RMD: RELAXATION-INSPIRED MASKED DIFFUSION LANGUAGE MODEL FOR CRYSTAL GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models offer a promising approach to crystal generation through unified sequence representations and flexible conditional modeling. However, standard autoregressive generation follows a unidirectional, left-to-right order, restricting predictions to the generated prefix and typically preventing earlier outputs from being revised using later structural information. To address these limitations, we present the first application of a discrete masked diffusion language model to crystal structural text generation, enabling bidirectional prediction and iterative recovery. To further address the challenge of identifying token positions that require continued adjustment, we propose *Relaxation-inspired Masked Diffusion (RMD)*. Inspired by the assess–update–reassess organization of physical relaxation, RMD combines relaxation-score-driven position selection with selective state updates, allowing existing content to be revised as the visible context evolves. State-of-the-art results on the evaluated de novo generation and composition-conditioned crystal structure prediction benchmarks, support RMD's potential to produce high-quality candidates for stable crystal discovery. Our code and datasets are available at `https://anonymous.4open.science/r/RMD`.

1 INTRODUCTION

Discovering crystalline materials with desirable structures and properties is of fundamental importance to a wide range of scientific domains, including energy storage Zhao et al. (2020), carbon capture Osman et al. (2021), and catalysis Zhao et al. (2019). Large language models provide a flexible framework for sequence modeling and conditional generation. Inspired by this modeling paradigm, crystal language models such as CrystaLLM and Mat2Seq represent structures as discrete sequences and learn dependencies among structural fields, enabling candidate generation and conditioning through specific prefixes Antunes et al. (2024); Yan et al. (2024). These representations organize composition, lattice parameters, atomic sites, and symmetry information within a common sequence. They connect crystal data with general sequence modeling techniques and provide an interface for incorporating material properties and domain knowledge. Together, structural serialization and prefix conditioning provide a unified way to encode structures and specify generation requirements, offering a flexible basis for extending conditional crystal generation.

Despite this progress, standard autoregressive (AR) models use unidirectional, left-to-right generation and face two limitations during standard sampling (Figure 1, middle). First, each prediction depends only on the generated prefix and cannot directly use structural information that has yet to be generated. Second, emitted tokens are typically not revised using subsequently formed context. These limitations matter because composition, lattice parameters, atomic positions, and symmetry jointly constrain atomic environments under periodic boundary conditions. Their dependencies do not naturally follow a textual order, so early decisions may become inconsistent with later content. This motivates crystal sequence models that can use currently visible structural information on both sides of a position and revise earlier outputs as the context develops.

Discrete masked diffusion Austin et al. (2021) provides a foundation for these capabilities through bidirectional prediction and iterative denoising. However, a key challenge in applying it to crystal generation is determining which token positions require further adjustment. Position selection must distinguish content that can be retained from positions that should remain masked or be remasked

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.

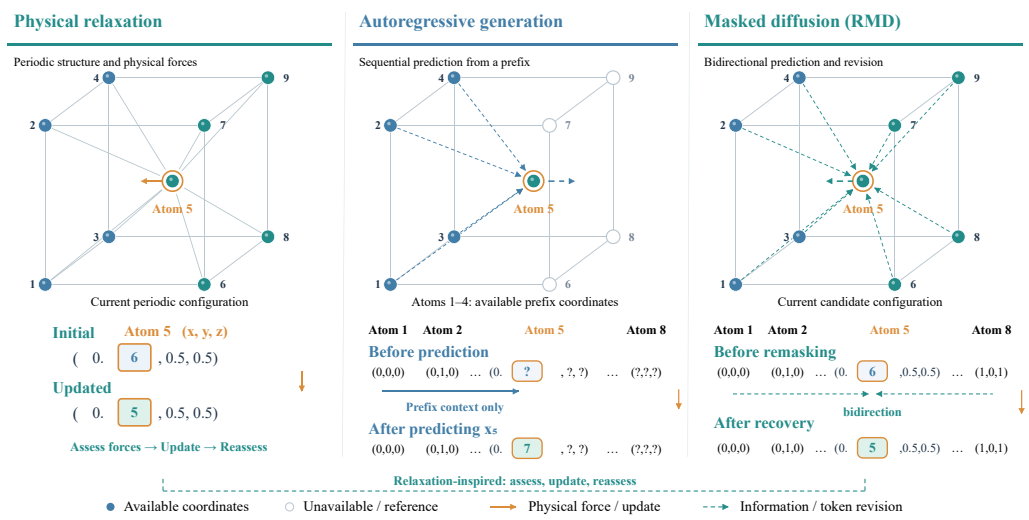

Figure 1: **Physical relaxation and crystal sequence generation.** From left to right, physical relaxation assesses forces and updates atomic positions, whereas standard autoregressive generation predicts from the available prefix and typically leaves previous outputs unchanged. RMD uses bidirectional visible context for selective iterative revision.

for recovery under new context. Coordinating this selection with state updates is therefore central to organizing selective revision of interdependent structural fields.

Physical structural relaxation assesses atomic forces and, when needed, cell stresses to determine how structural degrees of freedom should be adjusted. It updates the configuration and then reassesses the resulting state (Figure 1, left). Inspired by this organization, we propose **Relaxation-inspired Masked Diffusion (RMD)**, a discrete masked diffusion language-modeling approach to crystal sequence generation. RMD organizes generation into two functional stages: **relaxation-score-driven position selection** and **selective state updates**. A bidirectional network uses the current visible context to provide both candidate tokens and the predictive information needed to compute relaxation scores. These scores identify positions requiring continued adjustment. Selective updates combine the selection decisions with candidate predictions and current token states to retain content, accept candidates, or maintain masking. Previously generated content can also be remasked and recovered in later iterations using updated visible context, as illustrated in Figure 1 (right). The resulting sequence then provides the context for rescoring and further revision.

In summary, this work makes the following contributions:

1. To the best of our knowledge, this is the first work to apply a discrete masked diffusion language model to the generation of crystal structural text containing composition, lattice parameters, and atomic sites.
2. We propose RMD, which organizes crystal generation as a cycle of assessing adjustment needs, updating the sequence state, and reassessing it using the updated context.
3. Experiments on composition-conditioned crystal structure prediction and de novo generation demonstrate state-of-the-art results on the evaluated benchmarks, with valid and diverse candidates and stable crystal structures.

2 Related Work and Preliminaries

2.1 Crystal Generative Models

Geometric and latent models generate crystals through complementary representations. CDVAE Xie et al. (2021) combines variational modeling with iterative atomic denoising, whereas DiffCSP Jiao et al. (2024a) jointly diffuses lattices and fractional coordinates, primarily for composition-conditioned

prediction. DiffCSP++ Jiao et al. (2024b) incorporates space-group constraints through lattice parameterization and Wyckoff positions. MatterGen Zeni et al. (2025) combines categorical species diffusion with continuous geometric diffusion and property-conditioned fine-tuning. FlowMM Miller et al. (2024) uses Riemannian flow matching, while ADiT Joshi et al. (2025) generates shared continuous latent representations of molecules and crystals. These methods model structural degrees of freedom directly or through learned embeddings and can incorporate periodicity and symmetry.

Sequence models offer an alternative interface for organizing structural fields. CrystaLLM Antunes et al. (2024) autoregressively models CIF text with composition and optional space-group prefixes, while Mat2Seq Yan et al. (2024) constructs invariant crystal sequences and supports composition and property conditioning. Their unified sequence representations connect structural data to language-modeling techniques. Standard autoregressive sampling, however, depends on the generated prefix and ordinarily does not revise emitted tokens. FlowLLM Sriram et al. (2024) bridges sequence and geometric modeling: language-model samples form a learned base distribution for flow matching, which refines coordinates and lattices while keeping species fixed. Its complete generation pipeline therefore extends beyond purely autoregressive sampling.

2.2 Crystal Sequence Representation

A crystal is represented as $\mathcal{C} = (\mathbf{A}, \mathbf{Z}, \mathbf{F})$, where $\mathbf{A} \in \mathbb{R}^{3\times3}$ is the lattice matrix, $\mathbf{Z}$ specifies atomic species, and $\mathbf{F}$ contains fractional coordinates. Preprocessing organizes composition, lattice, atomic-site, and symmetry information into structural text, yielding

$$\mathbf{x}_0 = \mathrm{Tokenize}\big(\mathrm{Preprocess}(\mathcal{C})\big) = (x_0^1, \ldots, x_0^N). \tag{1}$$

Here, N is sequence length; superscripts index positions and subscripts index time, with 0 denoting the clean sequence. In this work, the vocabulary $\mathcal{V}$ contains 371 base tokens and four additional special tokens, totaling 375, including [MASK] and [PAD]. Structural text need not be a complete standard CIF, so generated sequences require decoding and postprocessing.

Let $\mathcal{I} \subseteq \{1, \ldots, N\}$ denote updatable positions and $\mathcal{F} = \{1, \ldots, N\} \setminus \mathcal{I}$ fixed positions with contents $\mathbf{c}_\mathcal{F}$. Padding is excluded from corruption and recovery. Conditioning on sequence length, position sets, and fixed contents is implicit throughout. Numerical values and atomic records may span multiple tokens; selected token sets therefore need not correspond to contiguous atomic regions.

3 Methodology

Standard autoregressive generation uses only the generated prefix and typically leaves emitted tokens unchanged. Early decisions about coupled structural fields can therefore become inconsistent with later information. Masked diffusion supports bidirectional prediction and iterative recovery, but these capabilities alone do not specify selective refinement. We propose *Relaxation-inspired Masked Diffusion (RMD)* for crystal sequence generation, organizing refinement through position selection and selective state updates (Fig. 2). The following sections introduce the reference corruption and recovery processes, derive the training objective, and describe score-driven generation.

3.1 Forward and Reverse Processes

The forward process progressively hides structural information, constructing the sequence states that the model must recover. Let T be the number of diffusion steps. For each updatable position $i \in \mathcal{I}$, define

$$q(x_t^i \mid x_{t-1}^i) = \beta_t \delta_{x_{t-1}^i}(x_t^i) + (1-\beta_t)\delta_{[\mathtt{MASK}]}(x_t^i), \qquad t = 1, \ldots, T, \tag{2}$$

where β_t is the one-step retention probability. The mask is an absorbing state of the forward process, whereas fixed positions satisfy $x_t^i = c^i$ throughout. Define the cumulative retention probability as $\alpha_t = \prod_{s=1}^{t} \beta_s$, with $\alpha_0 = 1$. We use the linear schedule $\alpha_t = 1 - t/T$, which produces an all-mask terminal state on the updatable positions. Positions are corrupted independently given the clean sequence. Appendix A.1 derives the marginal distribution and its relation to the one-step transition.

