SLIM: Sparse Latent Steering for Interpretable and Property-Directed LLM-Based Molecular Editing

Anonymous ACL submission

Abstract

Large language models possess strong chemical reasoning capabilities, making them effective molecular editors. However, property-relevant information is implicitly entangled across their dense hidden states, providing no explicit handle for property control: a substantial fraction of edits fail to improve or even degrade target properties. To address these issues, we propose **SLIM** (Sparse Latent Interpretable Molecular editing), a plug-and-play framework that decomposes the editor's hidden states into sparse, property-aligned features via a Sparse Autoencoder with learnable importance gates. Steering in this sparse feature space precisely activates property-relevant dimensions, improving editing success rate without modifying model parameters. The same sparse basis further supports interpretable analysis of editing behavior. Experiments on the MolEditRL benchmark across four model architectures and eight molecular properties show consistent gains over baselines, with improvements of up to 42.4 points.

1 Introduction

Molecular editing, the task of modifying a given molecule to optimize specific properties while preserving its core structure, is a fundamental step in drug discovery and materials science (Hughes et al., 2011; Jiménez-Luna et al., 2020). In real-world pharmaceutical development, a promising lead compound often requires iterative structural refinements to improve druglikeness, synthetic accessibility, binding affinity, or other pharmacological properties. The ability to perform such targeted modifications efficiently and reliably holds great potential for accelerating the drug design pipeline.

Early computational approaches to molecular optimization have relied on reinforcement learning (Loeffler et al., 2024; Zhuang et al., 2025) and diffusion models (Lee et al., 2023; Igashov et al., 2024). Recently, large language models (LLMs) have demonstrated strong potential for molecular editing (Ye et al., 2025; Dey et al., 2025; Fang et al., 2024; Pei et al., 2023). LLMs acquire rich chemical priors, including structure–activity relationships, functional group chemistry, and pharmacophore patterns, through pretraining on massive corpora of chemical literature (Zhang et al., 2025b,a). Post training methods then bridge this chemical knowledge with SMILES-level structural transformations, positioning LLM-based editors to produce chemically meaningful modifications grounded in domain knowledge.

Despite these advances, existing LLM-based editors provide no explicit handle for property control. Property-relevant information is implicitly entangled across the model's dense hidden states, making it impossible to selectively amplify a target property without retraining. As a result, supervised fine-tuning offers only coarse control: a substantial fraction of generated edits fail to improve, or even degrade, the target property. Improving a specific property currently requires collecting new paired data and retraining the model, a costly process that scales poorly across properties and models.

To address this, we propose SLIM, a plug-and-play framework that decomposes the editor's hidden states into sparse, property-aligned features via a task-oriented Sparse Autoencoder (SAE) (Cunningham et al., 2024; Bricken et al., 2023). Unlike standard SAEs trained for reconstruction alone, our SAE is jointly trained with contrastive, predictive, and gradient-alignment objectives that shape the sparse basis to be property-aware. Per-property Importance Gates then learn to select, from this shared basis, the subset of features most relevant to each target property. To steer the model, we project a gradient-derived causal direction through the top-k selected features of the SAE, producing a sparse steering vector that precisely activates property-relevant dimensions at inference time without modifying model parameters. The sparse basis further

supports interpretable analysis, as individual SAE features encode recognizable chemical semantics.

In summary, our contributions are as follows:

- We propose SLIM, a plug-and-play framework that decomposes dense LLM hidden states into sparse, property-aligned features for property-directed steering, requiring no retraining of the base molecular editor.
- We introduce task-oriented SAE with contrastive, predictive, and gradient-alignment objectives and per-property Importance Gates, ensuring that the learned features are causally aligned with target properties.
- Experimental results on the MolEditRL benchmark across four model architectures and eight molecular properties show consistent improvements of up to 42.4%. Case studies further show that individual SAE features encode interpretable chemical semantics.

2 Related Work

Molecular Editing. Molecular editing aims to modify a given molecule to optimize target properties while preserving structural similarity. Early approaches use reinforcement learning with scoring functions (Loeffler et al., 2024) or combine discrete diffusion with RL (Zhuang et al., 2025). More recently, LLM-based methods have shown strong results. DrugAssist (Ye et al., 2025) fine-tunes LLaMA-2 (Touvron et al., 2023) on paired molecular data, framing editing as instruction following. GeLLM3O (Dey et al., 2025) extends this to LLaMA-3 (Llama Team, AI @ Meta, 2024) and Mistral (Jiang et al., 2023), demonstrating cross-model transferability. MolGen (Fang et al., 2024) adopts a BART-based (Lewis et al., 2020) encoder-decoder trained on SELFIES (Krenn et al., 2020) with chemical feedback. These methods optimize molecular properties through training but offer no mechanism to interpret or steer the learned representations at inference time.

