

PairIT: Autoregressive Transformers for Low-Data Molecule Optimization

Hannah Lawrence^{1,2}Bodhi P. Vani¹Ji Won Park¹Natasa Tagasovska³Samuel Don Stanton⁴Andrew Martin Watkins¹Michael Maser¹Ewa M. Nowara^{1,*}¹Genentech, South San Francisco, CA, USA²Massachusetts Institute of Technology, Cambridge, MA, USA³Prescient Design, Genentech, Basel, Switzerland⁴Prescient Design, Genentech, New York, NY, USA

*Corresponding author: nowara.ewa@gene.com

Abstract

Generating novel molecules with desired properties is a central challenge in drug discovery. Although foundational work focused on *de novo* generation, practical use cases require a more complex setting: conditioning on an initial seed molecule, optimizing multiple properties, and operating in data-sparse regimes. We introduce *PairIT*, a flexible training framework for autoregressive transformers over SMILES sequences that enables *seeded* generation by learning from pairs of structurally similar molecules. By leveraging matched pairs, the model learns to generate local, chemically meaningful edits that improve a property of the seed molecule. Prompt tokens extend this framework by specifying the target property, desired direction, and approximate magnitude of change. Multiple prompt tokens can be concatenated to enable multi-property optimization at inference time. For properties where pair supervision alone is insufficient, reinforcement learning post-training can further improve optimization, providing a tunable trade-off between proximity to the seed and the optimization objective. *PairIT* naturally supports pretraining on a large dataset and fine-tuning on a smaller dataset, which is critical for performance in data-sparse regimes. Surprisingly, we observe diminishing returns to the scale of the pretraining dataset. We demonstrate empirically on MoleculeNet and Therapeutics Data Commons datasets that, with appropriate paired-data curation, a standard transformer model of modest size (68M parameters) can achieve robust seeded molecular optimization in low-data regimes.

1 Introduction

Generative models enable efficient exploration of chemical space, making them increasingly important tools for molecular discovery [Bohacek et al., 1996]. In practical drug discovery, an important task is *molecular optimization*. The goal is to propose small, local modifications to a known seed molecule that improve one or more (bio)chemical properties. Maintaining structural similarity to a seed molecule increases the likelihood of preserving biological function [Bender and Glen, 2004]. However,

Generative AI tools were used to assist with manuscript editing and formatting, and with code development and debugging. All AI-assisted outputs were reviewed and verified by the authors.

Code is available at <https://github.com/prescient-design/pairit>.

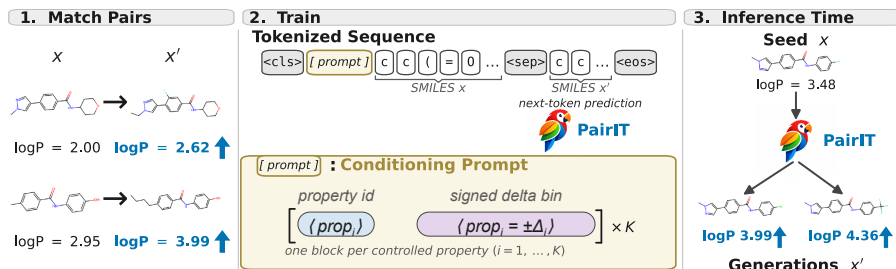


Figure 1: Overview of PairIT. We (1) construct training pairs of structurally similar molecules with differing property values and (2) train an autoregressive transformer to predict the target SMILES given the seed SMILES and structured property-control prompt tokens. (3) At inference time, the model generates property-improved candidates from a new seed and prompt. Multiple prompt blocks can be concatenated for multi-property control.

structure–function response surfaces are notoriously jagged [Aldeghi et al., 2022]; molecules differing by even a single atom may have drastically different properties, making optimization challenging. In practice, computational and medicinal chemists often approach this problem through local molecular edits informed by cheminformatics tools and domain expertise [Kramer et al., 2014]. A full discovery campaign has an “outer loop” in which candidates are synthesized and experimentally measured and an “inner loop” in which candidates are proposed and scored against a surrogate model. We focus on seeded generation⁰ in this “inner loop”, in the scarce-label regime common in early-stage discovery, where experimental measurements are expensive and datasets often contain only hundreds to thousands of labeled molecules [Wu et al., 2018].

While substantial prior literature exists on molecular optimization, the small-dataset regime remains under-addressed, with few works evaluating seeded optimization on biological properties. Many recent models are developed and evaluated either in *de novo* settings, where there is no seed molecule [Pinheiro et al., 2023, Joshi et al., 2025, Xu et al., 2023], or with far more labeled data than is typically available in early-stage discovery [Dey et al., 2025, Lee et al., 2025, Jin et al., 2020]. In contrast, we focus on seeded optimization in the low-data regime. Generated molecules must remain close enough to the seed to preserve relevant structure while introducing enough novelty to improve the desired property. Moreover, constraining generation to a local search space is restrictive, by definition: the model must introduce enough novelty to improve the requested properties while still remaining close to the seed, and such conditioning can collapse toward near-reconstruction rather than diverse one-to-many outputs [Kumar et al., 2024]. In small labeled datasets, limited examples further increase the risk of overfitting and memorized edits [Wu et al., 2018]. Yet even small datasets can contain useful local structure–property relationships, motivating training objectives that expose these transformations directly.

One way to expose such local transformations directly is to match structurally similar molecules from a dataset and train the model to “translate” from one molecule to the other. If the pairs are directed so that the target molecule has an improved value for a property of interest, the model learns local, property-improving edits. For example, Damani et al. [2019] apply this sequence-to-sequence approach to small-molecule optimization, while Tagasovska et al. [2024], Pinheiro et al. [2025], Chen et al. [2025] use related pair-based approaches for protein sequence optimization. Recent work also applies pair-based training to large language models for controllable molecular optimization [Dey et al., 2025].

We propose PairIT (Figure 1), a decoder-only autoregressive transformer over SMILES¹ trained on matched seed–target pairs of structurally similar molecules with differing properties. The base model learns to generate local, property-improving edits directly from pair supervision. To extend optimization beyond the property changes learned in a single generation step, we also consider iterative re-seeding, in which high-scoring generations are fed back to the model as new seeds for further optimization. We refer to this variant as PairIT-Beam. For properties that remain difficult to optimize from pair supervision alone, we further explore reinforcement learning (RL) post-training

⁰In this work, we use the terms *molecular optimization* and *seeded generation* interchangeably, and likewise for *seed molecule* and *input molecule*.

¹SMILES strings are text representations of molecular structure [Weininger, 1988].

with a property-model reward when a reliable property predictor is available, which we refer to as PairIT-RL.

However, pair-only approaches tie the optimization objective to the constructed pair set. Changing the target property or direction, or composing multiple objectives, can require constructing new pairs and retraining. This becomes increasingly restrictive in small datasets, where suitable pairs for specific combinations of property changes may be sparse or absent. To provide finer control over how much a seed property is changed, we additionally introduce structured control tokens specifying the property, direction, and approximate magnitude of change. This also extends naturally to multi-property optimization, where constructing similar pairs that improve several properties simultaneously can be especially difficult in very small datasets. Multiple property prompts can instead be composed at inference time without constructing a separate pair set and retraining for each combination.

Because the downstream pair sets are very small, we first pretrain PairIT on ZINC-250K [Irwin et al., 2012] before fine-tuning on each target dataset. This raises two practical questions, whether the gains come simply from molecular pretraining or specifically from pretraining on directed pairs, and how much paired pretraining is needed in terms of pair diversity and training duration. We therefore compare no pretraining, single-molecule pretraining, and directed-pair pretraining, vary the amount and diversity of paired data, and track validation and downstream performance over the course of training. We find that directed-pair pretraining outperforms pretraining on individual molecules, but that its benefits emerge surprisingly early, with the best validation checkpoint reached before one full epoch over the pair dataset. We study this behavior with training-dynamics and data-scale ablations, finding that repeated molecular coverage across many pairs helps explain the early saturation and that additional pair data yield diminishing returns.

PairIT deliberately combines existing components in a way tailored to low-data seeded molecular optimization, rather than introducing a new architecture or dataset. Rather than relying on a large general-purpose language model, we use a compact 68M-parameter transformer with molecule-specific tokenization, matched-pair construction, and the training and inference strategies described above. We evaluate this setting on MoleculeNet, where, to our knowledge, seeded generation has not previously been studied systematically. We formulate the task, define metrics that capture property improvement, novelty, and fidelity to the seed, and compare to diverse baseline methods under the same limited-data setting.

Our main contributions are (i) a compact 68M-parameter matched-pair autoregressive model for low-data seeded molecular optimization that is competitive with substantially larger or more elaborate baselines; (ii) iterative re-seeding and optional RL post-training that can further improve property optimization when additional inference compute or a reliable property predictor is available; (iii) structured property-control tokens that provide control over the direction and magnitude of property change and support multi-property optimization without retraining for each objective combination; and (iv) an analysis of paired-pretraining dynamics showing that pair structure provides benefits beyond exposure to the same molecules individually, and characterizing why useful pretraining can emerge before a full pass over the pair dataset.

2 Related Work

The methods most closely related to our approach are matched-pair methods for seeded molecular optimization. PropEn [Tagasovska et al., 2024] is a non-autoregressive encoder² trained to predict the output sequence in a pair; the authors show that this objective approximates the property-gradient direction. Follow-up work extends the framework to conditional density estimation [Pinheiro et al., 2025] and multi-objective optimization via a non-domination ranker applied during pair construction [Tagasovska et al., 2025, Park et al., 2024]. Earlier small-molecule work frames optimization as graph-to-graph or sequence-to-sequence translation [Jin et al., 2019, 2020, Kim et al., 2025]; Damani et al. [2019] introduce recursive inference, in which generated molecules are repeatedly fed back as inputs.

GeLLM4O-C [Dey et al., 2025] is the closest language-model baseline. It instruction-tunes 7–8B-parameter LLMs on approximately 256K molecule pairs with natural-language instructions specifying which properties to improve or maintain. PairIT similarly makes the requested property

²While Tagasovska et al. [2024] describe their architecture as encoder-decoder, they do not use an autoregressive decoder, which is the more common usage of the term today.

change explicit at inference time, but uses structured control tokens in a compact 68M-parameter autoregressive SMILES transformer rather than a general-purpose natural-language LLM, with only 88–662 downstream fine-tuning pairs per MoleculeNet task. Other molecular language-model work includes DrugAssist [Ye et al., 2023] and SmileyLlama [Cavanagh et al., 2024].

RL provides another route for molecular optimization. REINVENT [Olivecrona et al., 2017], LLOME [Chen et al., 2025], and MOLRL [Haddad et al., 2025] apply reinforcement learning to molecular or biological sequence generation under property-based rewards. Preference optimization, including DPO [Rafailov et al., 2023], provides a related pairwise-learning approach, with multi-objective and molecular extensions [Xiong and Singh, 2025, Zhou et al., 2024, Yang et al., 2024, Gupta et al., 2025, Park et al., 2023, Calanzone et al., 2025]. Appendix C compares the training signal of PairIT’s conditional likelihood objective with DPO as pair differences increase.

Property-model-guided search provides a complementary approach, including latent-space Bayesian optimization [Gómez-Bombarelli et al., 2018, Lee et al., 2025] and gradient-based steering [Lee et al., 2023]. These methods could in principle be combined with PairIT, while vanilla PairIT learns the optimization objective from paired data and does not require a property oracle in the generation loop.

3 Proposed Method: PairIT

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ be a dataset of molecules, where each x_i is a molecular representation (e.g., SMILES string) and $y_i \in \mathcal{Y}$ is its property, which may take discrete, continuous, or mixed values. We refer to the unknown property function as $g : \mathcal{X} \rightarrow \mathcal{Y}$ such that $y_i = g(x_i)$. Our goal is *seeded molecular generation*: given an *initial* molecule x , we seek to generate one or more *candidate* molecules x' such that $s(x', x) \geq \tau$ and $g(x') > g(x)$, where $s(\cdot, \cdot)$ is a molecular similarity measure (e.g., Tanimoto similarity) and τ is the chosen similarity threshold. This formulation reflects a realistic task in drug discovery where the aim is to propose local, structure-preserving modifications that improve a property of interest while remaining in the chemical neighborhood of the input molecule. During training, we instantiate this objective with directed matched seed-target pairs; the corresponding dataset-specific pair-construction procedures are described in Section 4.

3.1 PairIT objective and interpretation

Given a seed molecule x and optional conditioning variables c , PairIT models a conditional distribution $q_\theta(x' | x, c)$ over target molecules x' , represented as SMILES strings. Let $p_{\text{pair}}(x, x', c)$ denote the empirical distribution over the directed training pairs introduced above. PairIT parameterizes an autoregressive distribution over the target SMILES sequence $x' = (x'_1, \dots, x'_T)$, where T is the target length in tokens: $q_\theta(x' | x, c) = \prod_{t=1}^T q_\theta(x'_t | x'_{<t}, x, c)$.

Training minimizes the conditional next-token negative log-likelihood, $\mathcal{L}(\theta) = -\mathbb{E}_{(x, x', c) \sim p_{\text{pair}}} [\log q_\theta(x' | x, c)]$. Because p_{pair} is supported on directed pairs that are both structurally similar and property-improved, minimizing this objective fits $q_\theta(x' | x, c)$ to the empirical distribution of local property-improved neighbors of x . Thus, pair construction encodes both structural locality and the desired direction of property change, while the conditional model allows multiple plausible improved targets per seed.

PairIT provides a base matched-pair generator that can be extended in two complementary ways: iterative re-seeding or RL post-training can strengthen property optimization, while structured prompts provide explicit control over which properties should change and by how much.

3.2 PairIT implementation and variants

Model architecture. PairIT uses a 68M-parameter decoder-only autoregressive transformer over SMILES strings, instantiated with a Llama-style architecture [Touvron et al., 2023] and adapted for molecular sequence generation following prior implementations [Frey et al., 2024, Ismail et al., 2025]. We deliberately use a small model: SMILES generation has a substantially narrower vocabulary than natural language, and our model-size ablation shows little benefit from scaling across the 11M–200M range tested (Appendix B.2). For paired training, we concatenate seed and target SMILES separated by a `<sep>` token, with generation conditioned on the seed. Full architecture and training details are in Appendix B.1.

PairIT and iterative PairIT-Beam. Vanilla PairIT samples property-improved candidates directly from the matched-pair model. PairIT-Beam additionally applies iterative refinement: at each of three rounds, it generates 8 candidates per seed and retains the top 4 as seeds for the next round, yielding up to 72 candidates per original seed. This follows the recursive-inference idea of Damani et al. [2019].

PairIT-RL. When pair supervision alone provides insufficient signal for property improvement, we optionally post-train PairIT using a property model as a reward. We use Group Relative Policy Optimization (GRPO) [Shao et al., 2024] with a property-model-based reward and seed-similarity regularization to limit exploitation of the surrogate model. Full reward and training details are provided in Appendix F.

3.3 Prompt-conditioning

Single-property conditioning. Without prompting, the desired property change is implicit in each training pair. To make this control explicit, we prepend structured property tokens to the seed molecule [Kotsias et al., 2020]. For a single conditioned property, the prompt specifies the property identity and a signed delta bin: the sign gives the desired direction and the bin magnitude gives the approximate requested change. Prompting changes only the conditioning sequence and training uses the same autoregressive objective as above.

Multi-property conditioning. The same mechanism extends directly to multiple properties by concatenating one prompt block per requested property. During pretraining, a generic target slot can be trained using available properties and reassigned to a new downstream property during fine-tuning. For example, we pretrain on ZINC-250K using only RDKit properties and assign the generic target slot to the downstream MoleculeNet property during fine-tuning. We also apply property dropout during training so that subsets of the trained properties can be specified at inference time.

4 Experimental Results

Datasets and pair construction. We evaluate on three low-data MoleculeNet benchmarks (BACE, BBBP, HIV) using the class-balanced splits of GraphDiT [Liu et al., 2024] which are binary tasks corresponding to β -secretase 1 inhibition, blood-brain barrier permeability, and HIV antiviral activity, respectively. We additionally evaluate on hERG [Wang et al., 2016], a low-data toxicity task from Therapeutics Data Commons (TDC) [Huang et al., 2021] (Appendix D.6). For auxiliary pretraining, we construct ZINC-250K pairs by 2D Tanimoto similarity on Morgan fingerprints (ECFP4), directed toward a minimum logP improvement of $\Delta\log P > 1$, using 100 target neighbors per seed. For MoleculeNet fine-tuning, we construct matched molecular pairs (MMPs) using the mmpdb toolkit [Dalke et al., 2018], directed from inactive (class 0) to active (class 1) molecules. The resulting MoleculeNet training sets contain only 88–662 pairs from 517–1,420 molecules per dataset. Full pair construction details, including the rationale for the dataset-specific pairing strategies, and dataset statistics are in Appendix B.4.

Inference and evaluation. Each method produces 64 candidates per seed (PairIT-Beam: up to 72). We canonicalize and deduplicate the candidates, then rank them using validity and novelty together with property predictions from the random forest used in GraphDiT [Liu et al., 2024] and Tanimoto similarity to the seed, retaining the top five candidates per seed for each method. These top-five selections are then independently scored with MiniMol [Kläser et al., 2024] for the final reported results. Both property models are trained only on the corresponding MoleculeNet training splits. The same random forest is also used for NF-BO search, PairIT-Beam refinement, and PairIT-RL rewards. Agreement between the property predictions of these two oracles on held-out MoleculeNet test molecules and on generated molecules is analyzed in Appendix B.5.1. Inference uses temperature 0.5; see Appendix B.3 for a hyperparameter sweep.

Baselines. We compare against seeded generation baselines spanning hierarchical graph-to-graph translation (HierG2G) [Jin et al., 2020], discrete diffusion Schrödinger bridge matching (DDSBM) [Kim et al., 2025], latent Bayesian optimization with normalizing flows (NF-BO) [Lee et al., 2025], and the 7B-parameter instruction-tuned LLM GeLLM4O-C [Dey et al., 2025]. We also include MMP enumeration [Kramer et al., 2014] as a non-learning local-edit reference and GraphDiT [Liu et al., 2024] as a *de novo* reference baseline. HierG2G and DDSBM are pretrained on the same ZINC-250K pair sets as PairIT; NF-BO uses the released GuacaMol checkpoint [Brown

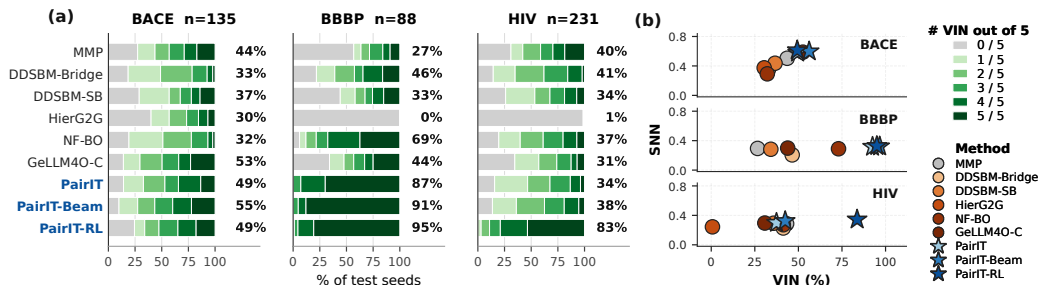


Figure 2: MoleculeNet seeded bioactivity optimization. (a) Per-seed VIN (valid, novel, improved) distribution: each stacked bar gives the percentage of seeds for which k of 5 retained generations are valid, novel, and improved. (b) Aggregate VIN-SNN (similarity to nearest neighbor in the target class) tradeoff across methods.

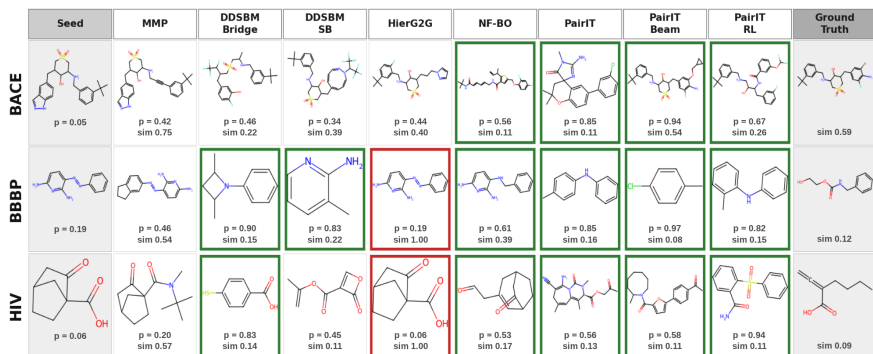


Figure 3: Example generations from PairIT and baselines. Leftmost column: the class-0 seed, rightmost column: the corresponding *ground-truth* class-1 MMP partner from the test set. Green = predicted class probability > 0.5 (success); red = seed duplicate.

et al., 2019].³ All baselines are finetuned on the same MoleculeNet pairs as PairIT after pretraining. GeLLM40-C uses the released 7B Mistral backbone with task-specific LoRA fine-tuning. Full baseline implementation details are in Appendix B.6.

Metrics. We report *Valid* (RDKit-parseable), *Novel* (absent from the fine-tune training set), *Improved* (class-1 generation from a class-0 seed on MoleculeNet), *QED* [Bickerton et al., 2012], *Unique* (distinct canonical SMILES across retained generations), and *Diverse* (within-seed scaffold diversity). Our aggregate metric *VIN* is the fraction of the top-5 retained generations per seed that are simultaneously Valid, Novel, and Improved. *SNN* is the mean Tanimoto similarity to the nearest target-class (class-1 for MoleculeNet corresponding to active molecules) training molecule over valid, non-memorized generations, measuring proximity to known target-class chemical space. Appendix B.5 provides a broader distributional evaluation of the generated molecules relative to the target-class chemical space using additional metrics including average Tanimoto similarity to the seed, Fréchet ChemNet Distance, fragment and scaffold similarity, and atom type coverage, allowing generation quality to be interpreted beyond the main VIN-SNN results.

4.1 Single-property optimization from pair supervision.

We first evaluate local single-property bioactivity optimization from class-0 seeds, with the goal of generating class-1 molecules, without prompting. Figure 2 summarizes the per-seed VIN distribution and aggregate VIN-SNN tradeoff. Full VIN-seed-similarity distributions are shown in Appendix D.3 and additional failure-mode and pair-filtering analyses are in Appendices D.4 and E.4, respectively.