For the posterior conditioned on the clean sequence $\mathbf{x}_0$, define the training corruption indicator as $z_t^i = \mathbb{I}[x_t^i \neq x_0^i]$. Clean data contain no mask tokens at updatable positions, and the forward process

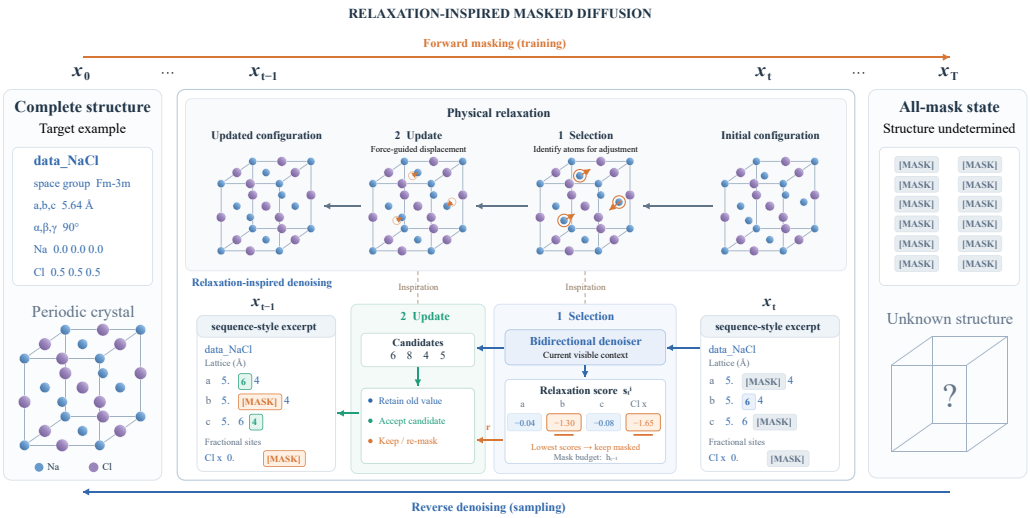

Figure 2: **Relaxation-inspired masked diffusion for crystal generation.** Forward masking transforms complete sequences into all-mask states from left to right, while reverse denoising recovers crystal sequences in the opposite direction. Inspired by adjustment assessment and position updates in physical relaxation, denoising combines relaxation-score-driven position selection with selective token updates.

only retains or masks the original token. Consequently, on forward samples, $z_t^i = \mathbb{I}[x_t^i = [\texttt{MASK}]]$. Define the posterior recovery probability as $\eta_t = (\alpha_{t-1} - \alpha_t)/(1 - \alpha_t)$ and the probability of remaining masked as $\rho_t = (1 - \alpha_{t-1})/(1 - \alpha_t) = 1 - \eta_t$. The reference posterior is

$$q(x_{t-1}^i \mid x_t^i, x_0^i) = \begin{cases} \delta_{x_t^i}(x_{t-1}^i), & z_t^i = 0, \\ \eta_t \delta_{x_0^i}(x_{t-1}^i) + \rho_t \delta_{[\texttt{MASK}]}(x_{t-1}^i), & z_t^i = 1. \end{cases} \tag{3}$$

Equation 3 retains visible tokens and either recovers masked positions to their original tokens or leaves them masked. Under the linear schedule, $\eta_t = 1/t$ and $\rho_t = (t-1)/t$. This posterior depends on the clean sequence and does not remask visible positions.

Since $\mathbf{x}_0$ is unavailable during generation, a learnable prediction distribution is required to provide recovery content. Given the current sequence $\mathbf{x}$, a bidirectional network outputs $f_\theta^i(a \mid \mathbf{x}) = \mathrm{softmax}\big(\boldsymbol{\ell}_\theta^i(\mathbf{x})\big)_a$ for $a \in \mathcal{V}$, where $\boldsymbol{\ell}_\theta^i$ denotes the logits at position i. Predictions use the structural information currently visible in the sequence.

Physical structural relaxation evaluates atomic forces and, when needed, cell stress to determine how structural degrees of freedom should be adjusted before applying an update. Inspired by this evaluation–update pattern, RMD organizes sequence denoising into routing and conditional recovery. Routing determines whether a position remains subject to adjustment, while conditional recovery specifies how the selected branch retains the current state or uses predicted content. Selection governs candidate acceptance and state updates; it does not require candidate prediction to occur after routing.

To express this organization probabilistically, introduce a binary routing variable r_{t-1}^i controlling the transition $t \to t-1$. A value of 1 selects the branch that remains masked, whereas 0 selects retention or recovery. In this subsection, the reference route is restricted to the absorbing-mask process:

$$q(r_{t-1}^i \mid \mathbf{x}_t) = \begin{cases} \delta_0(r_{t-1}^i), & x_t^i \neq [\texttt{MASK}], \\ \mathrm{Bern}(r_{t-1}^i; \rho_t), & x_t^i = [\texttt{MASK}], \end{cases} \tag{4}$$

where $\mathrm{Bern}(\cdot; \rho_t)$ is a Bernoulli distribution with success probability ρ_t. The sequence-level reference route factorizes over updatable positions. Its explicit dependence on the clean sequence can be removed because the corruption state is observable from mask membership on forward training samples.

We impose a fixed matching parameterization for the model route, $p_\theta(\mathbf{r}_{t-1} \mid \mathbf{x}_t) := q(\mathbf{r}_{t-1} \mid \mathbf{x}_t)$. This equality is a modeling constraint, not a consequence that the forward process imposes on every

reverse model. The model then replaces the clean token in the recovery branch with its prediction distribution:

$$p_\theta(x_{t-1}^i \mid r_{t-1}^i, \mathbf{x}_t) = \begin{cases} \delta_{x_t^i}(x_{t-1}^i), & x_t^i \neq [\texttt{MASK}], \\ \delta_{[\texttt{MASK}]}(x_{t-1}^i), & x_t^i = [\texttt{MASK}],\ r_{t-1}^i = 1, \\ f_\theta^i(x_{t-1}^i \mid \mathbf{x}_t), & x_t^i = [\texttt{MASK}],\ r_{t-1}^i = 0. \end{cases} \tag{5}$$

Conditional recovery factorizes over positions given the route and current sequence. Marginalizing the route yields

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \sum_{\mathbf{r}_{t-1}} p_\theta(\mathbf{r}_{t-1} \mid \mathbf{x}_t) p_\theta(\mathbf{x}_{t-1} \mid \mathbf{r}_{t-1}, \mathbf{x}_t). \tag{6}$$

This construction incorporates content prediction into the reverse probability model and supports the training decomposition below. Appendix A.2 establishes the connection between the simplified binary route and the full auxiliary routing representation, together with the required conditions. Generation subsequently uses prediction scores for adaptive selective refinement, with a routing construction distinct from the reference distribution defined here.

3.2 Training Objective

Training recovers clean structural tokens from corrupted sequences. For the augmented model with an all-mask prior p_T, the negative evidence lower bound is

$$\mathcal{L}_{\mathrm{VB}} = \mathbb{E}_{\mathbf{x}_0 \sim p_{\mathrm{data}}} \left[\mathcal{L}_{\mathrm{terminal}} + \mathcal{L}_{\mathrm{reconstruction}} + \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t \mid \mathbf{x}_0)} [\mathcal{J}_t] \right]. \tag{7}$$

The terminal term matches the forward terminal distribution to the generation prior, the reconstruction term handles the final recovery step, and $\mathcal{J}_t$ measures the one-step joint transition discrepancy conditioned on $\mathbf{x}_t, \mathbf{x}_0$. Appendix A.3 gives the augmented path model and complete derivation.

By the chain rule for KL divergence, the one-step joint term decomposes as

$$\mathcal{J}_t = \mathcal{J}_{\mathrm{route},t} + \mathcal{J}_{\mathrm{recovery},t}. \tag{8}$$

Here, $\mathcal{J}_{\mathrm{route},t}$ compares routing distributions, while $\mathcal{J}_{\mathrm{recovery},t}$ compares content recovery distributions given a route, averaged over the reference route. Their full expressions appear in equation A.21.

Under the fixed matching parameterization, the decomposition introduces no additional learnable routing module, separate loss evaluation, or training forward pass. Optimization reduces to the masked-position denoising loss

$$\mathcal{J}_t = \eta_t \sum_{i \in \mathcal{I}} z_t^i \big[-\log f_\theta^i(x_0^i \mid \mathbf{x}_t) \big]. \tag{9}$$

This objective trains the model to recover hidden structural content from the current context. Under the linear masking schedule, the theoretical coefficient is $\eta_t = 1/t$; the final step has $\eta_1 = 1$, and its reconstruction term takes the same form. Appendix A.4 gives the matching conditions, branchwise cancellations, and cross-entropy reduction. Appendix A.5 distinguishes the practical time reweighting and batch normalization from the theoretical objective.

3.3 Generation

RMD implements the relaxation-inspired evaluation–update cycle through position selection and selective state updates (Fig. 2). Each iteration evaluates positions using the current context, then retains, recovers, or remasks content to progressively complete the sequence.

Position selection using relaxation scores. Relaxation scores determine which positions require further adjustment, while the remaining-mask schedule controls their target number. Candidate prediction supplies the scoring information, so selection and updating are functional stages, not a requirement to run the network after selecting positions.

