Appendix

CreSTAL: An In-Silico Reinforcement Learning Environment for Targeted Materials Editing

A Methods and Reproducibility

A.1 The editing task and model inputs

CreSTAL trains a language model to propose edits and uses tools to apply and assess them. The model sees the crystal's formula, current property value, and objective; the environment retains its atomic coordinates and lattice vectors. During evaluation, the prompt also includes recent attempts.

This study targets band gap, the energy separation between valence and conduction bands. A target hit means a final recorded gap within 1.4 ± 0.05 eV. The target follows LLMatDesign's photovoltaic-motivated example (Jia, Zhang, and Fung 2024). The implementation also supports reducing formation energy, but that task is not evaluated here. Starting band gaps come from the dataset; after an executed edit, the predictor supplies the updated value. If no edit executes, the final recorded gap remains the starting dataset value. First-edit progress therefore compares a dataset value with a post-edit prediction.

The model proposes one of four edits (Table A1) and explains its choice in text called a hypothesis. The environment extracts the edit from the response, applies it, and checks the composition. It then relaxes the crystal—adjusting its geometry with a learned force model—and predicts the band gap with MatDeepLearn (Fung et al. 2021).

RL training rewards one proposed edit to a starting material. Evaluation allows up to seven successive attempts per material, updating the prompt after each attempt. An executed edit within the target band ends the episode; failed attempts leave the material unchanged. Target hits count materials whose final recorded gap is within the band.

Operator	Change before relaxation
`add`	Insert one atom of the selected element at a random position within the cell.
`remove`	Delete all atoms of the selected element.
`substitute`	Replace all atoms of the selected element with atoms of the proposed element.
`exchange`	Swap the two selected elements throughout the structure.

Table A1: How each edit changes the crystal in LLMatDesign. A single proposal can affect multiple atoms.

Composition checks. If the response cannot be parsed into an edit, the edit cannot be applied, or a chemistry check fails, the proposal stops before property prediction. The charge check tries to assign oxidation states whose charges balance. It accepts single-element compositions directly and permits metallic compositions when assignment fails. The configured relaxed mode also permits other failed assignments, so passing this check does not establish charge neutrality. A second check limits the change in composition, as defined in Section A.2.

Structural checks. Structural checks can reduce the reward without stopping the proposal. A steric clash—atoms placed too close together—or volume expansion above 20% removes the composition-check bonus and any positive band-gap reward. The separate stability check is disabled in these experiments.

What the model sees. The prompt contains a summary of up to five recent attempts: each proposed edit, its hypothesis, the resulting property value, and available changes from previous values. Detailed failure reasons remain in the logs and are not included in this summary (Table A2). The model is not asked to generate a separate reflection after each attempt. SFT and single-edit RL inputs contain no previous attempts.

Information	Visibility
Formula, current value, objective	Included in the model prompt.
Coordinates and lattice vectors	Retained by the environment only.
Previous attempts	Stored; the evaluation prompt includes a recent summary.
Detailed failure reasons	Logged; absent from the configured summary.

Table A2: Information shown to the model and retained by the environment.

Table A3 shows a representative first-step prompt, including the system instructions, material state, target, available edits, and required response format. SFT uses the same task and output structure, paired with a demonstrated response. RL prompts have no previous attempts. At evaluation, subsequent prompts replace the formula and property value with the current state and append the recent-attempt summary inside `<previous_attempts>` tags. Property values are displayed to three decimal places. The implementation also permits sending the accumulated conversation or omitting history; no controlled comparison of these modes is reported here.

A.2 Structured validation records and reward construction

For each proposed edit, CreSTAL records which checks ran, their results, and any predicted property change. This *validation record* also stores relaxation results and the reward components. The reward function uses the checks and outcomes to assign one numerical score for training. During evaluation, the next prompt receives a summary of previous attempts; detailed failure reasons remain in the record.

For example, a parsing failure receives a reward of -0.5 and has no post-edit property prediction. An unperformed or disabled check does not count as evidence of validation. Target success is assessed separately by checking whether the final recorded gap is within the target band.

System message

You are an expert material science assistant. Your goal is to design new materials with certain target properties by modifying their existing structure. Your response must always follow this XML output format:

`<hypothesis>` Your hypothesis for the modification `</hypothesis>`
`<modification>` Your modification in the following format: `[$TYPE, $ELEMENT_1, $ELEMENT_2]` `</modification>`.

Requirements:
1. `$HYPOTHESIS` should be your analysis and reason for choosing a modification.
2. `$TYPE` should be the modification type; one of "exchange", "substitute", "remove", "add".
3. `$ELEMENT` should be the selected element type to be modified. For "exchange" and "substitute", two `$ELEMENT` placeholders are needed. For "remove" and "add", one `$ELEMENT` placeholder is needed.

User message

I have a material and it's band gap, the minimum energy required to excite an electron that is stuck in its bound state into a free state where it can participate in conduction.

`(DyB12, 0.000)`

Please propose a modification to the material that results in a band gap of 1.400 eV. You can choose one of the four following modifications:

1. exchange: exchange two elements in the material
2. substitute: substitute one element in the material with another
3. remove: remove an element from the material
4. add: add an element to the material

Take a deep breath and work on this problem step-by-step. Your thoughtful and detailed analysis is highly appreciated.

