresearchclaw: Self-reinforcing autonomous research with human-ai collaboration. *arXiv preprint arXiv:2605.20025*, 2026.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

Xiaoshan Luo, Zhenyu Wang, Qingchang Wang, Xuechen Shao, Jian Lv, Lei Wang, Yanchao Wang, and Yanming Ma. CrystalFlow: A flow-based generative model for crystalline materials. *Nature Communications*, 16(1):9267, 2025.

Maya M Martirossyan, Thomas Egg, Philipp Hoellmer, George Karypis, Mark Transtrum, Adrian Roitberg, Mingjie Liu, Richard G Hennig, Ellad B Tadmor, and Stefano Martiniani. All that structure matches does not glitter. In *Advances in Neural Information Processing Systems*, 2025.

Benjamin Kurt Miller, Ricky T. Q. Chen, Anuroop Sriram, and Brandon M Wood. FlowMM: Generating materials with Riemannian flow matching. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2406.04713.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Amar Budhiraja, Despoina Magka, Vladislav Vorotilov, Gaurav Chaurasia, Dieuwke Hupkes, Ricardo Silveira Cabral, Tatiana Shavrina, Jakob Foerster, Yoram Bachrach, William Yang Wang, and Roberta Raileanu. MLGym: A new framework and benchmark for advancing AI research agents. *arXiv preprint arXiv:2502.14499*, 2025.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Artem R Oganov, Chris J Pickard, Qiang Zhu, and Richard J Needs. Structure prediction drives materials discovery. *Nature Reviews Materials*, 4(5):331–348, 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.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024.

Kiyoung Seong, Sungsoo Ahn, Sehui Han, and Changyoung Park. Multimodal crystal flow: Any-to-any modality generation for unified crystal modeling. *arXiv preprint arXiv:2602.20210*, 2026.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Anuroop Sriram, Benjamin Kurt Miller, Ricky T. Q. Chen, and Brandon M. Wood. FlowLLM: Flow matching for material generation with large language models as base distributions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Atsushi Togo, Kohei Shinohara, and Isao Tanaka. Spglib: a software library for crystal symmetry search. *Science and Technology of Advanced Materials: Methods*, 4(1):2384822, 2024.

Edan Toledo, Karen Hambardzumyan, Martin Josifoski, Rishi Hazra, Nicolas Baldwin, Alexis Audran-Reiss, Michael Kuchnik, Despoina Magka, Minqi Jiang, Alisia Maria Lupidi, Andrei Lupu, Roberta Raileanu, Jakob Nicolaus Foerster, and Yoram Bachrach. AI research agents for machine learning: Search, exploration, and generalization in MLE-bench. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Tin Hadži Veljković, Joshua Rosenthal, Ivor Lončarić, and Jan-Willem van de Meent. Crystalite: A lightweight transformer for efficient crystal modeling. *arXiv preprint arXiv:2604.02270*, 2026.

Jiaqi Wei, Yuejin Yang, Xiang Zhang, et al. From AI for science to agentic science: A survey on autonomous scientific discovery. *arXiv preprint arXiv:2508.14111*, 2025.

Hjalmar Wijk, Tao Lin, Joel Becker, Sami Jawhar, Neev Parikh, Thomas Broadley, Lawrence Chan, Michael Chen, Josh Clymer, Jai Dhyani, Elena Ericheva, Katharyn Garcia, Brian Goodrich, Nikola Jurkovic, Megan Kinniment, Aron Lajko, Seraphina Nix, Lucas Sato, William Saunders, Maksym Taran, Ben West, and Elizabeth Barnes. RE-bench: Evaluating frontier AI R&D capabilities of language model agents against human experts. *arXiv preprint arXiv:2411.15114*, 2024.

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

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10524–10533. PMLR, 2020.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 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. *Nature*, 639:624–632, 2025. URL https://arxiv.org/abs/2312.03687.

Ruiming Zhu, Wei Nong, Shuya Yamazaki, and Kedar Hippalgaonkar. WyCryst: Wyckoff inorganic crystal generator framework. *Matter*, 7(10):3469–3488, 2024.

Henry Peng Zou et al. LLM-based human-agent collaboration and interaction systems: A survey. *arXiv preprint arXiv:2505.00753*, 2025.

A TREE-BASED SEARCH AND ORCHESTRATION

Search tree and orchestrator. Each node in the search tree is a self-contained candidate: a complete training program that defines a generative model for CSP. Edges are search operators built on AIDE's (Jiang et al., 2025) tree-search loop, and a single orchestrator driven by Claude Opus 4.7 (Anthropic, 2026) decides which node to expand, dispatches the operator, evaluates the resulting candidate against validation METRe, and adds it to the tree, running up to eight candidates in parallel across eight RTX 3090 GPUs (Figure 2, main text). The orchestrator is also the sole interface for human steering: an intervention enters through the orchestrator, which folds it into the operator it dispatches.

Operators. The agent expands the tree with four main operators, three of them AIDE's (Jiang et al., 2025): *draft* instantiates a methodology into runnable code; *improve* makes one coherent change to an existing candidate; and *debug* repairs a candidate that failed. We add *idea*, dispatched ahead of *draft*, to propose the methodology itself. Before training, *draft* and *improve* each predict why their code should raise validation METRe. Two auxiliary operators support the search. The *smoke* operator runs a lightweight sanity check, using a few training and sampling steps on one batch. The *analyze* operator interprets a finished run against that prediction, judges whether the result is valid, and records a finding that guides what to change next. All operators use the same Opus 4.7 as the orchestrator.