Molecular Representation Learning. Understanding how models represent molecular structure and properties is central to interpretable editing. ChemATP (Zhang et al., 2025b) introduces a training-free chemical reasoning framework that leverages LLM internal representations. AtomDisc (Zhang et al., 2025a) proposes atom-level tokenization that reveals interpretable structure–property associations within LLM hidden states. More broadly, the *linear representation hypothesis* (Park et al., 2024) suggests that high-level concepts are encoded as directions in activation space, motivating methods that extract and manipulate these directions for controllable generation.

Sparse Autoencoders and Activation Steering. Neural networks represent more features than dimensions through superposition (Elhage et al., 2022). SAEs recover these features via an overcomplete basis (Sharkey et al., 2022; Cunningham et al., 2024; Bricken et al., 2023), with Gated SAEs (Rajamanoharan et al., 2024) improving feature quality. On the steering side, activation addition (Turner et al., 2024) and contrastive activation addition (CAA) (Rimsky et al., 2024) modify behavior by adding mean-difference directions, and representation engineering (Zou et al., 2025) extends this top-down. However, existing SAEs are reconstruction-only and used for post-hoc analysis, while steering methods rely on mean-difference directions that fail for molecular property control (§4.3). SLIM bridges these lines by integrating task-oriented SAE training with gradient-based steering.

3 Method

SLIM operates as a plug-and-play module on top of any molecular editing LLM. The pipeline consists of three stages: (1) a layer scan to identify the optimal intervention point, (2) task-oriented SAE training with gradient alignment, and (3) inference-time activation steering via interpretable sparse directions. Figure 1 illustrates the overall framework.

3.1 Layer Scan

As shown in Figure 1 (Stage 1), different layers encode different levels of chemical abstraction. We select the layer l^* whose hidden states are most informative about molecular properties by fitting ridge regression probes (Park et al., 2024) on activations extracted from 5,000 ZINC molecules:

$$l^* = \arg\max_{l} \frac{1}{|P|} \sum_{p \in P} R^2(l, p), \qquad (1)$$

where P is the set of target properties and $R^2(l, p)$ is the coefficient of determination for property p at layer l. The layer with the highest average R^2 concentrates the most property-relevant information in its hidden states, making it the most effective point for steering: interventions at this layer can

