MEGA: A Large-Scale Molecular Editing Dataset for Guided-Action Optimization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large language models show strong potential for molecular editing, but progress has been constrained by the limited scale and quality of available training data. To address this, we introduce MEGA, a large-scale dataset of 31.4 million molecule pairs, where each pair represents a single property-improving chemical edit annotated with an explicit action: *Replace*, *Insert*, or *Delete*. We demonstrate MEGA's utility in a controlled supervised fine-tuning (SFT) setting, where a model trained on MEGA outperforms models trained on existing datasets by up to +21.47 percentage points in hit ratio. Furthermore, we show that Group Relative Policy Optimization (GRPO) post-training with a similarity-aware reward achieves state-of-the-art performance and a remarkable $\sim 36\times$ improvement in data efficiency, while also preserving edit locality. We release MEGA in open access to the community to enable data-centric benchmarks and accelerate progress in molecular editing with generative models.

1 Introduction

Molecular optimization is critical to drug discovery, guiding chemists in turning initial molecular hits into drug-like candidates. Unlike unconstrained molecule generation [1, 2], molecular editing involves targeted modifications, such as scaffold decoration, fragment substitutions, or precise structural refinements, that carefully balance therapeutic properties with chemical feasibility and synthetic practicality [3, 4].

To assist chemists in this iterative lead optimization process, recent approaches leverage large language models (LLMs), either through fine-tuning or by using them as reasoning agents capable of interpreting textual prompts (e.g. "increase solubility") and proposing relevant molecular edits [5, 6]. Additionally, reinforcement learning (RL)-based post-training can align these models even more closely with practical constraints, improving both chemical plausibility and edit precision [7, 8]. Progress, however, is limited by data. Training and evaluating editing models requires goal-aligned edit datasets that pair a parent molecule with a proposed child and standardized outcomes, at a scale that supports both supervised fine-tuning and post-training [9, 10]. Nevertheless, existing corpora either lack the scale required for robust training or omit explicit edit annotations needed for guided policy learning.

To close this gap, we curate MEGA (Molecular Editing with Guided Action): a large-scale, molecule editing dataset composed of (parent, child) molecule pairs spanning 28 tasks. It contains 31.4M successful edits and a compute-friendly subset, MEGA-Small, with 522k positive samples. We also release an additional 41M valid and chemically close negative examples to enable contrastive learning and RL reward shaping [11, 12].

Submitted to in the AI for Science workshop (NeurIPS 2025).

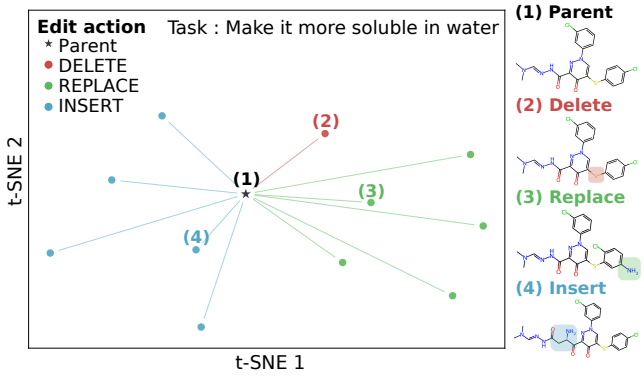


Figure 1: Morgan-fingerprint t-SNE for a parent SMILES and child molecules generated by fragment edits, delete, replace, insert. Colors encode the applied edit, highlighting neighborhood exploration under the given task.

Using a fixed LLM and a shared evaluation protocol, we first quantify the effect of data alone by fine-tuning on MEGA-Small versus other public datasets. We then show that post-training with GRPO [13], using a composite reward that combines a thresholded property gain term and a Tanimoto similarity term [14], yields further gains with reduced number of training samples.

Concretely, this work introduces the following contributions:

1. We release MEGA, a 31.4M-pair molecular editing dataset with fragment-level *Replace*, *Insert*, and *Delete* annotations across 28 property optimization tasks, alongside MEGA-Small (522k pairs) for compute-limited settings. MEGA is over an order of magnitude larger than any existing dataset for molecular editing.

2. We demonstrate that under fixed model and training protocol, fine-tuning on MEGA-Small subset increases hit ratios by up to +21.47 percentage points over established datasets on shared tasks, while its explicit edit labels enable per-action supervision and diagnostics.

3. We show that RL post-training on MEGA-Small with a similarity-aware reward further improves property alignment and edit minimality, sets strong performance on established benchmarks, and delivers large improvements in data efficiency. With only 14k training examples, GRPO matches the SFT model trained on the full 522k MEGA-Small set, corresponding to a $\sim 36\times$ improvement in data efficiency.

2 Related Work

2.1 Datasets for Molecular Editing

Public corpora vary in task formulation and scale. MoleculeSTM [15] trains a multimodal structure–text model on hundreds of thousands of molecule–caption pairs through contrastive learning and proposes instruction-guided retrieval and editing tasks, establishing a text-based benchmark for property-aware modification. Another example is MolOpt-Instructions [16], released alongside DrugAssist, which compiles a large instruction dataset to fine-tune language models for molecule optimization from natural language goals. Furthermore, MolEdit-Instruct [17] scales property-conditioned edits by pairing each parent molecule with an explicit edit instruction and target property change. The dataset is used to evaluate diffusion and RL models under joint constraints on molecular similarity and property improvement, reflecting a shift toward instruction-plus-constraint benchmarks. Together, these datasets illustrate the available range for training and evaluating molecular editing models, despite differences in construction, supervision signals, and scale.

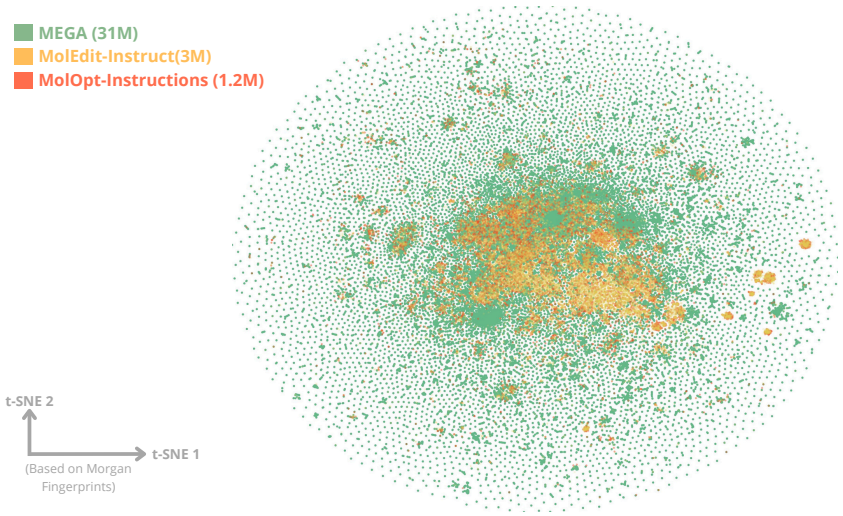


Figure 2: t-SNE projection of Morgan fingerprints showing chemical space coverage for MEGA (31M), MolEdit-Instruct (3M), and MolOpt-Instructions (1.2M).

2.2 LLMs for Chemistry

General-purpose language models trained on broad text data already exhibit useful zero-shot chemistry skills answering property prediction questions, translating line notations, or suggesting functional-group swaps straight out of the box [18, 19, 20]. When wrapped in a tool-calling framework, the same models can act as agents: ChemCrow, for example, prompts an off-the-shelf LLM to invoke cheminformatics utilities (parsers, property predictors, similarity search) and carry out multi-step design tasks from natural language instructions [21].

Researchers also adapt these open language models to chemistry via domain fine-tuning and task-specific supervision. For instance, LlamoLe trains on $\sim$128k USPTO reactions with textual descriptions to strengthen reasoning and route identification [22, 23], while DrugAssist uses MolOpt-Instructions to instruction-tune models for property-directed optimization from text in a single-shot fashion [16].

A further layer of refinement uses reinforcement learning such as with Ether0, trained on 640k experimentally-grounded chemistry problems across 375 tasks, to excel at tasks like retrosynthesis and solubility editing [24]. Another example is MolEditRL, which pairs property-conditioned prompts with structure-preserving edit operators and reinforcement-style objectives to promote local, similarity-respecting modifications [17].

2.3 Editors Beyond LLMs for Lead Optimization

While LLM-based editors are comparatively recent, lead optimization has a long history of non-LLM approaches that emphasize local, property-directed modifications to a given scaffold. Earlier rule-based strategies, such as matched molecular pairs (MMPs) [25] and fixed reaction templates, encoded medicinal-chemistry heuristics for systematic substitution. More recent machine learning methods operate directly on strings or graphs to propose minimal edits, including JT-VAE [2], GCPN [26], and MARS [27, 28]. In parallel, diffusion models adapt continuous generative dynamics to discrete molecular modifications: DiffLink designs linkers between fixed fragments [29], while DiffHop performs constrained scaffold hopping [30]. Taken together, these approaches chart a progression from rules to learned editors to diffusion frameworks, all aimed at controllable, chemically plausible edits central to lead optimization.