Cross-domain idea generation. The *idea* operator deliberately searches outside materials science. Rather than perturbing the current method, it surveys the broader generative-modeling literature, across vision, language, and other fields, and returns (generative framework, architecture) pairs that have not been applied to crystals but have a credible mechanism for competing on CSP. The operator selects each pair under two criteria: novelty to CSP and adaptability to periodic crystals. In our run, it proposed masked generative modeling (Chang et al., 2022), which the agent then instantiated and refined into MaskGXT (Appendix B).

Task and objective. The agent is asked to develop a model that generates crystal structures from chemical compositions. Its objective is to maximize validation METRe on the MP-20 polymorph split, which rewards recovery of any reference polymorph of a composition and so captures both structural accuracy and polymorph coverage. The entire development loop is confined to this validation split: framework proposals, component choices, and sampler refinements are selected only from its feedback.

Stages. The search proceeds in three stages of decreasing scope, each fixing what the previous one found (Figure 1). The discovery stage searches over the generative framework, training each candidate under a two-hour budget while the full operator set runs until validation METRe stops improving. The scale-up stage fixes the framework and optimizes the model under a 12-hour per-candidate budget. The sampling stage fixes the model and tunes the decoder under a 30-minute sampling budget.

Human steering. The human intervenes only through the orchestrator and only at the level of a mechanism or an objective, never as code. A mechanism supplies domain knowledge the agent lacks (Zou et al., 2025): that crystal symmetry should be represented explicitly, that augmentation should act at the level of symmetry orbits, and that independent draws under-cover a composition's polymorphs. An objective states a goal and leaves the method open (Kwon et al., 2023): that sub-bin coordinate precision should be recovered without leaving the discrete model. The orchestrator folds each intervention into the operator it dispatches, and the agent realizes it in code, including the crystallographic details (Appendix B).

Node evaluation and selection. The agent scores a node by training it under the stage-specific per-candidate budget and measuring validation METRe. It then expands promising nodes and prunes unproductive ones, so that strong lineages receive more compute over time. This mirrors the population- or tree-based search used by AI engineering agents (Lu et al., 2024; Jiang et al., 2025; Toledo et al., 2025), with the empirical metric as the primary basis for selection. Before com-

mitting a training budget to a candidate, the orchestrator sets up an isolated environment for it and screens it with the smoke operator, and it holds one node per GPU across the eight parallel slots.

Search-tree log. Table 4 logs the opening of the search in the order nodes were created: the 18 draft proposals of the discovery stage, of which Table 1 gives the best three, followed by their earliest debug and improve children. It shows the same pattern Figure 1 plots: most of the 18 frameworks plateau within a few points of their draft score, while the masked discrete diffusion lineage from node 18 pulls ahead and keeps improving through the rest of the search.

Table 4: **Search-tree log, nodes 1–72.** One row per node, in the order the search created them: the 18 draft proposals of the discovery stage, followed by their debug and improve children. "Framework" and "architecture" repeat the lineage's choice at each node; **bold** marks the lineage MaskGXT is built on. Performance is validation METRe on the MP-20 polymorph split; it is left blank for a debug node, whose fix is scored by the node that follows it, and for a node whose lineage produced no scored run. The log continues for 451 further nodes across the discovery, scale-up, and sampling stages, not shown here; Figure 1 plots the full run, on which the masked discrete diffusion lineage (node 18) pulls decisively ahead of the rest.