The network first produces logits from the bidirectional context of the current generation sequence $\mathbf{y}_t$, where $t = K, \ldots, 1$ indexes the K generation iterations and K need not equal the number T

of training diffusion steps. Let $\mathcal{V}_{\text{gen}}$ be the candidate vocabulary, excluding masks, padding, and other non-generatable special symbols. Following the candidate sampling rule, add independent standard Gumbel perturbations to the candidate logits: $\widetilde{\ell}_t^i(a) = \ell_\theta^i(a \mid \mathbf{y}_t) + g_t^i(a)$, where $g_t^i(a) \sim \text{Gumbel}(0, 1)$. The candidate token and relaxation score are

$$\widehat{y}_t^i = \arg\max_{a \in \mathcal{V}_{\text{gen}}} \widetilde{\ell}_t^i(a), \qquad s_t^i = \log \frac{\exp\big(\widetilde{\ell}_t^i(\widehat{y}_t^i)\big)}{\sum_{a \in \mathcal{V}_{\text{gen}}} \exp\big(\widetilde{\ell}_t^i(a)\big)}. \tag{10}$$

The relaxation score s_t^i expresses the preference for retention or further adjustment in the current iteration. Higher scores favor temporary retention or candidate acceptance, whereas lower scores prioritize positions for further adjustment. The score is computed from the perturbed candidate distribution; it need not score the previously retained token (see Appendix A.6.3). Here, relaxation refers to an internal selection and refinement procedure: the score is not an atomic force or energy gradient, and sequence updates are not assumed to decrease physical energy.

To progressively reduce the amount of content requiring adjustment, let $n = |\mathcal{I}| \geq 1$ and set the next-state target mask count to $h_{t-1} = \lfloor n(t-1)/K \rfloor$. Let $s_{t,(j)}$ denote the j-th smallest score among updatable positions. Define the set selected for further adjustment as $\mathcal{R}_{t-1} = \varnothing$ when $h_{t-1} = 0$, and $\mathcal{R}_{t-1} = \{i \in \mathcal{I} : s_t^i < s_{t,(h_{t-1}+1)}\}$ when $h_{t-1} > 0$. The current routing decision is $r_{t-1}^i = \mathbb{I}[i \in \mathcal{R}_{t-1}]$. A decision $r_{t-1}^i = 1$ selects masking in the transition $t \to t-1$, whereas $r_{t-1}^i = 0$ selects retention or recovery. With distinct scores, the set contains the h_{t-1} lowest-scoring positions; ties at the threshold can produce a smaller set. The route has the same decision semantics as in Section 3.1, but is determined by current prediction scores rather than sampled directly from the fixed reference distribution.

Selective state updates. The pre-update state m_t^i, with 1 indicating further adjustment and 0 temporary retention, and route r_{t-1}^i jointly determine how selection changes the sequence:

$$y_{t-1}^i = \begin{cases} c^i, & i \in \mathcal{F}, \\ [\texttt{MASK}], & i \in \mathcal{I},\ r_{t-1}^i = 1, \\ \widehat{y}_t^i, & i \in \mathcal{I},\ r_{t-1}^i = 0,\ m_t^i = 1, \\ y_t^i, & i \in \mathcal{I},\ r_{t-1}^i = 0,\ m_t^i = 0. \end{cases} \tag{11}$$

Selected positions remain masked or are remasked. Previously masked positions permitted to recover accept candidates, while retained visible positions preserve their old values even when new candidates have been computed.

Selecting a position for further adjustment therefore does not mean immediately filling it with a new token in the same iteration. In particular, a reactivated position is first masked and can recover in a later iteration using the updated context.

After applying the update, maintain the next state as $m_{t-1}^i := \mathbb{I}[y_{t-1}^i = [\texttt{MASK}]] = r_{t-1}^i$ for $i \in \mathcal{I}$. Here, m_t^i is the pre-update state, r_{t-1}^i is the current decision, and m_{t-1}^i is the resulting state. The final equality follows from the update rule and the exclusion of masks from accepted candidates, as proved in Appendix A.6. Temporary retention is an algorithmic status, not a determination that the token matches ground truth.

The updated sequence supplies context for rescoring and selection in the next iteration. The adjustment budget decreases with time, but selected positions can change, allowing earlier content to be reconsidered. Prediction training influences selection indirectly through the relaxation scores. Score-driven remasking is a generation strategy, rather than exact sampling from the absorbing-state reference posterior.

Generation performs exactly K iterations without additional stopping criteria based on physical convergence or EOS. At the final step, $h_0 = 0$, so all remaining masked positions accept candidates and retained positions preserve their values. The sequence is decoded into structural text $\widehat{S} = \text{Decode}_{\mathcal{T}}(\mathbf{y}_0)$ using tokenizer $\mathcal{T}$, then converted into a candidate CIF by postprocessing: $\widehat{\text{CIF}} = \mathcal{H}(\widehat{S})$. This conversion does not establish structural validity or physical stability. Appendix A.7 provides the complete generation algorithm and boundary conventions.

4 EXPERIMENTS

Table 1: Benchmarking crystal generation on MP20: validity and stability. The number following each method name indicates the size of its language model. Stable and metastable crystals are defined by DFT $E^{\text{hull}} < 0.0$ and $E^{\text{hull}} < 0.1$ eV/atom, respectively. Results marked with * are from MatterGen-MP using 1,024 samples; † denotes re-evaluation using the same DFT protocol as ADiT. Stability-related results are reproduced from ADiT Joshi et al. (2025) and Mat2Seq Yan et al. (2024). The best and second-best results are highlighted in **bold** and underlined, respectively.

Model	Validity (%) ↑		Metastability (%) ↑		Stability (%) ↑	
	Structure	Composition	M.S.	M.S.U.N.	S.	S.U.N.
CDVAE	**100.00**	86.70	–	–	1.6	1.4
FlowMM	96.85	83.19	$30.6^\dagger$	$22.5^\dagger$	$4.6^\dagger$	$2.8^\dagger$
Mat2Seq (200M)	88.6	81.7	46.6	–	4.5	3.2
CrystaLLM (200M)	95.54	93.07	–	–	–	–
LLaMA (70B)	99.60	**95.40**	–	–	–	–
DiffCSP	**100.00**	83.25	–	–	5.1	3.3
MatterGen	–	–	78^*	21^*	13^*	–
FlowLLM (70B)	99.94	90.84	$66.9^\dagger$	$26.3^\dagger$	$13.9^\dagger$	$4.7^\dagger$
ADiT	99.58	90.46	81.6	25.91	14.1	4.7
MD (150M)	96.38	91.02	63.25	22.47	10.2	3.9
RMD (150M)	99.96	92.17	**82.70**	**26.57**	**14.7**	**5.4**

4.1 DE NOVO CRYSTAL GENERATION

This experiment evaluates candidate validity and the yield of stable, unique, and novel structures after physical relaxation. The comparison with random position selection also assesses the effectiveness of RMD's score-guided refinement strategy.

4.1.1 EXPERIMENTAL SETTINGS

The evaluation uses MP-20 Jain et al. (2013) with the 60/20/20 training, validation, and test split established by CDVAE. RMD generates 10,000 samples, and the same collection serves both conventional quality evaluation and DFT-based assessment. The literature baselines include geometric and latent-space generators, crystal language models, and the hybrid generator FlowLLM (Table 1). Their results follow the sources listed in the table caption. An additional masked diffusion baseline (MD) randomly selects update positions at each iteration. MD and RMD use the same training configuration. RMD instead uses relaxation scores to select positions requiring further adjustment. This comparison evaluates score-guided selective refinement against random position selection in the configurations considered here.

Structural and compositional validity provide initial checks of basic plausibility. VASP structure relaxation and DFT energy evaluation against a reference convex hull complement these checks with a more direct physical assessment of stability. The reported metrics comprise stable (S.), metastable (M.S.), stable–unique–novel (S.U.N.), and metastable–unique–novel (M.S.U.N.) rates. Appendix B.1 provides their definitions and additional distributional evaluations.

4.1.2 EXPERIMENTAL RESULTS

RMD achieves near-leading structural validity and higher compositional validity than most reported baselines (Table 1). These results indicate that its generated candidates generally satisfy the basic geometric and chemical plausibility criteria, although neither validity measure is the highest reported.

After VASP relaxation and DFT assessment, RMD achieves the highest reported S., M.S., S.U.N., and M.S.U.N. rates. Its advantage therefore extends from producing low-hull-energy candidates to yielding structures that also satisfy the evaluation's novelty and uniqueness criteria. These findings support its potential for discovering stable candidates beyond repeated or reference-matching outputs.

Table 2: Comparison of generated structures and their corresponding CHGNet-relaxed structures. FlowMM and FlowLLM results are taken from the FlowLLM. RMD results are evaluated on the largest evaluable set for each metric. RMSD is computed only over successfully matched structures.

Method	Match Rate (%) ↑	RMSD ↓	Δ-Energy↓	Num Steps ↓
FlowMM	74.3	0.096	0.3031	191.98
FlowLLM	94.9	0.023	**0.0898**	37.97
CrystaLLM	86.27	0.0287	15.66	122.56
RMD (50%)	98.30	0.00428	3.0332	<u>19.1673</u>
RMD (75%)	<u>99.10</u>	<u>0.00246</u>	1.5746	19.2365
RMD (Full)	**99.99**	**0.00015**	<u>0.2789</u>	**13.9220**

The direct comparison with MD strengthens this evidence. RMD improves both validity measures and all four stability-related rates over the random-selection baseline. The gains span basic plausibility and the yield of physically screened, unique, and novel candidates. Together, they support the effectiveness of the score-guided selective refinement strategy relative to random position selection.

4.2 Comparison with Relaxed Structures

This evaluation examines how much further adjustment generated structures require under subsequent relaxation. Comparing models trained on different data fractions also assesses whether structural quality is retained with reduced training data.

4.2.1 Experimental Settings

The evaluation follows the CHGNet relaxation protocol of FlowLLM Sriram et al. (2024). Each generated structure is compared with its own relaxed counterpart using Match Rate, normalized RMSD, absolute energy difference (Δ-Energy), and Num Steps. These metrics are defined in Appendix B.2. This evaluation is separate from the VASP-based stability assessment in Table 1. FlowMM and FlowLLM results are taken from the FlowLLM paper, whereas CrystaLLM results are obtained using the authors' released code and checkpoints Antunes et al. (2024). For RMD, the parenthetical labels in Table 2 specify the training-data fraction: Full uses the entire training set, whereas 75% and 50% use randomly sampled subsets of the corresponding sizes.

4.2.2 Experimental Results

Structures generated by RMD Full remain closer to their CHGNet-relaxed counterparts than those of the listed baselines, with the highest Match Rate, lowest RMSD, and fewest average optimization steps (Table 2). These complementary measures indicate less geometric change and fewer subsequent optimization steps under the specified protocol. Overall, the results provide empirical support for the relaxation-inspired generation strategy.