Across MoleculeNet, PairIT and PairIT-Beam are competitive with the strongest baselines on BACE and substantially outperform them on BBBP, despite PairIT using 64 samples per seed versus 500

³On BACE, the train-only Random Forest predicts zero class-1 probability for every class-0 test seed, causing NF-BO’s GP surrogate to fail. Observation jitter did not resolve the issue, so we rerun NF-BO using a Random Forest trained on the full dataset, including the test set, giving NF-BO a slight advantage on BACE. See Appendix B.6.

property-model evaluations for NF-BO and only 68M parameters versus 7B for GeLLM4O-C [Dey et al., 2025]. HIV is more challenging for pair supervision in general: PairIT and most baselines are around or below the non-learning MMP baseline (40.4% VIN), which may reflect the greater heterogeneity of the active chemical space and the weaker structural context shared by available pairs. Here PairIT-RL with post-training substantially improves performance to 82.6% VIN. Because RL directly optimizes a learned property model and is therefore more sensitive to oracle quality, we treat PairIT-RL as an optional extension. Full RL training details and analyses are provided in Appendix F. Additional results on hERG, a distinct toxicity dataset from TDC, are reported in Appendix D.6.

4.2 Pretraining ablations and training dynamics

We first analyze the effects of pretraining and paired supervision separately. Without any ZINC-250K pretraining, training only on the small MoleculeNet pair sets leads to a large drop in VIN across all three tasks. Failure-mode analysis shows that these models generate many invalid SMILES (Appendix E.1, Figure 18), suggesting that a major role of pretraining in this low-data regime is to learn the SMILES grammar and generate valid molecules before downstream fine-tuning. Pretraining on molecules alone is not sufficient, however. Single-molecule pretraining on the same ZINC-250K distribution underperforms pretraining on directed pairs (Appendix E.1, Table 14). Thus, the benefit is not simply exposure to more molecules; pair supervision additionally teaches the seed-conditioned local transformations beneficial for molecular optimization. Within paired pretraining, performance tends to improve with the number of pairs per seed, although gains are not monotonic and eventually diminish. More diverse pair sets also increase seed similarity and expose the model to more unique molecules.

Surprisingly, the useful signal from paired ZINC-250K pretraining saturates very quickly. The best checkpoint by validation loss occurs before one full epoch over the pair dataset. Importantly, a pair epoch is not equivalent to a full pass over the unique molecules: because each molecule can be included in many matched pairs, the model repeatedly encounters the underlying molecules while seeing only a fraction of the constructed pair examples. Appendix E.2 explicitly quantifies the number of unique molecules encountered at the point of early stopping (Figure 19), separating molecular coverage from pair-dataset coverage. Thus, early saturation does not imply that the model has seen little chemical diversity; rather, it suggests that additional pair examples become redundant relatively quickly.

Consistent with this, increasing the size of the paired pretraining set yields rapid initial gains followed by diminishing downstream improvement (Figure 4). Once the model has learned to produce valid, similarity-preserving molecular edits, additional exposure to ZINC-250K pair examples appears to provide limited benefit, making aggressive early stopping important.

Importantly, ZINC-250K pretraining validation loss correlates with generation quality on the ZINC-250K validation set (Figure 20) and with downstream BACE VIN after fine-tuning (Figure 21). A similar relationship holds between pretraining validation loss and zero-shot logP optimization on MoleculeNet (Appendix E.6, Figure 24). We therefore select pretraining checkpoints by validation loss rather than training for a fixed number of epochs.

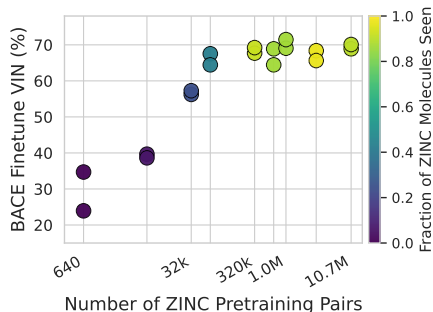


Figure 4: Scale of ZINC-250K pretraining dataset vs. downstream fine-tuned BACE performance. Saturation before one full epoch of pretraining is visible.

4.3 Prompt-conditioned property control

Pair supervision implicitly determines the property changes represented by the training pairs. Prompting instead makes the desired property change explicit by encoding its direction and magnitude in conditioning tokens, allowing finer control over which properties are changed and by how much. We evaluate prompt conditioning in three settings, including single-property magnitude control, joint two-property control, and conditioning on many properties simultaneously.

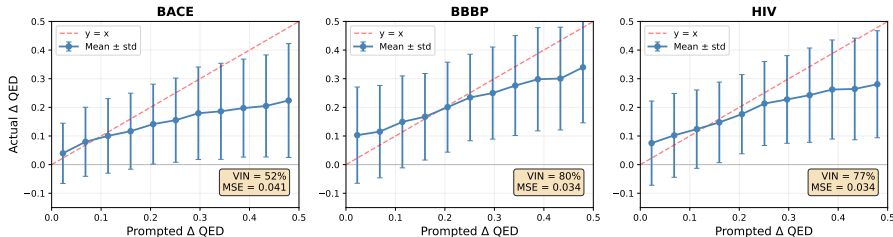


Figure 5: QED controllability on MoleculeNet, with upward prompts over delta bins in $[0, 0.5]$. Error bars show standard deviation across 5 generations per seed per bin. Realized changes scale with prompted magnitude, indicating coarse magnitude control. The corresponding logP results are in Appendix G.2 (Figure 37).

Table 1: Two-property controllability on MoleculeNet, using 64 raw generations per held-out matched pair without top-five selection. **Per-property VIN**: is the fraction of all attempts with an applicable nonzero request that are valid, novel, and satisfy that property’s criterion. **Combined VIN**: fraction of generations that are valid, novel, and move in the requested direction on *both* prompted properties; zero-delta properties impose no condition, only pairs with at least one non-zero change requested are counted. For bioactivity, success means an inactive seed yields a generation with train-only random-forest activity probability greater than 0.5. **Random**: independently resample each property’s signed-bin index uniformly from 00–15 while retaining property identities and block order. Both arms are scored against the original requests to serve as a baseline. Δ is prompted minus randomized Combined VIN in percentage points, positive Δ in bold.

Setting	Dataset	Combined VIN (%)	Per-property VIN (%)	Random (%)	Δ
Low-correlation pair	BACE	12	13 / 16	17	-4
	BBBP	32	38 / 35	21	+11
	HIV	64	70 / 76	32	+32
QED + logP	BACE	19	22 / 21	15	+3
	BBBP	30	33 / 43	17	+13
	HIV	42	55 / 63	30	+12
QED + bioactivity	BACE	8	12 / 10	7	+1
	BBBP	26	31 / 40	27	-1
	HIV	8	63 / 13	6	+2
TPSA + bioactivity	BACE	11	17 / 13	12	-1
	BBBP	48	53 / 55	37	+11
	HIV	8	76 / 10	9	0

Single-property magnitude control. We first test whether the signed delta token carries information beyond a binary increase-or-decrease signal already available from the training pairs. On the small MoleculeNet datasets, we fine-tune with both orderings of each pair and sweep 16 signed delta bins at inference. Figure 5 shows the QED results: across all three datasets, larger requested increases generally produce larger realized changes. Although the realized changes do not match the requested bins exactly, this suggests that the prompt can support graded control even in the low-data MoleculeNet setting, despite each delta bin being trained on only a small subset of the available examples. Appendix G.2 reports additional single-property results using logP, together with aggregate VIN and MSE values for both QED and logP (Figure 37, Table 15).

Two-property control. We next evaluate how well two requested property changes can be composed in the same prompt. On ZINC-250K, larger requested changes produce larger realized changes for both prompted properties, showing that graded control is retained under simultaneous conditioning (Appendix G.3, Figure 38). On MoleculeNet, both properties are prompted simultaneously using signed delta bins derived from held-out matched test pairs, which provides chemically realistic requests. Combined VIN is computed by considering every prompt with at least one valid non-zero change and including generations that are valid, novel, and move in the prompted direction for the non-zero change properties. To test whether the model is actually using the conditioning tokens rather than merely reproducing correlations in the paired data, we compare against randomized prompt assignments (Table 1).

Prompt-conditioned generation exceeds the randomized-prompt baseline most consistently for descriptor pairs selected to have low empirical correlation and for QED+logP, indicating that the model

Table 2: Direction adherence under multi-property prompting: fraction of generations whose property delta matches the prompted direction, averaged across the 9 non-activity properties. Higher is better.

Method	BACE	BBBP	HIV
PairIT (10-property prompt)	0.773	0.739	0.739
Randomized prompt (control)	0.433	0.499	0.559

can respond jointly to two property instructions. However, joint control involving bioactivity is less reliable. One likely reason is that, unlike continuously valued descriptors such as QED, logP, or TPSA, MoleculeNet bioactivity is binary and therefore provides a much coarser control signal. The small matched-pair datasets also contain relatively few examples of each joint property configuration. We therefore view the bioactivity results as evidence that joint control becomes harder when one objective provides only coarse supervision, rather than as a failure of the prompting mechanism itself. Detailed two-property scatter plots and per-property results are provided in Appendix G.4.

Many-property control. Finally, we evaluate whether prompt dependence is retained when the conditioning space becomes substantially larger. We train a single model on nine computable RDKit properties plus a tenth target slot that is assigned to downstream bioactivity during MoleculeNet fine-tuning. With all 10 properties prompted simultaneously, mean direction adherence across the nine non-activity properties reaches 0.739–0.773, compared with 0.433–0.559 under randomized prompts for matching the prompted increase/decrease direction (Table 2). Thus, even in the 10-property setting, the generated edits remain substantially sensitive to the requested property directions rather than ignoring the prompt.

Many-property control is nevertheless sensitive to which objectives are included. Appendix G.5 compares the full 10-property prompt with smaller property subsets and evaluates all $\binom{9}{3} = 84$ choices of three auxiliary RDKit properties alongside bioactivity. These results show that conditioning on more properties is not necessarily better: performance can improve when the prompt is restricted to a smaller subset, and varies substantially depending on which auxiliary properties are included.

5 Discussion

PairIT supports a progression from simple pair-supervised generation to more strongly guided optimization. Vanilla PairIT learns property-improving local edits directly from matched seed–target pairs. PairIT-Beam extends this by iteratively re-seeding high-scoring generations, allowing further property improvement at inference time at the cost of additional compute. When a reliable property oracle is available, PairIT-RL can push optimization further, but it is more sensitive to reward-model errors and can exhibit reward hacking under weak similarity constraints. Prompt conditioning adds a complementary form of control by making the desired direction and magnitude of property change explicit and extending the same framework to multi-property optimization.

PairIT also provides several ways to trade off property improvement against similarity to the seed. Sampling temperature changes this tradeoff without retraining (Fig. 7), while the similarity-reward weight in PairIT-RL directly controls the balance between property optimization and seed similarity.

Pretraining is particularly important in the low-data setting. ZINC-250K pretraining substantially improves reliable molecular generation, while directed-pair pretraining is more effective than pretraining on the same molecules individually. More diverse pretraining pairs also change the locality of the learned edits (Appendix E.1). At the same time, useful paired pretraining saturates quickly. In practice, this makes aggressive early stopping an important part of the training recipe.

On the surface, it may appear that the role of specialized domain knowledge has been diminished, yet it reappears in how the optimization problem is defined. The distance function determines what constitutes a “local” molecular edit, the construction of the paired data determines which transformations are learned, the prompted properties determine which objectives can be controlled, and the reward function determines what is optimized during post-training. As machine learning methods become increasingly effective at solving the problems we set before them, it becomes increasingly important that we formulate those problems well.

Limitations and broader impact. PairIT addresses the “inner loop” of generating high-scoring candidates under a scoring model, but lacks “outer loop” experimental validation. Any property oracle outside the wet lab is imperfect and susceptible to distribution shift, so oracle-dependent components inherit this uncertainty. Performance also depends on training-pair diversity, and multi-property control is less reliable when requested changes conflict with empirical property correlations or are weakly represented in the paired data (Table 1). We do not address synthesizability of generated candidate molecules, which we leave to future work. Generated candidates should be subject to expert review and safety screening before synthesis.

References

- Matteo Aldeghi, David E. Graff, Nathan Frey, Joseph A. Morrone, Edward O. Pyzer-Knapp, Kirk E. Jordan, and Connor W. Coley. Roughness of molecular property landscapes and its impact on modellability. *Journal of Chemical Information and Modeling*, 62(19):4660–4671, 2022. doi: 10.1021/acs.jcim.2c00903. URL <https://doi.org/10.1021/acs.jcim.2c00903>.
- Andreas Bender and Robert C. Glen. Molecular similarity: A key technique in molecular informatics. *Organic & Biomolecular Chemistry*, 2(22):3204–3218, 2004.
- G. Richard Bickerton, Gaia V. Paolini, J  r  my Besnard, Sorel Muresan, and Andrew L. Hopkins. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4(2):90–98, 2012. doi: 10.1038/nchem.1243.
- Regine S Bohacek, Colin McMartin, and Wayne C Guida. The art and practice of structure-based drug design: a molecular modeling perspective. *Medicinal research reviews*, 16(1):3–50, 1996.
- Nathan Brown, Marco Fiscato, Marwin H. S. Segler, and Alain C. Vaucher. GuacaMol: Benchmarking models for de novo molecular design. *Journal of Chemical Information and Modeling*, 59(3): 1096–1108, 2019. doi: 10.1021/acs.jcim.8b00839.
- Diego Calanzone, Pierluca D’Oro, and Pierre-Luc Bacon. Mol-MoE: Training preference-guided routers for molecule generation. *arXiv preprint arXiv:2502.05633*, 2025. doi: 10.48550/arXiv.2502.05633. URL <https://arxiv.org/abs/2502.05633>.
- Joseph M. Cavanagh, Kunyang Sun, Andrew Gritsevskiy, Dorian Bagni, Yingze Wang, Thomas D. Bannister, and Teresa Head-Gordon. SmileyLlama: Modifying large language models for directed chemical space exploration. *arXiv preprint arXiv:2409.02231*, 2024.
- Angelica Chen, Samuel Don Stanton, Frances Ding, Robert G Alberstein, Andrew Martin Watkins, Richard Bonneau, Vladimir Gligorijevic, Kyunghyun Cho, and Nathan C Frey. Generalists vs. specialists: Evaluating LLMs on highly-constrained biophysical sequence optimization tasks. In *International Conference on Machine Learning*, pages 9029–9072. PMLR, 2025.
- Kamran Chitsaz, Roshan Balaji, Quentin Fournier, Nirav Pravinbhai Bhatt, and Sarath Chandar. NovoMolGen: Rethinking molecular language model pretraining. In *International Conference on Learning Representations Generative AI and Biology (GenBio) Workshop*, 2025.
- Andrew Dalke, Jerome Hert, and Christian Kramer. mmpdb: An open-source matched molecular pair platform for large multiproperty data sets. *Journal of Chemical Information and Modeling*, 58(5):902–910, 2018. doi: 10.1021/acs.jcim.8b00173.
- Farhan Damani, Vishnu Sresht, and Stephen Ra. Black box recursive translations for molecular optimization. *arXiv preprint arXiv:1912.10156*, 2019.
- Vishal Dey, Xiao Hu, and Xia Ning. Large language models for controllable multi-property multi-objective molecule optimization. *arXiv preprint arXiv:2505.23987*, 2025.
- Nathan C Frey, Taylor Joren, Aya Abdelsalam Ismail, Allen Goodman, Richard Bonneau, Kyunghyun Cho, and Vladimir Gligorijevic. Cramming protein language model training in 24 GPU hours. In *ICML 2024 Workshop on Efficient and Accessible Foundation Models for Biological Discovery*, 2024.
- Rafael G  mez-Bombarelli, Jennifer N Wei, David Duvenaud, Jos   Miguel Hern  ndez-Lobato, Benjam  n S  nchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Al  n Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.
- Raghav Gupta, Ryan Sullivan, Yunxuan Li, Samrat Phatale, and Abhinav Rastogi. Robust multi-objective preference alignment with online DPO. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(26):27321–27329, 2025. doi: 10.1609/aaai.v39i26.34942. URL <https://ojs.aaai.org/index.php/AAAI/article/view/34942>.

- Ragy Haddad, Eleni E. Litsa, Zhen Liu, Xin Yu, Daniel Burkhardt, and Govinda Bhisetti. Targeted molecular generation with latent reinforcement learning. *Scientific Reports*, 15(1):15202, 2025. doi: 10.1038/s41598-025-99785-0.
- Emiel Hooeboom, Victor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3D. In *International conference on machine learning*, pages 8867–8887, 2022.
- 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. *Proceedings of Neural Information Processing Systems, NeurIPS Datasets and Benchmarks*, 2021.
- Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.
- John J Irwin, Teague Sterling, Michael M Mysinger, Erin S Bolstad, and Ryan G Coleman. ZINC: a free tool to discover chemistry for biology. *Journal of chemical information and modeling*, 52(7): 1757–1768, 2012.
- Aya Abdelsalam Ismail, Tuomas Oikarinen, Amy Wang, Julius Adebayo, Samuel Don Stanton, Hector Corrada Bravo, Kyunghyun Cho, and Nathan C Frey. Concept bottleneck language models for protein design. In *The Thirteenth International Conference on Learning Representations*, 2025.
- Wengong Jin, Kevin Yang, Regina Barzilay, and Tommi Jaakkola. Learning multimodal graph-to-graph translation for molecule optimization. In *International Conference on Learning Representations*, 2019.
- Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In *International Conference on Machine Learning*, pages 4839–4848, 2020.
- Chaitanya K. Joshi, Xiang Fu, Yi-Lun Liao, Vahe Gharakhanyan, Benjamin Kurt Miller, Anuroop Sriram, and Zachary W. Ulissi. All-atom diffusion transformers: Unified generative modelling of molecules and materials. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 28393–28417. PMLR, 2025.
- Jun Hyeong Kim, Seonghwan Kim, Seokhyun Moon, Hyeongwoo Kim, Jeheon Woo, and Woo Youn Kim. Discrete diffusion schrödinger bridge matching for graph transformation. In *The Thirteenth International Conference on Learning Representations*, 2025.
- Kerstin Kläser, Błażej Banaszewski, Samuel Maddrell-Mander, Callum McLean, Luis Müller, Ali Parviz, Shenyang Huang, and Andrew Fitzgibbon. MiniMol: A parameter-efficient foundation model for molecular learning. *arXiv preprint arXiv:2404.14986*, 2024.
- Panagiotis-Christos Kotsias, Josep Arús-Pous, Hongming Chen, Ola Engkvist, Christian Tyrchan, and Esben Jannik Bjerrum. Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. *Nature Machine Intelligence*, 2(5):254–265, 2020. doi: 10.1038/s42256-020-0174-5. URL <https://doi.org/10.1038/s42256-020-0174-5>.
- Christian Kramer, Julian E Fuchs, Steven Whitebread, Peter Gedeck, and Klaus R Liedl. Matched molecular pair analysis: significance and the impact of experimental uncertainty. *Journal of medicinal chemistry*, 57(9):3786–3802, 2014.
- Manoj Kumar, Neil Houlsby, and Emiel Hooeboom. Conditional diffusion on web-scale image pairs leads to diverse image variations. *arXiv preprint arXiv:2405.14857*, 2024.
- Seul Lee, Jaehyeong Jo, and Sung Ju Hwang. Exploring chemical space with score-based out-of-distribution generation. In *International Conference on Machine Learning*, pages 18872–18892, 2023.

- Seunghun Lee, Jinyoung Park, Jaewon Chu, Minseo Yoon, and Hyunwoo J Kim. Latent bayesian optimization via autoregressive normalizing flows. *arXiv preprint arXiv:2504.14889*, 2025.
- Gang Liu, Jiaxin Xu, Tengfei Luo, and Meng Jiang. Graph diffusion transformers for multi-conditional molecular generation. *Advances in Neural Information Processing Systems*, 37:8065–8092, 2024.
- Ewa Nowara, Pedro O Pinheiro, Sai Pooja Mahajan, Omar Mahmood, Andrew Martin Watkins, Saeed Saremi, and Michael Maser. NEBULA: Neural empirical bayes under latent representations for efficient and controllable design of molecular libraries. In *ICML 2024 AI for Science Workshop*, 2024.
- Ewa Nowara, Joshua A Rackers, Patricia Adriana Suriana, Pan Kessel, Max W Shen, Andrew Martin Watkins, and Michael Maser. Do we need equivariant models for molecule generation? In *International Conference on Learning Representations Generative AI and Biology (GenBio) Workshop*, 2025.
- Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9(1):48, 2017. doi: 10.1186/s13321-017-0235-x.
- Ji Won Park, Natasa Tagasovska, Michael Maser, Stephen Ra, and Kyunghyun Cho. BOTied: Multi-objective bayesian optimization with tied multivariate ranks. In *International Conference on Machine Learning*, pages 39813–39833. PMLR, 2024.
- Ryan Park, Ryan Theisen, Navriti Sahni, Marcel Patek, Anna Cichońska, and Rayees Rahman. Preference optimization for molecular language models. *arXiv preprint arXiv:2310.12304*, 2023. doi: 10.48550/arXiv.2310.12304. URL <https://arxiv.org/abs/2310.12304>.
- William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.
- Pedro O. Pinheiro, Joshua Rackers, Joseph Kleinhenz, Michael Maser, Omar Mahmood, Andrew Martin Watkins, Stephen Ra, Vishnu Sresht, and Saeed Saremi. 3D molecule generation by denoising voxel grids. *Advances in Neural Information Processing Systems*, 2023.
- Pedro O Pinheiro, Pan Kessel, Aya Abdelsalam Ismail, Sai Pooja Mahajan, Kyunghyun Cho, Saeed Saremi, and Natasa Tagasovska. Generative property enhancer: Implicit guided generation through conditional density estimation. In *Advances in Neural Information Processing Systems*, 2025.
- 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.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.
- Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. Unintentional unalignment: Likelihood displacement in direct preference optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.
- David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010. doi: 10.1021/ci100050t.
- 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. *arXiv preprint arXiv:2402.03300*, 2024. doi: 10.48550/arXiv.2402.03300. URL <https://arxiv.org/abs/2402.03300>.
- Max W Shen, Ewa Nowara, Michael Maser, and Kyunghyun Cho. Training dynamics of learning 3D-rotational equivariance. *arXiv preprint arXiv:2512.02303*, 2025.