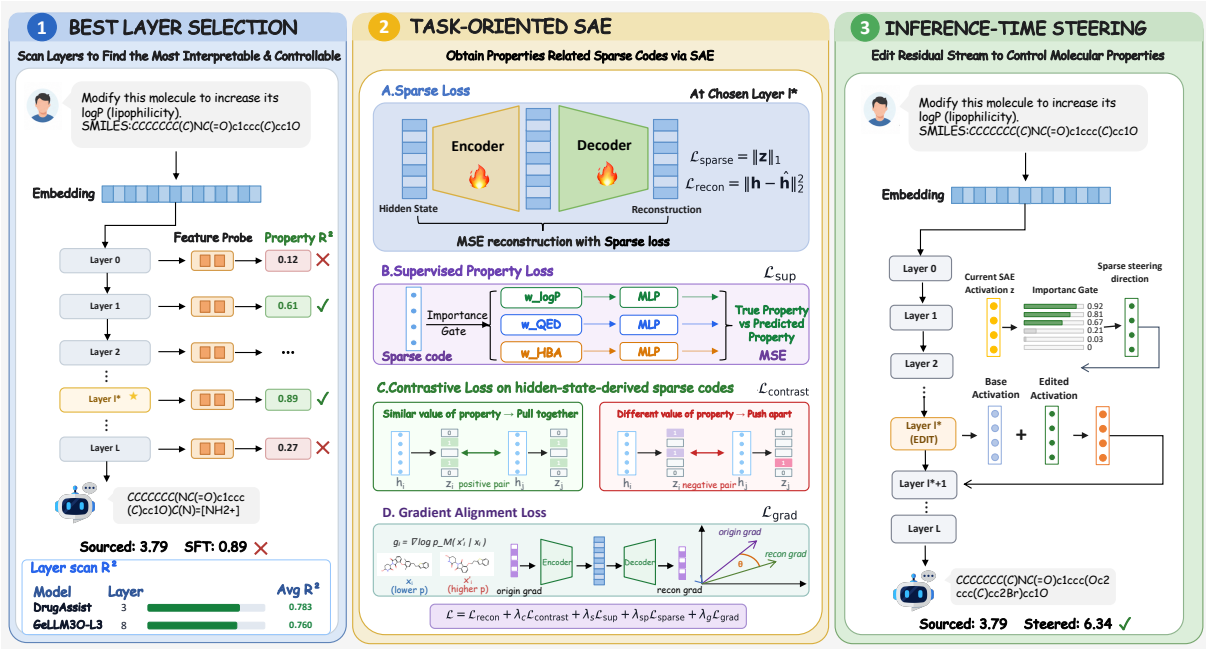


Figure 1: Overview of the SLIM framework. **Stage 1**: Ridge probes scan all layers to identify the optimal intervention point l^*. **Stage 2**: A task-oriented SAE is trained at layer l^* with four objectives: (A) sparse reconstruction, (B) supervised property prediction via per-property Importance Gates, (C) contrastive alignment of importance-gated sparse codes, and (D) gradient alignment to ensure the SAE basis faithfully represents causal steering directions. **Stage 3**: At inference time, a sparse steering vector is added to the residual stream at layer l^*, directing the model toward improved molecular properties without modifying model parameters.

maximally influence generation because the representation is richest in the signals we aim to amplify.

3.2 Task-Oriented SAE

As illustrated in Figure 1 (Stage 2A), we train a Gated SAE (Rajamanoharan et al., 2024) on hidden states $\mathbf{h} \in \mathbb{R}^d$ extracted at layer l^* from 50,000 molecules. The SAE uses $D = 8d$ latent dimensions. Encoding produces a sparse code $\mathbf{z} \in \mathbb{R}^D$:

$$\mathbf{z} = \sigma(\mathbf{W}_g\mathbf{h} + \mathbf{b}_g) \odot \mathrm{ReLU}(\mathbf{W}_m\mathbf{h} + \mathbf{b}_m), \quad (2)$$

where $\mathbf{W}_g, \mathbf{W}_m \in \mathbb{R}^{D\times d}$ are gate and magnitude matrices, σ is the sigmoid function, and $\odot$ denotes element-wise multiplication. Reconstruction is $\hat{\mathbf{h}} = \mathbf{W}_d\mathbf{z} + \mathbf{b}_d$.

Importance Gates. As shown in Figure 1 (Stage 2B), each property p has a learnable Importance Gate $\mathbf{w}_p \in \mathbb{R}^D$ that identifies which SAE features are relevant to p. The gated sparse code for property p is $\mathbf{z}_p = \mathbf{w}_p \odot \mathbf{z}$, which feeds into per-property contrastive and predictive heads. This enables the SAE to learn property-aligned feature decompositions while sharing the same underlying sparse basis across all properties.

Training Objective. Our task-oriented SAE is trained with a multi-objective loss:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_c\mathcal{L}_{\text{contrast}} + \lambda_s\mathcal{L}_{\text{sup}} + \lambda_{\text{sp}}\mathcal{L}_{\text{sparse}} + \lambda_g\mathcal{L}_{\text{grad}}, \quad (3)$$

where $\mathcal{L}_{\text{recon}} = \|\mathbf{h} - \hat{\mathbf{h}}\|^2$ ensures the SAE faithfully reconstructs activations, preserving property-related information needed for downstream generation. $\mathcal{L}_{\text{sparse}} = \|\mathbf{z}\|_1$ encourages activation sparsity, ensuring each molecule is represented by a small number of active features.

Supervised Loss. As shown in Figure 1 (Stage 2B), $\mathcal{L}_{\text{sup}}$ trains a per-property MLP head f_p to predict the property value from the gated code: $\mathcal{L}_{\text{sup}} = \frac{1}{|P|}\sum_{p\in P}\frac{1}{n}\sum_{i=1}^{n}(f_p(\mathbf{w}_p \odot \mathbf{z}_i) - y_i^{(p)})^2$, where $y_i^{(p)}$ is the oracle-computed property value. While $\mathcal{L}_{\text{contrast}}$ only requires the gate to distinguish high from low property values, $\mathcal{L}_{\text{sup}}$ demands that the selected features contain sufficient information to regress the precise numerical value, providing a stronger training signal for the Importance Gate.