The reduced-data variants retain strong geometric quality. Even with a randomly selected 50% of the training set, RMD outperforms all listed baselines in Match Rate, RMSD, and Num Steps. Increasing the training fraction improves structural matching and reduces geometric displacement and energy change, with Full training performing best among the RMD variants. These findings support effective data utilization in this setting: substantial geometric quality is retained with less training data, while additional examples improve agreement with relaxed configurations.

4.3 Crystal Structure Prediction

The CSP evaluation assesses whether RMD can recover reference structures from their compositions and whether this capability generalizes across crystal datasets.

Table 3: Comparison of Match Rate (MR) and root mean square error (RMSE) across different models and crystal datasets for CSP tasks. The baseline results are taken from CrystaLLM Antunes et al. (2024) and Mat2Seq Yan et al. (2024).

Model	Perov-5		Carbon-24		MP-20		MPTS-52	
	MR ↑	RMSE ↓	MR ↑	RMSE ↓	MR ↑	RMSE ↓	MR ↑	RMSE ↓
One Shot								
CDVAE	45.3%	0.114	17.1%	0.297	33.9%	0.105	5.34%	0.211
DiffCSP	52.0%	0.076	17.5%	0.276	51.5%	0.063	12.2%	0.179
CrystaLLM	46.1%	0.095	20.3%	0.176	58.7%	0.041	19.2%	0.111
Mat2Seq	50.0%	0.099	23.7%	0.169	61.3%	0.040	23.1%	0.109
RMD	**53.3%**	**0.071**	**24.2%**	**0.163**	**61.9%**	**0.035**	**24.3%**	**0.101**
20 Shots								
CDVAE	88.5%	0.046	88.4%	0.229	67.0%	0.103	20.8%	0.209
DiffCSP	98.6%	**0.013**	88.5%	0.219	77.9%	0.049	34.0%	0.175
CrystaLLM	97.6%	0.025	85.2%	0.151	74.0%	0.035	33.8%	0.106
Mat2Seq	98.5%	0.023	86.0%	0.148	75.3%	0.037	36.4%	0.088
RMD	**99.1%**	0.016	**89.3%**	**0.146**	**78.2%**	**0.033**	**37.5%**	**0.079**

4.3.1 Experimental Settings

Experiments cover Perov-5 Castelli et al. (2012a;b), Carbon-24 Pickard (2020), MP-20 Jain et al. (2013), and MPTS-52 Baird (2023), using their established benchmark splits. Each test instance receives either one or 20 generated candidates, evaluated according to the DiffCSP protocol Jiao et al. (2024a). Match Rate measures the fraction of test instances with at least one candidate matching the reference. RMSE measures the normalized geometric error of the best matching candidate for each successfully recovered instance. The two metrics capture recovery success and geometric accuracy, respectively. Dataset splits and matching details appear in Appendix B.3.

4.3.2 Experimental Results

Under one-shot sampling, RMD achieves the highest Match Rate and lowest RMSE on all four datasets (Table 3). Improvements in both metrics indicate that RMD recovers more reference structures while maintaining accurate geometry among successful matches. With 20 candidates, RMD retains the highest Match Rate on all four datasets and achieves the lowest RMSE except on Perov-5. Its MPTS-52 Match Rate increases to 37.5%, showing that multiple generation trajectories expand composition-conditioned structure recovery. Overall, the consistent recovery-rate advantage across datasets supports the effectiveness of RMD for composition-conditioned structure prediction.

5 Conclusion and Limitation

We presented Relaxation-inspired Masked Diffusion (RMD) for crystal sequence generation. Bidirectional context, relaxation-score-driven position selection, and selective state updates enable iterative revision of generated content. RMD achieves state-of-the-art results on the evaluated de novo generation and composition-conditioned crystal structure prediction benchmarks, showing its potential for stable crystal discovery.

Limitation. RMD follows the assess–update–reassess organization of physical relaxation, but its scores come from model predictions rather than calculated energies, forces, or stresses. Token updates therefore guarantee neither decreasing energy nor final physical stability. Future work could incorporate energy or force information into position selection and updates.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for [minor language polishing of a small number of sentences, without changing their intended meaning]. We have not used generative AI tools for [any other aspect of this work]. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Luis M Antunes, Keith T Butler, and Ricardo Grau-Crespo. Crystal structure generation with autoregressive large language modeling. *Nature Communications*, 15(1):10570, 2024.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Sam Baird. mp-time-split. `https://github.com/sparks-baird/mp-time-split`, 2023. Accessed: 2024.

Ivano E Castelli, David D Landis, Kristian S Thygesen, Søren Dahl, Ib Chorkendorff, Thomas F Jaramillo, and Karsten W Jacobsen. New cubic perovskites for one-and two-photon water splitting using the computational materials repository. *Energy & Environmental Science*, 5(10):9034–9043, 2012a.

Ivano E Castelli, Thomas Olsen, Soumendu Datta, David D Landis, Søren Dahl, Kristian S Thygesen, and Karsten W Jacobsen. Computational screening of perovskite metal oxides for optimal solar light capture. *Energy & Environmental Science*, 5(2):5814–5819, 2012b.

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.

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.

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

Rui Jiao, Wenbing Huang, Peijia Lin, Jiaqi Han, Pin Chen, Yutong Lu, and Yang Liu. Crystal structure prediction by joint equivariant diffusion. *Advances in Neural Information Processing Systems*, 36, 2024a.

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

Chaitanya K Joshi, Xiang Fu, Yi-Lun Liao, Vahe Gharakhanyan, Benjamin Kurt Miller, Anuroop Sriram, and Zachary W Ulissi. All-atom diffusion transformers: Unified generative modelling of molecules and materials. In *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*, volume 267, Vancouver, Canada, 2025. PMLR.

Benjamin Kurt Miller, Ricky TQ Chen, Anuroop Sriram, and Brandon M Wood. Flowmm: Generating materials with riemannian flow matching. In *Forty-first International Conference on Machine Learning*, 2024.

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.

Ahmed I Osman, Mahmoud Hefny, MIA Abdel Maksoud, Ahmed M Elgarahy, and David W Rooney. Recent advances in carbon capture storage and utilisation technologies: a review. *Environmental Chemistry Letters*, 19(2):797–849, 2021.

Chris J. Pickard. Airss data for carbon at 10gpa and the c+n+h+o system at 1gpa. https://archive.materialscloud.org/record/2020.0026/v1, 2020. Accessed: 2024-8-15.

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*, volume 37, 2024. URL https://arxiv.org/abs/2410.23405.

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.

Minkai Xu, Shitong Luo, Yoshua Bengio, Jian Peng, and Jian Tang. Learning neural generative dynamics for molecular conformation generation. In *International Conference on Learning Representations*, 2021a.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2021b.

Keqiang Yan, Xiner Li, Hongyi Ling, Kenna Ashen, Carl Edwards, Raymundo Arróyave, Marinka Zitnik, Heng Ji, Xiaofeng Qian, Xiaoning Qian, et al. Invariant tokenization of crystalline materials for language model enabled generation. *Advances in Neural Information Processing Systems*, 37: 125050–125072, 2024.

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

Qing Zhao, Sanjuna Stalin, Chen-Zi Zhao, and Lynden A Archer. Designing solid-state electrolytes for safe, energy-dense batteries. *Nature Reviews Materials*, 5(3):229–252, 2020.

Zhi-Jian Zhao, Sihang Liu, Shenjun Zha, Dongfang Cheng, Felix Studt, Graeme Henkelman, and Jinlong Gong. Theory-guided design of catalytic materials using scaling relationships and reactivity descriptors. *Nature Reviews Materials*, 4(12):792–804, 2019.

A More Details of Methods

This appendix develops the probability model and generation rules introduced in Section 3. The derivations follow the absorbing-mask process from its forward marginals to the training objective, then establish the state recursion used in score-driven generation. As in the main text, conditioning on the sequence length, position sets, and fixed contents is implicit.

A.1 Forward Marginals and Conditional Posteriors

The posterior in equation 3 determines the reference recovery process used for training. Its coefficients follow from the cumulative marginal of the one-step transition in equation 2. Deriving these quantities also establishes when the corruption indicator can be read directly from the current mask pattern.

Corruption is independent across updatable positions, and clean content satisfies $x_0^i \neq [\texttt{MASK}]$ for $i \in \mathcal{I}$. Starting from $\delta_{x_0^i}$, a position retains its original token at step t only if it has remained visible through all preceding steps and survives the current transition. Therefore,

$$\Pr(x_t^i = x_0^i \mid x_0^i) = \alpha_{t-1}\beta_t = \alpha_t, \tag{A.1}$$

where $\alpha_0 = 1$ and $\alpha_t = \prod_{s=1}^{t} \beta_s$. All remaining probability mass belongs to the absorbing mask state:

$$q(x_t^i \mid x_0^i) = \alpha_t \delta_{x_0^i}(x_t^i) + (1 - \alpha_t)\delta_{[\texttt{MASK}]}(x_t^i). \tag{A.2}$$

The sequence marginal is the product of these distributions over $\mathcal{I}$, together with point masses at c^i for fixed positions. The linear schedule $\alpha_t = 1 - t/T$ gives $\beta_t = \alpha_t/\alpha_{t-1} = (T-t)/(T-t+1)$ for $t = 1, \ldots, T$. In particular, $\alpha_T = 0$, so every updatable position is masked at the terminal step.

The forward support at position i is $\{x_0^i, [\texttt{MASK}]\}$. Since the clean token is not a mask,

$$\mathbb{I}[x_t^i \neq x_0^i] = \mathbb{I}[x_t^i = [\texttt{MASK}]]. \tag{A.3}$$

Thus, the training indicator z_t^i is observable from mask membership on forward samples. This identification uses both the absorbing-mask transition and mask-free clean content.