- Nataša Tagasovska, Vladimir Gligoričević, Kyunghyun Cho, and Andreas Loukas. Implicitly guided design with PropEn: Match your data to follow the gradient. *Advances in Neural Information Processing Systems*, 37:35973–36001, 2024.
- Nataša Tagasovska, Jan Ludwiczak, Andreas Loukas, Emily K Makowski, Homa Mohammadi-Peyhani, Sai Pooja Mahajan, Karina Zadorozhny, Donald Lee, Allen Goodman, Joshua Yao-Yu Lin, et al. Property enhancer—a data efficient multi-objective approach for functional antibody optimization. *bioRxiv*, 2025. doi: 10.1101/2025.10.08.681101. URL <https://www.biorxiv.org/content/10.1101/2025.10.08.681101v1>.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- Clement Vignac, Nagham Osman, Laura Toni, and Pascal Frossard. MiDi: Mixed graph and 3D denoising diffusion for molecule generation. *arXiv preprint arXiv:2302.09048*, 2023.
- Shuangquan Wang, Huiyong Sun, Hui Liu, Dan Li, Youyong Li, and Tingjun Hou. ADMET evaluation in drug discovery. 16. predicting hERG blockers by combining multiple pharmacophores and machine learning approaches. *Molecular pharmaceutics*, 13(8):2855–2866, 2016.
- Yuan Wang, Hanna Eckert, and Jürgen Bajorath. Apparent asymmetry in fingerprint similarity searching is a direct consequence of differences in bit densities and molecular size. *ChemMedChem*, 2(7):1037–1042, 2007. doi: 10.1002/cmdc.200700050.
- David Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.
- Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.
- Nuoya Xiong and Aarti Singh. Projection optimization: A general framework for multi-objective and multi-group RLHF. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 69017–69058. PMLR, 2025. URL <https://proceedings.mlr.press/v267/xiong25c.html>.
- Minkai Xu, Alexander S Powers, Ron O Dror, Stefano Ermon, and Jure Leskovec. Geometric latent diffusion models for 3D molecule generation. In *International Conference on Machine Learning*, pages 38592–38610. PMLR, 2023.
- Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. Rewards-in-Context: Multi-objective alignment of foundation models with dynamic preference adjustment. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 56276–56297. PMLR, 2024. URL <https://proceedings.mlr.press/v235/yang24q.html>.
- 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. *arXiv preprint arXiv:2401.10334*, 2023.
- Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10586–10613, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.630. URL <https://aclanthology.org/2024.findings-acl.630/>.

Appendix Contents

A	Related work comparison	16
A.1	Selected <i>de novo</i> generation related work	16
B	Implementation and evaluation details	16
B.1	Model architecture and training hyperparameters	16
B.2	Model-size ablation	17
B.3	Inference-temperature sweep	18
B.4	Pair construction details	20
B.5	Evaluation metrics	21
B.6	Baseline methods	21
C	Theoretical Comparison to DPO	24
D	Additional seeded-generation results and ablations	26
D.1	Detailed MoleculeNet results	26
D.2	Distribution-level results	28
D.3	VIN–Similarity trade-offs	28
D.4	Failure-mode analysis	30
D.5	Additional generation examples	32
D.6	Additional low-data evaluation on hERG	33
E	Pretraining and pair-construction ablations	34
E.1	Paired vs. single-molecule vs. no pretraining	34
E.2	Training dynamics and molecular coverage	35
E.3	Validation loss and downstream performance	36
E.4	Property filtering ablation	37
E.5	ZINC-250K logP optimization	38
E.6	Zero-shot logP generalization	38
F	PairIT-RL	39
F.1	Reward and training details	39
F.2	Similarity regularization and reward hacking.	39
F.3	Exploratory multi-property RL	39
F.4	Representative Generations	40
G	Prompt-conditioned property control	50
G.1	Prompt tokens	50
G.2	Single-property control	50
G.3	Two-property control on ZINC-250K	51
G.4	Two-property control on MoleculeNet	51

A Related work comparison

Table 3 summarizes how PairIT compares with related molecular generation and optimization methods along the dimensions most relevant to our setting. *property model-free* denotes methods that do not use iterative property model-guided search during generation.

Method	Seeded	Single Step	Multi-Property	Small Datasets	Property Model-Free	Optimization Mechanism
GraphDiT [Liu et al., 2024]	×	✓	✓	✓	✓	AdaLN
PropEn [Tagasovska et al., 2024]	✓	×	×	✓	✓	Pairs
LLOME [Chen et al., 2025]	✓	×	×	✓	✓	LLM + online black-box optimization
NF-BO [Lee et al., 2025]	✓	×	✓	×	×	Bayesian optimization
HierG2G [Jin et al., 2020]	✓	✓	×	×	✓	Graph translation
DDSBM [Kim et al., 2025]	✓	✓	×	×	✓	Schrödinger bridge
GeLLM4O-C [Dey et al., 2025]	✓	✓	✓	✓	✓	LLM prompt
PairIT	✓	✓	✓	✓	Optional	Pairs

Table 3: Comparison of molecular generation and optimization methods.

A.1 Selected *de novo* generation related work

A large body of work targets *de novo* property-conditioned generation, including equivariant diffusion [Hoogetboom et al., 2022, Vignac et al., 2023, Liu et al., 2024], flow-matching [Joshi et al., 2025], walk-jump-style methods [Pinheiro et al., 2023, Nowara et al., 2024], non-equivariant alternatives with augmentation [Nowara et al., 2025, Shen et al., 2025], and SMILES language models [Chitsaz et al., 2025]; property control in these models typically enters through activations (e.g., AdaLN-style modulation [Huang and Belongie, 2017, Peebles and Xie, 2023]) or inputs.

B Implementation and evaluation details

B.1 Model architecture and training hyperparameters

SMILES are tokenized using a regex-based pre-tokenizer that separates chemically meaningful SMILES units, followed by a fixed 1,225-token WordLevel vocabulary. No BPE or learned merges are used. The vocabulary includes the special tokens <cls>, <pad>, <eos>, <unk>, <mask>, and <sep>. All PairIT experiments were run on multi-GPU cluster nodes with NVIDIA B200 GPUs (180 GB per GPU). Pretraining runs took up to 1.5 hours per run.

Table 4: PairIT architecture and training hyperparameters for the 68M model used in the main experiments.

Setting	Value
Layers	12
Hidden size	648
Attention heads	12
KV heads	12 (MHA)
FFN size	2,048
Vocabulary size	1,225
Maximum sequence length	1,024
Optimizer	AdamW ($\beta_1=0.9$, $\beta_2=0.98$, $\epsilon=10^{-12}$)
Learning rate	10^{-4}
Weight decay	0.01
Warmup steps / fraction	1,000 steps
Scheduler	Constant with warmup
Batch size	64
Gradient accumulation	1
Precision	FP32
Pretraining duration	up to 1.5 hours
Fine-tuning epochs / steps	Best checkpoint by validation loss
Inference temperature	0.5

B.2 Model-size ablation

We analyze the effect of model scaling on generation quality and find that increasing model size yields little to no improvement across key metrics, including validity, VIN (valid, property improved and novel) fraction, target property optimization, and diversity. Performance plateaus already at medium scales, with the 68M model achieving comparable results to larger models. In contrast, the largest models show an increase in memorization, suggesting overfitting to the training data without meaningful gains in generation quality. Based on this trade-off, we use the 68M model as a practical default.

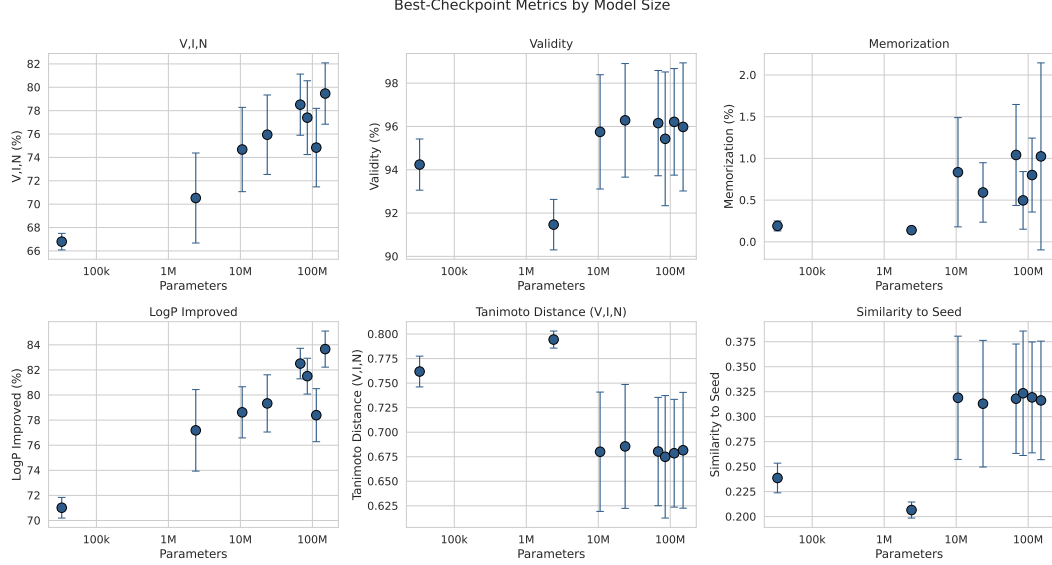


Figure 6: We observe negligible gains from scaling the model size; larger models show increased memorization, indicating overfitting without performance benefits.

B.3 Inference-temperature sweep

The temperature is an important sampling parameter for PairIT that must be tuned. We vary the temperature at inference across a range of 0.1–1.0. It modulates the randomness of softmax sampling during generation. As shown in Figure 7, lower temperatures reduce diversity and lead to near-repetition of the seed, while higher temperatures increase novelty but degrade validity. Higher temperatures tend to yield larger improvements in the target property, at the cost of reduced fidelity. We use the temperature of 0.5 for all datasets.

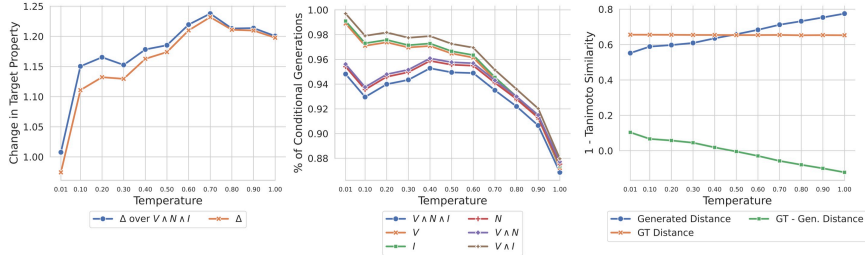


Figure 7: Inference-time temperature sweep (0.1–1.0) showing the trade-off between diversity and validity: lower temperatures favor similarity to the seed, while higher temperatures increase novelty and property improvement at the cost of validity.

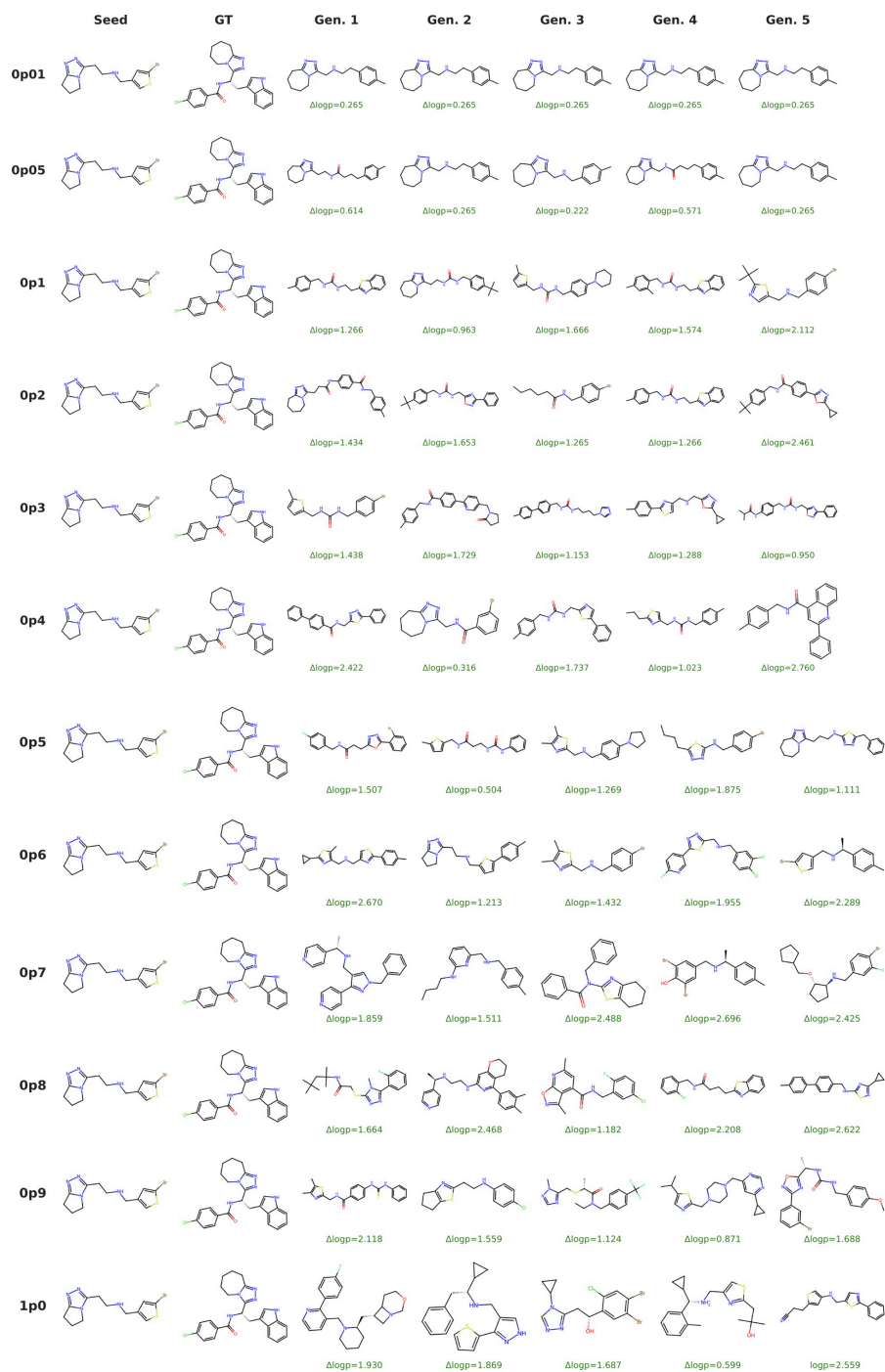


Figure 8: Examples of generations with different temperature settings.

B.4 Pair construction details

For ZINC-250K auxiliary pretraining, we construct directed logP-improvement pairs using 2D Tanimoto similarity [Bender and Glen, 2004] on Morgan fingerprints (ECFP4, radius 2, 2048-bit) [Rogers and Hahn, 2010]. A pair (x, x') is directed from seed x to target x' when the two molecules are structurally similar and x' has higher logP than x . We use this fingerprint-based pairing on ZINC-250K because it provides dense local neighborhoods and many improving examples per seed, which is useful for learning SMILES syntax and local property-improving edits before downstream fine-tuning.

For MoleculeNet fine-tuning and evaluation, we instead construct directed matched molecular pairs (MMPs), which share a common core and differ by a localized fragment replacement [Kramer et al., 2014]. For the binary BACE, BBBP, and HIV tasks, pairs are directed from inactive to active molecules, so the seed is class 0 and the target is class 1. We use MMPs on MoleculeNet to emphasize localized medicinal-chemistry-style edits in the low-data downstream setting, whereas fingerprint similarity is a broader notion of locality and can be affected by fingerprint bit density and molecular size [Wang et al., 2007]. We invoke `mmpdb fragment` and `mmpdb index` [Dalke et al., 2018] with default parameters (single-bond cuts), then post-filter to pairs whose seed and target have a heavy-atom ratio $\min(|x|, |x'|) / \max(|x|, |x'|) \geq 0.6$ to avoid pairing a small seed with a much larger target (or vice versa).

For the two- and ten-property signed-prompt models, MoleculeNet fine-tuning instead includes both orderings of every retained training pair, as in the augmented single-property training construction. Property changes are recomputed after swapping seed and target: a nonzero change becomes its negative, whereas a zero change remains zero. This supplies supervision for both signs of observed changes, without requiring every bin to be populated. For signed-prompt ZINC pretraining, we retain the original directed nearest-neighbor pairs without a positive-change filter; the reverse of every pair is not explicitly appended.

Table 5: Dataset statistics for fine-tuning and evaluation. Usable pairs are direction-improving transformations retained after filtering and used for fine-tuning. BACE, BBBP, and HIV use matched molecular pairs. For hERG, MMPs are too sparse (41 direction-improving training pairs), so we use Tanimoto-neighbor pairs, yielding 76 usable pairs after filtering. Test seeds are class-0 molecules for MoleculeNet and class-1 molecules (blockers) for hERG

	BACE	BBBP	HIV	hERG
	<i>Train/Test</i>	<i>Train/Test</i>	<i>Train/Test</i>	<i>Train/Test</i>
Molecules	798 / 267	517 / 175	1,420 / 475	414 / 111
Pairs	2,546 / 217	512 / 40	1,009 / 165	2,865 / 764
Usable pairs	662 / 51	88 / 14	256 / 46	76 / –
Test seeds	– / 135	– / 88	– / 231	– / 85

B.5 Evaluation metrics

Additional distributional metrics reported in Tables 10–12 are defined below.

Tanimoto similarity to seed (SimSeed, $\uparrow$). Mean Tanimoto similarity between generated molecules and their input seeds, computed using Morgan fingerprints. Higher values indicate greater structural fidelity to the seed.

Fréchet ChemNet Distance (FCD, $\downarrow$). Distance between the distributions of generated molecules and reference molecules in the ChemNet feature space [Preuer et al., 2018]. Lower values indicate that the generated set more closely matches the reference distribution.

Similarity to nearest neighbor (SNN, $\uparrow$). Mean Tanimoto similarity between each novel valid generated molecule and its nearest neighbor in the target-class training set, computed using Morgan fingerprints. Higher values indicate that generated molecules lie closer to the reference distribution in fingerprint space.

Fragment Cosine ($\uparrow$). Cosine similarity between fragment-frequency vectors computed on generated molecules and the reference set. Higher values indicate better agreement in fragment composition.

Scaffold Cosine ($\uparrow$). Cosine similarity between Bemis–Murcko scaffold-frequency vectors of generated molecules and the reference set. Higher values indicate better agreement in scaffold usage.

Coverage ($\uparrow$). Fraction of reference molecules covered by the generated set under the chosen descriptor-space neighborhood criterion. Higher values indicate broader support over the reference distribution.

B.5.1 Property-oracle agreement

To assess sensitivity to the choice of property oracle, we compare the random-forest and MiniMol predictions on benchmark test molecules and on generated molecules (Table 6). Agreement is consistently higher on benchmark molecules than on generated molecules, with the weakest generated-molecule agreement on HIV. For hERG, classification agreement should be interpreted alongside the correlation metrics because the benchmark test set is class-imbalanced.

Table 6: Agreement between the random-forest and MiniMol property oracles. Classification agreement is the fraction of molecules assigned the same active/inactive label by both models.

	Metric	BACE	BBBP	HIV	hERG
Benchmark test	n molecules	261	172	475	111
	Pearson r	0.791	0.819	0.576	0.783
	Spearman ρ	0.787	0.825	0.616	0.685
	Classification agreement (%)	83.5	84.9	69.7	87.6
Generated	n molecules	1618	1389	3751	1353
	Pearson r	0.438	0.631	0.290	0.595
	Spearman ρ	0.424	0.669	0.290	0.635
	Classification agreement (%)	63.0	64.7	51.9	71.8

B.6 Baseline methods

All baselines use the same task splits, task directions, metrics, and final selection procedure as PairIT; below we report method-specific implementation details.

NF-BO. We compare against NF-BO, a normalizing-flow-based Bayesian optimization method that combines a SELFIES-based autoregressive generative model with a Gaussian-process surrogate and trust-region Bayesian optimization. The generative model is an LSTM normalizing flow trained on SELFIES sequences, pretrained on GuacaMol [Brown et al., 2019], and fine-tuned separately for each task. Models were fine-tuned for 50 epochs using Adam with learning rate 10^{-4} and batch size 32.

For BACE, the Random Forest classifier returns $p_{\text{class } 1} = 0$ for every input class-0 test seed; this drives the initial Gaussian-process fit to a degenerate 1×1 kernel with NaN Cholesky, halting BO before any candidate is written. We therefore run NF-BO’s inner BO loop with the Random Forest

classifier trained on the full dataset, including training and test sets, for BACE only; this provides sufficient variation in the surrogate scores for BO to proceed. All 64 generations per BACE seed are otherwise produced by the same NF-BO procedure, and are re-scored downstream with the shared MiniMol oracle at evaluation time. BBBP and HIV use the correct train-only Random Forest classifier throughout.

At inference time, NF-BO performs iterative Bayesian optimization in the learned latent space with a property model budget of 500 evaluations per seed molecule. A deep-kernel Gaussian-process surrogate is trained for 5 epochs at each iteration on accumulated observations, and candidate molecules are proposed using TuRBO-based Thompson sampling with acquisition batch size 100; the top-10 current observations are used as trust-region centers at each step, and a running top-64 set of valid generations is maintained across the budget. Generated molecules are decoded via SELFIES and filtered for validity. Optimization objectives maximize either logP improvement or predicted class-1 probability, subject to a structural similarity constraint enforced via Tanimoto similarity (Morgan fingerprints, radius 2, 2048 bits) with a minimum threshold of 0.4 and a soft penalty below that threshold. Generation terminates upon exhausting the property model budget.

DDSBM (Bridge and SB variants). We compare against DDSBM, a discrete diffusion-based Schrödinger Bridge model for graph-to-graph molecular transformation. DDSBM operates directly on molecular graphs and learns stochastic forward and reverse diffusion processes that transport molecules from a source distribution to a target distribution using entropy-regularized optimal transport. The model is implemented with a graph transformer backbone.

We evaluate two DDSBM variants that differ only in the number of Schrödinger Bridge refinement steps. *DDSBM-Bridge* performs a single forward training pass from source to target (one outer loop), providing a fast but coarse approximation of the bridge. *DDSBM-SB* applies 5 outer iterative-proportional-fitting loops, corresponding to 10 training runs alternating backward and forward updates, yielding a tighter approximation to the true Schrödinger bridge at increased computational cost. Both variants are fine-tuned from a ZINC-250K paired pretraining checkpoint (Bridge and SB pretrained checkpoints are kept separate). Each training run is performed for 300 epochs, resulting in 300 and 3,000 effective training epochs for Bridge and SB, respectively.

At inference time, generation is performed via 100-step discrete reverse diffusion from the input molecule without explicit similarity constraints or property model budgets. We generate 64 stochastic samples per seed and evaluate generations using the metrics described in Appendix B.5.