Table A3: Representative first-step prompt reconstructed from the prompt template using a logged starting state: DyB_{12} (material ID `mp-1103476`), with a starting band gap of 0.000 eV and target 1.400 eV. Line breaks and punctuation are normalized for display; system-message placeholders define the required output format.

RLSF uses tool-generated diagnostic certificates to assign feedback to parts of a response (Jha et al. 2024). CreSTAL's RLSF-inspired feedback uses checks and outcomes to score the response as a whole. The reported experiments use scalar rewards, not feedback assigned to individual tokens.

Failure penalties and outcome bonuses. If parsing, modification, or chemistry fails before a property prediction is available, the reward is the corresponding penalty in Table A4. For an executed edit without a steric clash or fatal volume expansion, the outcome reward is

$$r = b_p \mathbf{1}[\text{parse}] + b_e \mathbf{1}[\text{execute}] + b_d \mathbf{1}[\text{domain}] + u. \quad \text{(A1)}$$

The composition-check bonus, labeled "domain" in the equation, requires passing both the charge check and a limit on composition change. To measure that change, the environment represents each composition by numerical summaries of its elements (Magpie features). It compares the before/after feature vectors using their variation and correlations in the training data (Mahalanobis distance), with a threshold of 26.

Condition or component	Value
Parsing failure (early stop)	-0.5
Modification failure (early stop)	-1.0
Chemistry failure (early stop)	-1.0
Parsing bonus, b_p	0.2
Execution bonus, b_e	0.3
Composition-check bonus, b_d	0.5

Table A4: Original reward coefficients. Early-stop penalties replace the outcome reward; bonuses contribute to Eq. (A1).

Band-gap reward term. Let d_s and d_f be the absolute distances of the starting and post-edit band gaps from 1.4 eV. With distances in eV,

$$u = \tanh\left(\frac{\max(0, d_s - d_f)}{0.5}\right) \exp\left(-\frac{d_f^2}{2(0.5)^2}\right). \quad \text{(A2)}$$

The first factor rewards movement toward the target; the second favors final values closer to it. The term is positive only for progress. For a fixed starting distance, closer final values receive more reward.

Reward after relaxation. An edit can still receive reward if relaxation reaches its iteration limit before converging. We score the returned structure using the same composition checks, structural checks, and predicted band gap.

Reward follow-ups. The two Qwen follow-ups add explicit bonuses for ending within specified distances of the target. With distances in eV, normalized progress is

$$p = \text{clip}\left(\frac{d_s - d_f}{\max(d_s, 0.1)}, -1, 1\right). \quad \text{(A3)}$$

This measures the fraction of the starting error removed, limits the result to $[-1, 1]$, and uses a 0.1 eV floor to avoid division by very small errors. Progress–Success replaces u with

$$u_{\text{PS}} = p + 2\mathbf{1}[d_f \le 0.05]. \quad \text{(A4)}$$

Tiered Target replaces it with

$$u_{\text{TT}} = 0.2p + \begin{cases} 3, & d_f \le 0.05, \\ 1, & 0.05 < d_f \le 0.10, \\ 0.5, & 0.10 < d_f \le 0.20, \\ 0, & d_f > 0.20. \end{cases} \quad \text{(A5)}$$

Only one distance-band bonus applies. Progress–Success retains the original check bonuses. Tiered Target reduces the

parsing, execution, and composition-check bonuses to 0.04, 0.06, and 0.10. Thus, it changes more than the band-gap reward term. Both follow-ups are exploratory, single-run experiments designed after examining the original reward. Tiered Target has checkpoint evaluation results, but its training-attempt logs are absent from the supplied archive.

A.3 Data construction and model training

Data for demonstrations and RL. SFT learns from examples of proposed edits and their reasoning; RL evaluates new proposals on a separate set of starting materials. Table A5 lists the stored material sets for SFT, RL, and evaluation. No material ID appears in more than one of these sets.

The candidate-edit dataset contains 2,614,404 records of proposed edits, starting formulas and properties, and resulting formulas and properties. The demonstration-generation procedure adds reasoning text to edit records using DeepSeek-R1-Distill-Llama-70B.

Stored material set	Unique IDs
SFT source materials	8,000
RL source materials	19,136
Held-out evaluation sample	1,000

Table A5: Separate material sets for SFT, RL, and evaluation. These counts refer to materials, not demonstrated edits or training steps.

The SFT dataset contains 13,085 demonstrations: 6,377 for band gap and 6,708 for formation energy, drawn from 7,394 of the 8,000 SFT source materials. Each supplies a material description, a proposed edit, and generated reasoning for that edit. These form single-response examples: a system instruction, a user request, and one demonstrated model response. RL and evaluation in this study focus on band-gap targeting.

SFT data preparation randomly splits these demonstrations by row (80%/10%/10%, seed 42), yielding 10,468 training, 1,308 validation, and 1,309 test examples.

The RL material set contains 770 records without a usable structure. Removing them leaves 18,366 input examples, matching the count in Qwen's training log. This is a dataset size, not a count of training steps.

From individual edits to successive attempts. RL begins with the SFT model and rewards individual proposals made from starting-material states, without previous-edit history. Evaluation then uses that model to make successive attempts on the updated material. This tests whether learning from individual edits improves targeting over an episode.