Node	Op.	What the node did	Generative framework	Architecture	METRe (%)
1	draft	propose framework	Score-based EBM + LJ prior	GemNet MPNN	
2	draft	propose framework	Mirror Flow Matching	MACE	
3	draft	propose framework	Stochastic Localization	Diffusion Transformer (DiT)	
4	draft	propose framework	Fisher-Flow + Riemannian Flow Matching	Geometric Algebra Transformer	
5	draft	propose framework	Discrete-token AR (VQ-VAE tokens)	Decoder-only causal Transformer	3.65
6	draft	propose framework	Diffusion-amortized EBM	Clebsch-Gordan Transformer	2.99
7	draft	propose framework	Walk-Jump Sampling (NEB)	SE(3) Vector Neuron MLP	8.80
8	draft	propose framework	Riemannian Flow Matching	Invariant Point Attention (IPA)	12.55
9	draft	propose framework	PFGM++	SE(3) tensor-product net	1.39†
10	draft	propose framework	Multimarginal stochastic interpolant	Mamba/SSM backbone	18.95
11	draft	propose framework	Coupling normalizing flow	Fourier Neural Operator backbone	20.60
12	draft	propose framework	Latent-EBM (cooperative)	3D periodic voxel U-Net	8.64
13	draft	propose framework	Bayesian Flow Network	Perceiver-IO set transformer	19.75
14	draft	propose framework	Latent AR set generation	Set Transformer, pointer decoding	23.57
15	draft	propose framework	Schrödinger Bridge matching	Equivariant graph neural ODE	21.88
16	draft	propose framework	Consistency model	Diffusion Transformer (DiT)	36.48
17	draft	propose framework	GFlowNet (trajectory-balance)	AR Transformer	2.83†
18	draft	propose framework	**Masked discrete diffusion**	**Bidirectional Transformer**	19.86†
19	improve	retry after debug	Mirror Flow Matching	MACE	
20	improve	retry after debug	Stochastic Localization	Diffusion Transformer (DiT)	
21	improve	retry after debug	Fisher-Flow + Riemannian Flow Matching	Geometric Algebra Transformer	
22	improve	scale hidden dimension	Walk-Jump Sampling (NEB)	SE(3) Vector Neuron MLP	8.43
23	improve	scale depth, add cosine schedule	Riemannian Flow Matching	Invariant Point Attention (IPA)	27.28
24	debug	fix residual wrapping	PFGM++	SE(3) tensor-product net	
25	improve	scale hidden dimension	Multimarginal stochastic interpolant	Mamba/SSM backbone	
26	improve	scale hidden width	Coupling normalizing flow	Fourier Neural Operator backbone	4.57
27	improve	fix learning rate	Coupling normalizing flow	Fourier Neural Operator backbone	12.43
28	improve	scale model depth	Bayesian Flow Network	Perceiver-IO set transformer	41.53
29	improve	add temperature annealing	Latent AR set generation	Set Transformer, pointer decoding	17.10
30	improve	sort by Z-order	Latent AR set generation	Set Transformer, pointer decoding	20.41
31	improve	fix pointer ordering	Latent AR set generation	Set Transformer, pointer decoding	9.67
32	improve	scale hidden width	Schrödinger Bridge matching	Equivariant graph neural ODE	32.30
33	improve	predict velocity	Schrödinger Bridge matching	Equivariant graph neural ODE	8.73
34	improve	add self-consistency	Consistency model	Diffusion Transformer (DiT)	36.95
35	improve	scale model depth	Consistency model	Diffusion Transformer (DiT)	45.03
36	debug	bound reward function	GFlowNet (trajectory-balance)	AR Transformer	
37	debug	fix broadcast bug	**Masked discrete diffusion**	**Bidirectional Transformer**	
38	improve	raise lattice loss weight	Riemannian Flow Matching	Invariant Point Attention (IPA)	12.83
39	improve	scale batch size	Riemannian Flow Matching	Invariant Point Attention (IPA)	12.86
40	improve	scale model width	Riemannian Flow Matching	Invariant Point Attention (IPA)	13.33
41	debug	fix scale instability	Multimarginal stochastic interpolant	Mamba/SSM backbone	
42	improve	add validity reranker	Bayesian Flow Network	Perceiver-IO set transformer	42.24
43	improve	raise sampling steps	Bayesian Flow Network	Perceiver-IO set transformer	41.15
44	improve	add self-consistency rerank	Bayesian Flow Network	Perceiver-IO set transformer	27.80
45	improve	scale batch size, raise learning rate	Bayesian Flow Network	Perceiver-IO set transformer	37.13
46	improve	scale hidden width	Bayesian Flow Network	Perceiver-IO set transformer	32.76
47	improve	switch to an SDE sampler	Schrödinger Bridge matching	Equivariant graph neural ODE	34.70
48	improve	scale model, add EMA	Consistency model	Diffusion Transformer (DiT)	34.36
49	improve	raise sampling steps	Consistency model	Diffusion Transformer (DiT)	38.90
50	improve	scale depth and width	Consistency model	Diffusion Transformer (DiT)	37.28
51	improve	scale capacity, add Gumbel sampler	**Masked discrete diffusion**	**Bidirectional Transformer**	57.78
52	improve	upweight coordinate loss	Multimarginal stochastic interpolant	Mamba/SSM backbone	25.71
53	improve	scale model depth	Multimarginal stochastic interpolant	Mamba/SSM backbone	23.15
54	improve	scale model depth	Bayesian Flow Network	Perceiver-IO set transformer	36.42
55	improve	raise candidates per composition	Bayesian Flow Network	Perceiver-IO set transformer	41.72
56	improve	raise coordinate noise	Schrödinger Bridge matching	Equivariant graph neural ODE	28.00
57	improve	scale model, raise noise	Schrödinger Bridge matching	Equivariant graph neural ODE	29.89
58	improve	add best-of-K decoding	**Masked discrete diffusion**	**Bidirectional Transformer**	60.02
59	improve	scale model, upweight coordinates	**Masked discrete diffusion**	**Bidirectional Transformer**	61.96
60	improve	add nucleus filtering	**Masked discrete diffusion**	**Bidirectional Transformer**	60.01
61	improve	port decoding to the larger model	**Masked discrete diffusion**	**Bidirectional Transformer**	63.91
62	improve	scale model depth	**Masked discrete diffusion**	**Bidirectional Transformer**	63.45
63	improve	scale batch size	**Masked discrete diffusion**	**Bidirectional Transformer**	61.35
64	improve	raise decoding steps	**Masked discrete diffusion**	**Bidirectional Transformer**	65.26
65	improve	lower sampling temperature	**Masked discrete diffusion**	**Bidirectional Transformer**	63.05
66	improve	raise decoding steps	**Masked discrete diffusion**	**Bidirectional Transformer**	63.19
67	improve	scale hidden width	**Masked discrete diffusion**	**Bidirectional Transformer**	63.18
68	improve	retune candidates and steps	**Masked discrete diffusion**	**Bidirectional Transformer**	65.29
69	improve	scale model depth	**Masked discrete diffusion**	**Bidirectional Transformer**	63.53
70	improve	retune candidates and steps	**Masked discrete diffusion**	**Bidirectional Transformer**	64.49
71	improve	raise candidates per composition	**Masked discrete diffusion**	**Bidirectional Transformer**	65.38
72	improve	raise learning rate	**Masked discrete diffusion**	**Bidirectional Transformer**	65.10

B TRANSCRIPT EXCERPTS FROM THE AGENT-GUIDED SEARCH

We summarize five turning points from the search as a representative transcript. Teal boxes are autonomous **Agent** turns, labeled with the search operator that produced them (*idea*, *draft*, *improve*); orange boxes are the **Human** interventions; gray boxes are the **Orchestrator**, the sole interface for human steering, which routes each intervention into the operator it dispatches. Nodes are shown in the order they occurred along the winning lineage. Interventions take two forms: an *objective* intervention introduces a new target the agent was not yet pursuing and leaves the method open, whereas a *mechanism* intervention supplies a causal lever toward the existing objective and lets the agent operationalize it.