Table 1: Comparison of molecular editing datasets used in this study. Reported sizes count only successful (positive) parent-child pairs. Unique molecules counts distinct SMILES across both parents and children. Action provided indicates whether a dataset records the edit label.

Dataset	Size	Unique Molecules	# Tasks	Action Provided
MoleculeSTM	280K	250K	34	No
MolEdit-Instruct	3.03M	967K	20	No
MolOpt-Instructions	1.24M	1.596M	16	No
MEGA-Small	522K	372K	28	Yes
MEGA	31.4M	22.126M	28	Yes

3 MEGA Dataset

3.1 Dataset Construction Overview

MEGA contains 31.4 million parent–child SMILES pairs, where each child comes from applying a single functional-group edit to a ZINC250K parent, without a constraint to preserve the scaffold [31]. Candidate modification sites are located with established retrosynthetic slicing rules (BRICS [32], Hussain–Rea (HR) [33] and RECAP [34]) and exactly one action is applied at a chosen site: *Delete*, *Insert*, or *Replace* a functional group. The child is rebuilt and sanitized in RDKit [35], and task properties are computed deterministically. We adopt the MoleculeSTM protocol for task labeling: for each objective (e.g. "increase solubility"), we use RDKit to verify whether the child clears the threshold for that task. The computational budget for MEGA amounted to approximately 184k CPU-hours on a 128-core cluster.

Each record includes parent SMILES, child SMILES, a coarse action tag (*Insert*/*Delete*/*Replace*), the task identifier and threshold level, and the parent/child property vectors. For efficient training, we also release MEGA-Small (522k positives), drawn uniformly from MEGA, which mirrors the full set's action distribution (14% *Delete*, 39% *Insert*, 46% *Replace*). In addition to the positives, we also release 41 million valid, chemically close negative pairs. While they fail to meet the threshold, they are useful as hard negatives for contrastive or RL setups.

To emphasize drug discovery relevance, our tasks target widely used objectives—aqueous solubility, drug-likeness (QED), H-bond donors/acceptors, permeability proxies, and topological polar surface area (TPSA)—each evaluated at two thresholds. Restricting edits to a single modification per pair enables controlled exploration of the parent's local chemical neighborhood. A parent molecule may appear in multiple pairs if it contains eligible sites for several actions across tasks. For each edit–task combination, we retain up to five successful and five near-miss children, ranked to maximize diversity while avoiding redundancy. Further details on tasks and dataset composition are provided in Appendix A.

3.2 Dataset Coverage

Figure 1 shows a representative parent alongside three children, one per action. The edits are local and chemically rational: removing an atom (*Delete*), adding a small moiety (*Insert*) or swapping one group for another (*Replace*). Together they illustrate the targeted nature of MEGA's pairs; in this example, all children satisfy the "increase aqueous solubility" objective.

Figure 2 visualizes a statistically significant subset of MEGA in the 2048-bit Morgan-fingerprint space [36] using t-SNE [37]. The overlay shows that MEGA occupies the shared high-density core with existing molecular editing datasets and also reaches beyond it, consistent with its scale and edit policy. Moreover, Table 1 quantifies this comparison: in terms of successful (positive) edits, MEGA is roughly an order of magnitude larger than the next largest dataset. Furthermore, unlike other datasets, MEGA and MEGA-Small include a coarse action label (*Insert*/*Delete*/*Replace*) for every pair, supporting per-action supervision, diagnostics, and reproducibility.

Table 2: Performance comparison of SFT models on shared molecular editing tasks. We report the mean and std of five random seeds. The best results are marked in bold. Llama 3.

Task	Description	Threshold	Dataset		
			MEGA-Small	MolEdit Instruct	MolOpt Instruction
103	*More like a drug*	0.0	**62.46 ± 2.18**	23.92 ± 0.99	16.38 ± 2.03
		0.1	**28.43 ± 1.38**	12.85 ± 0.58	8.38 ± 0.53
104	*Less like a drug*	0.0	97.81 ± 0.91	**98.97 ± 0.33**	96.87 ± 0.82
		0.1	83.94 ± 3.43	**98.86 ± 0.51**	94.43 ± 1.47
107	*More H-bond acceptors*	0.0	**99.28 ± 0.25**	94.96 ± 1.70	89.33 ± 1.18
		1.0	**93.06 ± 0.66**	43.35 ± 1.65	34.06 ± 0.58
108	*More H-bond donors*	0.0	**99.80 ± 0.25**	97.66 ± 0.59	96.21 ± 0.91
		1.0	**99.29 ± 0.25**	67.57 ± 1.78	56.67 ± 1.10
Average			**83.01**	67.27	61.54

4 Experiments

We evaluate MEGA in a two-stage protocol: (1) supervised fine-tuning (SFT) to benchmark performance under identical model and training settings against existing datasets, and (2) RL post-training with a hybrid reward combining property gains and structural similarity. We also analyze edit action distributions, locality, and sample efficiency in single- and multi-objective tasks.

4.1 Supervised Fine-Tuning

Protocol. We fine-tune a Llama-3 8B model [38] with LoRA adapters [39] on MolOpt Instruction, MolEdit Instruct, and MEGA-Small datasets. All runs use the same hyperparameters, training schedule, and LoRA configuration. Training last approximately 23 A100-equivalent hours per model until the validation loss no longer improves.

Evaluation follows the MoleculeSTM protocol [15] and is restricted to the 4 single-objective tasks shared by all three datasets. The test set contains 200 unique parent SMILES not present in any of the training sets. For each task, we assess performance at two property thresholds (loose and strict) and report the hit ratio, defined as the fraction of generated molecules that achieve the required property improvement. Each experiment is repeated five times, with a decoding temperature of 1.0, and we report the mean and standard deviation of the hit ratio across runs. For further comparisons and training settings details see Appendix B.

Results. Table 2 shows that the LLM trained on MEGA-Small outperforms the same architecture trained on MolEdit Instruct and MolOpt Instruction by +15.74 (pp) and +21.47 (pp), respectively. The largest gain occurs in the "more like a drug" objective, a target known to be particularly challenging due to its composite nature [18]. Variance is low and comparable to the other benchmarks, indicating that improvements are stable across repeated evaluations.

4.2 Reward-Guided Post-Training

Protocol. We further refine the best MEGA-Small SFT checkpoint using Group Relative Policy Optimization (GRPO) [13] to improve property alignment while preserving local edits. During training, for each parent SMILES, the model generates a batch of multiple candidates, which are scored relative to each other. This feedback is used for updating the model weights. The scalar reward is defined as:

$$R = \underbrace{1[\Delta p(\text{parent}, \text{child}) \geq \tau]}_{\text{property hit}} + \gamma \underbrace{1_{\text{valid}}(\text{child})}_{\text{validity hit}} + \lambda \underbrace{h_{\text{tan}}(\text{parent}, \text{child})}_{\text{Tanimoto hit level}}$$

Table 3: Comparison on DrugAssist benchmark. MEGA-Small GRPO (522k) outperforms DrugAssist and Gemini 2.5 Pro across five shared tasks under loose and strict thresholds.

Task	Description	Threshold	Model		
			DrugAssist	**Gemini 2.5 Pro**	**MEGA-Small GRPO**
101	*More soluble in water*	0	80.00	82.23	**97.49**
		0.5	41.00	59.45	**91.10**
103	*More like a drug*	0	76.00	60.14	**83.49**
		0.1	**63.00**	23.46	50.00
107	*More H-bond acceptors*	0	71.00	64.97	**98.60**
		1	67.00	5.57	**86.74**
108	*More H-bond donors*	0	72.00	73.54	**99.31**
		1	76.00	6.32	**91.45**
201	*More soluble & more HBA*	0 - 0	50.00	80.32	**95.19**
		0.5 - 1	27.00	24.43	**84.21**
Average			62.30	48.05	**87.76**

$$h_{\tan}(\text{parent}, \text{child}) = \begin{cases} 1.0, & \text{if } T > 0.65, \\ 0.5, & \text{if } 0.4 \leq T \leq 0.65, \\ 0.0, & \text{otherwise}, \end{cases}$$

where the first term awards a hit when the property change Δp meets or exceeds the task threshold τ, the second term rewards valid and sanitized child smiles, and the third rewards scaffold-local modifications via Tanimoto coefficient discretization. The coefficients γ and λ were selected empirically to 1.0. We train with 3,000 rollouts per task under a KL-constrained objective. To assess the data efficiency of the post-training stage, we repeat this experiment with training sets ranging from 1.4k parent SMILES up to the full MEGA-Small dataset (522k). The resulting models are referred to as MEGA-Small GRPO. Complete experimental details are provided in Appendix C.

We first compare MEGA-Small GRPO against DrugAssist [17], a state-of-the-art specialized LLM, and Gemini 2.5 Pro [40], a strong general-purpose LLM, on five single- and multi-objective molecular editing tasks. For this evaluation, we use the 500-SMILES test set provided by DrugAssist and report hit ratios under both loose and strict thresholds in Table 3.