Training on full episodes would require sampling groups of complete editing sequences and deciding how their rewards apply to individual responses. It would also require repeated tool calls and handling longer histories. The reported experiments use single-edit training; they do not compare it with training on full episodes.

Training settings. The base models are Qwen3-4B-Instruct-2507, Llama-3.2-3B, Phi-4-mini-reasoning, and Minitron-4B-Base. SFT model snapshots (checkpoints) are selected by loss on validation demonstrations, independently of target hits. Multiple examples are packed into each training sequence. Each selected model then receives 1,000 RL training steps using the Dr. GRPO objective (Shao et al. 2024; Liu et al. 2025). GRPO compares rewards for multiple responses to the same starting-material prompt. Training uses low-rank adaptation (LoRA) (Hu et al. 2022); Table A6 gives its rank and scaling along with the other shared settings.

RL setting	Value
Training steps	1,000
Checkpoint interval	100 steps
Learning rate	5×10^{-6}
KL coefficient, β	0.08
Completions per group	32
Batch size	16
LoRA rank / scaling	32 / 64

Table A6: Shared training settings for the reported RL runs.

Models	Additional recorded settings
Phi, Minitron	Token-level importance-sampling weights capped at 10; Liger fused loss disabled.
Qwen, Llama	Shared TRL library defaults.
Minitron	Fused AdamW optimizer; bfloat16 arithmetic.

Table A7: Implementation settings. Minitron's optimizer setting was not independently established for all four runs.

Evaluated checkpoints. Table A8 identifies the selected SFT checkpoints and the final RL checkpoints. All four RL runs use the scalar reward in Section A.2 and the shared settings in Table A6. Intermediate evaluations are available for Qwen at RL step 200 on 944 materials and Llama at step 300 on 500 materials; the primary SFT and RL-1000 comparisons each use the full 1,000-material sample.

Model	SFT checkpoint	Final RL step
Qwen3-4B	174	1,000
Llama-3.2-3B	174	1,000
Phi-4-mini	172	1,000
Minitron-4B	174	1,000

Table A8: Checkpoints used for the primary comparisons. SFT checkpoint numbers identify weights selected by validation loss; RL checkpoints are evaluated after 1,000 training steps.

A.4 Evaluation and reproducibility protocol

Each SFT and final RL checkpoint is evaluated on the same 1,000 held-out materials, with 250 drawn from each starting-gap group (Table A9). Each material receives one episode of

at most seven attempts. The primary result counts materials ending within the target band. The balanced sample gives the four groups equal representation. Where specified, diagnostic plots instead weight the groups by their proportions in the held-out pool; those percentages need not equal the primary hit count divided by 1,000.

Group	Starting gap (eV)	Pool	Sample
Metal	$= 0$	5,542	250
Narrow	$(0, 1]$	954	250
Moderate	$(1, 3]$	1,377	250
Wide	> 3	807	250
Total		8,680	1,000

Table A9: Starting-gap groups and counts from the evaluation manifest (sampling seed 42).

Generation setting	Value
Temperature	0.6
Top-p	0.9
Top-k	50
Repetition penalty	1.15
Maximum new tokens	1,024

Table A10: Generation settings recorded in the shared run configuration. These fields were not independently recorded in every evaluation worker's metadata.

Scope of the evidence. Each evaluated model–reward setting has one training run, so the results do not measure variation across repeated training runs. Post-edit target hits rely on the property predictor; the edited materials have not been independently validated for band gap or physical stability. The hypotheses are model-generated reasoning, not verified explanations. Changes in hit counts and edit choices describe the evaluated runs; they do not establish what caused an improvement or whether the model learned to plan across multiple edits.

B Additional Results

Executed edits and target hits. These measures answer different questions. Execution rate is the fraction of attempted edits recorded as executed. Target-hit rate is the fraction of starting materials whose final recorded gap is within the target band. Table A11 gives both denominators.

The execution counts come from the log fields `successful_modifications` and `total_steps`; they are not an independent check of every execution. Target hits count final recorded gaps in the target band, including unchanged materials that started there. Thus, the base models' 12 hits with zero executed edits are not newly achieved hits.

Paired evaluation. We use McNemar's test because SFT and RL hit/miss outcomes are paired by material, and Holm correction to account for the four model comparisons. For Qwen, both checkpoints hit on 20 materials and both miss on 761; RL gains hits on 128 materials and loses hits on 91. The net gain is 37/1,000, or 3.7 percentage points. McNemar's test evaluates the imbalance between gains and losses: under the null hypothesis, the two directions are equally likely among pairs whose outcomes differ.

Exact two-sided tests give Qwen $p = 0.0148$, Minitron $p = 0.376$, Llama $p = 0.943$, and Phi $p = 0.341$. Testing four models creates multiple opportunities for a false positive; Holm adjustment controls the probability of any false rejection across this family of tests. It gives adjusted $p = 0.059$ for Qwen and $p = 1.000$ for the other models. Thus, Qwen's observed gain meets the 0.05 significance threshold before correction but not after correction. This does not establish equal performance, and the adjustment does not change the observed effect size. These tests concern material-level outcomes for the evaluated checkpoints; they do not measure variation across independent training runs.

