CreSTAL: An In-Silico Reinforcement Learning Environment for Targeted Materials Editing (Student Abstract)

Jordan Madison[1]*, **Amandeep Gupta**[1], **Piyush Jha**[1], **and Vijay Ganesh**[1]

[1]Georgia Institute of Technology, USA
{jmadison9, alnu46, piyush.jha, vganesh45}@gatech.edu

Abstract

We introduce CreSTAL, an in-silico reinforcement learning (RL) environment for training language models to edit crystal structures toward target properties. It extends LLMatDesign's editing loop using RLSF's principle of tool-generated feedback. Execution checks, failure reasons, and property predictions form a validation record and determine one numerical training reward.

Models first learn from demonstrated edits and then receive scalar RL rewards for individual proposals. We evaluate four models on 1,000 held-out materials, allowing up to seven evaluation attempts to reach a predicted band gap of 1.4 ± 0.05 eV. RL has mixed effects across models: Qwen3-4B increases from 111 to 148 target hits, while other models improve less or decline. On Qwen3-4B, two exploratory reward variants reach 156 and 157 hits. These results suggest that executable materials feedback can support RL training, while also highlighting the sensitivity of performance to reward design.

Motivation

We study whether language models can learn to make crystal edits that bring a material closer to a desired property. Our focus is band gap—the minimum energy needed to excite an electron into a state where it can contribute to electrical conduction. Band gap influences a semiconductor's electrical and optical behavior, making it important across electronic and optoelectronic applications. We adopt LLMatDesign's 1.4 eV target as a concrete example of property-directed crystal editing (Jia, Zhang, and Fung 2024).

LLMatDesign uses pretrained language-model agents to propose crystal edits and evaluate their outcomes with materials-design tools (Jia, Zhang, and Fung 2024). CreSTAL builds on this framework by using edit outcomes to train the model's future choices. Models first learn from demonstrated edits through supervised fine-tuning (SFT). RL then rewards their proposals using execution checks and predicted property changes, connecting the training objective to whether an edit can be carried out and moves the material toward the target.

*Primary author address: Klaus Advanced Computing Building, 266 Ferst Dr NW, Atlanta, GA 30332.

Copyright © 2027, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

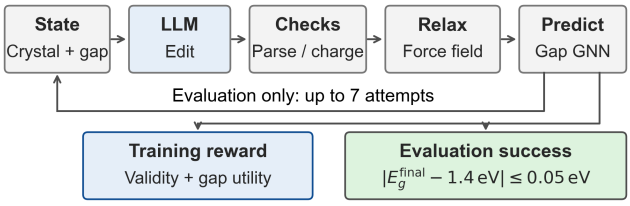


Figure 1: Single-edit RL assigns rewards without attempt history. Only evaluation repeats the loop, using recent attempts in the next prompt. E_g^{final} is the final recorded gap.

This approach is inspired by reinforcement learning from symbolic feedback (RLSF), which uses tool-generated diagnostics to provide feedback on parts of a response (Jha et al. 2024). CreSTAL combines its checks and outcomes into one scalar reward per proposed edit. We test whether learning from individual edits improves band-gap targeting beyond SFT when models make successive editing attempts.

Environment and Study

An evaluation episode consists of attempts to edit one starting crystal:

- **State:** the crystal and its current band gap. The model sees its formula, gap, target, and a summary of previous attempts.
- **Action:** an `add`, `remove`, `substitute`, or `exchange` proposal, with reasoning called a hypothesis.
- **Transition:** extract the proposed edit from the response, apply it, and check the composition. Relaxation adjusts the crystal geometry before MatDeepLearn predicts the gap (Fung et al. 2021).
- **Reward:** one numerical score based on the checks and outcomes (Eq. (1)).
- **Target hit:** the final recorded gap is within ± 0.05 eV of 1.4 eV.

An episode ends at an executed target hit or after seven attempts. Failed attempts leave the structure unchanged. Chemistry checks limit composition changes and check charge balance with exceptions (see the separate appendix).

Model	Target hits (SFT→RL)	Execution % (SFT→RL)
Qwen3-4B	111→148	64.1→55.3
Minitron-4B	121→135	65.8→59.9
Llama-3.2-3B	121→119	61.2→48.9
Phi-4-mini	118→104	59.8→74.7

Table 1: Target hits count materials out of 1,000 whose final recorded gap is in the target band. Execution is the percentage of attempted edits recorded as successfully executed.