We then compare MEGA-Small GRPO against ChatDrug Turbo, a strong in-context learning LLM, and MoleculeSTM, a contrastive-trained encoder–decoder, on the full 28-task suite of the MEGA dataset. For this evaluation, we follow the protocol described in the SFT section and report results in Table 4. We verified that none of the test SMILES appeared in our training data to maintain evaluation integrity.

Results. MEGA-Small GRPO outperforms both DrugAssist and Gemini 2.5 Pro on the DrugAssist benchmark (Table 3), achieving the highest hit ratio in 9 of 10 settings. The most pronounced gains appear on the dual-objective solubility + HBA task (201), where it reaches 95.19% under loose and 84.21% under strict thresholds, substantially ahead of both baselines. The only case where MEGA-Small GRPO underperforms is the strict drug-likeness objective, where DrugAssist retains an edge. Gemini 2.5 Pro consistently trails, particularly under strict thresholds, underscoring the difficulty of zero-shot general-purpose LLMs in molecular editing.

On the 28-task MoleculeSTM benchmark (Table 4), MEGA-Small GRPO attains the best mean hit ratio on all task/threshold pairs. It reaches $\geq$95% on most single-property edits under loose thresholds (e.g., 101–102, 104, 106–108) and remains strong under stricter criteria. The notable hard case is Task 103 (drug-likeness), where absolute rates drop for all methods; even so, MEGA-Small GRPO leads by 14 pp (62.60 vs. 48.65) at loose and 7 pp (26.75 vs. 19.37) at strict. MEGA-Small GRPO's advantage is most pronounced on multi-objective tasks (201, 203, and 206), indicating better balancing of potentially competing constraints. Variance across runs is small (typically ≤ 1.5),

Table 4: Performance comparison of MEGA-Small GRPO (522K) against editing methods across single and multi objective tasks and thresholds. We report the mean and standard deviation over five runs. The best results are shown in bold.

Task	Threshold	Random	MoleculeSTM	ChatDrug Turbo	MEGA-Small GRPO
101	0	35.33±1.31	61.87±2.67	94.13±1.04	**99.31±0.10**
	0.5	11.04±2.40	49.02±1.84	88.67±0.95	**94.43±0.24**
102	0	43.36±3.06	52.71±1.67	96.86±1.10	**99.71±0.21**
	0.5	19.75±1.56	30.47±3.26	70.08±3.44	**95.52±0.51**
103	0	38.06±2.57	36.52±2.46	48.65±3.39	**62.60±2.41**
	0.1	5.27±0.24	8.81±0.82	19.37±5.54	**26.75±1.64**
104	0	36.96±2.25	58.59±1.01	70.75±2.92	**97.55±0.64**
	0.1	6.16±1.87	37.56±1.76	30.99±2.66	**93.63±0.56**
105	0	25.23±2.13	57.74±0.60	56.56±1.84	**90.19±1.34**
	10	17.41±1.43	47.51±1.88	43.08±2.95	**87.88±0.94**
106	0	16.79±2.54	34.13±0.59	77.35±1.98	**100.00±0.00**
	10	11.02±0.71	26.48±0.97	66.69±2.74	**99.43±0.01**
107	0	12.64±1.64	54.01±5.26	95.35±0.62	**99.86±0.29**
	1	0.69±0.01	27.33±2.62	72.60±2.51	**92.35±0.50**
108	0	2.97±0.61	28.55±0.76	96.54±1.31	**98.45±0.83**
	1	0.00±0.00	7.69±0.56	76.43±3.32	**95.22±0.34**
201	0 – 0	9.88±1.03	27.87±3.86	79.62±0.64	**98.53±0.44**
	0.5 – 1	0.23±0.33	8.80±0.04	49.64±2.66	**90.34±0.47**
202	0 – 0	2.99±0.38	8.55±2.75	51.59±3.79	**97.24±0.92**
	0.5 – 1	0.45±0.32	2.93±0.30	24.92±4.85	**92.04±0.53**
203	0 – 0	2.28±1.15	33.51±4.08	89.34±0.96	**99.64±0.48**
	0.5 – 1	0.00±0.00	9.98±1.03	53.64±5.81	**98.35±0.90**
204	0 – 0	0.69±0.58	17.03±2.75	39.90±3.86	**92.60±1.44**
	0.5 – 1	0.00±0.00	2.59±1.14	24.19±2.19	**60.06±1.83**
205	0 – 0	5.06±1.21	35.69±3.19	12.85±2.68	**89.30±0.93**
	0.5 – 10	1.16±0.68	19.15±0.73	10.44±5.75	**82.86±0.75**
206	0 – 0	12.17±1.05	44.35±0.68	65.33±2.16	**99.54±0.43**
	0.5 – 10	6.20±0.64	28.67±2.22	52.90±2.23	**94.31±0.23**

suggesting the gains are stable across several runs. Overall, MEGA-Small GRPO establishes a robust state-of-the-art baseline for both single- and multi-objective molecular editing. These outcomes reflect the synergy between the MEGA-Small dataset and locality-aware GRPO training. MEGA-Small provides informative and diverse demonstrations of guided optimization through single-local edits, while GRPO further aligns the model's behavior with task-specific reward signals.

Data Efficiency. Figure 3 shows that GRPO with Tanimoto reward outperforms SFT across all data regimes while maintaining scaffold edits within our targeted Tanimoto similarity range (0.6–0.8). With only 14k training examples, MEGA-Small GRPO (14K) matches the performance of MEGA-Small SFT trained on 522k by +2.11 pp, achieving $\sim 36\times$ data efficiency multiplier with the same Llama 3 base model. More details in Appendix C.

Guided-Action Editing. Figure 4 shows the distribution of fragment-level edit actions across tasks. Models trained with MEGA-Small SFT roughly reproduce the action distribution of the MEGA-Small (522k) dataset. This indicates internalization of single-fragment edit patterns (replace, insert, delete) present in the demonstrations. In contrast, MEGA-Small GRPO learns, via RL, heavily favors *replace* actions, reflecting an optimization bias towards minimal yet property-aligned functional

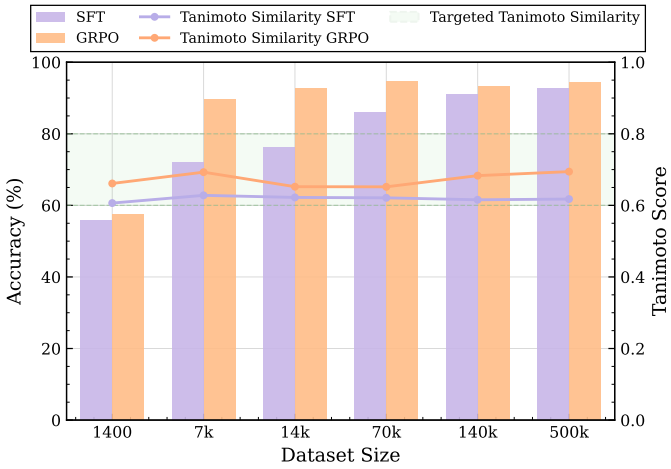


Figure 3: Data efficiency comparison of SFT and GRPO across training set sizes (based on loose threshold). GRPO consistently outperforms SFT while keeping edits within the targeted Tanimoto similarity range (0.6–0.8). Remarkably, MEGA-Small GRPO trained on only 14k examples matches SFT trained on the full 522k dataset, demonstrating a $\sim$36$\times$ improvement in data efficiency.

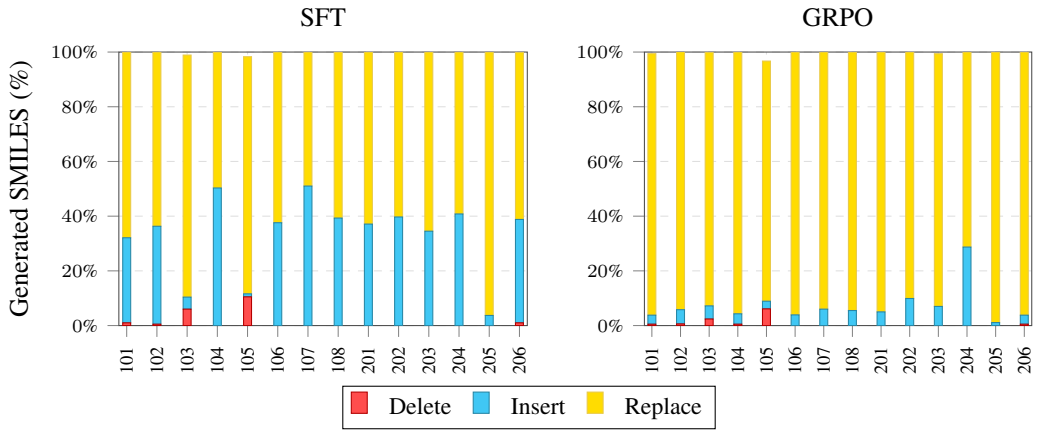


Figure 4: Distribution of fragment-level edit actions across tasks using the 200 held-out test SMILES. SFT models replicate the dataset's mixture of replace, insert, and delete operations, whereas GRPO strongly favors replace actions. This shift reflects an RL-driven bias toward minimal, property-aligned substitutions, which align with the improved performance reported in earlier results.

group modifications. The performance increase of the GRPO model, suggest that replace-dominant strategies yield, on average, better results than the dataset's action distribution.