Reward follow-ups. Two Qwen follow-ups evaluated on the same 1,000 materials and target produced 156/1000 hits for Progress–Success and 157/1000 for Tiered Target, compared with 148/1000 for the original reward. Their execution rates were 62.7% $(4060/6479)$ and 60.9% $(3927/6443)$, respectively, using the same logged execution counter. Each follow-up has one training run.

Offline reward rescoring. We applied two reward rules to the same 16,000 logged training edits to check how closely their scores aligned with target hits. Pearson correlation with the hit indicator was $r = 0.257$ for the original reward and $r = 0.596$ for normalized progress plus a hit bonus. These are new scores for fixed edits, not results from another trained model. The follow-up hit counts above come from evaluations of trained checkpoints.

Specialized crystal generators. For context, CDVAE yields 142 target hits that also pass geometry checks among 999 starting materials, using the best of ten candidates per material (Xie et al. 2021). DiffCSP yields 169 target hits that pass geometry checks among 15,000 candidates generated without a starting material (Jiao et al. 2023). These budgets and evaluation units differ from sequential editing and do not support a matched ranking.

C Target Hits by Starting Material and Attempt

These plots examine which starting-gap groups reach the target and when hits occur during an episode. Each SFT and RL-1000 evaluation uses the same 1,000 held-out materials; Qwen RL-200 has 944 completed episodes. Counts were reconstructed from attempt histories and checked against episode summaries.

D Distance from the Target

The histograms show where final recorded gaps fall. The tolerance curves show how many materials count as hits as the allowed distance from 1.4 eV increases. Together they distinguish changes near the target from changes farther away.

Model	Stage	Hits	Executed/attempted	Execution rate
Qwen3-4B	SFT	111/1000	4229/6600	64.1%
	RL-1000	148/1000	3558/6432	55.3%
Minitron-4B	SFT	121/1000	4346/6607	65.8%
	RL-1000	135/1000	3923/6551	59.9%
Llama-3.2-3B	SFT	121/1000	4023/6579	61.2%
	RL-1000	119/1000	3230/6600	48.9%
Phi-4-mini	SFT	118/1000	3985/6662	59.8%
	RL-1000	104/1000	4988/6674	74.7%

Table A11: Held-out evaluation counts reconstructed from episode logs. Action rates use attempted steps as the denominator; hits use starting materials.

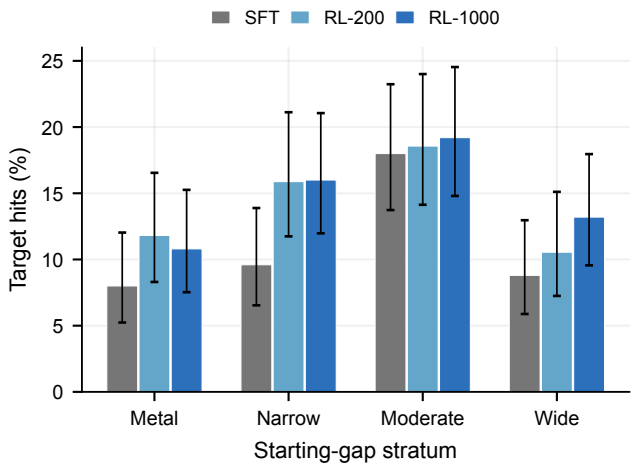


Figure A1: Qwen target hits by starting-gap group, with Wilson 95% confidence intervals. SFT and RL-1000 each have 250 materials per group; RL-200 has 237, 233, 237, and 237 (metal, narrow, moderate, wide). Each bar uses the number of completed materials in that group.

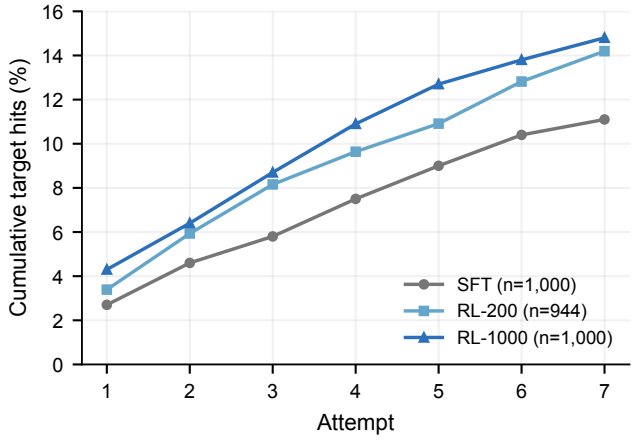


Figure A2: Qwen cumulative target hits over seven attempts. Final counts are 111/1,000 (SFT), 134/944 (RL-200), and 148/1,000 (RL-1000). A hit is first recorded when an executed edit reaches the target band.

Reward	Hits	≤ 0.10 eV	Median d_f
Original	148/1000	176/1000	0.634
Progress–Success	156/1000	190/1000	0.562
Tiered Target	157/1000	188/1000	0.588

Table A12: Qwen reward follow-ups. The second outcome counts materials within 0.10 eV of the target; the last gives the median final distance in eV. The primary measure remains the ± 0.05 eV target-hit count.

All plots use the same 1,000 materials per SFT/RL-1000 evaluation. Histograms count those materials equally. Tolerance curves weight the starting-gap groups by their proportions in the held-out pool, so their percentages need not equal the main text's hit counts divided by 1,000.