If $x_t^i \neq [\texttt{MASK}]$, absorption rules out a masked previous state, and both x_{t-1}^i and x_t^i equal x_0^i. The posterior is consequently $\delta_{x_t^i}$. If $x_t^i = [\texttt{MASK}]$, the previous state was either still clean or already masked. The first event has joint probability

$$\begin{aligned} &\Pr(x_{t-1}^i = x_0^i,\ x_t^i = [\texttt{MASK}] \mid x_0^i) \\ &\quad = \alpha_{t-1}(1 - \beta_t) = \alpha_{t-1} - \alpha_t. \end{aligned} \tag{A.4}$$

Dividing by $\Pr(x_t^i = [\texttt{MASK}] \mid x_0^i) = 1 - \alpha_t$ gives the posterior recovery probability

$$\Pr(x_{t-1}^i = x_0^i \mid x_t^i = [\texttt{MASK}], x_0^i) = \frac{\alpha_{t-1} - \alpha_t}{1 - \alpha_t} = \eta_t. \tag{A.5}$$

An already masked position remains masked with probability one in the forward transition, so the other posterior probability is

$$\Pr(x_{t-1}^i = [\texttt{MASK}] \mid x_t^i = [\texttt{MASK}], x_0^i) = \frac{1 - \alpha_{t-1}}{1 - \alpha_t} = \rho_t. \tag{A.6}$$

These probabilities sum to one and yield equation 3. Under the adopted schedule, $1 - \alpha_t = t/T > 0$ for every $t \geq 1$, $\eta_t = 1/t$, and $\rho_t = (t-1)/t$. The boundary values $\eta_1 = 1$ and $\rho_1 = 0$ will determine the final reconstruction term.

A.2 Binary Routing and Posterior Consistency

The posterior above separates the decision to remain masked from the content recovered at a position. This section derives the binary reference route in equation 4, verifies that marginalizing it recovers the posterior, and establishes the fixed matching parameterization used in the training model.

A.2.1 THE BINARY REFERENCE DECISION

The routing variable records which branch of the posterior is taken, rather than the identity of the recovered token. Its distribution therefore follows directly from the probabilities derived in Appendix A.1.

For the transition $t \to t-1$, $r^i_{t-1} = 1$ selects the masked branch and $r^i_{t-1} = 0$ selects retention or recovery. A visible position is retained with probability one. At a masked position, the probability of selecting the masked branch is $\rho_t = 1 - \eta_t$. Hence,

$$q(r^i_{t-1} = 1 \mid \mathbf{x}_t, \mathbf{x}_0) = \begin{cases} 0, & x^i_t \neq [\texttt{MASK}], \\ 1 - \eta_t, & x^i_t = [\texttt{MASK}]. \end{cases} \tag{A.7}$$

On the forward support, the right-hand side depends only on the current mask pattern and the time step. Thus, $q(r^i_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = q(r^i_{t-1} \mid \mathbf{x}_t)$, as used in equation 4. Independent corruption gives the sequence-level route $q(\mathbf{r}_{t-1} \mid \mathbf{x}_t) = \prod_{i \in \mathcal{I}} q(r^i_{t-1} \mid \mathbf{x}_t)$. The removal of $\mathbf{x}_0$ from this conditional relies on equation A.3; the clean sequence remains necessary to specify the recovered content.

A.2.2 POSTERIOR RECOVERY BY ROUTE MARGINALIZATION

To connect the binary decision to the original reverse process, the route must be paired with a conditional content distribution. The following construction reproduces equation 3 exactly and supplies the reference distribution for the conditional model in equation 5.

Define

$$q(x^i_{t-1} \mid r^i_{t-1}, \mathbf{x}_t, \mathbf{x}_0) = \begin{cases} \delta_{x^i_t}(x^i_{t-1}), & x^i_t \neq [\texttt{MASK}], \\ \delta_{[\texttt{MASK}]}(x^i_{t-1}), & x^i_t = [\texttt{MASK}],\ r^i_{t-1} = 1, \\ \delta_{x^i_0}(x^i_{t-1}), & x^i_t = [\texttt{MASK}],\ r^i_{t-1} = 0. \end{cases} \tag{A.8}$$

At a visible position, the route is deterministically zero, so summing over it leaves $\delta_{x^i_t}$. At a masked position,

$$\begin{aligned} &\sum_{r^i_{t-1}} q(r^i_{t-1} \mid \mathbf{x}_t) q(x^i_{t-1} \mid r^i_{t-1}, \mathbf{x}_t, \mathbf{x}_0) \\ &\qquad = \rho_t \delta_{[\texttt{MASK}]}(x^i_{t-1}) + \eta_t \delta_{x^i_0}(x^i_{t-1}). \end{aligned} \tag{A.9}$$

These two cases recover equation 3. Taking products over positions establishes the corresponding sequence-level identity. In particular, the joint reference transition is

$$q(\mathbf{x}_{t-1}, \mathbf{r}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = q(\mathbf{r}_{t-1} \mid \mathbf{x}_t) q(\mathbf{x}_{t-1} \mid \mathbf{r}_{t-1}, \mathbf{x}_t, \mathbf{x}_0).$$

The model preserves the copying and masking branches, replacing only $\delta_{x^i_0}$ by $f^i_\theta(\cdot \mid \mathbf{x}_t)$ in the recovery branch. Its route marginal is therefore the reverse transition in equation 6.

A.2.3 FIXED MATCHING OF REFERENCE AND MODEL ROUTES

The training model fixes the route probabilities while learning the recovered content. This separates the known masking schedule from the prediction task. The binary parameterization also preserves matching when retention and recovery are represented by separate auxiliary decisions.

Specifically, Section 3.1 imposes $p_\theta(\mathbf{r}_{t-1} \mid \mathbf{x}_t) := q(\mathbf{r}_{t-1} \mid \mathbf{x}_t)$. The right-hand side is fully specified by the current mask pattern and ρ_t, so this parameterization introduces no learnable routing module.

To express its consistency with a two-branch auxiliary representation, let $\mathbf{v}^i_{t-1} = (v^{\text{keep},i}_{t-1}, v^{\text{recover},i}_{t-1})$. The first component indicates retention when $z^i_t = 0$, and the second indicates recovery when $z^i_t = 1$. Only the component selected by the current state affects the transition:

$$r^i_{t-1} = (1 - z^i_t)(1 - v^{\text{keep},i}_{t-1}) + z^i_t (1 - v^{\text{recover},i}_{t-1}). \tag{A.10}$$

Retention has reference probability one at visible positions, and recovery has probability η_t at masked positions. Thus, this mapping induces equation A.7 regardless of the inactive component.

Let $Q_t(\mathbf{v} \mid \mathbf{x}_t)$ be a full auxiliary law whose active decisions have the factorized reference distribution specified above, and let $p_\theta(\mathbf{v} \mid \mathbf{x}_t) = Q_t(\mathbf{v} \mid \mathbf{x}_t)$. By equation A.3, the mapping in equation A.10 can be written as $\mathbf{r} = G_t(\mathbf{v}, \mathbf{x}_t)$ on forward samples. Applying it to both laws gives

$$\begin{aligned} p_\theta(\mathbf{r} \mid \mathbf{x}_t) &= \sum_{\mathbf{v}: G_t(\mathbf{v}, \mathbf{x}_t) = \mathbf{r}} p_\theta(\mathbf{v} \mid \mathbf{x}_t) \\ &= \sum_{\mathbf{v}: G_t(\mathbf{v}, \mathbf{x}_t) = \mathbf{r}} Q_t(\mathbf{v} \mid \mathbf{x}_t) \\ &= q(\mathbf{r} \mid \mathbf{x}_t). \end{aligned} \tag{A.11}$$

Conditional recovery depends only on the active decision, and the matching full laws also have matching conditional laws for the inactive components. Those components therefore contribute zero conditional KL when marginalized. The binary route retains the information needed for the training derivation, which uses the directly imposed matching parameterization above.

A.3 The Joint Variational Objective and KL Decomposition

Posterior consistency permits the reference routes to augment the diffusion path without changing its sequence marginals. This gives the path distribution underlying equation 7. The derivation below separates the terminal and reconstruction terms, then decomposes each intermediate joint transition into routing and recovery terms.

A.3.1 The Augmented Path Model and Variational Bound

The one-step construction in Appendix A.2 extends to a normalized model over complete sequences and auxiliary routes. Introducing a reference distribution over these latent paths yields the training bound in Section 3.2.

The augmented model is

$$P_\theta(\mathbf{x}_{0:T}, \mathbf{r}_{0:T-1}) = p_T(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{r}_{t-1} \mid \mathbf{x}_t) p_\theta(\mathbf{x}_{t-1} \mid \mathbf{r}_{t-1}, \mathbf{x}_t), \tag{A.12}$$

where p_T places masks at all updatable positions and preserves the prescribed fixed contents. Its data likelihood $p_\theta(\mathbf{x}_0)$ is obtained by summing over $\mathbf{x}_{1:T}$ and $\mathbf{r}_{0:T-1}$. The corresponding reference path distribution is

$$\begin{aligned} &Q(\mathbf{x}_{1:T}, \mathbf{r}_{0:T-1} \mid \mathbf{x}_0) \\ &\quad = q(\mathbf{x}_T \mid \mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_{t-1}, \mathbf{r}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)\, q(\mathbf{r}_0 \mid \mathbf{x}_1, \mathbf{x}_0). \end{aligned} \tag{A.13}$$

Marginalizing its routes gives the forward path written in reverse conditional order. At $t = 1$, $\rho_1 = 0$, so the reference and model both assign $\mathbf{r}_0 = \mathbf{0}$. Consequently,

$$p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1) = p_\theta(\mathbf{x}_0 \mid \mathbf{r}_0 = \mathbf{0}, \mathbf{x}_1). \tag{A.14}$$

For a fixed clean sequence, the variational identity is

$$\begin{aligned} \mathbb{E}_Q \left[\log \frac{Q}{P_\theta} \right] = &- \log p_\theta(\mathbf{x}_0) \\ &+ D_{\mathrm{KL}}(Q(\mathbf{x}_{1:T}, \mathbf{r}_{0:T-1} \mid \mathbf{x}_0) \,\|\, P_\theta(\mathbf{x}_{1:T}, \mathbf{r}_{0:T-1} \mid \mathbf{x}_0)), \end{aligned}$$

where the ratio on the left uses the joint model in equation A.12. Nonnegativity of KL therefore gives

$$-\log p_\theta(\mathbf{x}_0) \le \mathbb{E}_Q \left[\log \frac{Q(\mathbf{x}_{1:T}, \mathbf{r}_{0:T-1} \mid \mathbf{x}_0)}{P_\theta(\mathbf{x}_{0:T}, \mathbf{r}_{0:T-1})} \right]. \tag{A.15}$$

Substituting the path factorizations expands the log ratio as

$$\begin{aligned} \log \frac{Q}{P_\theta} = &\log \frac{q(\mathbf{x}_T \mid \mathbf{x}_0)}{p_T(\mathbf{x}_T)} \\ &+ \sum_{t=2}^{T} \log \frac{q(\mathbf{x}_{t-1}, \mathbf{r}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{r}_{t-1} \mid \mathbf{x}_t) p_\theta(\mathbf{x}_{t-1} \mid \mathbf{r}_{t-1}, \mathbf{x}_t)} \\ &- \log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1). \end{aligned}$$

The final route factors cancel because they are identical point masses. Taking expectations identifies

$$\mathcal{L}_{\text{terminal}}(\mathbf{x}_0) = D_{\text{KL}}(q(\mathbf{x}_T \mid \mathbf{x}_0) \,\|\, p_T(\mathbf{x}_T)), \tag{A.16}$$

$$\mathcal{L}_{\text{reconstruction}}(\mathbf{x}_0) = \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}[-\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)], \tag{A.17}$$

and, for $t > 1$,

$$\mathcal{J}_t = D_{\text{KL}}(q(\mathbf{x}_{t-1}, \mathbf{r}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \,\|\, p_\theta(\mathbf{x}_{t-1}, \mathbf{r}_{t-1} \mid \mathbf{x}_t)). \tag{A.18}$$

Each $\mathcal{J}_t$ is conditional on $\mathbf{x}_t, \mathbf{x}_0$. Averaging it over $q(\mathbf{x}_t \mid \mathbf{x}_0)$, summing over $t = 2, \ldots, T$, and averaging the bound over the data distribution yields equation 7. The terminal term is zero because $\alpha_T = 0$ and the prior matches the resulting all-mask distribution, including fixed contents.