HierG2G. We compare against Hierarchical Graph-to-Graph (HierG2G), a variational graph translation model that generates molecules by hierarchically assembling substructures into molecular graphs. HierG2G represents molecules using coupled tree and graph representations and performs conditional translation via a hierarchical variational graph neural network. We use the public HierG2G implementation and fine-tune models separately for each task, starting from a checkpoint pretrained on ZINC-250K.

For MoleculeNet tasks (BACE, BBBP, and HIV), we fine-tune task-specific translation models for 12 epochs using Adam (learning rate 10^{-3} , batch size 32, latent dimension 4), following default hyperparameters from prior work. At inference time, conditional translation is performed by sampling multiple candidate molecules per input via stochastic decoding in the learned latent space, with an increased maximum decoding length to reduce fragment generation. We generate 64 candidates per seed molecule using the checkpoint at epoch 11 and evaluate generations using the metrics described in Appendix B.5.

MMP baseline. We include a property-aware matched molecular pair baseline (MMP-PROP), a non-learning cheminformatics enumeration method. Starting from each seed molecule, we queried an *mmpdb* database of matched molecular transformations built from the training set. We augmented the database with training-set property scores from a task-specific classifier, so that each transformation carried an estimated property shift. For each test seed, we first enumerated candidate edits using *mmpdb transform -property*, which ranks transformations by their learned average property change; when needed, we supplemented the candidate pool with additional *mmpdb generate* proposals. This produced up to 64 candidate molecules per seed. As in the other 64-sample baselines, we then evaluated the candidate pool with the common benchmark metrics and reported the best-5 subset under the same selection procedure used for all methods.

GraphDiT. We used the publicly released pretrained GraphDiT checkpoints for the MoleculeNet datasets (BACE, BBBP, and HIV) as released by the original authors, without any retraining or fine-tuning on our side. The pretrained models use dropout for classifier-free guidance regularization during training (`drop_condition` = 0.01). For generation, we sampled 10,000 molecules per dataset using classifier-free guidance with `guide_scale` = 2. These generations are de novo samples from random noise and are therefore not conditioned on an input molecule.

GeLLM4O-C. We compare against GeLLM4O-C [Dey et al., 2025], an instruction-tuned 7B-parameter Mistral LLM (Mistral-7B-Instruct-v0.3) trained with LoRA adapters on the C-MuMOInstruct multi-property optimization corpus. We use the authors’ released BBBP, logP, and QED adapters untouched. For BACE and HIV, we fine-tune task-specific LoRA adapters ($r=16$, $\alpha=16$, dropout 0.05, targeting q/k/v/o/gate/up/down projections and the LM head; 0.58% trainable parameters, no quantization) using their trainer on directed mmpdb pairs converted to C-MuMOInstruct JSON format (`source` = inactive molecule, `target` = active molecule; direction encoded categorically as "increase" via the instruction template, with numeric activity from the RF oracle not surfaced in the prompt). Models were trained with AdamW at learning rate 10^{-4} , cosine schedule, warmup ratio 0.1, effective batch size 16 (per-device 1 with 16-step gradient accumulation), and `cutoff_len=4096`. Training uses `train_on_inputs=False` so loss is computed only on the response SMILES. At inference, we sample 64 candidates per seed via stochastic decoding and evaluate under the shared selection protocol described in Appendix B.5.

C Theoretical Comparison to DPO

Because PairIT is trained on paired examples, a preference objective such as DPO [Rafailov et al., 2023] is a natural alternative. We instead use conditional likelihood: prior work has found DPO to underperform SFT and SFT followed by RL on related biomolecular sequence-optimization tasks [Chen et al., 2025], and DPO’s contrastive objective can shift probability mass toward outputs outside the observed preference pair [Razin et al., 2025]. Below, we provide a complementary analytical comparison of the training signals produced by the two objectives.

We compare DPO’s pair-level parameter gradient to an aggregate tokenwise supervision proxy for PairIT as matched molecules become more different. The comparison is intentionally asymmetric: DPO naturally produces a single sequence-level gradient through one sigmoid margin term, whereas PairIT distributes supervision across multiple edited token positions. Because token-level parameter gradients in PairIT can partially cancel when summed in parameter space, we track an aggregate tokenwise proxy rather than the norm of the full PairIT parameter gradient.

Let x denote the seed molecule, x' the matched target molecule, and c the conditioning information. PairIT is trained with

$$\mathcal{L}_{\text{PairIT}}(\theta) = -\log q_{\theta}(x' | x, c) = \sum_{t=1}^T \ell_t(\theta), \quad \ell_t(\theta) := -\log q_{\theta}(x'_t | x'_{<t}, x, c). \quad (1)$$

That is, PairIT uses autoregressive cross-entropy with the loss computed only on the target tokens of the second molecule x' , conditioned on the seed molecule x and the control tokens c .

Note that

$$\nabla_{\theta} \mathcal{L}_{\text{PairIT}}(\theta) = \sum_{t=1}^T \nabla_{\theta} \ell_t(\theta). \quad (2)$$

These token-level gradients are vector-valued and live in the same shared parameter space, so they may partially cancel. For this reason, we will compare DPO’s pair-level parameter gradient to a tokenwise supervision proxy for PairIT rather than to $\|\nabla_{\theta} \mathcal{L}_{\text{PairIT}}\|$ directly.

Let $d_{\text{tok}}(x, x')$ denote a token edit distance between x and x' , and let $E(x, x') \subseteq \{1, \dots, T\}$ denote the set of target positions affected by these edits. We assume there exists a constant $c_{\text{edit}} > 0$ such that $|E(x, x')| \geq c_{\text{edit}} d_{\text{tok}}(x, x')$. That is, the number of target positions affected by the edits grows at least linearly with the token edit distance.

For DPO, we use the same conditioning input (x, c) and a preference pair (x^+, x^-) , where x^+ is the preferred output and x^- is the less preferred output. In the matched-pair setting, x^+ corresponds to the improved target molecule x' .

Let $z_t(\theta) \in \mathbb{R}^V$ denote the decoder logits at target position t , where V is the vocabulary size, and let

$$p_t(\theta) := \text{softmax}(z_t(\theta)). \quad (3)$$

We measure the PairIT supervision strength by the following aggregate logit-space tokenwise supervision proxy:

$$S_{\text{PairIT}}^{\text{logit}}(\theta; x, x', c) := \sum_{t \in E(x, x')} \|\nabla_{z_t} \ell_t(\theta)\|_2. \quad (4)$$

Assume:

1. **Non-saturated edited tokens.** There exists $p_{\max} < 1$ such that for every edited position $t \in E(x, x')$,

$$p_t(\theta)_{x'_t} \leq p_{\max}. \quad (5)$$

That is, on edited positions, the model is not already assigning probability arbitrarily close to 1 to the correct target token.

2. **Bounded score gradients for DPO.** There exists $M > 0$ such that

$$\|\nabla_{\theta} \log q_{\theta}(z | x, c)\| \leq M \quad \text{for all relevant } (x, c, z). \quad (6)$$

3. **Margin growth with pair difference.** For a pair consisting of a more preferred output x^+ and a less preferred output x^- under the same conditioning (x, c) , define

$$Z_\theta := \log q_\theta(x^+ | x, c) - \log q_\theta(x^- | x, c) - \log q_0(x^+ | x, c) + \log q_0(x^- | x, c). \quad (7)$$

Assume there exists $\alpha > 0$ such that

$$Z_\theta \geq \alpha d_{\text{tok}}(x^+, x^-). \quad (8)$$

For cross-entropy, the gradient with respect to the logits at position t is

$$\nabla_{z_t} \ell_t(\theta) = p_t(\theta) - e_{x'_t}, \quad (9)$$

where $e_{x'_t} \in \mathbb{R}^V$ is the one-hot vector of the target token x'_t . Therefore,

$$\|\nabla_{z_t} \ell_t(\theta)\|_2 = \|p_t(\theta) - e_{x'_t}\|_2 \quad (10)$$

$$\geq |(p_t(\theta) - e_{x'_t})_{x'_t}| \quad (11)$$

$$= 1 - p_t(\theta)_{x'_t} \quad (12)$$

$$\geq 1 - p_{\max} \quad \text{for every } t \in E(x, x'). \quad (13)$$

Summing over edited positions gives

$$S_{\text{PairIT}}^{\text{logit}}(\theta; x, x', c) \geq (1 - p_{\max}) |E(x, x')|. \quad (14)$$

Using $|E(x, x')| \geq c_{\text{edit}} d_{\text{tok}}(x, x')$, we obtain

$$S_{\text{PairIT}}^{\text{logit}}(\theta; x, x', c) \geq (1 - p_{\max}) c_{\text{edit}} d_{\text{tok}}(x, x'). \quad (15)$$

Thus, the aggregate logit-space tokenwise supervision proxy grows at least linearly with the sequence difference between the matched molecules.

For scalar ordered properties, define the matched-pair property gap as

$$\Delta y_{\text{pair}} := g(x') - g(x) > 0. \quad (16)$$

If, in addition, the property gap is controlled by edit size,

$$\Delta y_{\text{pair}} \leq \ell d_{\text{tok}}(x, x'), \quad (17)$$

then the PairIT bound implies

$$S_{\text{PairIT}}^{\text{logit}}(\theta; x, x', c) \geq \frac{(1 - p_{\max}) c_{\text{edit}}}{\ell} \Delta y_{\text{pair}}. \quad (18)$$

For the DPO preference pair (x^+, x^-) , the DPO loss is

$$\mathcal{L}_{\text{DPO}}(\theta) = -\log \sigma(\beta Z_\theta), \quad (19)$$

where σ is the sigmoid function and $\beta > 0$ is the DPO inverse-temperature parameter. Since q_0 is fixed,

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\theta) = -\beta \sigma(-\beta Z_\theta) [\nabla_\theta \log q_\theta(x^+ | x, c) - \nabla_\theta \log q_\theta(x^- | x, c)]. \quad (20)$$

Using the bounded score-gradient assumption,

$$\|\nabla_\theta \mathcal{L}_{\text{DPO}}\| \leq 2M\beta e^{-\beta Z_\theta}. \quad (21)$$

For the DPO preference pair, define

$$\Delta y_{\text{DPO}} := g(x^+) - g(x^-) > 0. \quad (22)$$

Likewise, if the DPO margin grows with property gap, $Z_\theta \geq \alpha \Delta y_{\text{DPO}}$, then

$$\|\nabla_\theta \mathcal{L}_{\text{DPO}}\| \leq 2M\beta e^{-\beta \alpha \Delta y_{\text{DPO}}}. \quad (23)$$

Takeaway. PairIT uses autoregressive cross-entropy on matched pairs, with the loss computed on the target tokens of the second molecule. As matched molecules become more different, more target positions are affected, so PairIT receives supervision at more edited token positions. This yields an aggregate tokenwise supervision proxy that grows at least linearly with sequence difference. Under the margin-growth assumption above, DPO instead applies a single sigmoid margin term to the entire pair, and the corresponding pair-level parameter gradient becomes exponentially small as the learned preference margin grows. Thus, PairIT is less prone to this particular sequence-level saturation effect.

D Additional seeded-generation results and ablations

D.1 Detailed MoleculeNet results

Figure 9 reports aggregate VIN, SimSeed, and QED with bootstrap confidence intervals. Tables 7–9 provide the corresponding numeric results. Entries are means across seeds after within-seed canonical-SMILES de-duplication and final top-5 selection; metrics are defined in Appendix B.5.

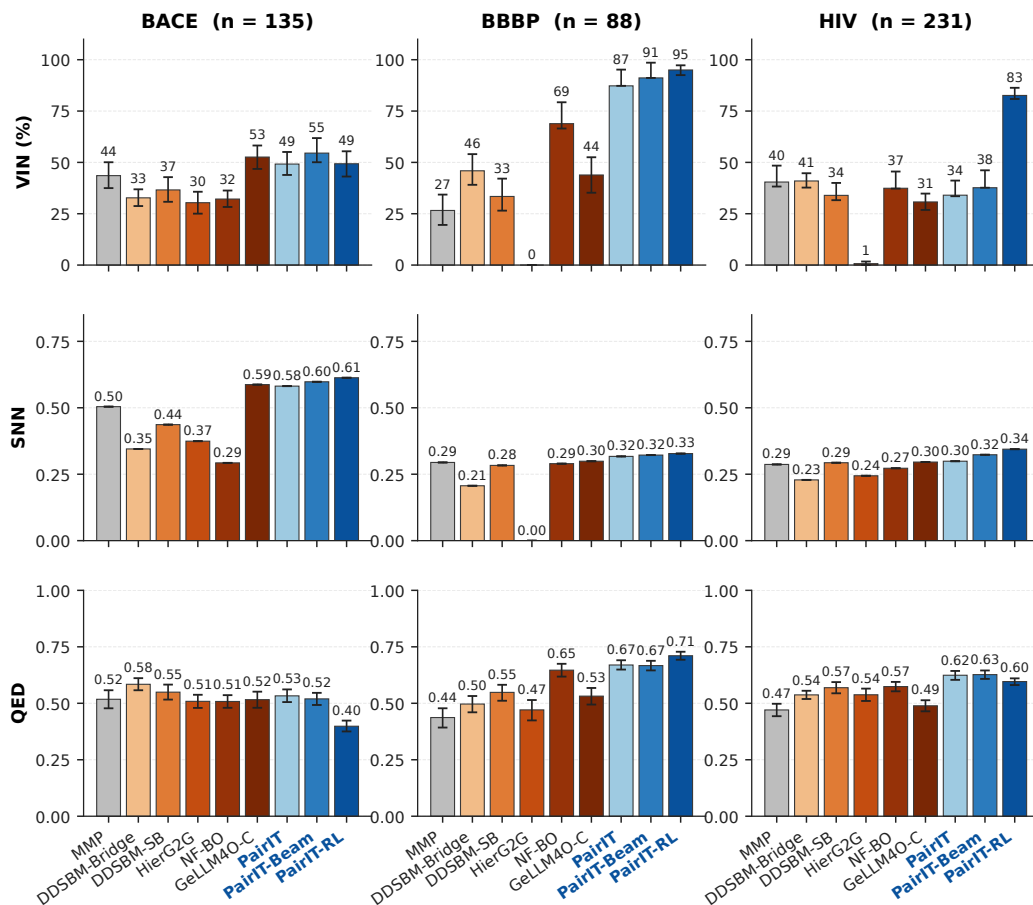


Figure 9: Aggregate MoleculeNet seeded-generation metrics corresponding to Figure 2. Bars show mean performance across test seeds after within-seed de-duplication and final top-5 selection; error bars are bootstrap 95% confidence intervals over seeds. VIN is the fraction of retained generations that are valid, novel, and improved. Similarity to nearest neighbor (SNN) is computed over novel and valid generations only. QED is the quantitative estimate of drug-likeness.

Table 7: MoleculeNet BACE bioactivity improvement ($0 \rightarrow 1$) under seeded generation. Best 5 of 64 per seed selected after within-seed canonical-SMILES de-duplication, so each retained molecule is distinct within its seed. Per-seed bootstrap 95% CIs for VIN and SNN are shown in Fig. 9. SNN is the mean per-generation Tanimoto to the nearest class-1 training molecule, computed over novel valid generations only.

Model	VIN	QED	Valid	Novel	Unique	Diverse	Improved	SNN
MMP	43.56	0.52	100.00	99.11	100.00	38.00	44.30	0.50
GraphDiT	–	–	87.32	100.00	11.28	–	86.60	–
DDSBM-Bridge	32.74	0.58	100.00	<u>99.85</u>	100.00	79.00	32.89	0.35
DDSBM-SB	36.59	<u>0.55</u>	92.89	<u>97.77</u>	92.15	46.00	41.16	0.44
HierG2G	30.37	0.50	79.85	93.69	74.81	62.00	40.59	0.37
NF-BO	32.15	0.51	100.00	100.00	100.00	<u>74.00</u>	32.15	0.29
GeLLM4O-C	<u>52.59</u>	0.51	<u>96.89</u>	92.35	<u>96.74</u>	42.00	<u>60.34</u>	0.59
PairIT	49.19	0.54	96.74	87.90	96.00	54.00	58.80	0.58
PairIT-Beam	54.52	0.53	94.52	88.71	94.52	51.00	65.52	0.60
PairIT-RL	49.33	0.39	92.30	100.00	92.30	27.00	53.45	0.61

Table 8: MoleculeNet BBBP bioactivity improvement ($0 \rightarrow 1$) under seeded generation. Best 5 of 64 per seed selected after within-seed canonical-SMILES de-duplication, so each retained molecule is distinct within its seed. Per-seed bootstrap 95% CIs for VIN and SNN are shown in Fig. 9. SNN is the mean per-generation Tanimoto to the nearest class-1 training molecule, computed over novel valid generations only.

Model	VIN	QED	Valid	Novel	Unique	Diverse	Improved	SNN
MMP	26.59	0.44	99.55	100.00	99.55	43.00	26.71	0.29
GraphDiT	–	–	83.93	100.00	20.33	–	82.30	–
DDSBM-Bridge	45.91	0.51	91.14	100.00	91.14	88.00	50.37	0.21
DDSBM-SB	33.41	0.55	88.86	95.91	85.91	57.00	39.68	0.28
HierG2G	0.00	0.47	20.00	0.00	0.00	–	0.00	–
NF-BO	68.86	0.65	90.45	98.74	89.32	70.00	77.10	0.29
GeLLM4O-C	43.86	0.53	100.00	100.00	100.00	50.00	43.86	0.30
PairIT	87.27	<u>0.67</u>	94.09	<u>99.76</u>	93.86	<u>78.00</u>	92.98	0.32
PairIT-Beam	<u>90.91</u>	0.67	94.32	100.00	94.32	73.00	96.39	0.32
PairIT-RL	95.00	0.71	<u>99.55</u>	100.00	<u>99.55</u>	66.00	<u>95.43</u>	0.33

Table 9: MoleculeNet HIV bioactivity improvement ($0 \rightarrow 1$) under seeded generation. Best 5 of 64 per seed selected after within-seed canonical-SMILES de-duplication, so each retained molecule is distinct within its seed. Per-seed bootstrap 95% CIs for VIN and SNN are shown in Fig. 9. SNN is the mean per-generation Tanimoto to the nearest class-1 training molecule, computed over novel valid generations only.

Model	VIN	QED	Valid	Novel	Unique	Diverse	Improved	SNN
MMP	40.43	0.47	92.03	99.91	91.95	41.00	43.97	0.29
GraphDiT	–	–	80.16	100.00	16.55	–	95.55	–
DDSBM-Bridge	<u>40.95</u>	0.54	97.66	<u>99.91</u>	97.66	85.00	42.02	0.23
DDSBM-SB	33.94	0.56	87.53	97.03	86.15	61.00	40.70	0.29
HierG2G	0.69	0.53	21.04	7.00	1.47	71.00	<u>47.06</u>	0.24
NF-BO	37.40	0.57	87.01	98.31	85.54	70.00	43.72	0.27
GeLLM4O-C	30.74	0.48	95.76	88.16	92.73	56.00	41.55	0.30
PairIT	33.94	0.63	87.79	99.70	87.53	<u>73.00</u>	38.77	0.30
PairIT-Beam	37.75	<u>0.63</u>	87.36	100.00	87.36	64.00	43.21	0.32
PairIT-RL	82.60	0.60	<u>97.06</u>	99.82	<u>96.88</u>	52.00	85.25	0.34

D.2 Distribution-level results

Tables 10–12 report distribution-level metrics comparing generated molecules to the class-1 train reference set, as the test set is too small to provide a stable comparison and would result in high variance in the estimated metrics. Definitions of these metrics are given in Appendix B.5.

Table 10: MoleculeNet BACE bioactivity improvement ($0 \rightarrow 1$): distribution metrics on best-5-of-64 real-only selection.

Model	SimSeed $\uparrow$	FCD $\downarrow$	FragCos $\uparrow$	ScafCos $\uparrow$	Cov $\uparrow$
MMP	0.71	10.80	0.92	0.54	1.00
DDSBM-Bridge	0.22	10.32	0.85	0.46	1.00
DDSBM-SB	0.44	10.16	0.90	0.41	1.00
HierG2G	0.33	15.56	0.89	0.38	1.00
NF-BO	0.30	21.19	0.88	0.17	1.00
GeLLM4O-C	0.54	7.16	0.96	0.71	1.00
PairIT	0.37	9.27	0.96	0.61	1.00
PairIT-Beam	0.31	10.29	0.93	0.65	1.00
PairIT-RL	0.24	27.47	0.88	0.19	1.00

Table 11: MoleculeNet BBBP bioactivity improvement ($0 \rightarrow 1$): distribution metrics on best-5-of-64 real-only selection.

Model	SimSeed $\uparrow$	FCD $\downarrow$	FragCos $\uparrow$	ScafCos $\uparrow$	Cov $\uparrow$
MMP	0.67	24.07	0.88	0.54	0.89
DDSBM-Bridge	0.12	17.61	0.78	0.51	0.89
DDSBM-SB	0.39	19.13	0.83	0.50	0.89
HierG2G	—	—	—	—	—
NF-BO	0.27	12.90	0.91	0.74	0.89
GeLLM4O-C	0.56	21.06	0.89	0.64	0.89
PairIT	0.17	16.49	0.88	0.72	0.78
PairIT-Beam	0.12	19.90	0.87	0.71	0.78
PairIT-RL	0.23	17.75	0.88	0.72	0.78

Table 12: MoleculeNet HIV bioactivity improvement ($0 \rightarrow 1$): distribution metrics on best-5-of-64 real-only selection.

Model	SimSeed $\uparrow$	FCD $\downarrow$	FragCos $\uparrow$	ScafCos $\uparrow$	Cov $\uparrow$
MMP	0.67	9.44	0.93	0.36	0.73
DDSBM-Bridge	0.17	13.09	0.87	0.35	0.50
DDSBM-SB	0.33	10.82	0.91	0.41	0.50
HierG2G	0.28	45.42	0.51	0.05	0.32
NF-BO	0.34	8.89	0.94	0.48	0.55
GeLLM4O-C	0.50	9.19	0.93	0.38	0.73
PairIT	0.23	14.91	0.89	0.30	0.45
PairIT-Beam	0.17	19.64	0.83	0.17	0.41
PairIT-RL	0.24	18.87	0.61	0.37	0.32

D.3 VIN–Similarity trade-offs

Figures 10–12 show the VIN–SimSeed trade-off across test seeds. As a random-reference baseline, for each inactive test seed we sample one molecule from the training set, either uniformly from all molecules or from the active class. Mean Tanimoto similarity to the seed is 0.158/0.159 for BACE, 0.109/0.101 for BBBP, and 0.093/0.093 for HIV (random molecule/random active, respectively). Thus, the similarities produced by the generative methods remain well above the random-reference regime.