E Changes in Model Outputs and Edit Choices

The training curves show how the model's outputs change during RL. KL divergence measures departure from the reference model's output probabilities. Entropy measures how spread out the choices are: lower operator entropy indicates greater concentration on a few edit types. The later figures examine proposed edits during held-out evaluation.

KL and available token entropy each have 100 recorded points at training steps 10–1,000; operator entropy has 500 at steps 0–990. Smoothing averages neighboring recorded points without filling missing values. The corresponding log fields are `train/kl`, `train/entropy`, and `policy/action_entropy_norm`.

The element plot compares which elements the models propose to introduce at the final RL checkpoint. It is a comparison between models, not a record of how element preferences change during training.

The next figure shows how often Qwen proposes each of the four edit types at different checkpoints. SFT and RL-1000 use 1,000 completed episodes each; RL-200 uses 944. Both edit-choice plots count proposals that could be parsed into edits, including those that later failed execution.

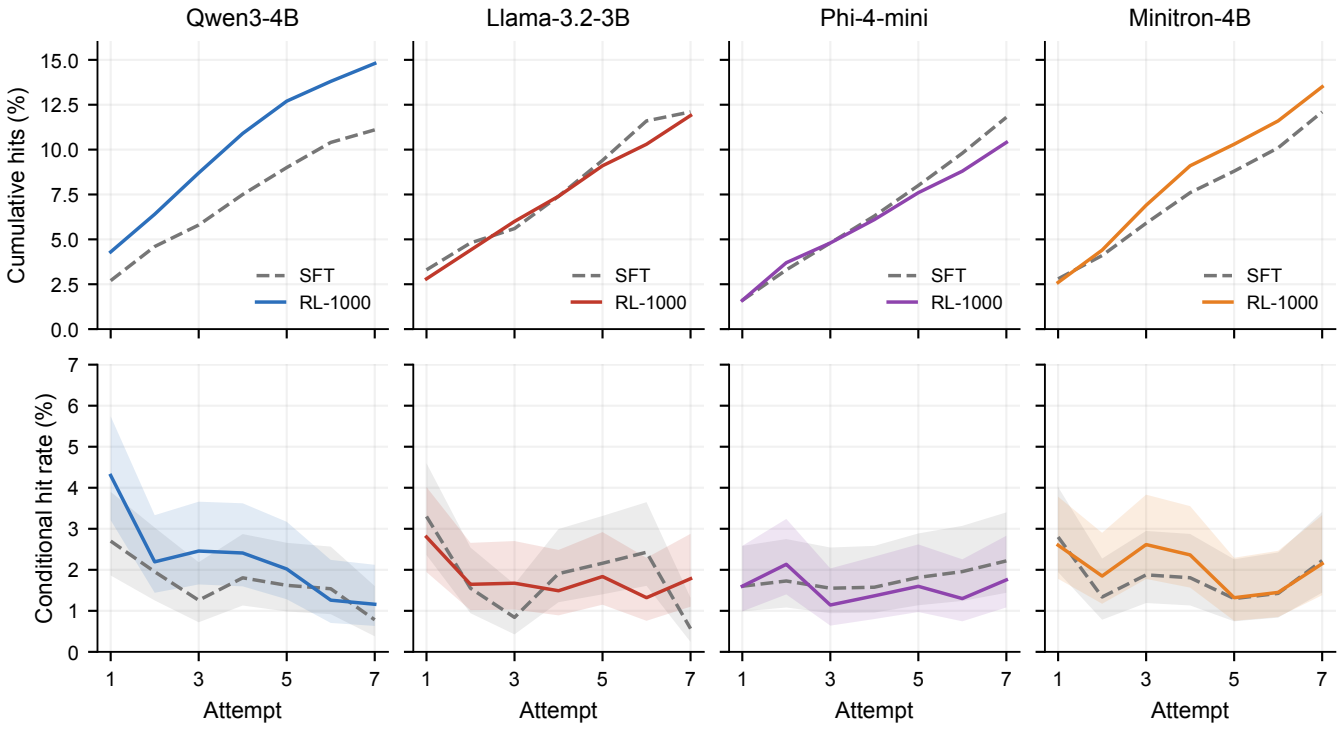


Figure A3: Cumulative target hits (top) and conditional hit rates (bottom) for SFT and RL-1000, with 1,000 materials per run. Conditional rates divide new hits at each attempt by materials without an earlier hit; shading shows Wilson 95% confidence intervals.