All four models use the same 1,000 held-out materials, sampled equally from four starting-gap groups. In the primary comparison, SFT is followed by one 1,000-step run per model of Group Relative Policy Optimization (GRPO) (Shao et al. 2024), which compares rewards for responses to the same prompt. RL trains on single-edit transitions without attempt history; sequential editing is used only during evaluation.

The reward combines bonuses for passing checks with a band-gap term, u. For executed edits without fatal structural defects,

$$R = b_p \, \mathbf{1}_{\text{parse}} + b_e \, \mathbf{1}_{\text{exec}} + b_d \, \mathbf{1}_{\text{domain}} + u, \qquad (1)$$

where the indicators denote extracting a valid proposal (parsing), execution, and passing the charge and composition-change checks (domain acceptance). The band-gap term is positive only for progress toward the target and weights that progress by final proximity. We use $b_p = 0.2$, $b_e = 0.3$, and $b_d = 0.5$. The appendix gives the full formula, failure penalties, and structural checks.

The original band-gap reward changes smoothly with distance from the target. The Qwen follow-ups add explicit bonuses for reaching the target band (Progress–Success) or the target and nearby bands (Tiered Target), as shown in Figure 2. Tiered Target also reduces the bonuses for passing checks. The appendix gives the full formulas.

Results

Base models executed no edits; their 12 hits are unchanged materials already in the target band. A target hit therefore does not require an executed edit. The SFT target-hit rate is 11.1–12.1% (Table 1). Following RL, Qwen3-4B gained target hits on 128 materials and lost hits on 91, increasing success from 11.1% to 14.8%. This paired improvement was significant before correction (exact McNemar $p = 0.015$), but not at the 0.05 level after Holm correction across four model comparisons (adjusted $p = 0.059$).

Rescoring 16,000 fixed training edits gives reward–hit correlations of $r = 0.257$ for the original reward and $r = 0.596$ for normalized progress plus a hit bonus; this does not establish training gains. Evaluated Qwen follow-ups reach 156 and 157/1000 hits versus 148/1000 (Table 2).

Limitations. Post-edit target hits rely on the environment's property predictor; edited materials have not been independently validated for band gap or physical stability. Each evaluated model–reward setting has one training run, leaving variation across repeated runs unmeasured.

Model setting	Target hits	Hit rate
Base Qwen3-4B	12/1000	1.2%
SFT Qwen3-4B	111/1000	11.1%
RL Qwen3-4B (original reward)	148/1000	14.8%
RL Qwen3-4B + Progress–Success	156/1000	15.6%
RL Qwen3-4B + Tiered Target	157/1000	15.7%

Table 2: Qwen3-4B target hits under the original reward and two exploratory reward follow-ups.

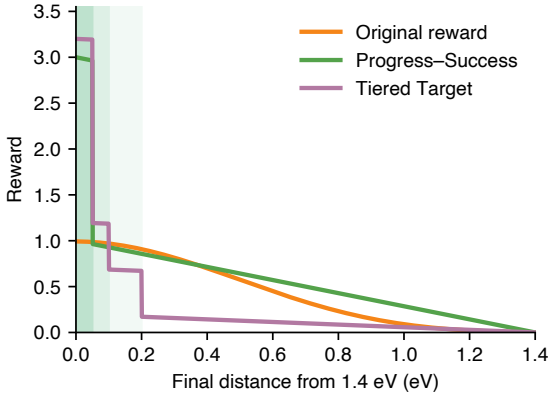


Figure 2: Band-gap reward terms versus final distance from 1.4 eV, with starting distance $d_s = 1.4$ eV and check bonuses omitted. Progress–Success jumps at 0.05 eV; Tiered Target steps at 0.05, 0.10, and 0.20 eV.

Takeaway

CreSTAL[1] makes LLMatDesign-style editing trainable using RLSF-inspired tool-generated feedback. Its validation records connect executable checks and predicted outcomes to RL rewards. Mixed gains across models and exploratory reward follow-ups motivate controlled tests of how reward design affects property targeting, with repeated training runs and independent property validation.

References

Fung, V.; Zhang, J.; Juarez, E.; and Sumpter, B. G. 2021. Benchmarking graph neural networks for materials chemistry. *npj Computational Materials*, 7(1): 84.

Jha, P.; Jana, P.; Suresh, P.; Arora, A.; and Ganesh, V. 2024. Rlsf: Fine-tuning llms via symbolic feedback. *arXiv preprint arXiv:2405.16661*.

Jia, S.; Zhang, C.; and Fung, V. 2024. LLMatDesign: Autonomous Materials Discovery with Large Language Models. arXiv:2406.13163.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

[1]Source code: https://github.com/jordanmadison/CreSTAL