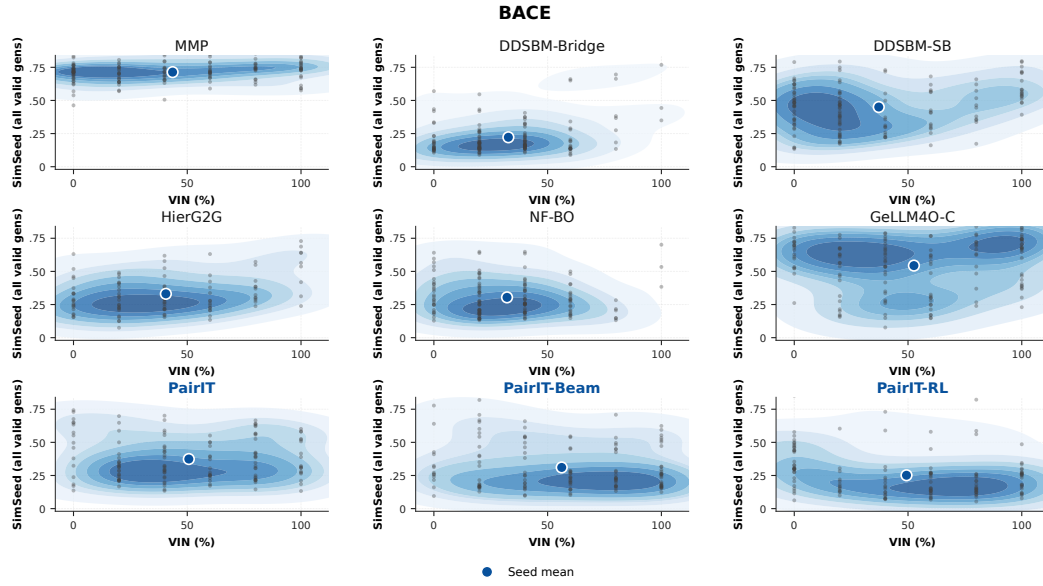


Figure 10: Scatter plots of VIN vs. SimSeed showing the distribution of all generations for all seeds on BACE.

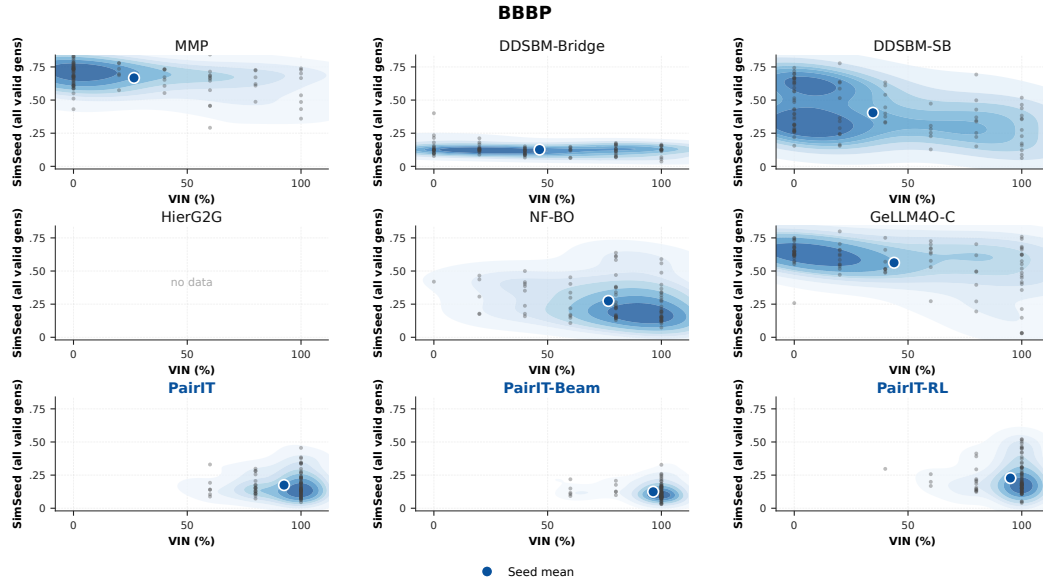


Figure 11: Scatter plots of VIN vs. SimSeed showing the distribution of all generations for all seeds on BBBP. HierG2G generations are excluded because they all contain duplicates of the seed.

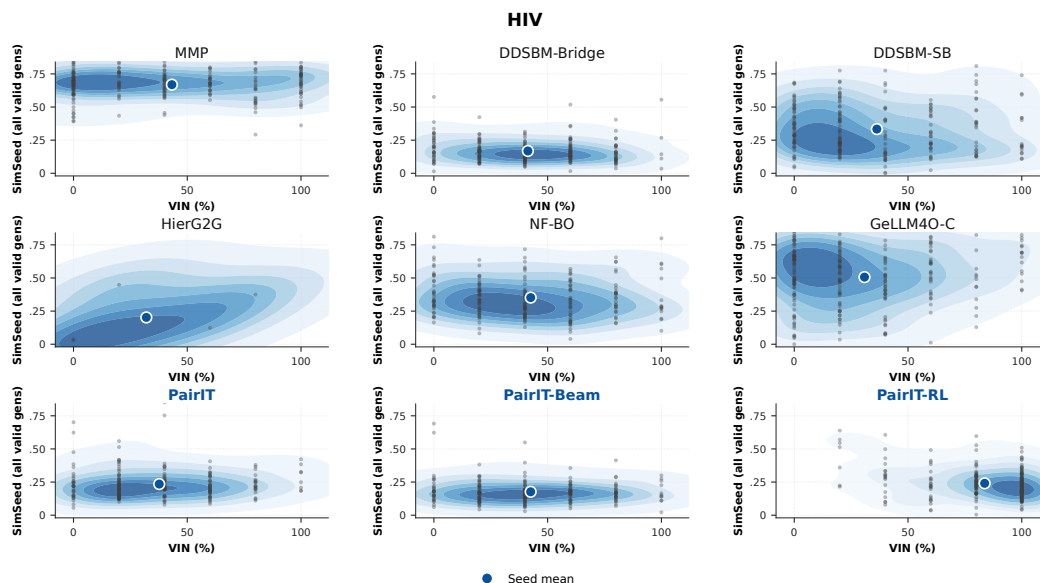


Figure 12: Scatter plots of VIN vs. SimSeed showing the distribution of all generations for all seeds on HIV. The majority of HierG2G generations contain duplicates of the seed.

D.4 Failure-mode analysis

Figures 13–15 show the sources of error for generated molecules that do not satisfy VIN. The green segment corresponds to VIN success; the remaining segments indicate why a retained generation was not counted as VIN. Categories are assigned using the priority order shown in the legend, so each generation is counted once.

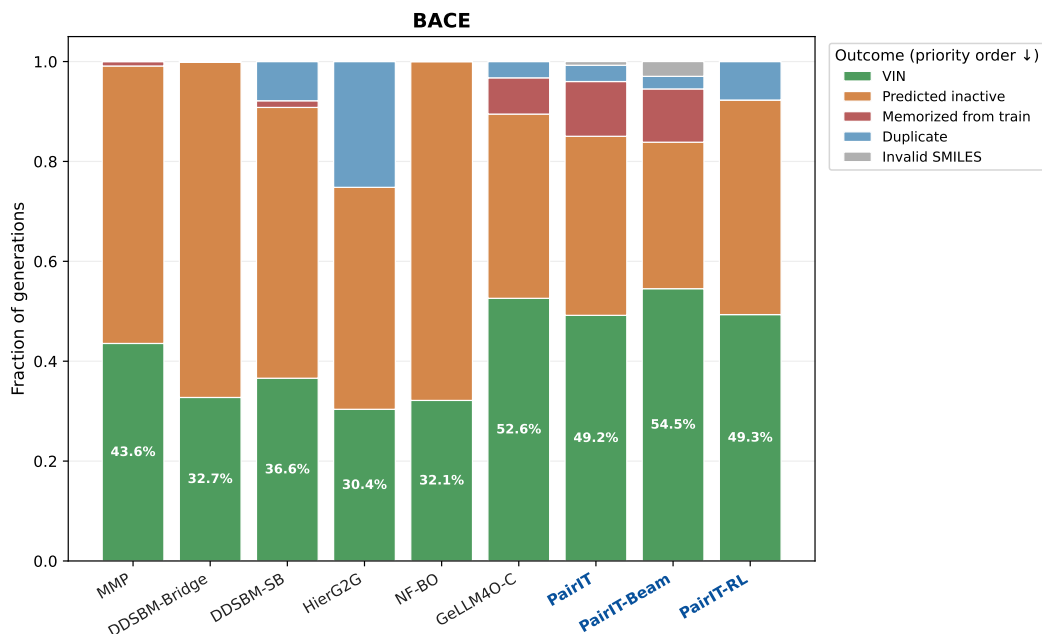


Figure 13: Failure-mode breakdown on MoleculeNet BACE. Bars show VIN success in green and the sources of error for generations not counted as VIN.

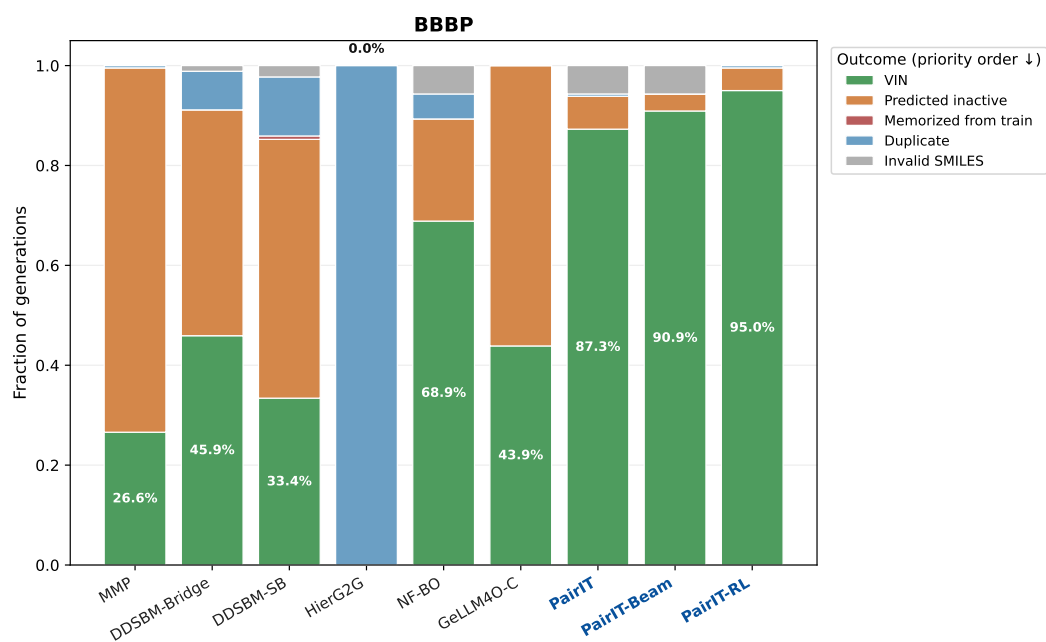


Figure 14: Failure-mode breakdown on MoleculeNet BBBP. Bars show VIN success in green and the sources of error for generations not counted as VIN.

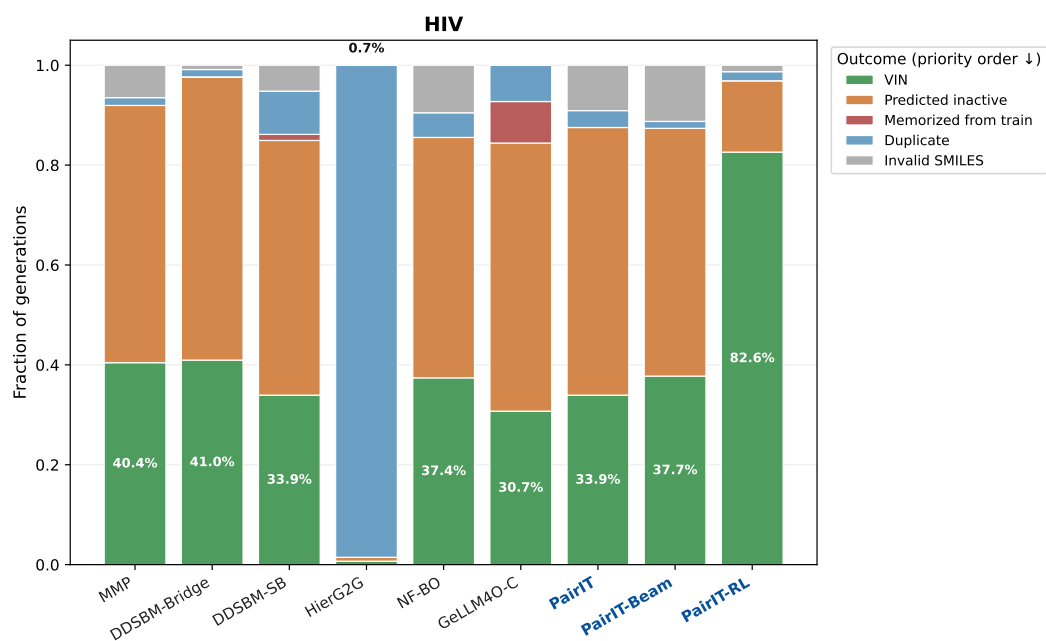


Figure 15: Failure-mode breakdown on MoleculeNet HIV. Bars show VIN success in green and the sources of error for generations not counted as VIN.

D.5 Additional generation examples

Figure 16 shows additional PairIT and PairIT-Beam generations on BACE, BBBP, and HIV.

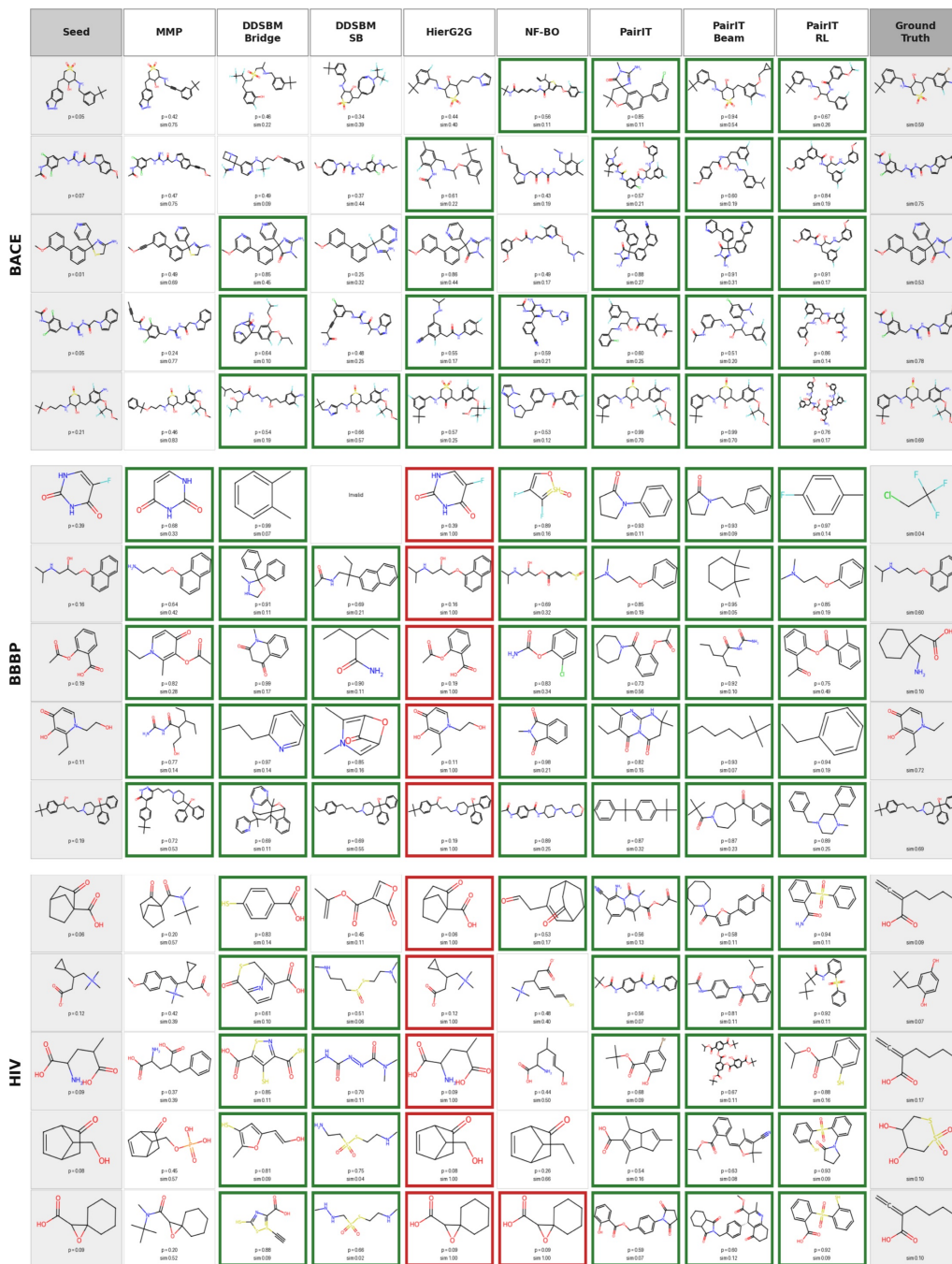


Figure 16: Additional example generations for MoleculeNet seeded optimization. Generations with predicted class probability greater than 0.5 are highlighted in green and seed duplicates in red.

D.6 Additional low-data evaluation on hERG

We additionally evaluate on hERG from Therapeutics Data Commons (TDC) [Huang et al., 2021, Wang et al., 2016] as a distinct low-data toxicity endpoint. hERG provides substantially sparser usable pair supervision than BACE. MMPs are especially sparse, yielding only 41 blocker-to-non-blocker training pairs, so we instead use local Tanimoto-neighbor pairs. Of 2,865 initial training pairs, 76 unique direction-improving pairs remain after canonical-SMILES de-duplication and ε -filtering (Table 5). This corresponds to approximately 0.18 usable pairs per training molecule, similar to HIV and substantially below BACE (0.83 pairs per molecule). Pairs are directed from blocker to non-blocker.

Table 13 reports the resulting seeded-generation performance.

Despite the sparse pair supervision, PairIT obtains 72.9% VIN and PairIT-Beam 79.8%, with PairIT-Beam performing on par with the strongest baselines. PairIT-RL does not improve over SFT on this task. hERG also illustrates the importance of considering structural fidelity together with property improvement: MMP and GeLLM4O-C remain substantially closer to the input molecule but achieve much lower VIN, whereas the highest-VIN methods have relatively low SimSeed. Thus, high VIN alone does not imply a localized molecular improvement in this setting.

This issue is especially important for hERG, where improvement is defined as blocker-to-non-blocker: methods can achieve high VIN while making large structural departures from the seed. Indeed, HierG2G, DDSBM-SB, and PairIT-Beam all achieve high VIN with SimSeed < 0.2 , illustrating that property improvement and localized molecular editing need not coincide.

Table 13: Seeded molecular optimization on hERG. SimSeed measures Tanimoto similarity to the input seed among VIN generations.

Method	Valid %	Improved %	Novel %	VIN %	SimSeed
MMP	91.76	10.77	100.00	9.88	0.74
DDSBM-Bridge	100.00	19.76	100.00	19.76	0.30
DDSBM-SB	93.88	87.91	97.74	80.71	0.19
HierG2G	99.06	81.67	<u>99.76</u>	80.71	0.14
NF-BO	98.59	22.30	99.52	21.88	0.47
GeLLM4O-C	100.00	29.18	100.00	29.18	<u>0.60</u>
PairIT	100.00	72.94	100.00	72.94	0.20
PairIT-Beam	94.12	<u>84.75</u>	100.00	<u>79.76</u>	0.13
PairIT-RL	<u>99.76</u>	40.19	<u>99.76</u>	40.00	0.39

E Pretraining and pair-construction ablations

E.1 Paired vs. single-molecule vs. no pretraining

To isolate the contribution of paired-molecule supervision during ZINC pretraining, we compare PairIT against single-molecule pretraining variants and against no pretraining at all. All variants share the same architecture (CLM-68M) and BACE fine-tuning pipeline; only the ZINC pretraining objective differs. Results are averaged across five replicate training seeds (four for Tanimoto $k=2$; two for the no-pretraining reference).

Single-molecule (uniform) refers to pretraining with single molecules from ZINC-250K (not pairs), sampled uniformly at random — the model sees the full ZINC distribution. *Single-molecule ($\varepsilon=1$ seeds)* refers to pretraining with single low-property molecules from ZINC-250K restricted to the subset that appears as a seed molecule in the $\varepsilon=1$ paired set (i.e. the first elements of the training pairs used for the paired variants). *Tanimoto (k nbr)* refers to pretraining with k -nearest neighbor pairs, where each seed is paired with its up to k Tanimoto-nearest neighbors that satisfy $\Delta \log P > 1$; “all nbr” uses our default of up to 100 neighbors per seed.

Table 14 shows that pretraining on pairs of similar molecules is central to PairIT; single-molecule pretraining from the same dataset is less effective. Performance is not monotonic in the number of neighbors, but more diverse ZINC-250K pairs modestly increase seed similarity and expose the model to more unique molecules before early stopping, motivating our default of up to 100 neighbors per seed.

Two failure modes are visible. Single-molecule (uniform) inflates Improved (45.5%) because the model emits seed-agnostic drug-like molecules that the oracle scores as active, but SimSeed collapses to 0.20 and VIN drops to 40.0%. Small- k Tanimoto pairing (e.g., $k = 1$) keeps generations close to the seed (SimSeed 0.49), but produces enough near-duplicate canonicals that fewer than five unique valid, improved, novel candidates remain to fill the top-5 slots, giving 28.0% VIN. Thus, pretraining with diverse pairs and early stopping is the key component.

Table 14: Downstream BACE molecule-improvement with varying ZINC pretraining objectives. Valid, Improved, and Novel are computed over all 64 raw generations per seed, while VIN and SimSeed are computed over the top-5 selected candidates per seed.

Method	Valid %	Improved %	VIN %	SimSeed	Novel %
<i>No pretraining</i>	40.9	25.3	20.0	0.42	36.3
Single-molecule (uniform)	75.2	45.5	40.0	0.20	74.8
Single-molecule ($\varepsilon=1$ seeds)	69.3	37.0	44.0	0.21	70.0
Tanimoto (1 neighbor)	87.7	44.6	28.0	0.49	63.5
Tanimoto (2 neighbor)	89.2	42.9	45.0	0.52	71.0
Tanimoto (3 neighbor)	89.0	39.6	32.0	0.52	73.2
Tanimoto (5 neighbor)	88.8	42.9	40.0	0.55	69.3
Tanimoto (10 neighbor)	91.2	41.0	48.0	0.55	75.8
Tanimoto (20 neighbor)	92.0	36.2	36.0	0.57	79.0
Tanimoto (all neighbor)	91.6	40.8	64.0	0.56	79.2

Figure 17 compares the main pretrained PairIT variants with a PairIT model trained from random initialization using only the MoleculeNet directed pairs. The no-pretraining model uses the same 68M architecture and the same MoleculeNet fine-tuning data, but does not receive ZINC-250K auxiliary pretraining. Without pretraining, VIN collapses on all three MoleculeNet tasks, indicating that the small MoleculeNet pair sets alone are insufficient for learning valid, property-improving SMILES edits.