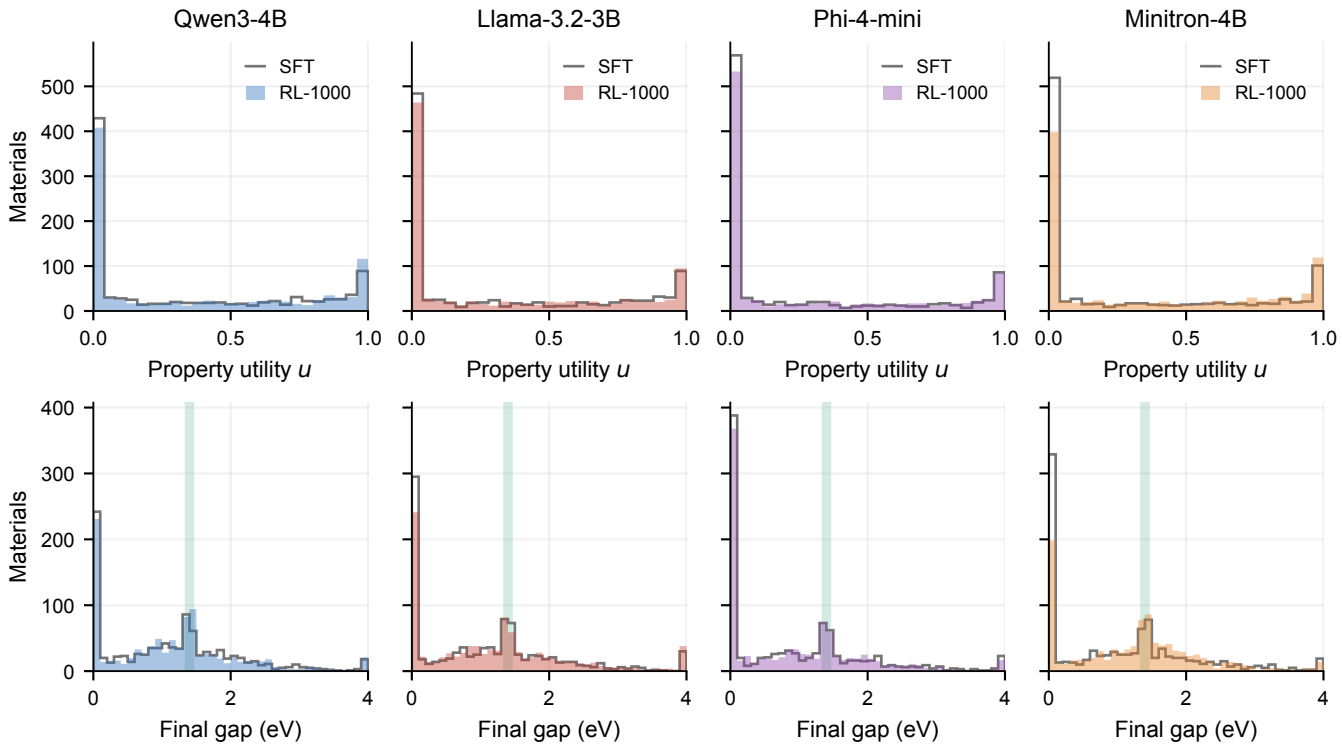


Figure A4: Band-gap reward term u (top, labeled property utility) and final predicted gaps (bottom), with each material counted equally. Here u is recomputed from the dataset starting gap and final value using Eq. (A2). It excludes check bonuses and structural reductions, so it is not the training reward for an individual edit. Histograms use 25 bins for u over $[0, 1]$ and 40 gap bins over $[0, 4]$ eV, clipping gaps to that range for display only. Green marks the 1.4 ± 0.05 eV target band.