A.3.2 Decomposition into Routing and Recovery

The intermediate loss in equation A.18 compares joint transitions over content and routes. Factoring these transitions separates the contribution of the routing decision from the conditional recovery error, giving the two terms in equation 8.

Fix $\mathbf{x}_t, \mathbf{x}_0$, and abbreviate $\mathbf{u} = \mathbf{x}_{t-1}$ and $\mathbf{r} = \mathbf{r}_{t-1}$. With the shared conditions temporarily suppressed,

$$q(\mathbf{u}, \mathbf{r}) = q(\mathbf{r})q(\mathbf{u} \mid \mathbf{r}), \qquad p_\theta(\mathbf{u}, \mathbf{r}) = p_\theta(\mathbf{r})p_\theta(\mathbf{u} \mid \mathbf{r}).$$

Substitution gives

$$D_{\text{KL}}(q\|p_\theta) = \sum_{\mathbf{r},\mathbf{u}} q(\mathbf{r})q(\mathbf{u} \mid \mathbf{r}) \left[\log \frac{q(\mathbf{r})}{p_\theta(\mathbf{r})} + \log \frac{q(\mathbf{u} \mid \mathbf{r})}{p_\theta(\mathbf{u} \mid \mathbf{r})}\right]. \tag{A.19}$$

The first logarithm is independent of $\mathbf{u}$. Using $\sum_{\mathbf{u}} q(\mathbf{u} \mid \mathbf{r}) = 1$ gives

$$\begin{aligned} D_{\text{KL}}(q\|p_\theta) = {} & D_{\text{KL}}(q(\mathbf{r})\|p_\theta(\mathbf{r})) \\ & + \mathbb{E}_{q(\mathbf{r})} D_{\text{KL}}(q(\mathbf{u} \mid \mathbf{r})\|p_\theta(\mathbf{u} \mid \mathbf{r})). \end{aligned} \tag{A.20}$$

Restoring the conditions, and using the observable reference route from Appendix A.2.1, yields

$$\begin{aligned} \mathcal{J}_t = {} & \underbrace{D_{\text{KL}}(q(\mathbf{r}_{t-1} \mid \mathbf{x}_t) \,\|\, p_\theta(\mathbf{r}_{t-1} \mid \mathbf{x}_t))}_{\mathcal{J}_{\text{route},t}} \\ & + \underbrace{\mathbb{E}_{q(\mathbf{r}_{t-1}|\mathbf{x}_t)} D_{\text{KL}}(q(\mathbf{x}_{t-1} \mid \mathbf{r}_{t-1}, \mathbf{x}_t, \mathbf{x}_0) \,\|\, p_\theta(\mathbf{x}_{t-1} \mid \mathbf{r}_{t-1}, \mathbf{x}_t))}_{\mathcal{J}_{\text{recovery},t}}. \end{aligned} \tag{A.21}$$

This identity follows from the KL chain rule before applying fixed matching. Both terms refer to the augmented transition; marginalizing the route generally changes the KL being evaluated.

The full objective additionally averages the route term over clean data and forward states:

$$\mathbb{E}_{\mathbf{x}_0 \sim p_{\text{data}}} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}[\mathcal{J}_{\text{route},t}]. \tag{A.22}$$

The recovery term has the same outer expectations and the additional route expectation displayed in equation A.21.

A.4 Cancellation Under Fixed Matching and Cross-Entropy Reduction

The joint decomposition identifies the two possible sources of training loss. Under the fixed route distribution and conditional recovery model of Section 3.1, only recovery of masked content contributes. The following branchwise calculation establishes equation 9 and includes the final reconstruction step in the full theoretical objective.

A.4.1 Cancellation of the Routing Term

Fixed matching removes the routing contribution pointwise in the conditioning state. This establishes why the decomposition in equation 8 requires no separate routing optimization.

For every state on the absorbing forward support, $p_\theta(\mathbf{r}_{t-1} \mid \mathbf{x}_t) = q(\mathbf{r}_{t-1} \mid \mathbf{x}_t)$ by parameterization. Therefore,

$$\begin{aligned}\mathcal{J}_{\text{route},t} &= \sum_{\mathbf{r}} q(\mathbf{r} \mid \mathbf{x}_t) \log \frac{q(\mathbf{r} \mid \mathbf{x}_t)}{p_\theta(\mathbf{r} \mid \mathbf{x}_t)} \\ &= \sum_{\mathbf{r}} q(\mathbf{r} \mid \mathbf{x}_t) \log \frac{q(\mathbf{r} \mid \mathbf{x}_t)}{q(\mathbf{r} \mid \mathbf{x}_t)} = 0.\end{aligned} \tag{A.23}$$

Zero-probability events contribute zero under the usual KL convention. The equality holds for each reference state, so the outer expectation in equation A.22 also vanishes. This cancellation follows from fixing both route laws to the same schedule, rather than learning to match them.

A.4.2 Reduction of Conditional Recovery to Cross-Entropy

With the routing term removed, the remaining loss compares the reference and model content distributions within each branch. Their shared copying and masking rules leave only the clean-token prediction error on the recovery branch.

Given the route and current sequence, both conditional distributions factorize over positions:

$$\begin{aligned}q(\mathbf{x}_{t-1} \mid \mathbf{r}, \mathbf{x}_t, \mathbf{x}_0) &= \prod_{i \in \mathcal{I}} q(x^i_{t-1} \mid r^i, \mathbf{x}_t, \mathbf{x}_0), \\ p_\theta(\mathbf{x}_{t-1} \mid \mathbf{r}, \mathbf{x}_t) &= \prod_{i \in \mathcal{I}} p_\theta(x^i_{t-1} \mid r^i, \mathbf{x}_t).\end{aligned} \tag{A.24}$$

Here, $\mathbf{r} = \mathbf{r}_{t-1}$, and the identical point masses at fixed positions are omitted. Reference factorization follows from independent corruption and routing; model factorization is specified in Section 3.1. Each prediction can still use the entire visible sequence as context. The product structure makes the conditional joint KL equal to the sum of positionwise KL terms.

The reference distribution in equation A.8 and model distribution in equation 5 select the same branch from the observable mask pattern. At a visible position, $z^i_t = 0$, both copy the current token:

$$D_{\mathrm{KL}}(\delta_{x^i_t} \| \delta_{x^i_t}) = 0. \tag{A.25}$$

At a masked position with $r^i_{t-1} = 1$, both remain masked:

$$D_{\mathrm{KL}}(\delta_{\texttt{[MASK]}} \| \delta_{\texttt{[MASK]}}) = 0. \tag{A.26}$$

At a masked position with $r^i_{t-1} = 0$, the reference recovers x^i_0, while the model predicts its content:

$$\begin{aligned}D_{\mathrm{KL}}\left(\delta_{x^i_0} \| f^i_\theta(\cdot \mid \mathbf{x}_t)\right) &= \sum_{a \in \mathcal{V}} \delta_{x^i_0}(a) \log \frac{\delta_{x^i_0}(a)}{f^i_\theta(a \mid \mathbf{x}_t)} \\ &= -\log f^i_\theta(x^i_0 \mid \mathbf{x}_t).\end{aligned} \tag{A.27}$$

Only $a = x^i_0$ contributes to the sum. Averaging over the reference route weights this branch by $1 - \rho_t = \eta_t$, giving the positionwise contribution

$$z^i_t \left[\rho_t \cdot 0 + \eta_t\big(-\log f^i_\theta(x^i_0 \mid \mathbf{x}_t)\big)\right]. \tag{A.28}$$

Summing over positions and using equation A.23 yields

$$\mathcal{J}_t = \mathcal{J}_{\text{recovery},t} = \eta_t \sum_{i \in \mathcal{I}} z^i_t[-\log f^i_\theta(x^i_0 \mid \mathbf{x}_t)],$$

which is equation 9. The clean sequence supplies the recovery targets, while branch selection uses the input mask pattern through equation A.3.

A.4.3 Final Reconstruction and the Full Theoretical Objective

The intermediate losses alone do not include the final transition to the observed sequence. The boundary values of the posterior allow its reconstruction term to be written in the same cross-entropy form, completing the reduction of equation 7.

Since $\eta_1 = 1$ and $\rho_1 = 0$, the last reference route selects retention at visible positions and recovery at masked ones. Equation equation A.14 and the conditional factorization therefore give

$$\mathcal{L}_{\text{reconstruction}}(\mathbf{x}_0) = \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \sum_{i \in \mathcal{I}} z_1^i [-\log f_\theta^i(x_0^i \mid \mathbf{x}_1)]. \tag{A.29}$$

Combining this term with the zero terminal loss and the intermediate losses gives

$$\mathcal{L}_{\text{VB}} = \mathbb{E}_{\mathbf{x}_0 \sim p_{\text{data}}} \sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \left[\eta_t \sum_{i \in \mathcal{I}} z_t^i [-\log f_\theta^i(x_0^i \mid \mathbf{x}_t)] \right]. \tag{A.30}$$

The $t = 1$ contribution appears once as reconstruction; the remaining terms correspond to the joint transition KLs. Under the linear schedule, their theoretical coefficient is $\eta_t = 1/t$. Thus, all learnable terms train the same clean-token predictor, without a separate routing loss evaluation.