(a) Agent-originated framework choice. (Node 18)



Agent idea

Every strong published CSP model evolves the lattice and coordinates through a continuous process, diffusion, flow matching, or a stochastic interpolant, each requiring iterative integration at sampling time. Surveying generative frameworks outside materials science for an alternative: *generative framework choice*: masked generative modeling corrupts a sequence by masking rather than by continuous noise, decodes in a handful of parallel unmasking steps instead of hundreds of denoising steps, and exposes a categorical posterior at every position, none of which the continuous baselines offer. *Architecture choice*: a bidirectional transformer, since predicting a masked token needs attention over every visible token at once, and a crystal's variable-length, set-like atom sequence fits a fixed grid better than an autoregressive alternative. Novelty: this combination is absent from the published-CSP catalogue. Feasibility hinges on one open question: whether a crystal's periodicity survives quantization into bins.



Agent draft

Instantiating the cell in code. I discretize each fractional coordinate axis into K bins (a resolution within the structure-matching tolerance) to form coordinate tokens, and Niggli-parameterize the unit cell into six scalars $(a, b, c, \alpha, \beta, \gamma)$, each binned over data-derived ranges, as lattice tokens; six scalars are a cleaner target than the nine-entry lattice matrix. The composition is given, so I embed the atomic numbers as fixed per-site side information rather than generating them, leaving six lattice tokens plus $3N$ coordinate tokens to predict. I keep equivariance implicit in the representation: fractional coordinates are rotation-invariant and the lattice tokens carry only invariant Niggli scalars, so a plain bidirectional transformer over the tokens suffices without vector features. I train it with absorbing-state masking and a cross-entropy loss on the masked positions, sample by iterative parallel unmasking, and train against the validation METRe.

(b) Mechanism intervention: represent crystal symmetry explicitly. (Node 176)



Human

The lattice and coordinate tokens leave crystal symmetry implicit, and several failed candidates miss on symmetry rather than on individual coordinates. Represent the space group and each site's Wyckoff position explicitly, rather than leaving the model to infer them from N correlated coordinates.



Orchestrator

I dispatch the *improve* operator on the current best node, folding this mechanism into its prompt.

Agent improve

Nothing in the sequence currently says what space group it is generating, so the model has to reconstruct global symmetry from $3N$ noisy coordinate predictions, a much harder credit-assignment problem than one categorical label. Crystallography already gives that label: 230 space groups, and a finite set of Wyckoff positions per site within each one. I add a space group token and one Wyckoff token per site to the sequence, fuse them into the same per-site projection as the other streams, and supervise both with the same masked cross-entropy as the rest of the sequence, no new loss term and no change to the training recipe.

(c) Mechanism intervention: restrict augmentation to symmetry orbits. (Node 274)

Human

Now that Wyckoff labels exist, the free atom-order permutation used for augmentation can swap sites across orbits, which does not correspond to any real symmetry operation. Restrict the permutation to symmetry orbits, not the full atom ordering.

Orchestrator

I dispatch the *improve* operator on the current best node, folding this mechanism into its prompt.

Agent improve

A cross-orbit swap moves two sites with different Wyckoff labels into each other's sequence position without any crystal symmetry justifying it, so it trains the model on relabelings that contradict the labels just added. The permutations that are actually free are the order in which orbits appear and the order of atoms within one orbit, since every atom in an orbit shares its Wyckoff letter. I group sites into symmetry orbits under the canonical ordering and shuffle only at those two levels during training, never across an orbit boundary.

(d) Objective intervention: recover sub-bin coordinate precision. (Node 340)

Human

Quantizing coordinates into bins caps geometric precision at half a bin width, and the matched structures are landing right at that ceiling. Recover the lost precision without leaving the discrete token model.

Orchestrator

I dispatch the *improve* operator on the current best node, folding this objective into its prompt.

Agent improve

The categorical token only says which bin a coordinate falls in, so the best the current model can do is place it at the bin center, half a bin width from the true value on average. I add a small regression head that reads the trunk output at each coordinate and lattice position and predicts a bounded sub-bin offset, supervised against the signed residual about the bin center and kept out of the masked cross-entropy objective so it does not interfere with the categorical prediction it rides on. At decode time, I place each value at its bin center shifted by the predicted offset; precision is no longer floored at the bin width.

(e) Mechanism intervention: cover multiple polymorphs. (Node 391)

Human

i.i.d. sampling draws independently, so repeated draws for the same composition collapse onto its single most likely structure and its other polymorphs go uncovered. Make the sampler spread its draws across those polymorphs.

Orchestrator

I dispatch the *improve* operator in the sampling-optimization stage, folding this mechanism into its prompt.

Agent improve

Independent draws all start from the same posterior, so they keep landing on the same dominant mode instead of spreading across a composition's distinct polymorphs. Since the space group is now an explicit token, I can read its posterior directly from one forward pass on the fully masked sequence, conditioned only on the composition. I pin the most confident draw to its argmax space group and assign the remaining draws to distinct high-posterior space groups without replacement, so diversity comes from space group conditioning rather than sampling noise.