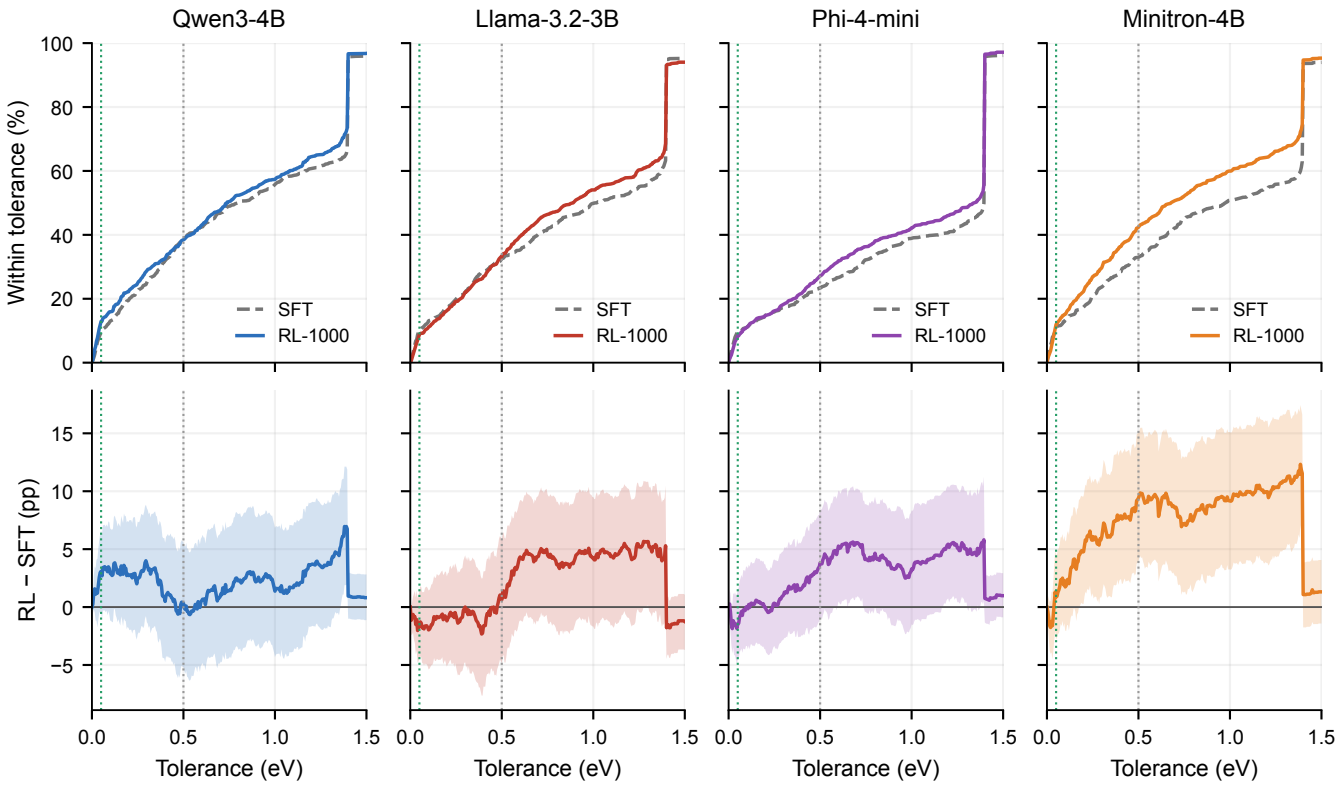


Figure A5: Fraction within tolerance of 1.4 eV (top) and RL-1000 minus SFT in percentage points (bottom), weighted by the starting-gap groups' proportions in the held-out pool. SFT and RL are paired on the same 1,000 materials for each model. The grid runs from 0 to 1.5 eV in 0.005 eV increments, including exactly 0.05 eV (green); gray marks the 0.5 eV scale in the band-gap reward term. Shading shows 95% intervals from 2,000 bootstrap resamples (seed 0), preserving SFT/RL pairs and starting-gap groups. Intervals apply at each tolerance separately, not simultaneously across the whole curve.

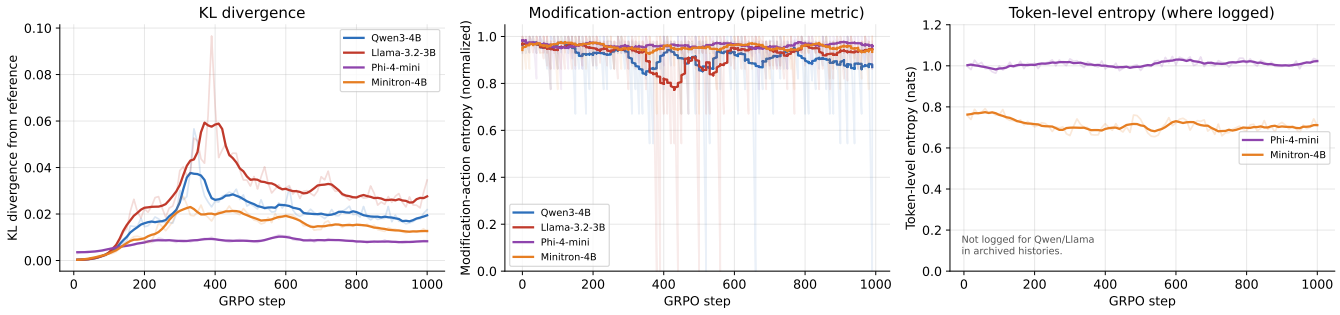


Figure A6: Logged KL divergence, normalized action entropy, and token entropy for the four RL runs. Faint curves show raw values; solid curves use centered moving averages over 7, 25, and 7 logged points, respectively. Token entropy was logged only for Phi and Minitron.

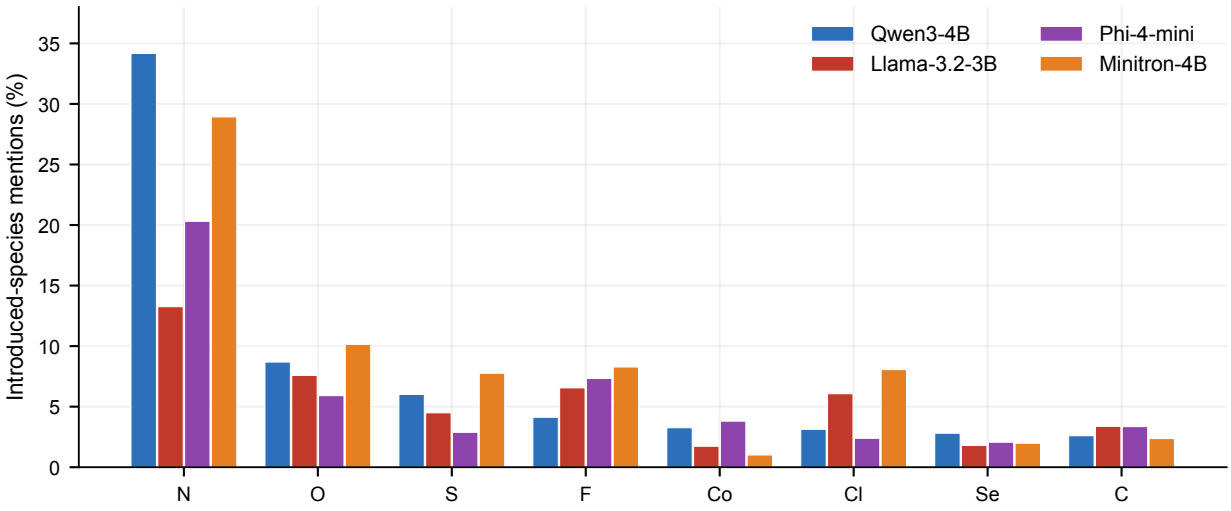


Figure A7: Elements selected in parsed RL-1000 edit proposals. Each `add` or `substitute` proposal contributes one mention of its introduced element, regardless of execution success or atoms affected. Percentages use all such mentions for that model; the eight most frequent Qwen elements are displayed for all models.