Contrastive Loss. As shown in Figure 1 (Stage 2C), $\mathcal{L}_{\text{contrast}}$ is a group contrastive loss that operates on importance-gated sparse codes. For each property p, molecules in a batch are ranked by oracle-computed property values;

the top 25% form the positive group $\mathcal{P}$ and the bottom 25% form the negative group $\mathcal{N}$. The gated codes $\tilde{\mathbf{z}}_i = \text{norm}(\mathbf{w}_p \odot \mathbf{z}_i)$ are L2-normalized, and the loss is: $\mathcal{L}^{(p)}_{\text{contrast}} = -\frac{1}{|\mathcal{P}|+|\mathcal{N}|}\sum_i \log \frac{\sum_{j\in\mathcal{G}(i), j\neq i}\exp(\tilde{\mathbf{z}}_i^\top \tilde{\mathbf{z}}_j/\tau_c)}{\sum_{k\neq i}\exp(\tilde{\mathbf{z}}_i^\top \tilde{\mathbf{z}}_k/\tau_c)}$, where $\mathcal{G}(i)$ is the group containing sample i. Same-group pairs are positives and cross-group pairs are negatives.

Gradient-Alignment Loss. As shown in Figure 1 (Stage 2D), the key innovation is $\mathcal{L}_{\text{grad}}$, which ensures the SAE basis faithfully represents property-relevant steering directions. For each property p, we pre-compute a gradient-based direction from the SFT training pairs $\{(\mathbf{x}_i, \mathbf{x}'_i)\}_{i=1}^N$:

$$\mathbf{d}^{(p)}_{\text{grad}} = \frac{1}{N}\sum_{i=1}^{N} \frac{\nabla_{\mathbf{h}_{l^*}} \log p_{\mathcal{M}}(\mathbf{x}'_i \mid \mathbf{x}_i)}{\|\nabla_{\mathbf{h}_{l^*}} \log p_{\mathcal{M}}(\mathbf{x}'_i \mid \mathbf{x}_i)\|} \tag{4}$$

where the gradient is averaged across all token positions and $\mathbf{x}'_i$ has improved property value. This direction captures the perturbation to $\mathbf{h}_{l^*}$ that maximally increases the model's probability of generating the property-improved target. The gradient-alignment loss then encourages the SAE's encode-then-decode pipeline to preserve this direction:

$$\mathcal{L}_{\text{grad}} = \frac{1}{|P|}\sum_{p\in P}\left(1 - \cos\left(\mathbf{W}_d \text{top}_k\left(\text{enc}(\mathbf{d}^{(p)}_{\text{grad}})\right), \mathbf{d}^{(p)}_{\text{grad}}\right)\right) \tag{5}$$

Joint Objective. These five losses serve complementary roles. $\mathcal{L}_{\text{recon}}$ and $\mathcal{L}_{\text{sparse}}$ learn a faithful sparse decomposition of model hidden states. $\mathcal{L}_{\text{contrast}}$ and $\mathcal{L}_{\text{sup}}$ shape the Importance Gates to select property-relevant features. $\mathcal{L}_{\text{grad}}$ ensures the resulting sparse basis can faithfully reconstruct causal steering directions for downstream tasks.

3.3 Inference-Time Activation Steering

As shown in Figure 1 (Stage 3), after training we extract a steering direction for each property p. We compute the per-feature mean-difference $\boldsymbol{\delta}^{(p)}$ between high- and low-property quartiles in SAE feature space, and use the Importance Gate to select the top-k features:

$$S_k = \text{top}_k\left(\sigma(\mathbf{w}_p) \odot |\boldsymbol{\delta}^{(p)}|,\ k\right), \tag{6}$$
$$\mathbf{d}^{(p)}_{\text{steer}} = \textstyle\sum_{i\in S_k} \delta^{(p)}_i\, \mathbf{w}_{d,i} \Big/ \left\|\sum_{i\in S_k} \delta^{(p)}_i\, \mathbf{w}_{d,i}\right\|$$

where $\mathbf{w}_{d,i}$ is the i-th decoder column. The Importance Gates determine *which* features contribute to steering at inference, while $\mathcal{L}_{\text{grad}}$ ensures the SAE basis is aligned with causal property directions during training.