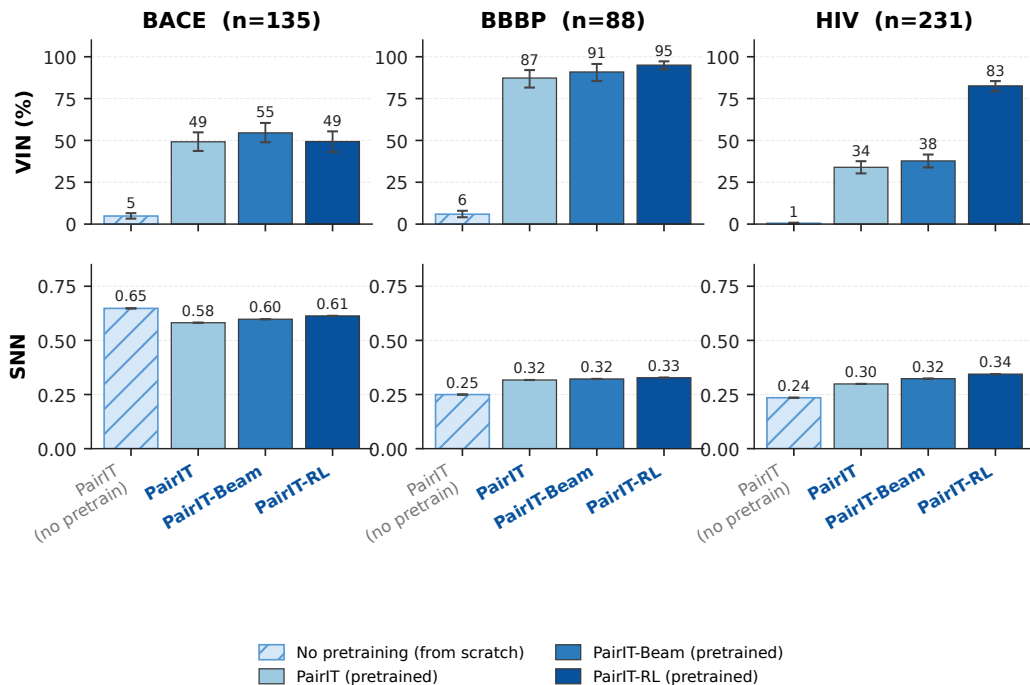


Figure 17: Effect of ZINC-250K pretraining on PairIT. Top row: VIN; bottom row: SNN computed over VIN molecules. The no-pretraining model uses the same architecture and MoleculeNet directed pairs as PairIT, but is trained from random initialization.

Figure 18 shows that the main failure mode without pretraining is invalid SMILES, whereas the pretrained PairIT variants rarely fail validity. This suggests that pretraining on a moderate-size dataset is necessary for the model to learn the SMILES grammar.

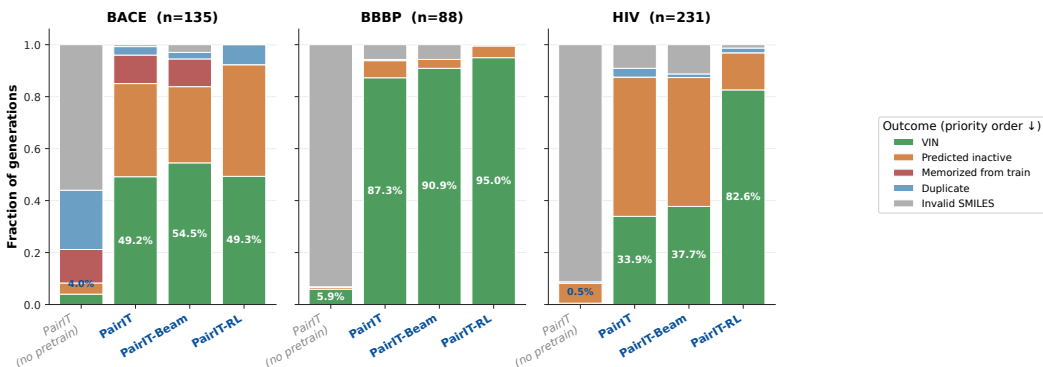


Figure 18: Failure-mode breakdown for the pretraining ablation. Without ZINC-250K pretraining, PairIT is dominated by invalid SMILES; pretrained PairIT variants rarely fail validity.

E.2 Training dynamics and molecular coverage

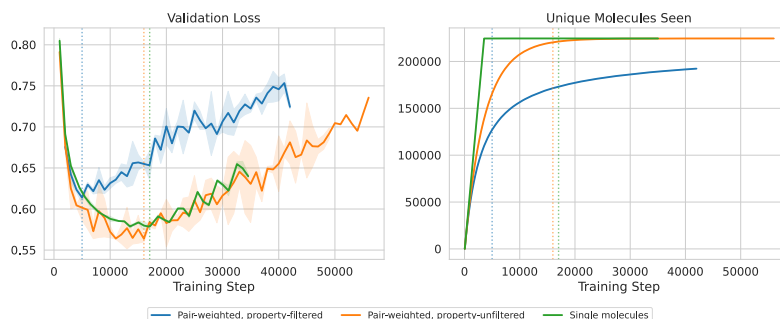


Figure 19: Variance in overfitting behavior, and the number of unique molecules seen at the time of overfitting, for different ZINC-based data distributions (pairs with and without property-based filtering, and unconditional single-molecule generation). Dotted lines indicate the point of early stopping, i.e. minimal validation loss.

E.3 Validation loss and downstream performance

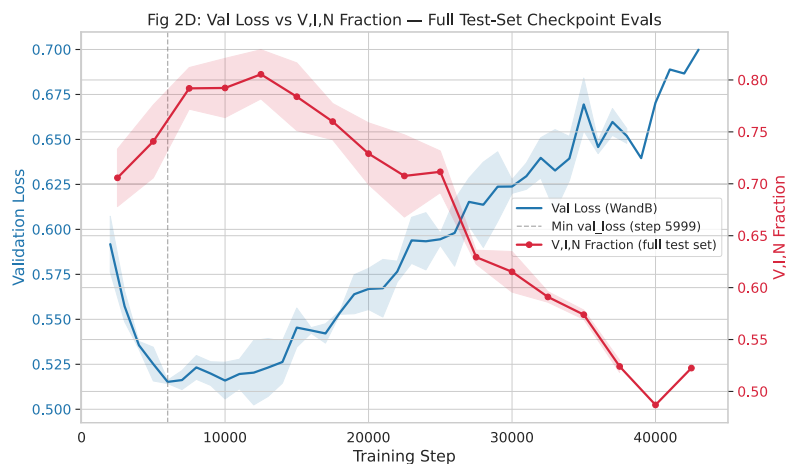


Figure 20: ZINC pretraining validation loss is well-correlated with chemical metrics, specifically its own generation rate of valid, logP-improved molecules, and novel molecules (VIN), justifying our early stopping criterion based on validation loss.

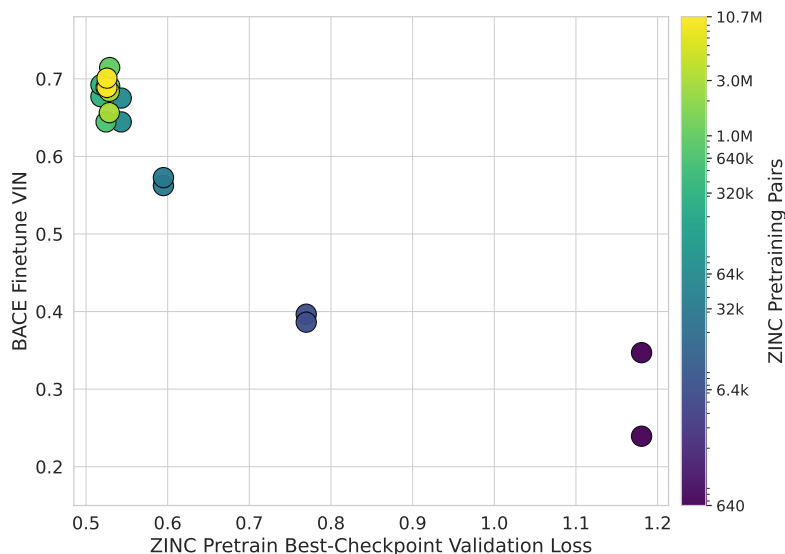


Figure 21: ZINC pretraining validation loss is well-correlated with downstream fine-tuned performance on BACE, further justifying our early stopping criterion based on validation loss. Here, each point is a separate checkpoint, pretrained on ZINC pair datasets of varying sizes and fine-tuned on the full BACE dataset.

E.4 Property filtering ablation

We vary the property-filtering cutoff used to construct training pairs and find that more aggressive cutoffs induce correspondingly stronger property changes in the generated molecules, even without prompt conditioning.

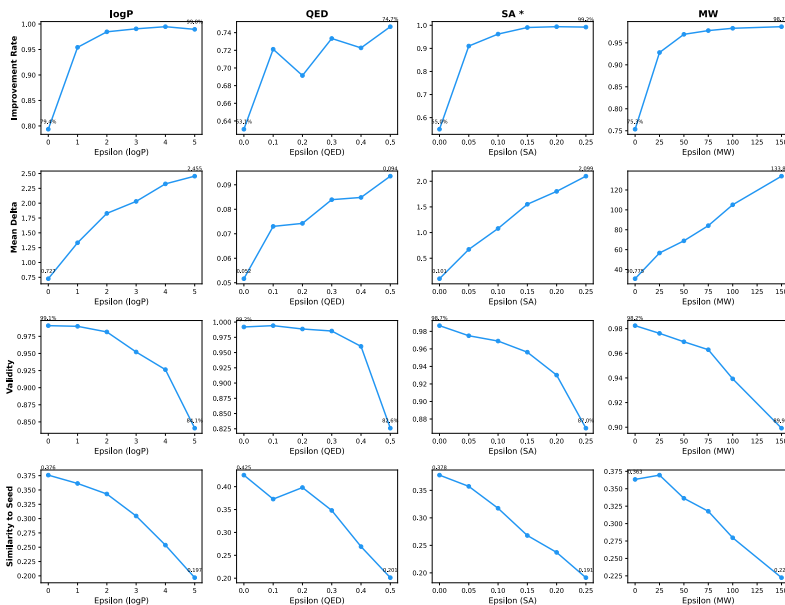


Figure 22: Filtering with more aggressive property cutoffs results in corresponding learned behavior. (This is in the basic setting without any prompting.)

E.5 ZINC-250K logP optimization

Figure 23 reports seeded logP optimization on ZINC-250K. PairIT variants differ by the pair-construction threshold ε , defined as the minimum logP increase between the seed and target molecules in a training pair. Increasing ε makes the training pairs more strongly property-improving, which increases the Improved rate but decreases similarity to the seed. Overall, PairIT achieves the strongest joint valid–novel–improved performance at intermediate ε , while maintaining high validity and novelty.

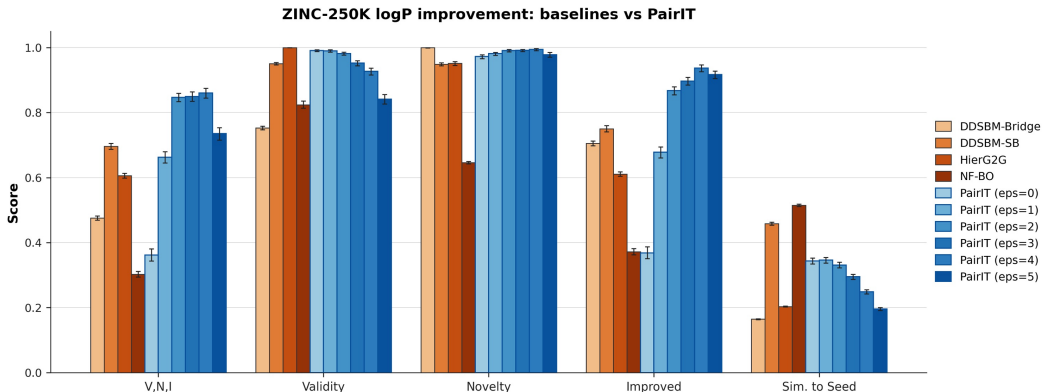


Figure 23: ZINC-250K logP optimization. Bars report joint valid–novel–improved success (V,N,I), validity, novelty, improvement, and SimSeed. PairIT variants sweep the pair-construction threshold $\varepsilon \in \{0, \dots, 5\}$; larger ε corresponds to training on pairs with larger logP increases. Error bars are bootstrap 95% confidence intervals over seeds.

E.6 Zero-shot logP generalization

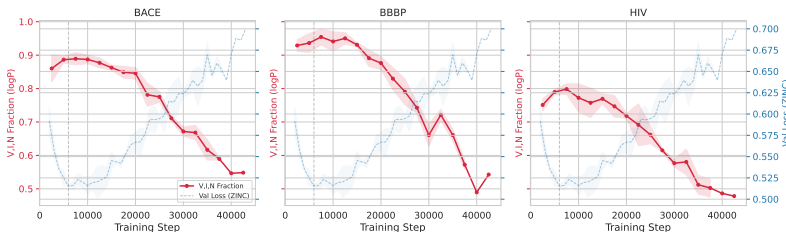


Figure 24: A model pretrained to optimize logP on ZINC transfers zero-shot to logP optimization on MoleculeNet. We track this ability over the course of training, and see that it correlates well with the pretraining validation loss.

F PairIT-RL

F.1 Reward and training details

Reward Model For each generation g produced from seed s , the reward has a gated structure that assigns fixed penalties to degenerate outputs before scoring valid, non-trivial molecules:

$$R_{\text{gated}}(g, s) = \begin{cases} -1.0 & \text{if } g \text{ is not a valid SMILES} \\ -0.5 & \text{if } g = s \text{ (exact copy of the seed)} \\ -0.5 & \text{if } g \in \mathcal{D}_{\text{train}} \text{ (memorised from any training set)} \\ -0.5 & \text{if } \text{Tan}(g, s) < \tau_{\text{sim}} \text{ (below similarity gate)} \\ r_{\text{obj}}(g, s) & \text{otherwise.} \end{cases}$$

$$r_{\text{obj}}(g, s) = \Delta_{\text{property model}}(g, s) + \lambda_{\text{act}} \mathbf{1}[p_{\text{property model}}(g) > 0.5] + w_{\text{sim}} \text{Tan}(g, s).$$

The first three gates enforce basic quality, assigning fixed penalties to invalid, copied, or memorized outputs rather than property-model rewards. The fourth gate (controlled by threshold $\tau_{\text{sim}} \in \{0, 0.2\}$) optionally penalises generations that diverge too far from the seed before any positive reward is computed.

For molecules that pass all gates, the reward combines three terms. $\Delta_{\text{property model}}(g, s) = p_{\text{property model}}(g) - p_{\text{property model}}(s)$ is the change in predicted activity probability relative to the seed, rewarding improvement in the target property. $\lambda_{\text{act}} \cdot \mathbf{1}[p_{\text{property model}}(g) > 0.5]$ is a threshold bonus ($\lambda_{\text{act}} = 2.0$) for generating molecules the property model classifies as active, providing a discrete incentive beyond the continuous Δ signal. $w_{\text{sim}} \cdot \text{Tan}(g, s)$ rewards structural similarity to the seed, where w_{sim} is the primary hyperparameter we sweep ($w_{\text{sim}} \in \{1, 3, 5, 10\}$). This term serves both its obvious purpose — preserving structural fidelity — and an implicit one: keeping generations within the distribution where the activity property model was trained and can be trusted.

For multi-property models, the total reward adds a prompt-adherence term:

$$R(g, s) = R_{\text{gated}}(g, s) + 2.0 \cdot R_{\text{adhere}}(g, s)$$

where R_{adhere} averages, over each prompted property, whether the generation changed that property in the prompted direction and by the prompted magnitude. The activity property is not prompted. It is driven entirely by the property model reward — so R_{adhere} covers only the nine RDKit descriptors (logP, QED, SA, MW, TPSA, HBD, HBA, rotatable bonds, fraction sp^3 carbons).

The key point is that w_{sim} and τ_{sim} are doing the work that KL regularisation alone is insufficient to do. The KL penalty (with $\beta_{\text{KL}} = 0.01$) constrains the policy to stay near the SFT distribution in token space, but it doesn’t directly constrain the output molecules to stay near the seed in chemical space. The Tanimoto terms provide that constraint explicitly, and the right operating point ($w_{\text{sim}} = 5$, $\tau_{\text{sim}} = 0.2$) is effectively a statement about how far from the seed the property model remains reliable.

Training setup. PairIT-RL is initialized from the real-only SFT checkpoint and fine-tuned with GRPO [Shao et al., 2024] separately for each dataset. Common hyperparameters: temperature 1.0, $\beta_{\text{KL}} = 0.01$, $\text{lr} = 5 \times 10^{-5}$, batch size 32, num_gen 8, LoRA rank 16, 10 epochs.

F.2 Similarity regularization and reward hacking.

Without a similarity penalty ($w_{\text{sim}} = 0$), the policy collapses to property model-fooling attractors at near-zero seed similarity, with BBBP VIN dropping to 0% and HIV VIN to 22%. Setting $w_{\text{sim}} = 5$ recovers seed similarity while maintaining high activity; a hard penalty that rejects generations with $\text{Tanimoto} < 0.2$ works similarly and trades a small amount of activity for greater similarity.

F.3 Exploratory multi-property RL

We additionally explored a multi-property RL variant initialized from the 10-property SFT model, using the activity reward above together with a prompt-adherence term. We treat this as an exploratory

extension rather than part of the primary PairIT-RL evaluation; qualitative examples are shown in Appendix F.4.

F.4 Representative Generations

F.4.1 Single-property RL

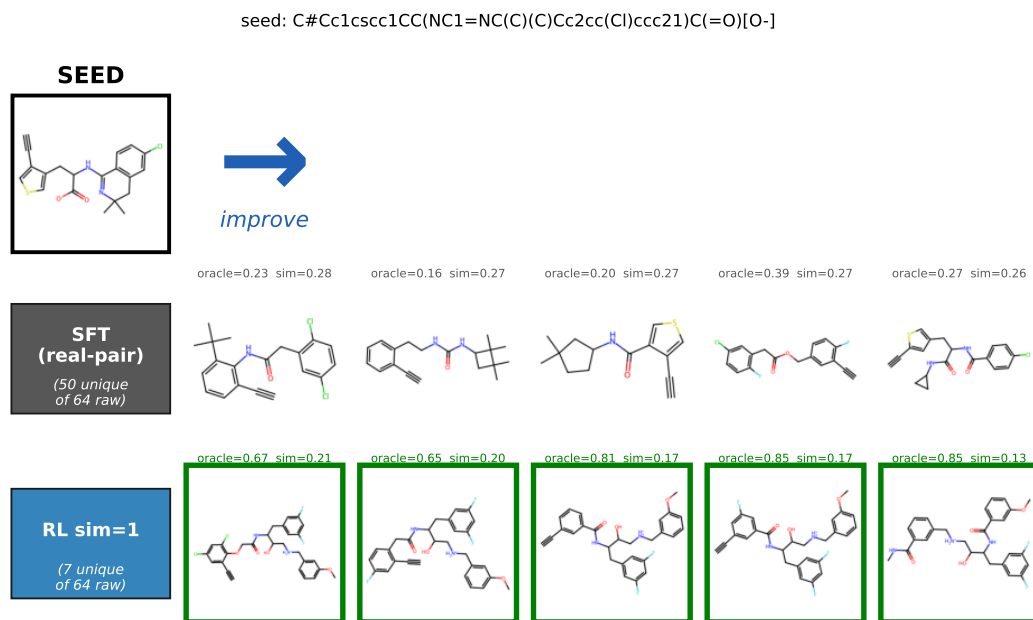


Figure 25: BACE single-property example, seed 1 of 2. Top row: seed and ground-truth improved partner from the test pair. Each subsequent row: top-5 deduplicated generations for one method. Box color: green = V·N·I (valid $\wedge$ novel $\wedge$ improved); orange = V·I but training-memorized; red = V·I but seed-copy.

seed: C=C(c1cccc(C2([NH2+][C@H](O)[C@H](Cc3cc(F)cc(F)c3)NC(...

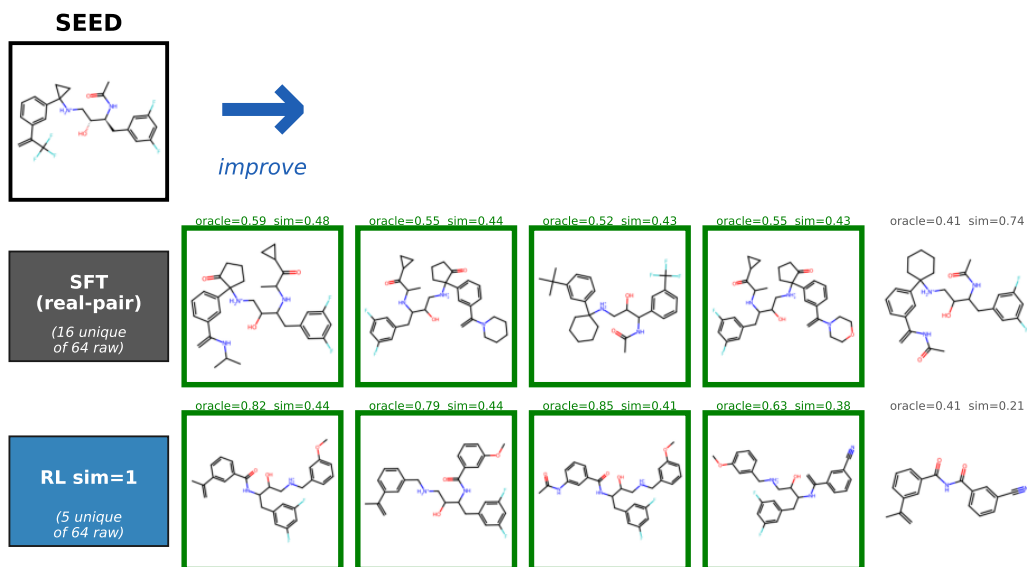


Figure 26: BACE single-property example, seed 2 of 2. Top row: seed and ground-truth improved partner from the test pair. Each subsequent row: top-5 deduplicated generations for one method. Box color: green = V·N·I (valid ∧ novel ∧ improved); orange = V·I but training-memorized; red = V·I but seed-copy.

seed: C=CC1=C(C(=O)O)N2C(=O)[C@H](NC(=O)/C(=N\OCC(=O)O)c3csc...