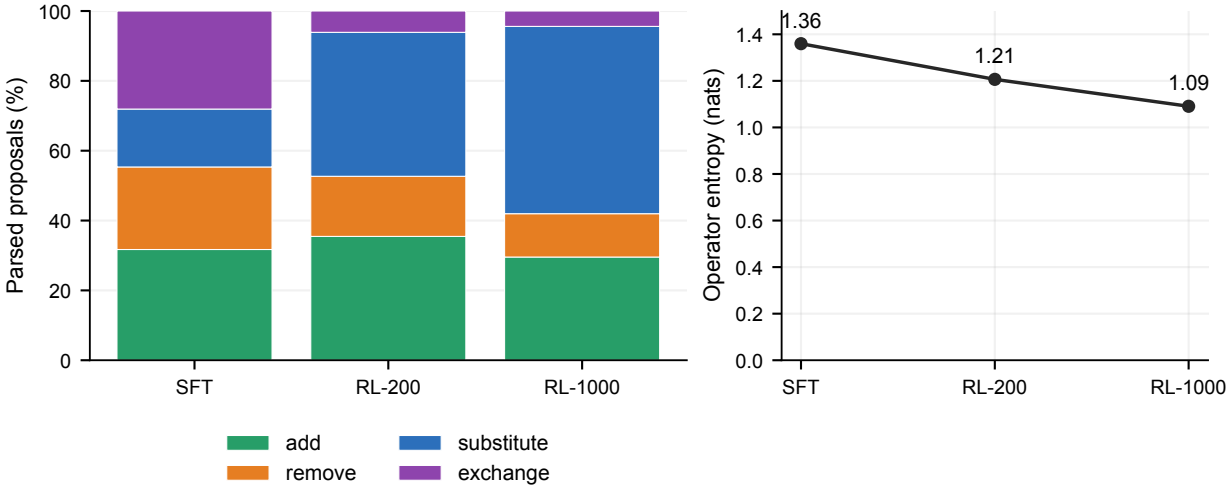


Figure A8: Qwen's choice of edit type (left) and the spread across those four choices, measured by Shannon entropy in nats (right). Both panels use the same counts of parsed proposals, including those that failed execution.

F Related Work and Scope of the Contribution

Executable materials interactions. LLMatDesign uses language-model agents to interpret instructions, modify materials, and evaluate the resulting candidates with tools, including self-reflection over previous decisions (Jia, Zhang, and Fung 2024). It applies pretrained models to materials-design tasks without task-specific training. CreSTAL makes this style of materials editing trainable through supervised demonstrations and reinforcement learning from executable outcomes. It turns demonstrated edits into training examples and uses tool checks and predicted outcomes to reward new proposals. The trained models are then evaluated over successive editing attempts.

RLSF and structured validation records. RLSF obtains diagnostic certificates from symbolic tools and maps them to feedback to particular tokens or parts of a response (Jha et al. 2024). These certificates include practical diagnostics such as compiler errors and chemical-parser feedback. CreSTAL uses RLSF-inspired tool-generated feedback for crystal editing through the validation record defined in Section A.2, combining programmatic checks with learned relaxation and property estimates.

In the reported experiments, the checks and outcomes determine one scalar reward per completion for GRPO (Shao et al. 2024). Section A.2 explains how the validation record supports this score. Assigning feedback to individual parts of a response remains an open experiment.

Specialized crystal generators. CDVAE combines a variational autoencoder with diffusion-based crystal generation and supports property optimization (Xie et al. 2021). DiffCSP jointly models lattice parameters and fractional atomic coordinates with equivariant diffusion (Jiao et al. 2023). Our CDVAE comparison uses targeted generation from starting materials, whereas our DiffCSP comparison uses samples generated without a starting material. These are experimental configurations of the cited methods, not descriptions of their complete capabilities. The methods generate different numbers of candidates, receive different inputs, and count success over different units. Their results therefore do not provide a direct ranking against the editing models.

Questions left open. Three questions follow from this study. Does showing the model detailed failure reasons help it choose better edits? Does training on complete episodes improve targeting compared with training on individual edits? Do the gains persist across repeated training runs and independent property predictors? These experiments would test which parts of the feedback and training procedure matter for targeting.

Appendix 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.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

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.

Jiao, R.; Huang, W.; Lin, P.; Han, J.; Chen, P.; Lu, Y.; and Liu, Y. 2023. Crystal Structure Prediction by Joint Equivariant Diffusion. *arXiv preprint arXiv:2309.04475*.

Liu, Z.; Chen, C.; Li, W.; Qi, P.; Pang, T.; Du, C.; Lee, W. S.; and Lin, M. 2025. Understanding R1-Zero-Like Training: A Critical Perspective. *arXiv preprint arXiv:2503.20783*.

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*.

Xie, T.; Fu, X.; Ganea, O.-E.; Barzilay, R.; and Jaakkola, T. 2021. Crystal Diffusion Variational Autoencoder for Periodic Material Generation. *arXiv preprint arXiv:2110.06197*.