At inference, the steering vector is added to the residual stream at all token positions in layer l^*:

$$\mathbf{h}'_{l^*,t} = \mathbf{h}_{l^*,t} + \alpha \cdot \mathbf{d}^{(p)}_{\text{steer}} \quad \forall\, t \tag{7}$$

where $\alpha > 0$ controls steering strength. This requires *no retraining*: the base model weights remain frozen, and only the added direction changes. For multi-property steering, directions combine via vector addition: $\mathbf{d}_{\text{multi}} = \sum_p \alpha_p \cdot \mathbf{d}^{(p)}_{\text{steer}}$.

4 Experiments

In this section, we empirically validate SLIM's effectiveness and design choices. We aim to answer four research questions: **RQ1**: Does SLIM improve molecular editing across diverse model architectures and properties? (§4.2) **RQ2**: Are gradient-based steering directions essential, or do simpler alternatives like mean-difference (CAA) suffice? (§4.3) **RQ3**: Does the task-oriented SAE training provide gains beyond a vanilla reconstruction-only SAE? (§4.4) **RQ4**: Can SLIM's steering directions compose for multi-property optimization? (§4.5)

4.1 Experimental Setup

Models. We evaluate SLIM on four molecular editing models spanning diverse architectures (Table 9): **DrugAssist** (Ye et al., 2025) (LLaMA-2-7B, $l^* = 3$), **GeLLM3O-LLaMA3** (Dey et al., 2025) (LLaMA-3.1-8B, $l^* = 8$), **GeLLM3O-Mistral** (Mistral-7B, $l^* = 14$), and **MolGen** (Fang et al., 2024) (BART-Large, $l^* = 5$). This covers three decoder-only causal LMs and one encoder-decoder.

Properties and Metrics. We evaluate on eight molecular properties from the MolEditRL benchmark (Zhuang et al., 2025): QED (Bickerton et al., 2012), DRD2, logP, MW, RotBond, SA (Ertl and Schuffenhauer, 2009), HBA, and HBD (full descriptions in Appendix A). We report Acc@τ: the percentage of test molecules for which at least one generated candidate improves the target property and has Tanimoto similarity $\geq \tau$ to the input, using Morgan fingerprints (Rogers and Hahn, 2010). We use $\tau = 0.15$ (permissive) and $\tau = 0.65$ (strict). The test set contains 500 molecules, with $n = 5$ candidates per molecule.

	Acc@0.15 (%)								Acc@0.65 (%)							
	DA		L3		Mi		MG		DA		L3		Mi		MG	
Property	SFT	+S	SFT	+S	SFT	+S	SFT	+S	SFT	+S	SFT	+S	SFT	+S	SFT	+S
QED ↑	77.0	**77.6**	95.8	**97.2**	71.0	**98.2**	42.4	**76.2**	60.4	**61.8**	32.4	**34.4**	52.2	48.6	19.4	**36.6**
DRD2 ↑	47.6	**50.6**	88.2	**88.8**	52.4	**90.8**	37.2	**83.6**	35.8	**42.6**	29.6	**30.4**	40.2	**47.2**	23.0	**33.4**
logP ↑	55.8	**61.2**	88.2	**89.6**	58.0	**94.4**	46.2	**86.4**	43.4	**51.4**	43.2	**48.6**	44.0	**60.2**	27.2	**46.2**
MW ↑	41.8	**46.8**	86.8	**89.8**	51.4	**93.0**	48.2	**87.0**	32.2	**38.6**	44.6	**48.4**	37.0	**63.8**	22.6	**46.4**
RotBond ↑	36.4	**41.2**	55.8	**64.2**	38.6	**75.0**	43.2	**85.6**	27.0	**33.8**	26.4	**29.8**	29.6	**43.6**	18.0	**35.4**
SA ↓	67.6	67.6	94.0	**94.8**	65.4	**95.8**	44.4	**49.4**	54.8	**55.6**	32.6	**35.4**	49.8	**55.4**	25.2	**31.2**
HBA ↑	50.6	**77.4**	45.0	**47.6**	49.4	**54.2**	42.6	**55.0**	38.4	**56.0**	20.8	**27.2**	37.0	30.6	23.0	**32.4**
HBD ↑	49.6	**74.2**	28.8	**42.2**	37.4	**47.4**	33.8	**40.8**	37.2	**55.2**	13.8	**23.8**	25.4	**30.0**	16.8	**18.2**
Avg	53.3	**62.1**	72.8	**76.8**	52.9	**81.1**	42.3	**70.5**	41.4	**49.4**	30.4	**34.8**	39.4	**47.4**	21.9	**35.0**
Δ	**+8.8**		**+3.9**		**+28.1**		**+28.2**		**+8.0**		**+4.3**		**+8.0**		**+13.1**	