5 Conclusion

In this work, we introduce MEGA, a new large-scale dataset of 31.4 million molecule pairs designed to advance property-guided molecular editing. By systematically generating single chemically rational edits that improves a target property (*replace*, *insert*, *delete*), MEGA provides dense, high-quality supervision for exploring local chemical space. Our experiments demonstrate its value: a model fine-tuned on a small subset, MEGA-Small, significantly outperforms models trained on existing datasets in supervised settings. Furthermore, when combined with reinforcement learning, models trained on MEGA achieves state-of-the-art performance on established benchmarks and demonstrates a remarkable $\sim$ 36$\times$ improvement in data efficiency. By providing controlled, high-quality examples at scale, MEGA facilitates the development of better models and optimization workflows.

References

[1] Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018. PMID: 29532027.

[2] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation, 2019.

[3] Seonghwan Seo, Jaechang Lim, and Woo Youn Kim. Molecular generative model via retrosynthetically prepared chemical building block assembly. *Adv Sci (Weinh)*, 10(8):e2206674, January 2023.

[4] Shao Jinsong, Jia Qifeng, Chen Xing, Yajie Hao, and Li Wang. Molecular fragmentation as a crucial step in the AI-based drug development pathway. *Communications Chemistry*, 7(1):20, February 2024.

[5] Yoel Zimmermann, Adib Bazgir, Zartashia Afzal, Fariha Agbere, Qianxiang Ai, Nawaf Alampara, Alexander Al-Feghali, Mehrad Ansari, Dmytro Antypov, Amro Aswad, Jiaru Bai, Viktoriia Baibakova, Devi Dutta Biswajeet, Erik Bitzek, Joshua D. Bocarsly, Anna Borisova, Andres M Bran, L. Catherine Brinson, Marcel Moran Calderon, Alessandro Canalicchio, Victor Chen, Yuan Chiang, Defne Circi, Benjamin Charmes, Vikrant Chaudhary, Zizhang Chen, Min-Hsueh Chiu, Judith Clymo, Kedar Dabhadkar, Nathan Daelman, Archit Datar, Wibe A. de Jong, Matthew L. Evans, Maryam Ghazizade Fard, Giuseppe Fisicaro, Abhijeet Sadashiv Gangan, Janine George, Jose D. Cojal Gonzalez, Michael Götte, Ankur K. Gupta, Hassan Harb, Pengyu Hong, Abdelrahman Ibrahim, Ahmed Ilyas, Alishba Imran, Kevin Ishimwe, Ramsey Issa, Kevin Maik Jablonka, Colin Jones, Tyler R. Josephson, Greg Juhasz, Sarthak Kapoor, Rongda Kang, Ghazal Khalighinejad, Sartaaj Khan, Sascha Klawohn, Suneel Kuman, Alvin Noe Ladines, Sarom Leang, Magdalena Lederbauer, Sheng-Lun, Liao, Hao Liu, Xuefeng Liu, Stanley Lo, Sandeep Madireddy, Piyush Ranjan Maharana, Shagun Maheshwari, Soroush Mahjoubi, José A. Márquez, Rob Mills, Trupti Mohanty, Bernadette Mohr, Seyed Mohamad Moosavi, Alexander Moßhammer, Amirhossein D. Naghdi, Aakash Naik, Oleksandr Narykov, Hampus Näsström, Xuan Vu Nguyen, Xinyi Ni, Dana O'Connor, Teslim Olayiwola, Federico Ottomano, Aleyna Beste Ozhan, Sebastian Pagel, Chiku Parida, Jaehee Park, Vraj Patel, Elena Patyukova, Martin Hoffmann Petersen, Luis Pinto, José M. Pizarro, Dieter Plessers, Tapashree Pradhan, Utkarsh Pratiush, Charishma Puli, Andrew Qin, Mahyar Rajabi, Francesco Ricci, Elliot Risch, Martiño Ríos-García, Aritra Roy, Tehseen Rug, Hasan M Sayeed, Markus Scheidgen, Mara Schilling-Wilhelmi, Marcel Schloz, Fabian Schöppach, Julia Schumann, Philippe Schwaller, Marcus Schwarting, Samiha Sharlin, Kevin Shen, Jiale Shi, Pradip Si, Jennifer D'Souza, Taylor Sparks, Suraj Sudhakar, Leopold Talirz, Dandan Tang, Olga Taran, Carla Terboven, Mark Tropin, Anastasiia Tsymbal, Katharina Ueltzen, Pablo Andres Unzueta, Archit Vasan, Tirtha Vinchurkar, Trung Vo, Gabriel Vogel, Christoph Völker, Jan Weinreich, Faradawn Yang, Mohd Zaki, Chi Zhang, Sylvester Zhang, Weijie Zhang, Ruijie Zhu, Shang Zhu, Jan Janssen, Calvin Li, Ian Foster, and Ben Blaiszik. Reflections from the 2024 large language model (llm) hackathon for applications in materials science and chemistry, 2025.

[6] Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, Mehrdad Asgari, Juliane Eberhardt, Amir Mohammad Elahi, Hani M Elbeheiry, María Victoria Gil, Christina Glaubitz, Maximilian Greiner, Caroline T Holick, Tim Hoffmann, Abdelrahman Ibrahim, Lea C Klepsch, Yannik Köster, Fabian Alexander Kreth, Jakob Meyer, Santiago Miret, Jan Matthias Peschel, Michael Ringleb, Nicole C Roesner, Johanna Schreiber, Ulrich S Schubert, Leanne M Stafast, A D Dinga Wonanke, Michael Pieler, Philippe Schwaller, and Kevin Maik Jablonka. A framework for evaluating the chemical knowledge and reasoning abilities of large language models against the expertise of chemists. *Nature Chemistry*, 17(7):1027–1034, July 2025.

[7] Weichen Dai, Zijie Dai, Zhijie Huang, Yixuan Pan, Xinhe Li, Xi Li, Yi Zhou, Ji Qi, and Wu Jiang. Rldbf: Enhancing llms via reinforcement learning with database feedback, 2025.

[8] Xuefeng Liu, Songhao Jiang, Siyu Chen, Zhuoran Yang, Yuxin Chen, Ian Foster, and Rick Stevens. Drugimprovergpt: A large language model for drug optimization with fine-tuning via structured policy optimization, 2025.

[9] Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alán Aspuru-Guzik, and Alex Zhavoronkov. Molecular sets (MOSES): A benchmarking platform for molecular generation models. *Front Pharmacol*, 11:565644, December 2020.

[10] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W. Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development, 2021.

[11] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples, 2021.

[12] Wei Shen, Xiaoying Zhang, Yuanshun Yao, Rui Zheng, Hongyi Guo, and Yang Liu. Improving reinforcement learning from human feedback using contrastive rewards, 2024.

[13] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

[14] Dávid Bajusz, Anita Rácz, and Károly Héberger. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of cheminformatics*, 7(1):20, 2015.

[15] Shengchao Liu, Weili Nie, Chengpeng Wang, Jiarui Lu, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao, and Anima Anandkumar. Multi-modal molecule structure-text model for text-based retrieval and editing. *Nature Machine Intelligence*, 5(12):1447–1457, 2023.

[16] Geyan Ye, Xibao Cai, Houtim Lai, Xing Wang, Junhong Huang, Longyue Wang, Wei Liu, and Xiangxiang Zeng. Drugassist: a large language model for molecule optimization. *Briefings in Bioinformatics*, 26(1):bbae693, 01 2025.

[17] Yuanxin Zhuang, Dazhong Shen, and Ying Sun. Moleditrl: Structure-preserving molecular editing via discrete diffusion and reinforcement learning, 2025.

[18] Shengchao Liu, Jiongxiao Wang, Yijin Yang, Chengpeng Wang, Ling Liu, Hongyu Guo, and Chaowei Xiao. Chatgpt-powered conversational drug editing using retrieval and domain feedback, 2023.

[19] Andres M Bran, Theo A Neukomm, Daniel P Armstrong, Zlatko Jončev, and Philippe Schwaller. Chemical reasoning in llms unlocks strategy-aware synthesis planning and reaction mechanism elucidation, 2025.

[20] Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, Mehrdad Asgari, Juliane Eberhardt, Amir Mohammad Elahi, Hani M Elbeheiry, María Victoria Gil, Christina Glaubitz, Maximilian Greiner, Caroline T Holick, Tim Hoffmann, Abdelrahman Ibrahim, Lea C Klepsch, Yannik Köster, Fabian Alexander Kreth, Jakob Meyer, Santiago Miret, Jan Matthias Peschel, Michael Ringleb, Nicole C Roesner, Johanna Schreiber, Ulrich S Schubert, Leanne M Stafast, A D Dinga Wonanke, Michael Pieler, Philippe Schwaller, and Kevin Maik Jablonka. A framework for evaluating the chemical knowledge and reasoning abilities of large language models against the expertise of chemists. *Nature Chemistry*, 17(7):1027–1034, July 2025.