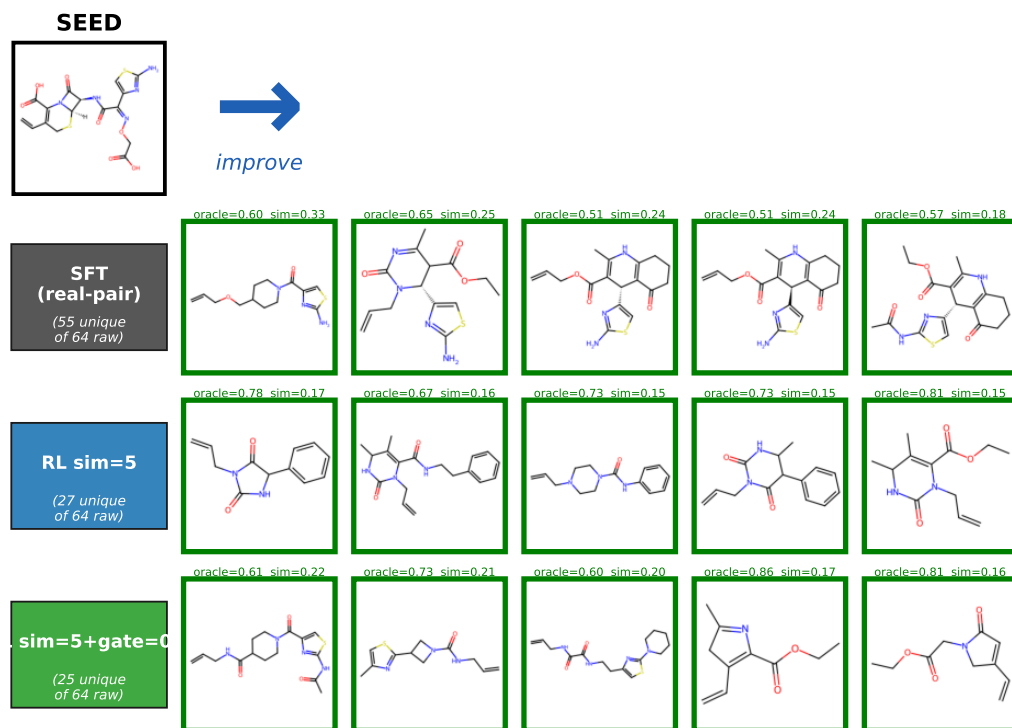


Figure 27: BBBP single-property example, seed 1 of 2.

seed: CC#C[C@]1(O)CC[C@H]2[C@@H]3CCCC4=CC(=O)CCC4=C3[C@@H](c3c...

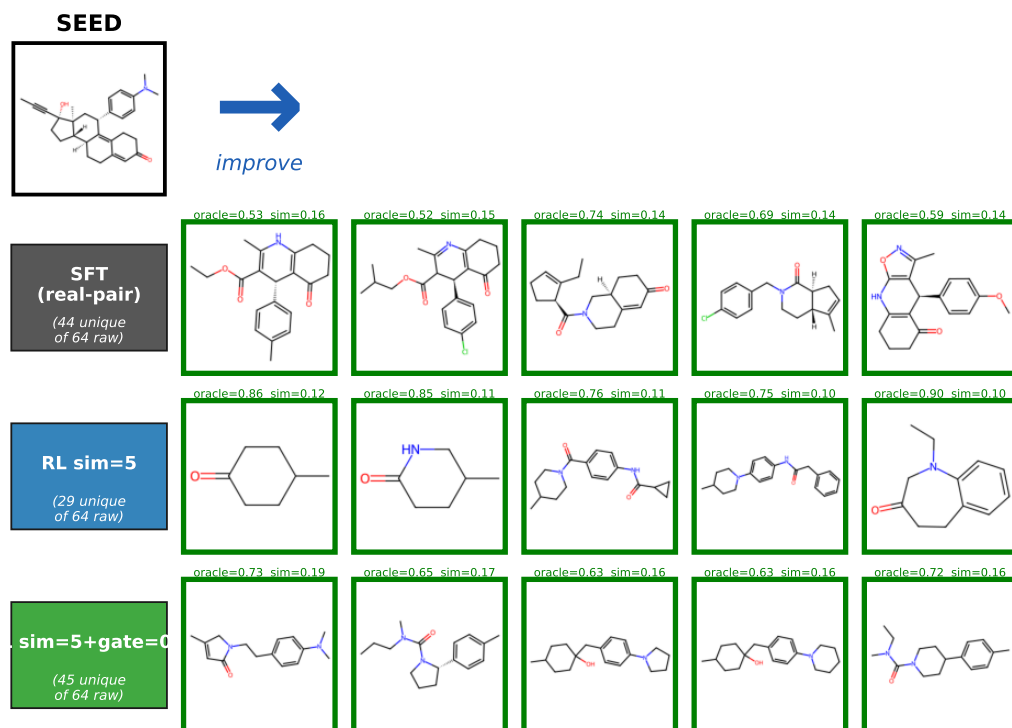


Figure 28: BBBP single-property example, seed 2 of 2.

seed: C#CC1(O)CCC2C3CCC4=CCCCC4C3CCC21C

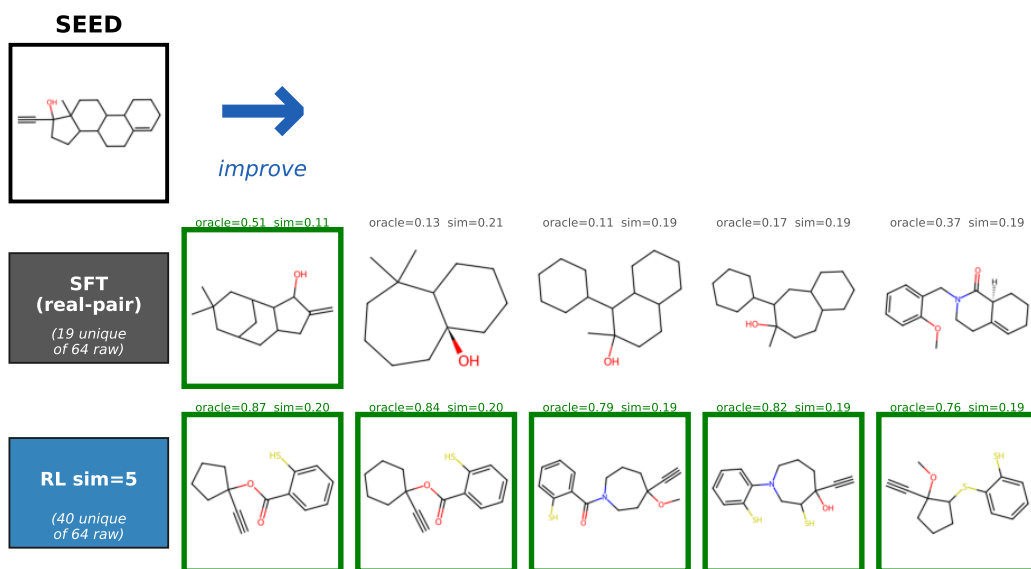


Figure 29: HIV single-property example, seed 1 of 2.

seed: C(#CCN1CCCCC1)CN1CCCCC1

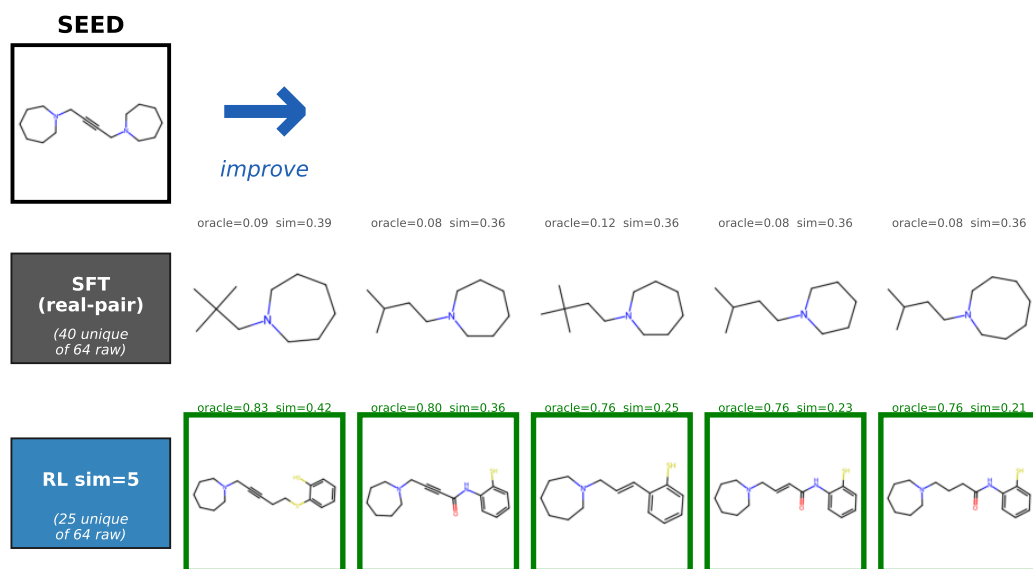


Figure 30: HIV single-property example, seed 2 of 2.

F.4.2 Multi-property RL

BACE — top-5 generations per method (seed: CC(=O)Nc1c(CI)cc(CNC(=[NH2+])NC(=O)Cn2ccc3cc(Br)cc...))
 Box: green = V-N-I (valid & novel & improved); orange = memorized from training set; red = seed-copy. (Both orange and red would be V-N-I if novel.)

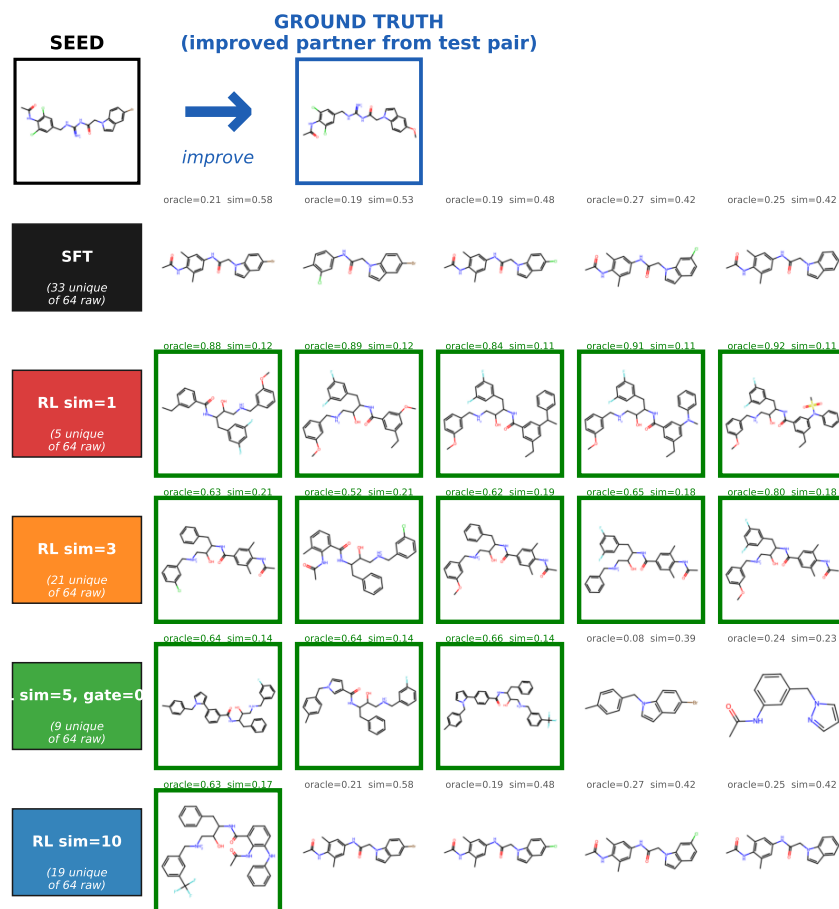


Figure 31: BACE multi-property example, seed 1 of 2. Methods compared: SFT, RL sim=1, RL sim=3, RL sim=5, gate=0.2, RL sim=10.

BACE — top-5 generations per method (seed: CC(=O)Nc1c(Cl)cc(CNC(=[NH2+])NC(=O)Cn2ccc3cc(C#N)c...)
 Box: green = V-N-I (valid $\wedge$ novel $\wedge$ improved); orange = memorized from training set; red = seed-copy. (Both orange and red would be V-N-I if novel.)

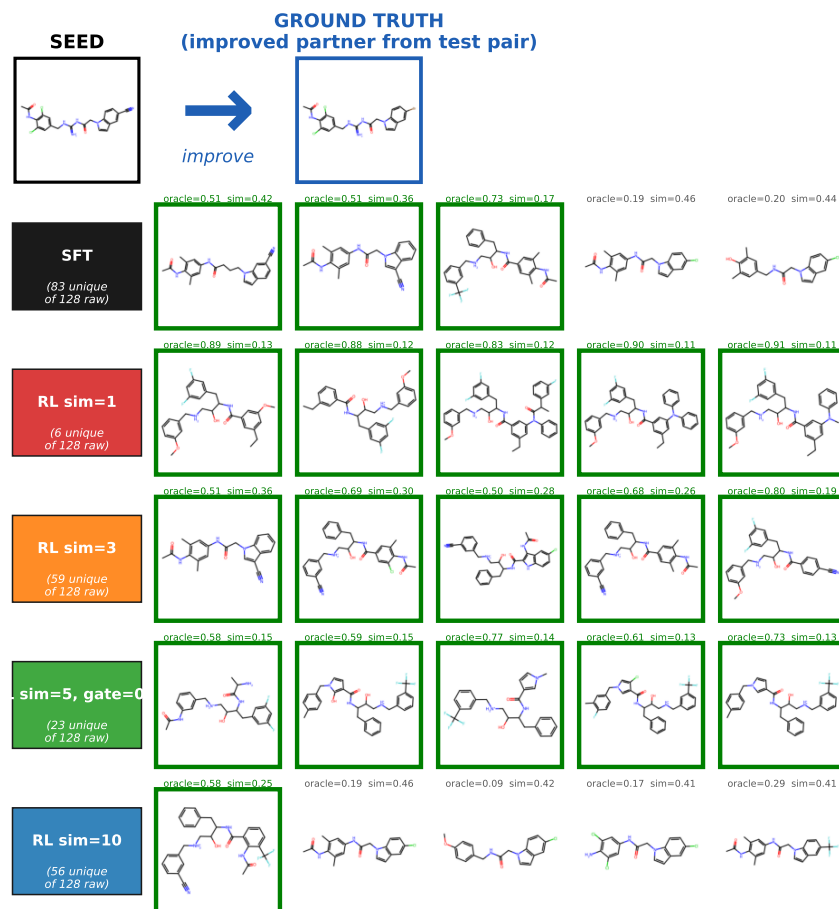


Figure 32: BACE multi-property example, seed 2 of 2. Methods compared: SFT, RL sim=1, RL sim=3, RL sim=5, gate=0.2, RL sim=10.

BBBP — top-5 generations per method (seed: CC(=O)OCC1=C(C(=O)O)N2C(=O)[C@@H](NC(=O)Cc3cccs3)[...])
 Box: green = V-N-I (valid $\wedge$ novel $\wedge$ improved); orange = memorized from training set; red = seed-copy. (Both orange and red would be V-N-I if novel.)

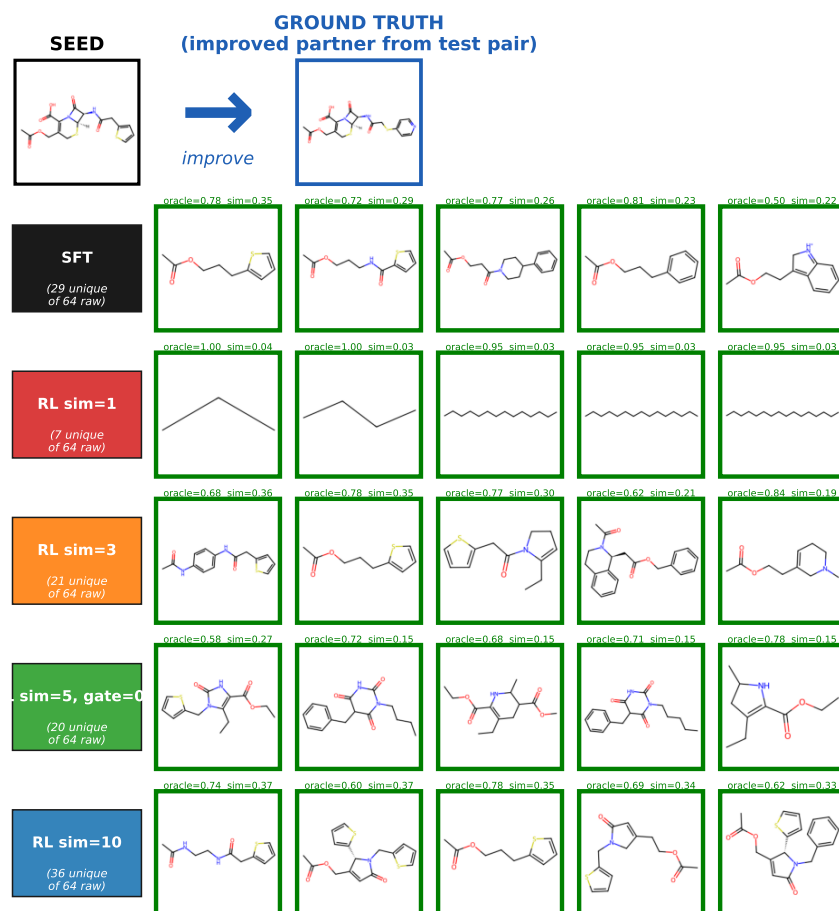


Figure 33: BBBP multi-property example, seed 1 of 2.

BBBP — top-5 generations per method (seed: CC(=O)Oc1ccccc1C(=O)O)
 Box: green = V-N-I (valid $\wedge$ novel $\wedge$ improved); orange = memorized from training set; red = seed-copy. (Both orange and red would be V-N-I if novel.)

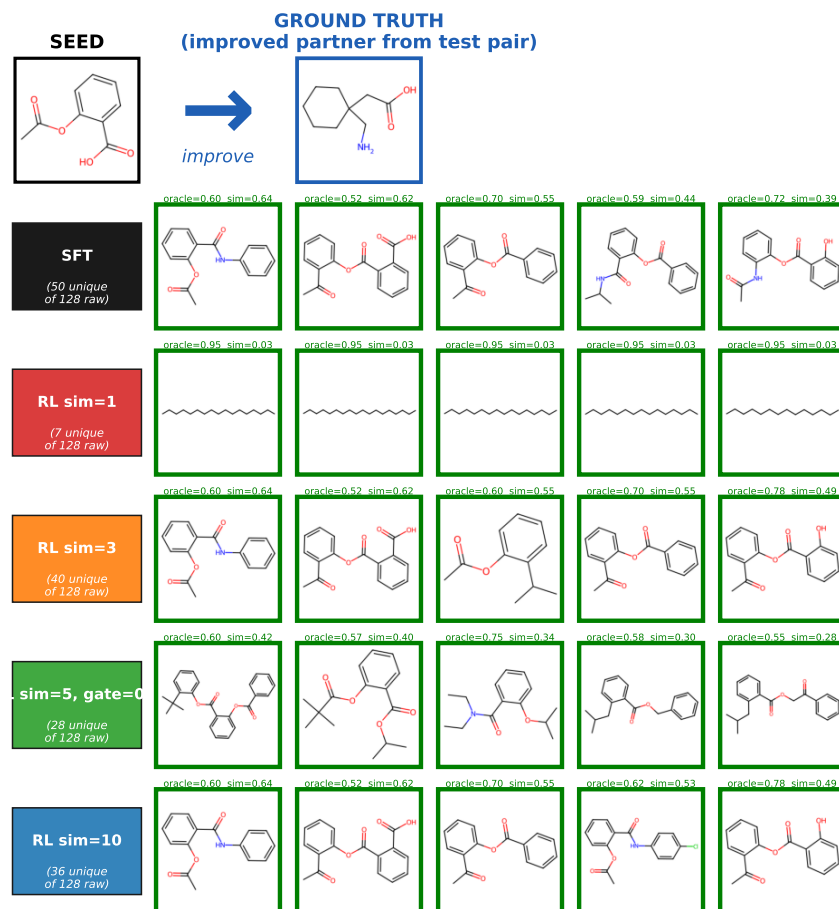


Figure 34: BBBP multi-property example, seed 2 of 2.

HIV — top-5 generations per method (seed: C(=NCCSSCCN=Cc1ccccc1)c1ccccc1)
 Box: green = V-N-I (valid $\wedge$ novel $\wedge$ improved); orange = memorized from training set; red = seed-copy. (Both orange and red would be V-N-I if novel.)

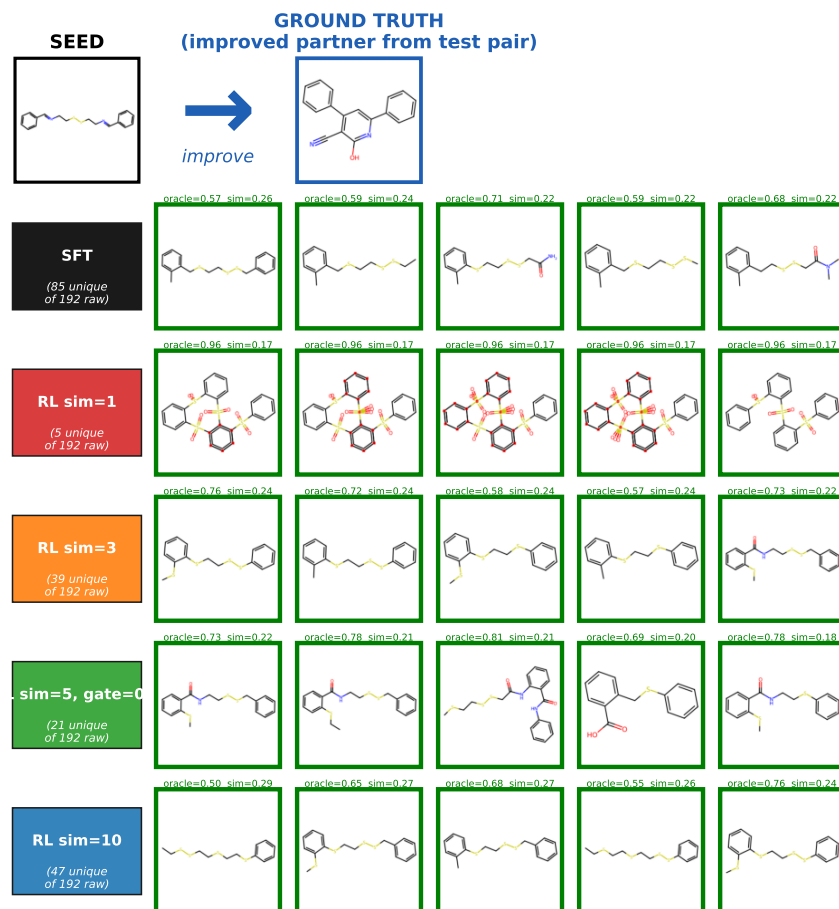


Figure 35: HIV multi-property example, seed 1 of 2.