Table 1: Forward-direction results on the MolEditRL benchmark (500 test molecules, $n = 5$ candidates). +S denotes SFT with SLIM steering. Δ = average gain over SFT. Bold = better of SFT vs. +S per cell.

4.2 Main Results

To answer **RQ1**, we evaluate SLIM on all four models across eight properties at both similarity thresholds (Table 1).

As shown in Table 1, SLIM improves 31/32 model-property pairs at $\tau = 0.15$. The gains are largest on models with weak SFT baselines: **Mistral** QED jumps from 71.0 to 98.2 (+27.2), and **MolGen** RotBond from 43.2 to 85.6 (+42.4), the single largest improvement. **DrugAssist** shows consistent moderate gains, with counting properties HBA (+26.8) and HBD (+24.6) as standouts. Even **LLaMA-3**, the strongest SFT model (72.8 avg), improves by +3.9 on average. At the strict threshold $\tau = 0.65$, all four models maintain positive average gains (DA +8.0, L3 +4.3, Mi +8.0, MG +13.1), indicating that SLIM improves target properties *without sacrificing molecular similarity*.

Comparison with external baselines. To position SLIM against methods from different paradigms, we compare with REINVENT4 (Loeffler et al., 2024) (RL-based) and MolEditRL (Zhuang et al., 2025) (discrete diffusion + RL) using their published results on the same 500-molecule test set. Table 2 reports the best SLIM result across our four base models for each property, compared against these external baselines on the six properties reported in both works (QED and DRD2 are omitted as MolEditRL does not report single-property results for these).

As shown in Table 2, SLIM outperforms MolEditRL on 4/6 properties at both thresholds and achieves higher averages (86.7 vs. 83.8 at $\tau = 0.15$; 55.7 vs. 51.1 at $\tau = 0.65$), despite being a training-free plug-and-play module. The gains are largest on MW at $\tau = 0.65$ (+23.4 points). MolEditRL retains an edge on counting properties HBA and HBD at $\tau = 0.15$, where its end-to-end RL optimization may better capture discrete structural changes.

	Acc@0.15 (%)			Acc@0.65 (%)		
Property	REINV.	MolEd.	+SLIM	REINV.	MolEd.	+SLIM
logP ↑	36.0	91.0	**94.4**	11.4	57.8	**60.2**
MW ↑	34.0	85.6	**93.0**	6.8	40.4	**63.8**
RotBond ↑	38.4	76.4	**85.6**	11.2	39.2	**43.6**
SA ↓	2.4	82.8	**95.8**	1.0	**62.8**	55.6
HBA ↑	40.0	**82.6**	77.4	19.0	48.4	**56.0**
HBD ↑	45.8	**84.2**	74.2	26.8	**58.2**	55.2
Avg	32.8	83.8	**86.7**	12.7	51.1	**55.7**

Table 2: Comparison with external baselines on six overlapping properties. REINVENT4 and MolEditRL results are from published papers on the same 500-molecule test set. +SLIM reports the best result across our four base models for each property. Bold = best per property.

4.3 Ablation: Steering Direction

To answer **RQ2**, we compare SLIM's gradient-based directions against simpler alternatives: **SFT** (no steering), a unit-norm **Random** vector, and **CAA** (Rimsky et al., 2024) (mean-difference between high- and low-property activations). All steered methods use the same per-property strength α. Table 3 shows per-property results on DrugAssist and Mistral.