[21] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5):525–535, May 2024.

[22] Daniel Lowe. Chemical reactions from US patents (1976-Sep2016). 6 2017.

[23] Gang Liu, Michael Sun, Wojciech Matusik, Meng Jiang, and Jie Chen. Multimodal large language models for inverse molecular design with retrosynthetic planning, 2024.

[24] Siddharth M. Narayanan, James D. Braza, Ryan-Rhys Griffiths, Albert Bou, Geemi Wellawatte, Mayk Caldas Ramos, Ludovico Mitchener, Samuel G. Rodriques, and Andrew D. White. Training a scientific reasoning model for chemistry, 2025.

[25] Ziyi Yang, Shaohua Shi, Li Fu, Aiping Lu, Tingjun Hou, and Dongsheng Cao. Matched molecular pair analysis in drug discovery: methods and recent applications. *Journal of Medicinal Chemistry*, 66(7):4361–4377, 2023.

[26] Jiaxuan You, Bowen Liu, Rex Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation, 2019.

[27] Yutong Xie, Chence Shi, Hao Zhou, Yuwei Yang, Weinan Zhang, Yong Yu, and Lei Li. Mars: Markov molecular sampling for multi-objective drug discovery, 2021.

[28] Hannes H Loeffler, Jiazhen He, Alessandro Tibo, Jon Paul Janet, Alexey Voronov, Lewis H Mervin, and Ola Engkvist. Reinvent 4: Modern AI–driven generative molecule design. *Journal of Cheminformatics*, 16(1):20, February 2024.

[29] Ilia Igashov, Hannes Stärk, Clément Vignac, Arne Schneuing, Victor Garcia Satorras, Pascal Frossard, Max Welling, Michael Bronstein, and Bruno Correia. Equivariant 3d-conditional diffusion model for molecular linker design. *Nature Machine Intelligence*, 6(4):417–427, April 2024.

[30] Jos Torge, Charles Harris, Simon V. Mathis, and Pietro Lio. Diffhopp: A graph diffusion model for novel drug design via scaffold hopping, 2023.

[31] John J Irwin and Brian K Shoichet. ZINC–a free database of commercially available compounds for virtual screening. *J Chem Inf Model*, 45(1):177–182, January 2005.

[32] Jörg Degen, Christof Wegscheid-Gerlach, Andrea Zaliani, and Matthias Rarey. On the art of compiling and using 'drug-like' chemical fragment spaces. *ChemMedChem*, 3(10):1503–1507, October 2008.

[33] Jameed Hussain and Ceara Rea. Computationally efficient algorithm to identify matched molecular pairs (mmps) in large data sets. *Journal of Chemical Information and Modeling*, 50(3):339–348, 2010. PMID: 20121045.

[34] X Q Lewell, D B Judd, S P Watson, and M M Hann. RECAP–retrosynthetic combinatorial analysis procedure: a powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry. *J Chem Inf Comput Sci*, 38(3):511–522, May 1998.

[35] Greg Landrum and RDKit Contributors. RDKit: Open-source cheminformatics, 2025. This concept DOI covers all RDKit versions on Zenodo and always resolves to the latest release.

[36] H. L. Morgan. The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service. *Journal of Chemical Documentation*, 5(2):107–113, 1965.

[37] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[38] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

[39] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[40] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[41] Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Gunter Klambauer. Fréchet chemnet distance: a metric for generative models for molecules in drug discovery. *Journal of chemical information and modeling*, 58(9):1736–1741, 2018.

[42] Tagir Akhmetshin, Arkadii I. Lin, Daniyar Mazitov, Evgenii Ziaikin, Timur Madzhidov, and Alexandre Varnek. ZINC 250K data sets. 12 2021.

A MEGA Dataset Details

Tasks. For curating MEGA we used single-objective tasks (101–108) that targets one property, and multi-objective tasks (201–206) for two properties. Table 5 lists the desired direction of change (↑ increase, ↓ decrease), variable name (consistent with RDKit), alongside the requirement in natural-language. For each task we evaluate 2 threshold with different levels of property change. Table 6 gives the evaluation thresholds under *loose* and *strict* criteria. For multi-objective tasks, each threshold vector follows the property order in the *Target(s)* column.

Task ID	Target(s)	Task Requirement 1	Task Requirement 2
101	$\downarrow \log P$	more soluble in water	None
102	$\uparrow \log P$	less soluble in water	None
103	$\uparrow$ QED	more like a drug	None
104	$\downarrow$ QED	less like a drug	None
105	$\downarrow$ TPSA	higher permeability	None
106	$\uparrow$ TPSA	lower permeability	None
107	$\uparrow$ HBA	more hydrogen bond acceptors	None
108	$\uparrow$ HBD	more hydrogen bond donors	None
201	$\downarrow \log P, \uparrow$ HBA	more soluble in water	more hydrogen bond acceptors
202	$\uparrow \log P, \uparrow$ HBA	less soluble in water	more hydrogen bond acceptors
203	$\downarrow \log P, \uparrow$ HBD	more soluble in water	more hydrogen bond donors
204	$\uparrow \log P, \uparrow$ HBD	less soluble in water	more hydrogen bond donors
205	$\downarrow \log P, \downarrow$ TPSA	more soluble in water	higher permeability
206	$\downarrow \log P, \uparrow$ TPSA	more soluble in water	lower permeability

Table 5: Task catalog for small-molecule property edits. All tasks require the output molecule to remain similar to the input. Arrows indicate desired property direction.

Task ID	Loose	Strict
101	`[0]`	`[0.5]`
102	`[0]`	`[0.5]`
103	`[0]`	`[0.1]`
104	`[0]`	`[0.1]`
105	`[0]`	`[10]`
106	`[0]`	`[10]`
107	`[0]`	`[1]`
108	`[0]`	`[1]`
201	`[0, 0]`	`[0.5, 1]`
202	`[0, 0]`	`[0.5, 1]`
203	`[0, 0]`	`[0.5, 1]`
204	`[0, 0]`	`[0.5, 1]`
205	`[0, 0]`	`[0.5, 10]`
206	`[0, 0]`	`[0.5, 10]`

Table 6: Evaluation thresholds per task. For multi-objective tasks, each vector's order follows the *Target(s)* order in Table 5.

Dataset Statistics. This subsection summarizes the scale and composition of MEGA (31M) and MEGA-Small (522K) and quantifies how representative the smaller split is of the full corpus. Table 7 reports dataset-level counts. MEGA-31M contains 246,532 unique parent molecules directly taken from the Zinc-250 dataset. In includes $72{,}366{,}584$ evaluated edits, of which $31{,}354{,}522$ are successful. MEGA-Small mirrors this profile at smaller scale with 4,105 unique parents and $1{,}205{,}430$ edits, including $522{,}058$.

Metric	MEGA (31M)	MEGA-Small (522K)
Unique parent molecules	246,532	4,105
Successful edits	31,354,522	522,058
Unique successful SMILES	21,879,431	367,954
Negative edits	41,012,062	683,372
Unique negative SMILES	8,129,138	137,012
Total SMILES	72,366,584	1,205,430

Table 7: Side-by-side summary of MEGA datasets.

Table 8 compares the distribution of successful edits by operation. The proportions are stable across scales: `delete` $\approx$3.1%, `insert` $\approx$43.6%, and `replace` $\approx$53.3% in both MEGA-Small (522K) and MEGA (31M). This alignment suggests that MEGA-Small preserves the operational mix of the full dataset and is suitable for compute-friendly budgets.

	MEGA-Small (522K)		MEGA (31M)	
Operation	**Count**	**%**	**Count**	**%**
delete	15,924	3.1%	960,992	3.1%
insert	227,789	43.6%	13,677,420	43.6%
replace	278,345	53.3%	16,716,110	53.3%
Total	522,058	100%	31,354,522	100%

Table 8: Distribution of successful edit operations for MEGA and MEGA-Small.

Table 9 reports successful edits per task for MEGA (31M) and MEGA-Small (522K). Counts are broadly balanced across tasks and per-task ranking is consistent across scales. Tasks 101/102/104 yield the largest winner pools, while 103 (increase QED) and 205 (reduce $\log P$& decrease TPSA) show markedly consistent with results from the literature. MEGA-Small preserves the relative task difficulty profile of the full corpus.

Task	MEGA	MEGA-Small
101	2,613,794	43,463
102	2,609,126	43,443
103	1,061,168	17,774
104	2,570,496	42,793
105	1,645,706	27,401
106	2,462,800	41,005
107	2,462,791	41,005
108	2,462,781	41,005
201	2,462,711	41,005
202	2,457,965	40,933
203	2,462,768	41,005
204	2,400,936	39,978
205	1,218,686	20,243
206	2,462,794	41,005
Total	**31,354,522**	**522,058**

Table 9: Number of successful edit examples per task for MEGA (31M) and MEGA-Small (522K).