A.5 Theoretical Weights, Practical Reweighting, and Batch Normalization

Equation equation A.30 specifies a time-summed variational objective. The practical loss retains its masked-token prediction task but uses a different time weighting and a batch-level normalization. Distinguishing these choices explains the relationship between the theoretical reduction in Section 3.2 and the optimized training loss.

If t is sampled from a distribution $\pi(t) > 0$ on $\{1, \dots, T\}$, estimating the theoretical time sum without changing its other normalization factors requires the weight

$$\frac{\eta_t}{\pi(t)}. \tag{A.31}$$

Indeed, the expectation over the sampled time cancels $\pi(t)$ and recovers the original sum. Uniform sampling, $\pi(t) = 1/T$, gives $T\eta_t$.

The practical objective instead uses the linear weight

$$w(t) = \frac{T - t + 1}{T}. \tag{A.32}$$

This is an optimization choice rather than a simplification of η_t or its importance-weighted counterpart. For an effective batch of B corrupted samples, let t_j be the sampled time and $\mathcal{I}_j$ the updatable set for sample j. Define

$$\begin{aligned} M_j^i &= \mathbb{I}[i \in \mathcal{I}_j]\, \mathbb{I}[x_{t_j,j}^i = [\texttt{MASK}]], \\ \ell_j^i &= -\log f_\theta^i(x_{0,j}^i \mid \mathbf{x}_{t_j,j}). \end{aligned} \tag{A.33}$$

Here, j indexes corrupted samples and i indexes token positions. The number of masked updatable positions is

$$D_{\mathcal{B}} = \sum_{j=1}^{B} \sum_{i} M_j^i, \tag{A.34}$$

and the practical loss is

$$\widehat{\mathcal{L}}_{\text{train}} = \frac{\displaystyle\sum_{j=1}^{B} \sum_{i} w(t_j) M_j^i \ell_j^i}{D_{\mathcal{B}}}, \qquad D_{\mathcal{B}} > 0. \tag{A.35}$$

Normalization uses the unweighted mask count across the batch. It therefore differs from dividing by the sum of time weights or averaging separately normalized sample losses. If several corrupted views are included, B counts those views. For $D_{\mathcal{B}} = 0$, the numerator also vanishes; the ratio requires a separate zero-denominator convention.

Time reweighting and normalization by a random mask count change the relative contributions of diffusion times and corrupted samples. Equation equation A.35 is therefore generally not an unbiased estimate of equation A.30, although both train recovery of the same masked structural content.

A.6 Generation States and Score-Driven Routing

The training derivation concerns forward samples with known clean targets. Generation instead applies the learned predictor to evolving sequences $\mathbf{y}_t$, using scores to choose which positions remain open to revision. This section establishes the state identity used after equation 11 and clarifies how score-driven routing relates to the reference construction. The generation index descends from K to 1; K is an iteration budget and need not equal the training horizon T.

A.6.1 Consistency of the Generation State Recursion

Section 3.3 distinguishes the pre-update state m_t^i, current decision r_{t-1}^i, and resulting state m_{t-1}^i. The identity $m_{t-1}^i = r_{t-1}^i$ follows from the masking and acceptance rules, while the pre-update state may differ from both.

At initialization, every updatable position satisfies $m_K^i = \mathbb{I}[y_K^i = [\texttt{MASK}]] = 1$. Assume $m_t^i = \mathbb{I}[y_t^i = [\texttt{MASK}]]$ at iteration t. The update in equation 11 has three cases for $i \in \mathcal{I}$:

1. If $r_{t-1}^i = 1$, the next token is [MASK], so $m_{t-1}^i = 1$.
2. If $r_{t-1}^i = 0$ and $m_t^i = 1$, the candidate is accepted. Since $\mathcal{V}_{\text{gen}}$ excludes [MASK], the next state is $m_{t-1}^i = 0$.
3. If $r_{t-1}^i = 0$ and $m_t^i = 0$, the current token is retained. It is not a mask by the induction hypothesis, so $m_{t-1}^i = 0$.

Hence,

$$\mathbb{I}[y_{t-1}^i = [\texttt{MASK}]] = r_{t-1}^i, \tag{A.36}$$

and the state definition gives $m_{t-1}^i = r_{t-1}^i$ at every iteration. The invariant depends on both mask-free accepted candidates and the prescribed initialization. It identifies temporary retention with a visible algorithmic state, not with correctness relative to an unknown clean structure.

A.6.2 Selective Acceptance and Delayed Recovery

The invariant makes explicit how position selection controls candidate acceptance in equation 11. This distinction explains why evaluating a candidate and selecting a position for further adjustment need not replace that position's content in the current iteration.

Candidates and scores are computed from $\mathbf{y}_t$ before the state update. If $r_{t-1}^i = 0$, a masked position accepts its candidate, whereas a visible position retains y_t^i, even if its newly computed candidate differs. If $r_{t-1}^i = 1$, the output is a mask regardless of the candidate. This either continues masking or returns a visible position to the adjustment state.

A remasked position is therefore still masked in $\mathbf{y}_{t-1}$. Recovery can occur only in a later iteration, after a new prediction has been made from the updated visible context and the new route permits acceptance. The previous token is then absent from the visible input at that position. Repeated selection and recovery allow existing content to be reconsidered without conflating remasking with immediate replacement.

A.6.3 Probabilistic Scope of Score-Based Routing

The state proof holds for the decisions produced by the generation rule, independently of how those decisions are distributed. Relating that rule to training requires examining the score in equation 10 and the selection budget in Section 3.3.

Each relaxation score is the log probability of the selected candidate under the Gumbel-perturbed distribution restricted to $\mathcal{V}_{\text{gen}}$. It need not equal the maximum probability of the unperturbed network distribution or score the token currently retained at that position. Higher scores favor retention or candidate acceptance, while lower scores prioritize continued adjustment.

Given $\mathbf{y}_t$ and the perturbations, ranking scores across positions determines the adjustment set and its routes. Although the perturbations are independent, the shared budget generally couples the resulting routing decisions. The reference route instead factorizes into the distributions in equation 4 and

Algorithm 1 Crystal Sequence Generation by Relaxation-Inspired Refinement

Require: Prediction network f_θ; sequence length N; position sets $\mathcal{I}, \mathcal{F}$; fixed contents $\mathbf{c}_\mathcal{F}$; candidate vocabulary $\mathcal{V}_{\text{gen}}$; iteration budget K

Ensure: Generated structural text $\widehat{S}$ and candidate $\widehat{\text{CIF}}$

Initialize $\mathbf{y}_K, \mathbf{m}_K$ using equation A.37.
for $t = K, K-1, \ldots, 1$ **do**
  Compute logits $\boldsymbol{\ell}_\theta^i(\mathbf{y}_t)$ for all $i \in \mathcal{I}$.
  Draw independent $g_t^i(a) \sim \text{Gumbel}(0,1)$ for $i \in \mathcal{I}$, $a \in \mathcal{V}_{\text{gen}}$.
  Form perturbed logits $\widetilde{\ell}_t^i(a) = \ell_\theta^i(a \mid \mathbf{y}_t) + g_t^i(a)$.
  Compute candidates $\widehat{\mathbf{y}}_t$ and relaxation scores $\mathbf{s}_t$ using equation 10.
  Compute h_{t-1} using equation A.38.
  Determine $\mathcal{R}_{t-1}$ and $\mathbf{r}_{t-1}$ using equation A.39.
  Compute $\mathbf{y}_{t-1}$ using equation 11 and the pre-update state $\mathbf{m}_t$.
  Set $m_{t-1}^i = \mathbb{I}[y_{t-1}^i = [\texttt{MASK}]]$ for $i \in \mathcal{I}$, and $m_{t-1}^i = 0$ for $i \in \mathcal{F}$.
end for
Decode and postprocess using equation A.40.
return $\widehat{S}, \widehat{\text{CIF}}$.

assigns zero probability to remasking visible content. Score-based generation can select such content for adjustment. The two routes share the meanings of masking and retention or recovery, but differ in their distributions and allowed transitions.

Fixed matching applies to the training probability model; score-driven remasking is the generation strategy. Training influences that strategy through the predictions used to compute scores, without a separately learned routing module. The relaxation analogy concerns the assess–update–reassess organization of sequence revision. Scores are not atomic forces or energy gradients, and the state recursion imposes no monotonicity condition on physical energy.

A.7 Complete Generation Algorithm

The prediction, selection, and update rules in Section 3.3 combine into Algorithm 1. The explicit initialization and boundary conditions below complete the specification and show why all updatable positions are unmasked after the final iteration.

The inputs are the prediction network, sequence length N, position sets $\mathcal{I}, \mathcal{F}$, fixed contents $\mathbf{c}_\mathcal{F}$, nonempty candidate vocabulary $\mathcal{V}_{\text{gen}}$, and integer budget $K \geq 1$. The vocabulary excludes masks, padding, and non-generatable special symbols. With $n = |\mathcal{I}| \geq 1$, initialize

$$y_K^i = \begin{cases} [\texttt{MASK}], & i \in \mathcal{I}, \\ c^i, & i \in \mathcal{F}, \end{cases} \qquad m_K^i = \mathbb{I}[i \in \mathcal{I}]. \tag{A.37}$$

For $t = K, \ldots, 1$, the target remaining-mask count is

$$h_{t-1} = \left\lfloor |\mathcal{I}| \frac{t-1}{K} \right\rfloor. \tag{A.38}$$

Writing $s_{t,(j)}$ for the j-th smallest score among updatable positions, the strict selection rule is

$$\mathcal{R}_{t-1} = \begin{cases} \varnothing, & h_{t-1} = 0, \\ \{i \in \mathcal{I} : s_t^i < s_{t,(h_{t-1}+1)}\}, & h_{t-1} > 0, \end{cases} \qquad r_{t-1}^i = \mathbb{I}[i \in \mathcal{R}_{t-1}]. \tag{A.39}$$

After the final update, tokenizer $\mathcal{T}$ and postprocessing map $\mathcal{H}$ produce

$$\widehat{S} = \text{Decode}_\mathcal{T}(\mathbf{y}_0), \qquad \widehat{\text{CIF}} = \mathcal{H}(\widehat{S}). \tag{A.40}$$

For every iteration, $0 \leq h_{t-1} < n$, so the required $(h_{t-1}+1)$-th order statistic exists whenever $h_{t-1} > 0$. Strict thresholding selects at most h_{t-1} positions and exactly that number when scores are distinct. Ties at the threshold can reduce the selected count. By equation A.36, this count is also the number of masks in the next state. Fixed positions retain c^i throughout.