HIV — top-5 generations per method (seed: CC(C(=O)O)C(C)(O)C(=O)O)
 Box: green = V-N-I (valid $\wedge$ novel $\wedge$ improved); orange = memorized from training set; red = seed-copy. (Both orange and red would be V-N-I if novel.)

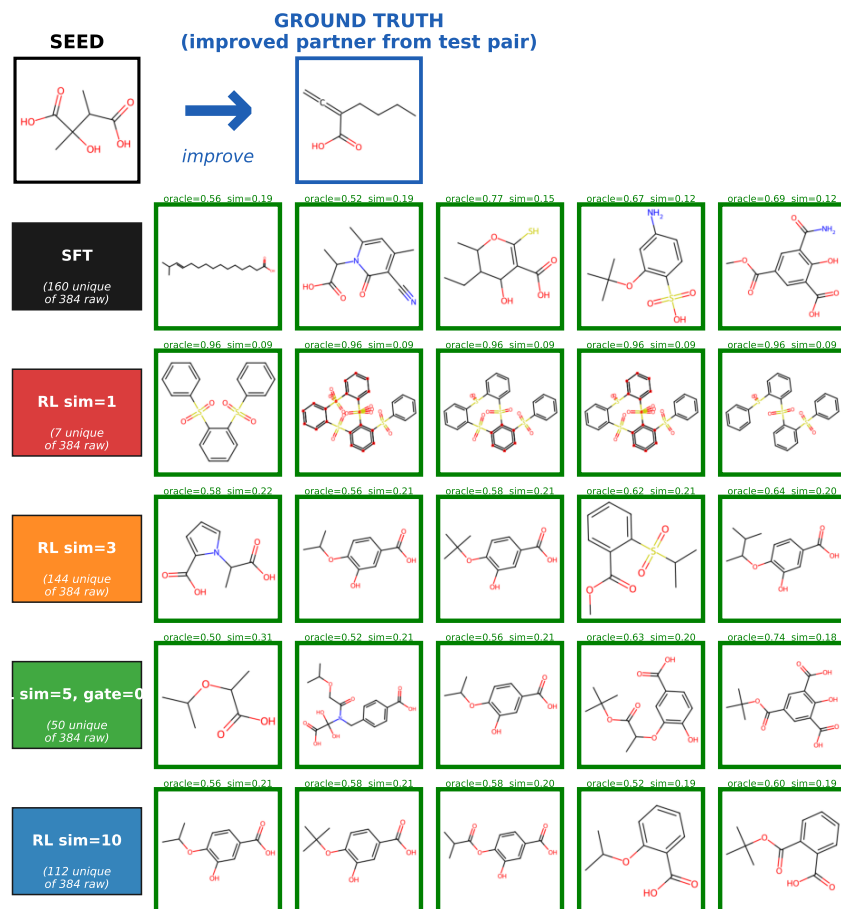


Figure 36: HIV multi-property example, seed 2 of 2.

G Prompt-conditioned property control

G.1 Prompt tokens

At inference time, the prompt tokens and seed SMILES are provided as context, and generation begins after `<sep>`. The prompt vocabulary consists of a start-of-prompt marker `<pc>`, a property identifier such as `<p_logp>`, and a property-specific signed delta-bin token encoding the desired change. The seed property value is not included in the prompt; direction and approximate magnitude are encoded jointly by the sign and magnitude of the delta bin.

G.2 Single-property control

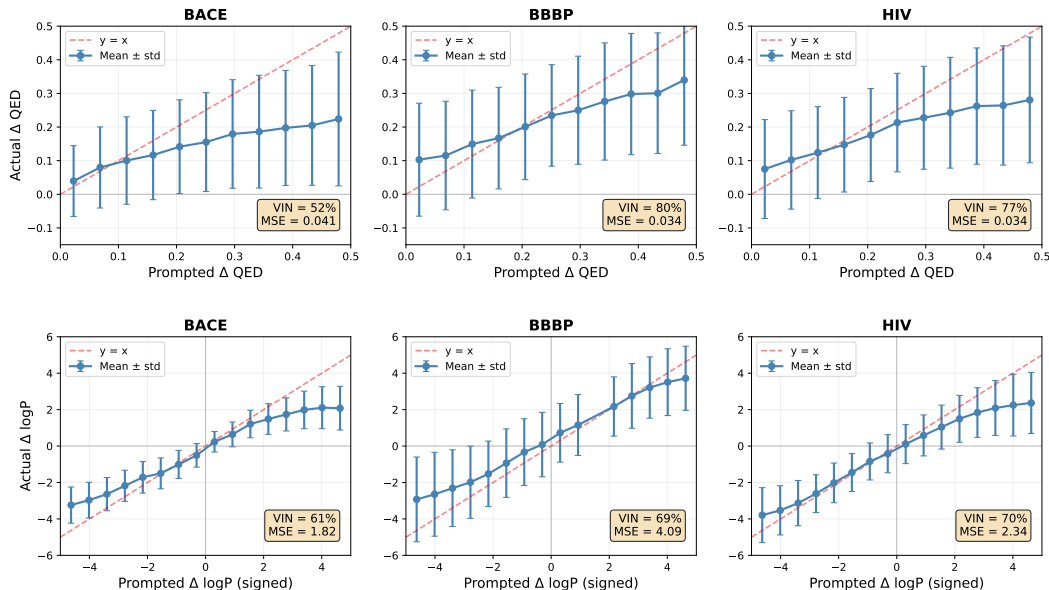


Figure 37: Single-property controllability on MoleculeNet for both QED and logP. Top: QED, upward prompts over $[0, 0.5]$. Bottom: logP, bidirectional prompts over $[-5, 5]$. Error bars show standard deviation across 5 generations per seed per bin. Realized changes scale with prompted magnitude in both cases, indicating coarse magnitude control. The QED-only version of the top row appears in the main body as Figure 5.

Table 15: Single-property controllability on MoleculeNet after fine-tuning. VIN is the fraction of valid, novel generations whose property moves in the prompted direction. MSE is computed relative to the midpoint of the prompted delta bin.

Property	Dataset	VIN%	MSE
QED ($\uparrow$ only)	BACE	52	0.041
	BBBP	80	0.034
	HIV	77	0.034
logP (bidirectional)	BACE	61	1.82
	BBBP	69	4.09
	HIV	70	2.34

G.3 Two-property control on ZINC-250K

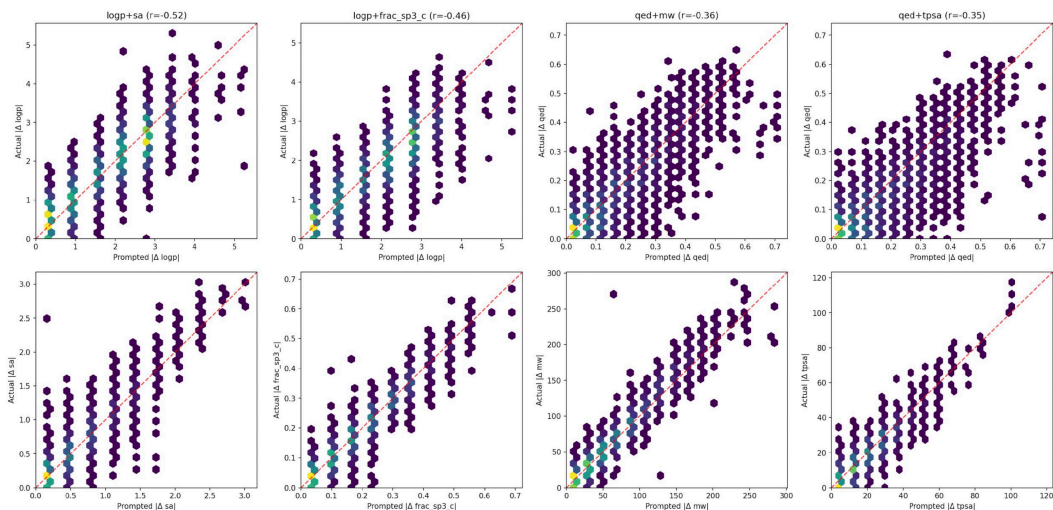


Figure 38: Prompted versus realized absolute property change under multi-property conditioning for four selected low-correlation property pairs from ZINC-250K. Each panel corresponds to one property within a pair and plots the prompted absolute delta on the x-axis against the realized absolute delta of the generated molecule on the y-axis; the dashed diagonal indicates perfect agreement. Across all pairs, larger requested edits produce larger realized changes, showing that approximate magnitude control is retained for both properties under simultaneous conditioning, although dispersion increases for large requested deltas and bounded properties such as QED show some saturation.

G.4 Two-property control on MoleculeNet

The figures below show the full scatter plots for selected low-correlation two-property control and joint QED+logP control. In both cases, both properties are prompted simultaneously using signed delta bins derived from matched test pairs. Each point shows a valid generation; points are not filtered by whether the other property also moves correctly.

We selected MW/HBD, TPSA/logP, and HBA/logP as weakly correlated descriptor pairs using Pearson correlations of molecular property values across the pooled ZINC training and validation molecules ($n = 249,455$; $r = -0.005$, -0.162 , and $+0.056$, respectively), and used them for BACE, BBBP, and HIV, respectively.

For each held-out matched test pair, we compute the requested property changes as reference-target minus seed values and convert them to signed-delta-bin tokens. Generation is conditioned only on the prompt tokens and seed SMILES; the reference-target SMILES is not provided to the model. The two-property evaluations use 51 BACE, 14 BBBP, and 46 HIV test pairs with inactive seeds and active reference targets. We generate 64 outputs per pair with the same prompt, without top-five

selection. Results pool generated attempts across pairs; prompts follow the observed pair distribution rather than a balanced bin sweep.

For each descriptor, VIN is the fraction of all generated attempts with a nonzero raw requested change that are valid, novel relative to the task’s training molecules, and move in the requested direction. Combined VIN includes attempts with at least one applicable requested change and requires every applicable criterion to succeed, together with validity and novelty. A zero requested descriptor change imposes no success condition, and pairs with both descriptor changes zero are excluded from Combined VIN. For bioactivity co-control, success means an inactive seed yields a generation with train-only random-forest activity probability greater than 0.5. The scatter plots show valid generations with a nonzero request for the displayed property, without requiring success on the other property. Their MSE pools squared errors against the raw matched-pair change across valid generations with finite values and nonzero target changes; novelty is not required. The per-pair medians are display summaries and do not enter MSE.

For the randomized-prompt control, we independently sample each property’s signed-bin index uniformly from 00–15, with replacement, while keeping the property identities and prompt-block order fixed. No separate direction token is sampled. One reproducible randomized assignment is fixed per ordered seed–target pair and shared by its 64 generations. The control uses the same trained checkpoint as the correctly prompted arm. Both arms are scored against the original requested changes.

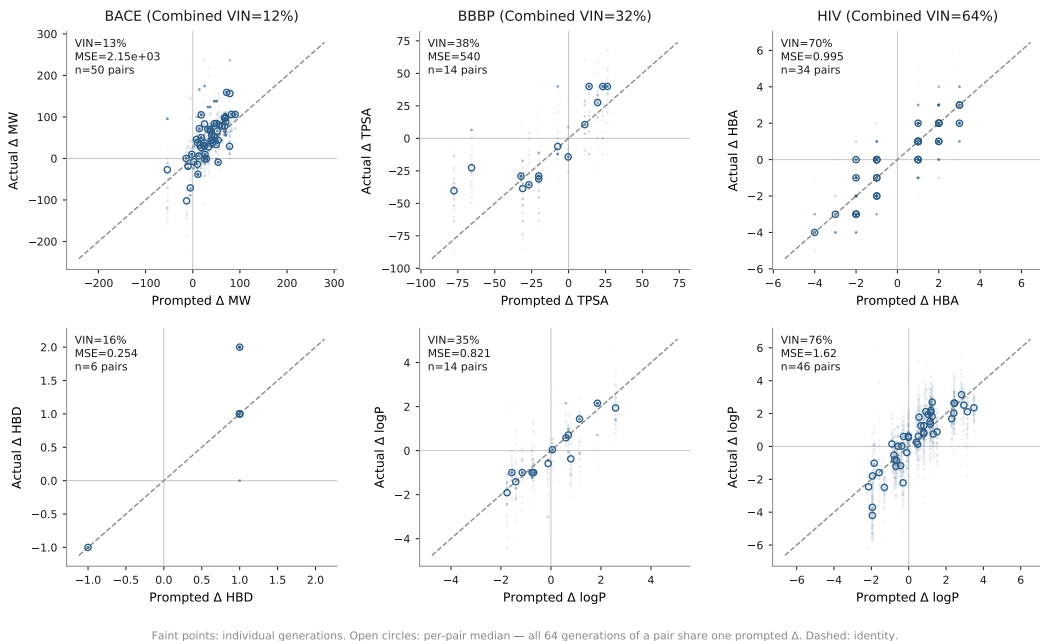


Figure 39: Selected low-correlation two-property control on MoleculeNet. Each column is one dataset; the top and bottom rows show the two prompted properties. Column titles report combined VIN% (valid, novel outputs satisfying every nonzero requested direction, divided by all attempts with at least one nonzero request). The model achieves 12–64% combined VIN, demonstrating independent control of selected property pairs with low empirical correlation.

Table 16: Two-property controllability for selected low-correlation property pairs on MoleculeNet, using 64 raw generations per matched test pair without top-five selection. Per-property VIN divides valid, novel, direction-matching outputs by all attempts with a nonzero requested change for that property. Combined VIN divides valid, novel outputs satisfying every nonzero requested direction by all attempts with at least one nonzero request. Invalid and non-novel attempts remain denominator failures; zero-delta properties impose no condition.

Dataset	Props	Combined VIN%	Prop1 VIN%	Prop2 VIN%
BACE	MW+HBD	12	13	16
BBBP	TPSA+logP	32	38	35
HIV	HBA+logP	64	70	76

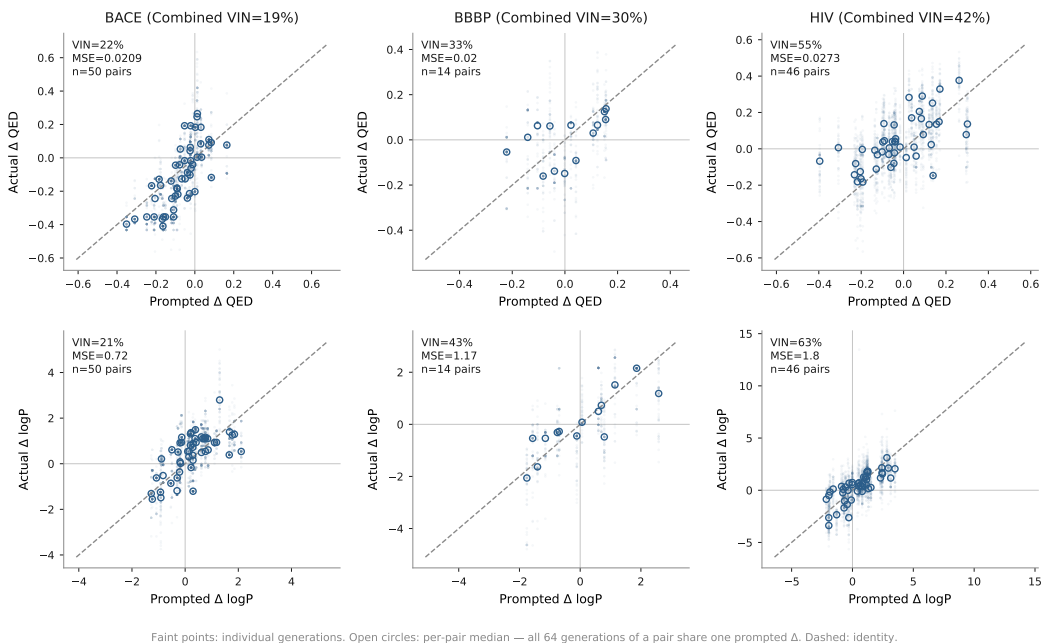


Figure 40: Joint QED+logP control, with both properties prompted using natural signed deltas from matched test pairs. The model achieves 19–42% combined VIN (valid, novel outputs satisfying every nonzero requested direction, divided by all attempts with at least one nonzero request). QED+logP is a harder joint-control setting than uncorrelated descriptor pairs.

Table 17: Two-property controllability for joint QED and logP control on MoleculeNet, using signed-bin requests from held-out matched pairs and 64 raw generations per pair without top-five selection. Per-property VIN requires validity, novelty, and the correct direction, using all attempts with a nonzero requested change for that property as its denominator. Combined VIN uses all attempts with at least one nonzero request and requires validity, novelty, and agreement with every nonzero requested direction. Zero-delta properties impose no condition.

Dataset	Combined VIN%	QED VIN%	logP VIN%
BACE	19	22	21
BBBP	30	33	43
HIV	42	55	63

G.5 Many-property control

We train a single model on all nine computable RDKit properties simultaneously (logP, QED, SA, MW, TPSA, HBD, HBA, rotatable bonds, and fraction sp³ carbons) plus a tenth *target* prompt field. During ZINC pretraining, the target prompt field mirrors a randomly chosen real property for each training example, without committing it to a fixed property during pretraining. Property dropout (rate = 0.3) is applied during pretraining, randomly omitting properties from the prompt so the model can be conditioned on only a subset of properties at inference time.

At MoleculeNet fine-tuning, the target prompt field is assigned to the dataset-specific bioactivity property (Class for BACE, p_np for BBBP, and HIV_active for HIV). The target-slot change is the reference target’s label minus the seed’s label, discretized with the activity bin range. The model is fine-tuned with this reassigned target meaning and the dataset-specific bin limits. Every retained training pair is used in both orderings, with all signed changes recomputed after reversal, and all ten property blocks remain included.

Table 18: Full multi-property prompted model evaluated with all 10 properties on MoleculeNet test pairs. Each property is prompted with the signed delta bin corresponding to the ground-truth pair. Active% is the fraction of valid, novel generations predicted active (class 1) by the property model; property model Up% is the fraction for which the property model score moves in the correct direction relative to the seed. SimSeed is the mean Tanimoto similarity to the input seed.

Dataset	VIN%	Valid%	Novel%	Active%	Property-model Up%	SimSeed
BACE	21.7	95.8	40.9	53.0	60.4	0.491
BBBP	83.0	99.8	91.6	90.6	72.2	0.290
HIV	12.3	92.3	89.4	13.8	47.9	0.162

For the ten-property model’s direction-adherence results in, we use only held-out pairs with an inactive seed and an activity-label request of $0 \rightarrow 1$. For each of the nine non-activity descriptors, we calculate the fraction of valid generations with a nonzero raw requested change whose realized change has the same sign. We pool eligible valid generations within each descriptor and then take the unweighted mean of the nine rates. These are raw generations without top-five selection.

Full multi-property context substantially helps BBBP, where RDKit descriptors are highly correlated with bioactivity (e.g., TPSA: $r = -0.50$, QED: $r = +0.48$).

Table 19: Full multi-property prompted model evaluated with the target bioactivity property plus logP, SA, and QED (three-property subset) on MoleculeNet test pairs.

Dataset	VIN%	Valid%	Novel%	Active%	property model Up%	SimSeed
BACE	25.5	95.8	61.3	41.4	67.1	0.401
BBBP	79.3	90.0	90.0	88.1	95.2	0.818
HIV	19.8	84.6	84.6	23.4	75.5	0.841

The logP+SA+QED subset yields 25.5% VIN on BACE, 79.3% on BBBP, and 19.8% on HIV, compared with 21.7%, 83.0%, and 12.3%, respectively, under the full 10-property prompt. Thus, reducing the prompt to this subset improves VIN on BACE and HIV while modestly reducing it on BBBP, consistent with the broader property-subset sensitivity in Table 20.

Table 20: Property-subset sensitivity for the full multi-property prompted model. A single trained model is evaluated across all $\binom{9}{3} = 84$ combinations of three RDKit properties prompted alongside the target bioactivity property (84 evaluation runs per dataset, not 84 trained models).

Dataset	Mean VIN%	Best VIN%	Worst VIN%	Spread (pp)
BACE	25.1 ± 2.2	30.4	21.1	9.3
BBBP	72.0 ± 6.3	85.0	57.1	27.9
HIV	13.0 ± 3.8	25.0	5.5	19.5

Property-subset selection has a significant impact, especially on BBBP (27.9 pp spread) and HIV (19.5 pp spread). The best HIV combination (logP + QED + TPSA, 25.0%) is $4.5\times$ better than the worst (QED + rotatable bonds + fraction sp^3 carbons, 5.5%). Consistently poor subsets contain rotatable bonds and fraction sp^3 carbons across all datasets; consistently strong subsets contain logP and/or SA for BACE and HIV, and SA, HBA, and TPSA for BBBP, reflecting the high correlation of these descriptors with blood–brain barrier permeability.

Table 21: Mean property delta (generated minus seed) for the full multi-property prompted model under 10-property prompting.

Property	BACE	BBBP	HIV
logP	+0.022	-0.037	+0.709
QED	-0.032	+0.018	+0.033
SA	+0.126	-0.132	-0.595
MW	+17.78	+3.11	+18.56
TPSA	+6.20	-7.09	-15.51
HBD	-0.041	-0.354	-0.719
HBA	+0.423	-0.921	-0.125
Rot. bonds	+0.661	+0.312	+0.306
Frac. sp^3	+0.057	+0.200	-0.006

BBBP generations systematically decrease TPSA (-7.09) and HBD (-0.354), consistent with the strong negative correlation between these properties and BBB permeability (TPSA: $r = -0.50$, HBD: $r = -0.47$). The model appears to learn the structure–activity relationship through multi-property prompting.