Mean shifts, Table 10, align with the instructions for every task. Examples: *LogP*$\downarrow$ (101) moves the mean by -0.975 (winners vs. parents) and separates winners from losers by -1.577; *LogP*$\uparrow$ (102)

shifts by $+0.965$ with a winner–loser gap of $+1.133$; *QED*↓ (104) shifts by -0.217; *TPSA*↑ (106) exhibits a large increase of $+31.611$; *HBA*↑ (107) and *HBD*↑ (108) increase by $+2.749$ and $+2.316$, respectively. The consistent sign and sizable winner–loser separations (last column) provide evidence of strong task-wise consistency on MEGA-Small.

Task	Property	Obj.	Parent $\bar{x}$	Winner $\bar{x}$	Δ W–P	Loser $\bar{x}$	Δ W–L
101	LogP	↓	2.475	1.501	-0.975	3.078	-1.577
102	LogP	↑	2.475	3.440	$+0.965$	2.307	$+1.133$
103	QED	↑	0.733	0.797	$+0.064$	0.614	$+0.183$
104	QED	↓	0.733	0.516	-0.217	0.727	-0.211
105	TPSA	↓	64.918	49.669	-15.249	77.022	-27.353
106	TPSA	↑	64.918	96.530	$+31.611$	61.857	$+34.673$
107	HBA	↑	3.990	6.739	$+2.749$	4.224	$+2.515$
108	HBD	↑	1.237	3.553	$+2.316$	1.248	$+2.305$

Table 10: MEGA-Small: mean target-property values and deltas. $\Delta_{\text{W–P}} = \bar{x}_{\text{W}} - \bar{x}_{\text{P}}$ (winners minus parents) and $\Delta_{\text{W–L}} = \bar{x}_{\text{W}} - \bar{x}_{\text{L}}$ (winners minus losers). "Winners" and "losers" correspond to successful and unsuccessful edits, on strict threshold respectively. Signs follow the task objective (increase/decrease).

Figure 5 visualizes the single-objective shifts via kernel density estimates of the target property for parent (orange) and edited child (blue) molecules. Across all eight tasks, the child distribution moves in the instructed direction (reduce/increase or count increase), demonstrating strong task-wise consistency in MEGA-Small.

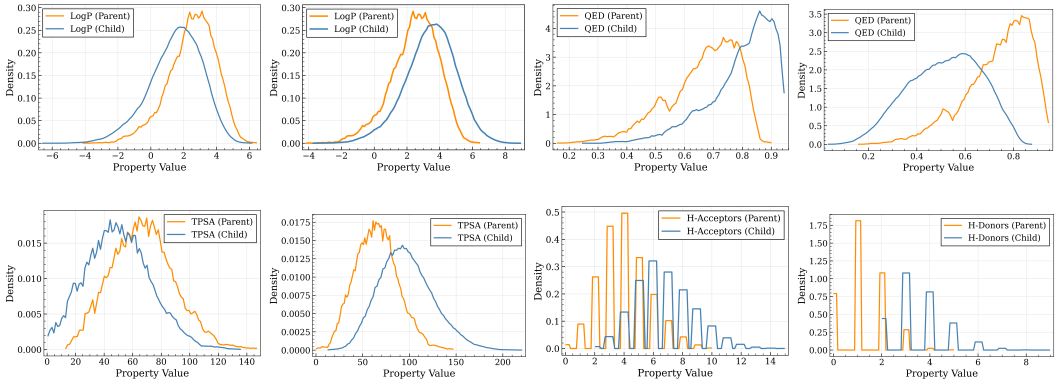


Figure 5: Molecular property distributions between parent and child molecules for MEGA-Small.

For comparison to prior datasets, we report the Fréchet ChemNet Distance (FCD; lower is closer) [41]. As shown in Table 11, the distance between MolEdit and MolOpt roughly 4x lower compared to MEGA-Small. This indicates that MEGA occupies a distinct region of the chemical space, while the incumbent datasets exhibit notable overlap, thus, expanding the resources available in the existing literature.

Table 11: Fréchet distance between datasets computed in Morgan-fingerprint space (lower is closer).

Dataset	MEGA-Small	MolEdit	MolOpt
MEGA-Small	0.000	2.790	2.738
MolEdit	2.790	0.000	0.696
MolOpt	2.738	0.696	0.000

Prompts. Unless otherwise stated, prompts request one candidate molecule in SMILES, with no extra explanation.

Single-objective prompts:

101: Reduce $\log P$

```
User: Can you make molecule <SMILES_PLACEHOLDER>more soluble in water? The
output molecule should be similar to the input molecule.
Output: One valid SMILES.
```

102: Increase $\log P$

```
User: Can you make molecule <SMILES_PLACEHOLDER>less soluble in water? The
output molecule should be similar to the input molecule.
Output: One valid SMILES.
```

103: Increase QED

```
User: Can you make molecule <SMILES_PLACEHOLDER>more like a drug? The output
molecule should be similar to the input molecule.
Output: One valid SMILES.
```

104: Reduce QED

```
User: Can you make molecule <SMILES_PLACEHOLDER>less like a drug? The output
molecule should be similar to the input molecule.
Output: One valid SMILES.
```

105: Decrease TPSA

```
User: Can you make molecule <SMILES_PLACEHOLDER>higher permeability? The
output molecule should be similar to the input molecule.
Output: One valid SMILES.
```

106: Increase TPSA

```
User: Can you make molecule <SMILES_PLACEHOLDER>lower permeability? The output
molecule should be similar to the input molecule.
Output: One valid SMILES.
```

107: Increase HBA

```
User: Can you make molecule <SMILES_PLACEHOLDER>with more hydrogen bond
acceptors? The output molecule should be similar to the input molecule.
Output: One valid SMILES.
```

108: Increase HBD

```
User: Can you make molecule <SMILES_PLACEHOLDER>with more hydrogen bond donors?
The output molecule should be similar to the input molecule.
Output: One valid SMILES.
```

Two-objective prompts:

201: Reduce $\log P$ & Increase HBA

```
User: Can you make molecule <SMILES_PLACEHOLDER>more soluble in water and more
hydrogen bond acceptors? The output molecule should be similar to the input
molecule.
Output: One valid SMILES.
```

202: Increase $\log P$ & Increase HBA

```
User: Can you make molecule <SMILES_PLACEHOLDER>less soluble in water and more
hydrogen bond acceptors? The output molecule should be similar to the input
molecule.
Output: One valid SMILES.
```

203: Reduce $\log P$ & Increase HBD

```
User: Can you make molecule <SMILES_PLACEHOLDER>more soluble in water and
more hydrogen bond donors? The output molecule should be similar to the input
molecule.
Output: One valid SMILES.
```

204: Increase $\log P$ & Increase HBD

```
User: Can you make molecule <SMILES_PLACEHOLDER>less soluble in water and
more hydrogen bond donors? The output molecule should be similar to the input
molecule.
Output: One valid SMILES.
```

205: Reduce $\log P$ & Decrease TPSA

```
User: Can you make molecule <SMILES_PLACEHOLDER>more soluble in water and
higher permeability? The output molecule should be similar to the input
molecule.
Output: One valid SMILES.
```

206: Reduce $\log P$ & Increase TPSA

```
User: Can you make molecule <SMILES_PLACEHOLDER>more soluble in water and lower
permeability? The output molecule should be similar to the input molecule.
Output: One valid SMILES.
```

B Supervised Fine-tuning (SFT) details

For all our fine-tuning experiments, we utilize a memory-efficient, 4-bit quantized LLaMA 3.1 8B Instruct model as the backbone. Our datasets are consistently formatted as prompt-completion pairs, where the prompts are detailed in the main text and the corresponding completions are the child SMILES.

To ensure a fair comparison across benchmarks, we trained three models, as detailed in Table 2, each on a different dataset that has been filtered to contain comparable tasks. For the MEGA-Small dataset, we retain tasks 101, 102, 103, 104, 107, and 108, resulting in 229K prompt-completion pairs. For MolEdit-Instruct, we use tasks 103, 104, 107, and 108 (as tasks 101 and 102 are not available), yielding 650K prompt-completion pairs. For MolOpt-Instructions, we include tasks 101, 102, 103, 104, 107, and 108, producing 301K prompt-completion pairs.

All models are trained using Low-Rank Adaptation (LoRA) with a rank of r=32 and α=16, targeting all attention projection matrices and feed-forward layers. We use a training batch size of 16 with a gradient accumulation of 2 steps, resulting in an effective batch size of 32. Optimization is performed with an 8-bit quantized AdamW optimizer for memory efficiency. The learning rate is set to $1e-4$ with a cosine annealing scheduler and a linear warm-up period of 100 steps. For regularization, a weight decay of 0.01 is applied. All models are trained with a maximum sequence length of 512 tokens, using mixed-precision training (bfloat16) when supported. All trainings are conducted on a single A100 (40GB) GPU for approximately 23 hours.