Table 4: Evaluation Metrics and Descriptions

Evaluation Metrics	Description
Structural Validity	A structure is considered valid if the shortest distance between any pair of atoms is greater than 0.5 Å Court et al. (2020).
Composition Validity	A composition is valid if the total charge is neutral Davies et al. (2019).
COV-R (Recall)	Measures the percentage of test structures covered by the generated collection Xu et al. (2021a).
COV-P (Precision)	Measures the percentage of generated structures covered by the test collection Xu et al. (2021b). COV-R and COV-P assess the similarity between the generated crystal materials and the test set.
Density EMD	Describes the Earth Mover's Distance (EMD) between the density distributions of the generated data and the test set data.
Element EMD	Describes the Earth Mover's Distance (EMD) between the unique element quantities distributions of the generated data and the test set data.

At $t = 1$, $h_0 = 0$ and $\mathcal{R}_0 = \varnothing$. All remaining masked positions accept candidates, and visible positions retain their current tokens. Since accepted candidates exclude [MASK], the final sequence has no masks at updatable positions. Generation uses exactly K iterations, with no early stopping based on EOS or physical convergence. Decoding and postprocessing yield a candidate structure whose validity and stability are assessed separately.

B More Details of Experiments

B.1 De Novo Generation

Data and evaluation. MP-20 follows the 60/20/20 training, validation, and test split of CDVAE Xie et al. (2021). RMD generates 10,000 samples without specified target compositions or structures. The same collection serves both conventional quality evaluation and physical assessment. For physical assessment, VASP relaxes the structures, and the resulting DFT energies are evaluated relative to a reference convex hull. Validity and stability-related results appear in Table 1, with complementary distributional evaluations in Table 5.

Validity and distributional metrics. Structural validity checks for unphysical close contacts between atoms. Compositional validity uses the SMACT heuristic to assess compatibility with charge neutrality and related chemical rules. Following established crystal-generation benchmarks, COV-R measures the fraction of test structures covered by the generated collection, whereas COV-P measures the fraction of generated structures covered by the test collection Xie et al. (2021); Antunes et al. (2024). Coverage is evaluated using structural and compositional fingerprints. These metrics assess plausibility and distributional agreement rather than thermodynamic stability. DFT energies of the VASP-relaxed structures provide a separate, energy-based assessment relative to the reference convex hull.

Property distribution differences are measured by the one-dimensional Wasserstein distances between generated and test distributions:

$$d_\rho = W_1\left(\widehat{P}_{\rho,\mathrm{gen}}, \widehat{P}_{\rho,\mathrm{test}}\right), \qquad d_{\mathrm{elem}} = W_1\left(\widehat{P}_{n_{\mathrm{elem}},\mathrm{gen}}, \widehat{P}_{n_{\mathrm{elem}},\mathrm{test}}\right). \tag{B.1}$$

Here, ρ denotes crystal density, and n_{elem} is the number of distinct elements in a structure. Lower values indicate closer agreement with the corresponding test distributions. These distribution-level comparisons do not require paired generated and reference structures.

Coverage and property distribution results. RMD combines high coverage with competitive distribution matching (Table 5). It achieves the lowest reported distance for the distribution of the number of distinct elements, indicating particularly close agreement with the reference collection on this property. Its density distance is lower than most reported baselines, although FlowMM and

Table 5: Benchmarking crystal generation on MP20: coverage and property distribution distances. The number following each method name indicates the size of its language model. Coverage and property results are taken from CrystaLLM Antunes et al. (2024) and FlowLLM Sriram et al. (2024).

Model	Coverage (%) ↑		Property ↓	
	COV-R	COV-P	d_ρ	d_{elem}
CDVAE	99.15	99.49	0.6875	1.4320
FlowMM	<u>99.49</u>	99.58	**0.2390**	<u>0.0830</u>
Mat2Seq (200M)	–	–	–	–
CrystaLLM (200M)	97.22	96.40	0.5965	0.1709
LLaMA (70B)	85.80	98.90	0.8100	0.4400
DiffCSP	**99.71**	<u>99.76</u>	<u>0.3502</u>	0.3398
MatterGen	–	–	–	–
FlowLLM (70B)	96.95	**99.82**	1.1400	0.1500
ADiT	–	–	–	–
MD (150M)	96.38	98.27	0.7366	0.2928
RMD (150M)	99.40	99.54	0.5481	**0.0761**

DiffCSP remain better on this metric. Coverage is also competitive, with some baselines retaining higher recall or precision.

Compared with MD, RMD improves both coverage recall and precision while reducing both property distribution distances. These results indicate broader coverage of the reference collection and closer agreement in density and element-count distributions under the evaluated settings. They support the distributional benefits of score-guided refinement relative to random position selection. This evidence complements the validity and physical screening results in Table 1 by characterizing how well the generated collection represents the reference distribution.

Stability, uniqueness, and novelty. S. and M.S. report the fractions satisfying the corresponding energy criteria. S.U.N. and M.S.U.N. additionally require novelty relative to the reference structure collection and uniqueness within the generated collection. Structural matching identifies reference matches and duplicates, with repeated candidates counted only once for the unique-candidate yield. These metrics complement validity and coverage by assessing the output of physical screening Yan et al. (2024); Joshi et al. (2025). The criteria are defined below.

Stability (S.) and Metastability (M.S.). Under the thresholds in Table 1, stable candidates satisfy $E^{\text{hull}} < 0$, and metastable candidates satisfy $E^{\text{hull}} < 0.1\,\text{eV/atom}$. The latter is a cumulative criterion that includes stable candidates. Here, E^{hull} denotes the signed energy difference relative to the reference convex hull at the candidate composition.

Novelty (N.). Stable structures are compared against the training set using `StructureMatcher` under default settings. A shared element set alone does not establish a match; structural matching must also succeed. A stable structure is deemed novel if no training structure matches it.

Uniqueness (U.). Pairwise comparisons among stable and novel structures use `StructureMatcher`. Mutually matching structures are grouped into equivalence classes, each counted as a single unique structure.

The S.U.N. rate is

$$\text{S.U.N. Rate} = \frac{N_{\text{S.U.N.}}}{N_{\text{gen}}}, \tag{B.2}$$

where N_{gen} denotes the number of generated samples and $N_{\text{S.U.N.}}$ the number of structures that are simultaneously stable, unique, and novel. Table 1 expresses this fraction as a percentage. M.S.U.N. applies the same novelty and uniqueness criteria to candidates satisfying the metastability threshold.

B.2 Comparison with Relaxed Structures

Subsequent structural relaxation incurs computational cost. Comparing generated structures with their relaxed configurations therefore helps assess the amount of further adjustment required. Each generated structure is evaluated against its corresponding CHGNet relaxation endpoint. The tests use FlowLLM's public evaluation implementation without modifying the relevant evaluation functions Sriram et al. (2024). The data-efficiency comparison includes RMD trained on the full training set and on randomly sampled subsets containing 75% and 50% of the training examples. Table 2 denotes these settings by Full, 75%, and 50%, respectively.

1. **Match Rate.** The percentage of evaluated generated structures that successfully match their corresponding relaxed structures using `pymatgen`'s `StructureMatcher` Ong et al. (2013).
2. **RMSD.** The mean root-mean-square displacement over successfully matched structure pairs.
3. **Δ-Energy.** The absolute energy difference between each generated structure and its relaxed counterpart.
4. **Num Steps.** The mean number of steps recorded during CHGNet relaxation, following the FlowLLM evaluation convention. Lower values indicate a shorter recorded optimization trajectory.

Each metric is averaged over its evaluable samples. RMSD requires a successful structural match, whereas energy changes and step counts can be evaluated for unmatched pairs with valid relaxation outputs. These measurements describe changes under the specified relaxation protocol; they do not independently establish thermodynamic stability.

B.3 Crystal Structure Prediction

Datasets and sampling. Perov-5, Carbon-24, and MP-20 follow the 60/20/20 training, validation, and test splits established by CDVAE and DiffCSP Xie et al. (2021); Jiao et al. (2024a). MPTS-52 uses the chronological split containing 27,380 training, 5,000 validation, and 8,096 test structures Jiao et al. (2024a); Antunes et al. (2024).

The target composition is supplied as a fixed prefix, with one or 20 candidates generated for each test instance. Evaluation is performed per reference structure rather than per distinct chemical formula, since multiple structures may share a composition.

Structural matching. Following DiffCSP, structural matching uses `StructureMatcher` with `ltol=0.3`, `stol=0.5`, and `angle_tol=10` Jiao et al. (2024a). Geometric errors are computed after structural alignment and normalized by the average free length per atom, $(V/n)^{1/3}$.

Let Q denote the number of test instances, and let $\mathcal{A}_j$ contain the candidates that successfully match reference structure C_j^{ref}. Match Rate is

$$\mathrm{MR} = \frac{100}{Q} \sum_{j=1}^{Q} \mathbb{I}[\mathcal{A}_j \neq \varnothing], \tag{B.3}$$

where $\mathbb{I}[\cdot]$ is the indicator function. A test instance is successful if at least one candidate matches its reference. Instances without a match remain in the denominator.

For each successful instance, the smallest normalized geometric error among its matching candidates is selected:

$$e_j^* = \min_{C \in \mathcal{A}_j} d(C, C_j^{\mathrm{ref}}), \qquad \mathrm{RMSE} = \frac{\sum_{j:\mathcal{A}_j \neq \varnothing} e_j^*}{\sum_{j=1}^{Q} \mathbb{I}[\mathcal{A}_j \neq \varnothing]}. \tag{B.4}$$

Here, d is the normalized root-mean-square displacement returned by the matcher. RMSE is therefore dimensionless and averaged only over successfully recovered instances. Selecting the best matching

candidate uses the reference structure for benchmark evaluation; it is not a model-side candidate-ranking procedure.