C Symmetry Consistency

Table 5: **Symmetry consistency of generated structures.** How often spglib (Togo et al., 2024) confirms the symmetry MaskGXT generated, on the reported model's greedy samples at a symmetry tolerance of 0.1 (9,001 structures on MP-20, 8,987 on the polymorph split). Which Wyckoff letter a site carries depends on where the origin of the cell sits, so the Wyckoff rows accept the generated labels when they match what spglib writes under any origin the space group's normalizer translations allow, moved once for the whole structure.

Confirmed by spglib (%)	MP-20	Polymorph split
Space group	86.4	83.9
Wyckoff labels	83.7	81.1
Wyckoff labels, among structures whose space group is confirmed	96.9	96.6

MaskGXT decodes the space group token, the Wyckoff tokens and the coordinates as separate streams, so nothing forces a generated structure to carry the symmetry its tokens claim. Table 5 shows that the space group is the single point of failure: once spglib confirms it, that structure's Wyckoff labels are almost always confirmed too. Symmetrizing each generated structure to its detected space group leaves the match rate unchanged, so we report the structures without post-processing.

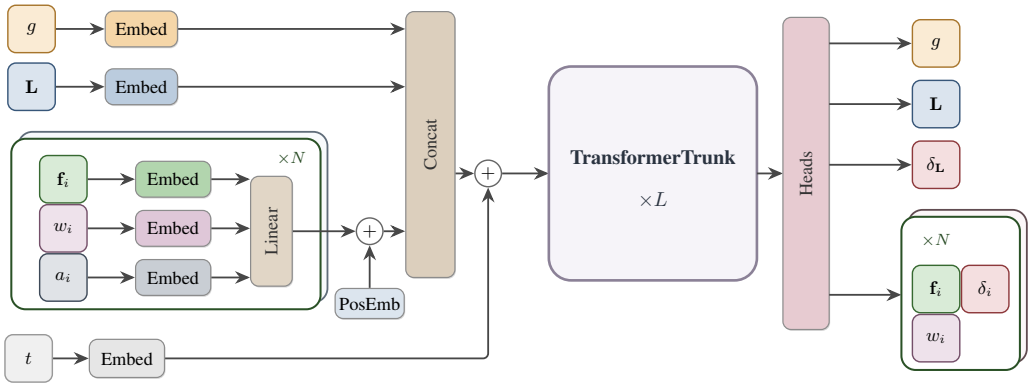


Figure 5: **MaskGXT architecture.** The per-site projection, the shared bidirectional trunk, and the per-stream heads of Equation (5), with the sub-bin offset head of Equation (4).

D MaskGXT Architecture Details

Forward pass. After embedding the $1{+}6{+}5N$ tokens, a per-site linear projection $\mathbf{W}_{\text{site}}$ fuses the five token embeddings of each site into one atom-wise feature, yielding a length-$1{+}6{+}N$ sequence of one space group, six lattice, and N atom-wise features. Each atom-wise feature adds a learned positional encoding over the canonical site order, and the whole sequence adds a mask-ratio embedding, before L trunk layers; linear heads then read from the shared trunk (Figure 5). The forward pass is

$$
\begin{aligned}
&\mathbf{h}_g = \mathbf{e}_g(g), \qquad \mathbf{h}_j^{\text{lat}} = \mathbf{e}(\ell_j) + \mathbf{a}_j^{\text{lat}}, \\
&\mathbf{h}_i^{\text{site}} = \mathbf{W}_{\text{site}}\big[\,\mathbf{e}(q(x_i)){+}\mathbf{a}_x;\ \mathbf{e}(q(y_i)){+}\mathbf{a}_y;\ \mathbf{e}(q(z_i)){+}\mathbf{a}_z;\ \mathbf{e}_Z(a_i);\ \mathbf{e}_w(w_i)\,\big] + \mathbf{p}_i, \\
&\mathbf{H}^{(0)} = \big[\,\mathbf{h}_g;\ \mathbf{h}_{1:6}^{\text{lat}};\ \mathbf{h}_{1:N}^{\text{site}}\,\big] + \mathbf{1}\,\boldsymbol{\tau}_t^{\top}, \qquad \boldsymbol{\tau}_t = \mathbf{W}_t\ \mathrm{SiLU}(\mathrm{SinEmb}(t))\,, \\
&\mathbf{H}^{(L)} = \mathrm{TransformerTrunk}\Big(\mathbf{H}^{(0)}\Big)\,, \qquad f_\theta(\mathcal{C}) = \big(\mathrm{Head}_m(\mathbf{H}^{(L)})\big)_{m\in\{\text{sg, lat, lat-off, coord, coord-off, wyk}\}},
\end{aligned}
\tag{5}
$$

where $\mathcal{C}$ is the token sequence, ℓ_j the j-th lattice bin token, $\mathbf{e}$ the shared bin embedding, $\mathbf{a}_j^{\text{lat}}$ and $\mathbf{a}_{\{x,y,z\}}$ per-axis encodings, and $\mathbf{e}_g, \mathbf{e}_Z, \mathbf{e}_w$ the space group, atom-type, and Wyckoff embeddings. The positional encoding factors as $\mathbf{p}_i = \mathbf{p}_{k_i}^{\text{elem}} + \mathbf{p}_{r_i}^{\text{within}}$, a learned element-block encoding at the rank k_i of atom i's element in the canonical order plus a within-element encoding at its rank r_i within that element. Each Head_m produces categorical logits for the four token streams and bounded $\frac{1}{2}\tanh$ sub-bin offsets for the lattice and coordinate positions (Section 4.2).