B.1 Evaluation

We perform a sanity check to ensure that test SMILES are not present in any of the training sets using canonical SMILES notation to prevent any data leakage. Importantly, to ensure the fairest possible evaluation, we evaluate each model using the prompt templates specific to their respective training datasets. This means models trained on MolEdit-Instruct data are evaluated with MolEdit-Instruct prompt templates, while models trained on MEGA-Small use MEGA templates, and models trained on MolOpt-Instructions use MolOpt-Instructions templates, eliminating any potential bias from prompt format differences.

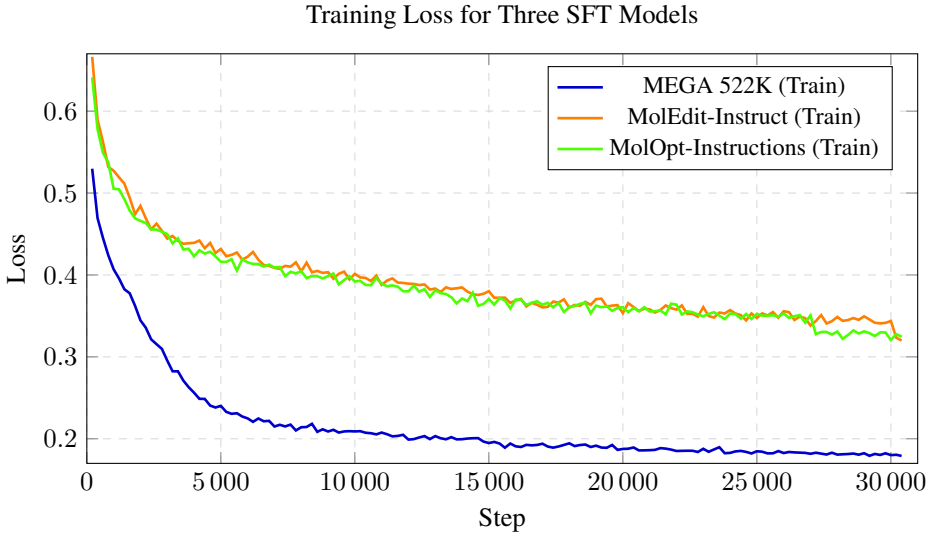


Figure 6: Training loss curves for three SFT models on MEGA-Small, MolEdit-Instruct, and MolOpt-Instructions. MEGA-Small achieves the lowest final loss (0.18), followed by MolOpt-Instructions and MolEdit-Instruct respectively.

As shown in Figure 6 the model trained on MEGA-Small exhibits significantly faster convergence and substantially lower final loss values. The better training dynamics observed with MEGA-Small indicates that our dataset leads to more sample-efficient learning, achieving better optimization faster.

In addition, for Table 3, we report hit ratio results comparing MEGA-Small GRPO against Gemini 2.5 Pro (June 17, 2025 official API release) and DrugAssist on the 500 test SMILES provided by DrugAssist.
This evaluation is performed over a single run, and we carefully verify that none of these 500 SMILES are included in our training set to avoid any possibility of data contamination.

B.2 Extra Comparisons

To further assess the utility of the MEGA dataset and extend the results in Table 2, we conducted an pair-wise comparison between MEGA and each external dataset on their overlapping task sets. Specifically, MEGA shares five tasks with MolEdit-Instruct and six with MolOpt-Instructions.

In the first experience, we trained models exclusively on the five tasks shared between MEGA and MolEdit-Instruct, namely tasks 103, 104, 107, 108, and 201 (Table 12). This setting corresponds to 678K training examples from MolEdit-Instruct and 183K examples from the MEGA-Small subset restricted to these five tasks. In the second, we trained models on the six tasks shared between MEGA and MolOpt-Instructions, namely tasks 101, 102, 103, 104, 107, and 108 (Table 13), which amounts to 301K training examples from MolOpt-Instructions and 229K examples from the filtered MEGA-Small subset. All training hyperparameters and conditions described in Appendix B were kept identical to ensure a fair and controlled comparison.

In these head-to-head evaluations, we found that models trained on the MEGA data partitions, in average, outperform those trained on the corresponding data from MolEdit-Instruct and MolOpt-Instructions. This finding further validates the quality and effectiveness of our dataset, demonstrating that its superior performance is not limited to a small task intersection, but holds true in expanded comparisons.

Table 12: Performance comparison: MEGA-Small vs MolEdit Instruct

Task	Threshold	MolEdit-Instruct	MEGA-Small
103	0.0	27.19 ± 0.84	**61.05 ± 2.88**
	0.1	14.37 ± 0.95	**24.36 ± 1.73**
104	0.0	**99.28 ± 0.52**	95.84 ± 0.89
	0.1	**97.94 ± 0.55**	80.95 ± 3.41
107	0.0	95.72 ± 0.61	**98.02 ± 0.90**
	1.0	43.05 ± 1.64	**94.58 ± 0.76**
108	0.0	98.10 ± 0.71	**99.80 ± 0.25**
	1.0	66.53 ± 2.05	**97.25 ± 0.60**
201	0.0	87.14 ± 1.99	**96.18 ± 1.03**
	0.5	81.66 ± 1.72	**87.86 ± 1.58**
Average		71.10	**83.59**

Table 13: Performance comparison: MEGA-Small vs MolOpt-Instructions

Task	Threshold	MolOpt-Instruction	MEGA-Small
101	0.0	96.71 ± 0.70	**98.04 ± 0.51**
	0.5	**96.41 ± 0.58**	92.47 ± 0.79
102	0.0	88.41 ± 1.85	**97.41 ± 0.74**
	0.5	88.41 ± 1.85	**92.53 ± 2.25**
103	0.0	16.82 ± 1.57	**59.71 ± 1.29**
	0.1	8.68 ± 1.16	**26.72 ± 2.31**
104	0.0	**97.92 ± 1.40**	97.42 ± 0.32
	0.1	**93.68 ± 1.88**	84.54 ± 2.15
107	0.0	92.33 ± 2.42	**98.35 ± 0.50**
	1.0	33.41 ± 3.03	**93.36 ± 0.57**
108	0.0	94.76 ± 0.91	**100.00 ± 0.00**
	1.0	56.10 ± 1.74	**98.56 ± 0.89**
Average		71.97	**86.59**

C GRPO Details

C.1 GRPO Algorithm for Molecural Editing

For each molecular editing prompt (x_{in}, x_t), GRPO operates as follows:

1. **Sample a group of candidate molecules**:
$$\{y_1, y_2, ..., y_G\} \sim \pi_\theta(\cdot|x_{\text{in}}, x_t) \tag{1}$$
where G is the number of generations by our policy model
2. **Compute rewards for all candidates** using batch molecular property evaluation:
$$r_i = R(y_i, x_{\text{in}}, x_t) \quad \text{for } i = 1, ..., G \tag{2}$$
3. **Calculate group-relative advantages**:
$$\hat{A}_i = \frac{r_i - \bar{r}}{\sigma_r + \epsilon} \tag{3}$$
where $\bar{r} = \frac{1}{G}\sum_{j=1}^{G} r_j$ and $\sigma_r = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_j - \bar{r})^2}$ are the mean and standard deviation of rewards within the group, and $\epsilon = 10^{-8}$ for numerical stability.
4. **Update the policy** using the GRPO objective:
$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\left[\min\left(\rho_{i,t}\hat{A}_i, \text{clip}(\rho_{i,t}, 1-\varepsilon, 1+\varepsilon)\hat{A}_i\right) - \beta D_{\text{KL}}[\pi_\theta \| \pi_{\text{ref}}]\right] \tag{4}$$
where:
   - $\rho_{i,t} = \frac{\pi_\theta(y_{i,t}|x_{\text{in}}, x_t, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x_{\text{in}}, x_t, y_{i,<t})}$ is the probability ratio
   - $\varepsilon = 0.2$ is the clipping parameter
   - $\beta = 0.0$ by default
   - If $\beta > 0$, the KL divergence is estimated as shown previously

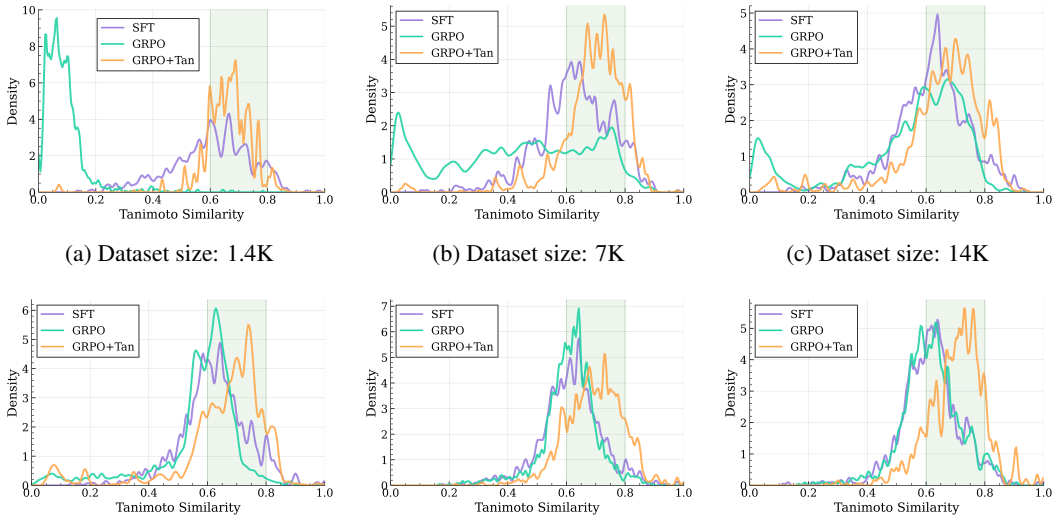
C.2 Experimental Details

For locality-aware GRPO training, we ensured strict consistency between supervised fine-tuning (SFT) and post-training data. For example, the MEGA-Small GRPO (14K) model used the same 14K SMILES for both SFT and GRPO. Similarly, the results in Table 4 were obtained from a policy model first fine-tuned on the full 522K prompt–completion pairs of MEGA-Small, with the same data reused during GRPO. In this phase, we sampled $G = 12$ generations per prompt and computed rewards for each candidate molecule.