Transformer trunk. The TransformerTrunk of Equation (5) is a Pre-LN Transformer (Xiong et al., 2020) with QK-normalized self-attention (Dehghani et al., 2023). It consists of L blocks, each applying multi-head self-attention followed by a SwiGLU feedforward network (Shazeer, 2020). For a block input sequence $\mathbf{x}$, let $\tilde{\mathbf{x}} = \mathrm{LN}(\mathbf{x})$ and $\mathbf{q}, \mathbf{k}, \mathbf{v} = \mathrm{proj}(\tilde{\mathbf{x}})$ denote the per-head query, key, and value projections, and $\mathbf{W}_O$ the attention output projection. Each block updates

$$
\mathbf{x} \leftarrow \mathbf{x} + \mathbf{W}_O\Big(\mathrm{softmax}\Big(s_h\,\hat{\mathbf{q}}\hat{\mathbf{k}}^{\top}\Big)\,\mathbf{v}\Big)\,, \qquad \hat{\mathbf{q}} = \frac{\mathbf{q}}{\|\mathbf{q}\|},\ \ \hat{\mathbf{k}} = \frac{\mathbf{k}}{\|\mathbf{k}\|}, \tag{6}
$$

$$
\mathbf{x} \leftarrow \mathbf{x} + \mathbf{W}_2(\mathrm{SiLU}(\mathbf{W}_1\ \mathrm{LN}(\mathbf{x})) \odot \mathbf{W}_3\ \mathrm{LN}(\mathbf{x}))\,. \tag{7}
$$

QK-normalization normalizes queries and keys along the head dimension, bounding their dot products, while a learnable per-head temperature s_h replaces the standard $1/\sqrt{d_h}$ scaling. The SwiGLU hidden width is $\frac{2}{3}\cdot 4d$, rounded to a multiple of 64.

E ALGORITHMS

Algorithm 1 MaskGXT training

Input: dataset $\mathcal{D}$, network f_θ, bins K, stream weights $\{w_m\}$, smoothing σ.
Precompute the circular ordinal soft-target matrix $p^\star$ (Equation (2)).
while not converged **do**
  Sample a crystal $(\mathbf{L}, \mathbf{F}, \mathbf{A}, g, \mathbf{W}_g) \sim \mathcal{D}$.
  Draw an origin shift $\mathbf{r} \sim \mathcal{U}[0,1)^3$, translate all coordinates by $\mathbf{r}$ (mod 1), and relabel Wyckoff letters by the normalizer coset nearest to $\mathbf{r}$.
  Order the sites canonically by symmetry orbit (electronegativity, then Wyckoff letter), and apply the intra/inter-orbit permutation augmentation.
  Quantize $\mathbf{L}, \mathbf{F}$ into K bins (Equation (1)) and assemble the token streams $\mathcal{C}$.
  Draw $t \sim \mathcal{U}(0,1)$ and mask each token independently with probability t, giving $\mathcal{C}_{\setminus\mathcal{M}}$.
  Predict token logits $f_\theta(\mathcal{C}_{\setminus\mathcal{M}})$ at the masked positions $\mathcal{M}$.
  Compute the masked cross-entropy $\mathcal{L}(\theta)$ against the soft targets (Equation (3)).
  Update θ with AdamW.
end while

Algorithm 2 MaskGXT sampling

Input: composition $\mathbf{A}$, steps T, network f_θ; optional fixed space group $g^\dagger$.
Initialize all lattice, coordinate, space group, and Wyckoff tokens to MASK; if a space group $g^\dagger$ is given, fix its token to $g^\dagger$ throughout decoding.
for $\tau = 1, \ldots, T$ **do**
  Predict the posterior $\mathrm{softmax}(f_\theta)$ for all tokens in parallel.
  At each masked position, draw a token $\hat{c} \sim \mathrm{softmax}(f_\theta)$.
  ▷ greedy decoding: $\hat{c} = \arg\max \mathrm{softmax}(f_\theta)$
  Score each position by the posterior probability (confidence) of its drawn token $\hat{c}$.
  Fix the drawn tokens at the highest-confidence positions and re-mask the rest.
end for
Dequantize lattice and coordinate refined by sub-bin offset prediction and **return** $(\mathbf{L}, \mathbf{F})$.

Algorithm 3 Polymorph-aware sampling

Input: composition $\mathbf{A}$, stratification count S, steps T, network f_θ.
Read the space group posterior $p(g) = \mathrm{softmax}\big(f_\theta(\mathcal{C}_{\mathrm{mask}})_{\mathrm{sg}}\big)$ from one forward pass on the fully masked sequence $\mathcal{C}_{\mathrm{mask}}$.
Take the S highest-posterior space groups $\mathcal{G} = \text{top-}\,S_g\, p(g)$.
For each $g \in \mathcal{G}$, decode $(\mathbf{L}_g, \mathbf{F}_g)$ by MaskGXT sampling (Algorithm 2) with the space group fixed to g.
return $\{(\mathbf{L}_g, \mathbf{F}_g)\}_{g \in \mathcal{G}}$.

Table 6: **Seed spread of the MaskGXT rows of Table 2.** Mean over three training seeds with the sample standard deviation in parentheses. Match rate and RMSE are given as unfiltered/filtered, as in the main table. Baseline rows are single published values and are omitted.