Our composite reward function is designed to guide the model toward valid, improved, and structurally related molecules using three distinct signals. First, the validity reward provides a binary signal that ensures chemical correctness through RDKit sanitization while rejecting any outputs that are unchanged or fragmented. Second, the property reward implements a task-specific evaluation using a dual-threshold mechanism to provide fine-grained control over property modifications. Strict thresholds (e.g., ΔLogP > 0.5, ΔQED > 0.1) yield a reward of 1.0, whereas loose thresholds that only require a correct directional change yield 0.5. This encourages the model to learn both conservative and substantial improvements. Third, the Tanimoto similarity reward enforces structural conservation, assigning a reward of 1.0 for high similarity (Tanimoto coefficient > 0.65), 0.5 for moderate modifications (coefficients $\in$ [0.4, 0.65]), and 0.0 for major scaffold modifications (coefficients < 0.4).

All GRPO training was conducted on a single A100 GPU, with convergence achieved in approximately 10 hours at around 3,000 steps. We used an 8-bit quantized AdamW optimizer with a learning rate of $\alpha = 5 \times 10^{-6}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, a weight decay of 0.01, and gradient norm clipping at 0.5. The learning rate followed a cosine annealing schedule with a 10% linear warmup. To ensure memory efficiency, the model incorporated 4-bit quantization and LoRA adaptation with a rank of $r = 32$. We used an effective batch size of 8 (4 samples per device with 2 gradient accumulation steps) and maximum sequence lengths of 256 and 128 for prompts and completions, respectively. All computations were performed using bfloat16 mixed precision.

D Impact of GRPO and Tanimoto Reward on Scaffold Similarity



(a) Dataset size: 1.4K (b) Dataset size: 7K (c) Dataset size: 14K

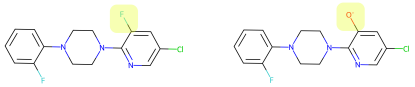
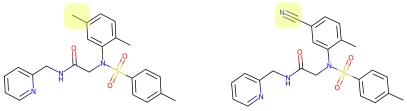
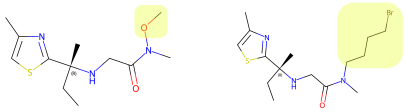
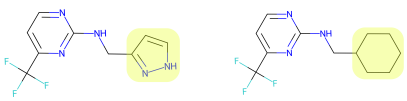
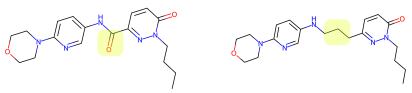
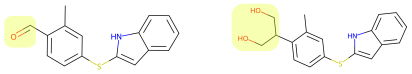
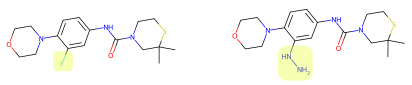
(d) Dataset size: 70K (e) Dataset size: 140K (f) Dataset size: 500K

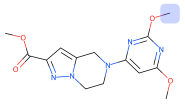
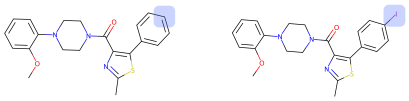
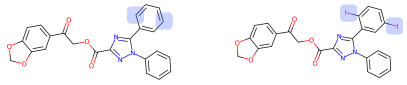
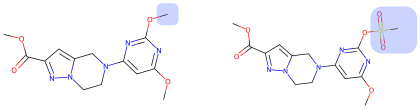
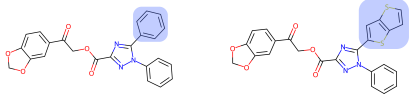
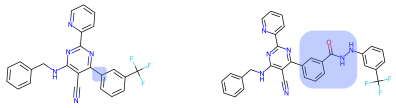
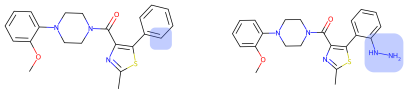
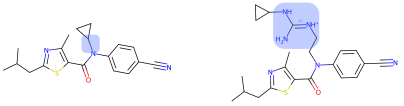
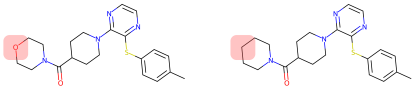
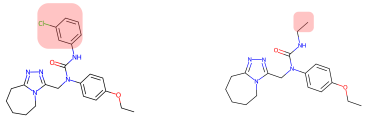
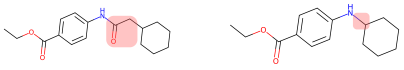
Figure 7: Tanimoto similarity distributions for different training data sizes. Each plot shows the distribution for SFT (purple), GRPO without Tanimoto reward (turquoise), and GRPO with Tanimoto reward (orange) models. The green shaded region (0.6–0.8) indicates the targeted tanimoto similarity range.

We trained MEGA across varying dataset sizes using GRPO, either with or without incorporating a Tanimoto similarity component into the reward system.
When trained without the Tanimoto reward on small datasets, the model achieves high benchmark performance but tends to alter the scaffold substantially, yielding molecules with low similarity to their parent compounds.
As the dataset size increases, however, the model implicitly recovers the similarity distribution observed in the SFT baseline (MEGA-Small), ultimately reaching the target similarity regime even without an explicit similarity signal. In contrast, when the Tanimoto reward is included, the model attains this small-edit regime with as few as 1.4k training examples (roughly 100 per task type).

E Qualitative examples

Table 14: Visualization of molecular editing with three actions: Replace, Insert, and Delete. The yellow regions indicate replaced substructures, the blue regions indicate inserted substructures, and the red regions indicate deleted substructures. Each example shows the transformation from the input molecule $\mathbf{x}_{\text{in}}$ to the output molecule $\mathbf{x}_{\text{out}}$.

Action: REPLACE	
(a) 101 (strict)	**(b) 106 (strict)**
Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$	Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$
	
LogP: 3.3398 $\rightarrow$ 2.2743	TPSA: 79.3700 $\rightarrow$ 103.1600
(c) 102 (strict)	**(d) 103 (loose)**
Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$	Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$
	
LogP: 1.6861 $\rightarrow$ 3.2998	QED: 0.8626 $\rightarrow$ 0.9025
(e) 105 (strict)	**(f) 107 (loose)**
Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$	Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$
	
TPSA: 89.3500 $\rightarrow$ 72.2800	H-Bond Acceptors: 2 $\rightarrow$ 3
(g) 108 (strict)	**(h) 205 (strict)**
Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$	Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$
	
H-Bond Donors: 1 $\rightarrow$ 3	LogP: 3.0216 $\rightarrow$ 1.3313, TPSA: 44.81 $\rightarrow$ 32.18

Action: INSERT	
(i) 101 (strict)	**(j) 102 (strict)**
Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$	Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$
	
LogP: 0.4971 → -0.0816	LogP: 4.0895 → 4.6941
(k) 104 (strict)	**(l) 201 (strict)**
Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$	Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$
	
QED: 0.3421 → 0.1626	LogP: 0.4971 → -0.1731, H-Acceptors: 9 → 11
(m) 202 (strict)	**(n) 204 (strict)**
Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$	Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$
	
LogP: 3.7027 → 4.9789, H-Acceptors: 8 → 10	LogP: 5.7082 → 6.4651, H-Donors: 1 → 3
(o) 206 (strict)	**(p) 108 (strict)**
Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$	Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$
	
LogP: 4.0895 → 3.3751, TPSA: 45.67 → 83.72	H-Bond Donors: 0 → 3
Action: DELETE	
(q) 102 (strict)	**(r) 103 (strict)**
Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$	Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$
	
LogP: 3.0114 → 3.5138	QED: 0.5656 → 0.8620
(s) 105 (strict)	**(t) 105 (strict)**
Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$	Input Molecule $\mathbf{x}_{\text{in}} \rightarrow$ Output Molecule $\mathbf{x}_{\text{out}}$
	
TPSA: 31.3500 → 18.4600	TPSA: 55.4000 → 38.3300

F Dataset License

We used the ZINC 250K dataset [42] available here , which is distributed under the GNU General Public License v3 or later (GPL-3.0+). In accordance with this license, we release our derived dataset under the same terms, preserving the freedoms to use, share, and modify the data.