Model	Dataset	Metric	Value
MaskGXT	MP-20	MR (%) ↑	73.97 (0.19) / 67.36 (0.17)
		RMSE ↓	0.0342 (0.0011) / 0.0331 (0.0012)
	MPTS-52	MR (%) ↑	37.21 (0.47) / 33.97 (0.41)
		RMSE ↓	0.0974 (0.0013) / 0.0930 (0.0011)
	polymorph split	METRe (%) ↑	79.75 (0.21)
		cRMSE ↓	0.1282 (0.0006)

F Datasets and Evaluation Protocol

Datasets. MP-20 contains materials from the Materials Project (Jain et al., 2013) with at most 20 atoms per unit cell and energy above the convex hull below 0.08 eV/atom, under the standard 60/20/20 train/validation/test partition. MPTS-52 (Jiao et al., 2023) extends this to structures with up to 52 atoms per cell under a chronological split by publication year, with a 27,380/5,000/8,096 partition. We derive the space group and Wyckoff label of every training structure with spglib (Togo et al., 2024) at a tolerance of 0.01. We count the following ourselves. Under the standard random split, 21% of MP-20 test references share a reduced composition with a training structure, so a model can be trained on one polymorph and scored on a different one of the same composition; under the polymorph split of Martirossyan et al. (2025) this overlap is zero, and 28% of test references belong to a composition with more than one reference, against 10% in the standard split.

Evaluation metrics. We compare the generated and reference structures directly with `StructureMatcher` (Ong et al., 2013) (tolerances ltol $= 0.3$, stol $= 0.5$, angle_tol $= 10°$), following the CSP protocol of Jiao et al. (2023); Miller et al. (2024). The *validity-filtered* form of match rate and RMSE counts a generated structure as unmatched when it fails standard chemical or structural validity checks. METRe (Martirossyan et al., 2025) pools generated structures by composition and counts a reference as matched if any generated structure of that composition matches it; cRMSE assigns each matched reference its best matched RMSE and each unmatched reference the tolerance cap stol.

G HYPERPARAMETERS

Table 7: MaskGXT configuration.

Component	Value
Coordinate / lattice bins K	64 (32 for MPTS-52)
Transformer layers	34
Hidden dimension	768
Attention heads	12 (QK-normalized)
Feed-forward	SwiGLU
Parameters	$\approx$ 248M
Ordinal label-smoothing σ	1.0 (coordinate and lattice)
Stream weights $(w_{\mathrm{lat}}, w_{\mathrm{coord}}, w_{\mathrm{sg}}, w_{\mathrm{wyk}})$	$(0.10,\ 1.5,\ 0.10,\ 0.30)$
Offset-regression loss weight (lattice, coordinate)	$(0.2,\ 0.2)$
Euclidean-normalizer augmentation probability	0.7
Permutation augmentation probability	0.5
Dropout	0.05
Optimizer	AdamW (β_1=0.9, β_2=0.95, weight decay 0.05)
Learning rate	4×10^{-4}, cosine decay to 5% floor
Batch size	512 (128 for MPTS-52)
Gradient clipping	1.0
Weight EMA decay	0.9999
Training epochs	up to 3000 with early stopping on validation METRe
Sampling steps T	150

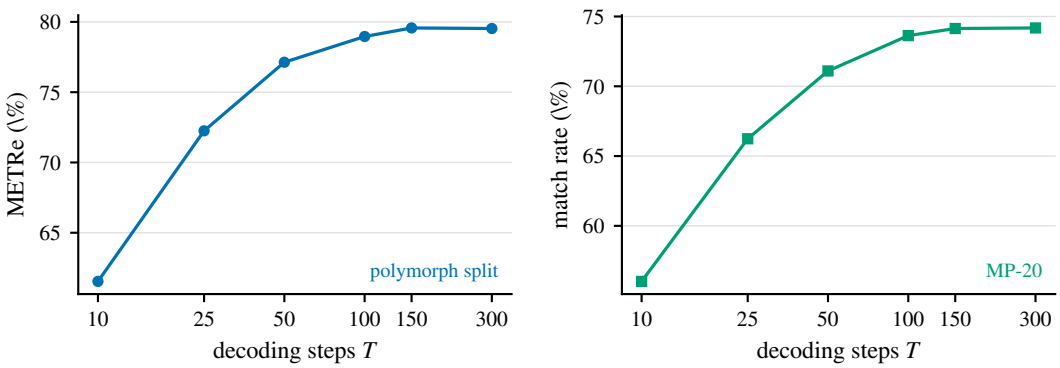


Figure 6: **Decoding steps.** METRe on the MP-20 polymorph split and match rate on MP-20, against the number of parallel decoding steps T, under greedy space group stratified decoding on one trained checkpoint per split. Both curves rise steeply up to $T{=}100$ and then saturate; every reported number uses $T{=}150$.

H Computational Cost

Training and sampling cost. MaskGXT is cheaper than Crystalite both to train and to sample from, at essentially the same parameter count. Both models select their checkpoint on a validation metric, and MaskGXT reaches its own in roughly five times less compute on the two MP-20 splits and eight times less on MPTS-52 (Table 8). At sampling time the saving comes from the number of network evaluations: each parallel decoding step costs one, against two for a Heun corrector step, and MaskGXT converges in fewer steps, leaving it about six times faster per structure (Table 9). Its coverage saturates well below the step count we report (Figure 6), so this margin does not come from undersampling.

Table 8: **Model size and training compute.** Compute is counted to the checkpoint each model evaluates, which both select on a validation metric rather than taking the last step: $6\,N\,T\,E$ over N parameters, sequence length T and examples presented E (epochs times the training set). GPU-hours are single-RTX-3090 wall-clock, given only for the models we trained ourselves (MaskGXT, and Crystalite on the polymorph split). Our rows are three-seed means. The table covers training the reported models; the search that produced MaskGXT ran 523 trials under the per-candidate budgets of Figure 1, at most 2,100 GPU-hours in total.

Dataset	Model	Params	Epochs	PFLOPs	GPU-h
MP-20	Crystalite	270M	6,604	6,098	–
	MaskGXT	248M	1,067	**1,162**	12.7
MP-20 polymorph split	Crystalite	270M	6,306	5,823	60.3
	MaskGXT	248M	1,033	**1,147**	**12.5**
MPTS-52	Crystalite	270M	10,752	25,282	–
	MaskGXT	248M	1,267	**3,044**	30.9

Table 9: **Sampling cost.** Each model at the setting it reports, timed on one NVIDIA H200 over the full MP-20 test set with only the sampler measured, and scored on the same set. NFE counts network evaluations, verified by instrumenting both samplers: $T{+}1$ for MaskGXT, which decodes every position in parallel once per step and then reads the sub-bin offsets of the committed tokens in a final pass at $t{=}0$; $2S{-}1$ for Crystalite, whose Heun correction evaluates the network twice per step bar the last. Timing depends on the architecture and the step count, not on the weights. Figure 6 shows how MaskGXT's coverage varies with T.

Model	Steps	NFE	ms / structure	MR (%)
Crystalite	400	799	73.59	72.10
MaskGXT	150	151	**12.37**	**73.97**

I Model Scale

Returns to scale. We trained the reported architecture at six sizes, from 10.4M to 317.4M parameters, to separate what the formulation contributes from what capacity contributes. On both splits the metric grows close to linearly in the log of the parameter count (Figure 7), and the slope is shallow: a model four times smaller than the reported one gives up about a point. The ladder crosses Crystalite, the strongest baseline, at about a tenth of its parameters on MP-20, and stays above it at every size on the polymorph split. We therefore read the margin in Table 2 as a property of the masked formulation rather than of model size, and the ladder continues above the reported model.

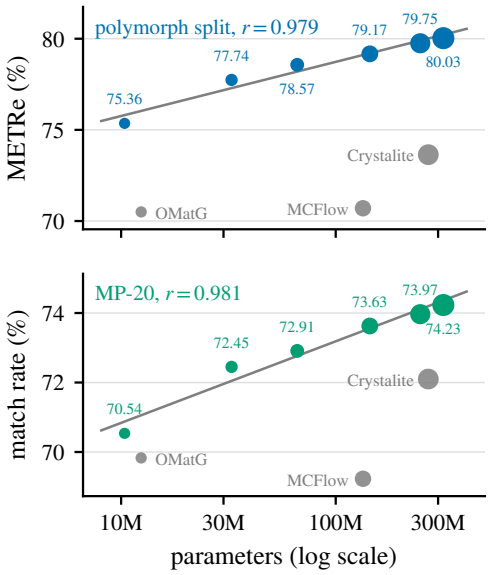


Figure 7: **Correlation between model size and match rate or METRe.** Each coloured point is one trunk size, trained on the split it is scored on, at the depth-to-width ratio of the reported model; the grey points are baselines at their own parameter counts. Marker area grows with the model and each coloured point carries its value. The solid line is a least-squares fit to the coloured points and r is the Pearson correlation.

J Limitations and Outlook

Scope and limitations. The success of agent-guided search in CSP can be understood through the structure of the development problem. CSP provides a rare combination of fixed datasets, executable model code, fast training and evaluation cycles, and a validation metric that is closely aligned with the final task. These properties turn algorithm design into a high-throughput empirical feedback problem: the agent can propose a modeling principle, instantiate it in code, train it under a controlled budget, and receive a reliable signal on whether the change improves the algorithm. This signal is budget-dependent: our fixed compute budget penalizes methods that need more compute to converge. More broadly, the loop is harder to close in domains with physical experiments, long training cycles, or weak proxy objectives, which may instead require surrogate models, stronger human steering, or more structured evaluation.

Future directions. The same agent and tokenization extend naturally to related crystal generation tasks, including de novo generation and structure-conditioned atom-type generation within a single model. More broadly, this development record highlights several requirements for making agent-guided search more generally useful.

First, future systems must address the human attention bottleneck in agent-guided search. A single search run can produce many hypotheses, code variants, logs, plots, failed attempts, and partial analyses, most of which are too raw for a human scientist to inspect. Rather than exposing this stream directly, an agent should maintain a compact research state: which ideas have been tried, why they succeeded or failed, which mechanisms remain unexplored, and which decisions require human judgment. This is also a goal-specification problem, because the human should be able to revise the scientific objective at a conceptual level while the agent translates that revision into coordinated parallel research threads with preserved provenance.

Second, agent search needs more principled proxy evaluation. Rather than selecting candidates only by short-budget runs, future systems should develop validation signals whose rankings extrapolate to larger budgets, for example through scaling-law analyses over model size, training time, data size, and sampling budget.

Third, the search loop should expand beyond metric-optimizing experiments. Model development also requires exploratory and analytic experiments: mapping failure modes, inspecting generated

samples, testing mechanistic hypotheses, designing ablations, and deciding which empirical signals are trustworthy. Equipping agents with these capabilities would move them from automated implementation search toward a fuller research workflow for carrying ideas across chemistry, biology, materials science, and other domains.