As shown in Table 3, on DrugAssist, Random and CAA are indistinguishable from SFT (avg $\Delta < 2$ at both thresholds), confirming that mean-difference directions fail to capture property-relevant steering information. On Mistral, Random and CAA inflate Acc@0.15 for continuous properties (logP +30.8, MW +35.4) but *decrease* it for

	DrugAssist (Layer 3)								GeLLM3O-Mistral (Layer 14)							
	Acc@0.15				Acc@0.65				Acc@0.15				Acc@0.65			
	SFT	Rand.	CAA	SLIM	SFT	Rand.	CAA	SLIM	SFT	Rand.	CAA	SLIM	SFT	Rand.	CAA	SLIM
QED	77.0	76.8	76.8	**77.6**	60.4	61.0	61.6	**61.8**	71.0	96.4	96.6	**98.2**	52.2	47.6	47.2	**48.6**
DRD2	47.6	48.0	47.0	**50.6**	35.8	36.4	37.4	**42.6**	52.4	87.2	88.0	**90.8**	40.2	40.6	39.2	**47.2**
logP	55.8	55.4	56.4	**61.2**	43.4	43.8	43.0	**51.4**	58.0	88.8	86.4	**94.4**	44.0	56.6	49.6	**60.2**
MW	41.8	43.6	43.8	**46.8**	32.2	31.6	30.2	**38.6**	51.4	86.8	85.8	**93.0**	37.0	57.8	55.0	**63.8**
RotBond	36.4	36.8	37.2	**41.2**	27.0	27.8	26.0	**33.8**	38.6	50.2	50.4	**75.0**	29.6	28.2	24.8	**43.6**
SA	67.6	67.8	**69.2**	67.6	54.8	55.6	**57.2**	55.6	65.4	**97.4**	97.0	95.8	49.8	53.0	51.0	**55.4**
HBA	50.6	61.0	59.6	**77.4**	38.4	43.8	37.6	**56.0**	49.4	33.8	39.4	**54.2**	37.0	18.8	21.4	**30.6**
HBD	49.6	63.8	63.0	**74.2**	37.2	47.2	48.4	**55.2**	37.4	28.4	19.8	**47.4**	25.4	16.6	13.2	**30.0**
Avg	53.3	54.2	54.5	**62.1**	41.4	43.4	42.7	**49.4**	52.9	71.1	70.4	**81.1**	39.4	39.9	37.7	**47.4**

Table 3: Steering direction ablation on DrugAssist and Mistral. Four steering directions compared: no steering (SFT), random unit vector (Rand.), contrastive activation addition (CAA), and SLIM. All steered methods use the same per-property strength α. Bold = best per row.

counting properties (HBA -15.6, HBD -17.6 with CAA), revealing that mean-difference directions trade property gain for structural disruption. SLIM is the only method that improves Acc@0.65 on both models (DA +8.0, Mi +8.0 vs. CAA's +1.3 and -1.7), demonstrating that gradient-based directions preserve molecular similarity while increasing target properties.

4.4 Ablation: SAE Training Objective

To answer **RQ3**, we compare **Vanilla SAE** ($\mathcal{L}_{\text{recon}}$ only) and our full **Task-Oriented SAE** (Eq. 3), both using the same pre-computed gradient direction. The key question is whether the SAE basis must be aligned with property-relevant axes, or whether reconstruction alone suffices. Table 4 shows per-property results.

As shown in Table 4, a consistent ordering emerges: **SFT** $<$ **Vanilla SAE** $<$ **SLIM**. Vanilla SAE already provides partial gains (DA: +2.4, Mi: +25.2 at Acc@0.15), suggesting that even a reconstruction-only sparse bottleneck filters noise from the gradient direction. The task-oriented training adds further improvement (DA: +6.4, Mi: +3.0 over Vanilla at Acc@0.15), with the largest gains on counting properties: HBA improves by +9.2 over Vanilla on DA, and RotBond by +11.2 on Mi. The cosine similarity between Vanilla and task-oriented directions is near zero (< 0.05 for DA, < 0.12 for Mi), confirming that our losses fundamentally reorganize the latent basis rather than merely refining it.

Gradient alignment loss. We further isolate the contribution of the gradient alignment loss $\mathcal{L}_{\text{grad}}$ by training a variant with $\lambda_{\text{grad}} = 0$ (denoted **w/o** $\mathcal{L}_{\text{grad}}$), retaining all other task-oriented losses ($\mathcal{L}_{\text{recon}}$, $\mathcal{L}_{\text{contrastive}}$, $\mathcal{L}_{\text{supervised}}$). Table 5 shows results on Mistral and MolGen.

The gradient alignment loss is essential for robust steering. On MolGen, a clean ordering **SFT** $<$ **w/o** $\mathcal{L}_{\text{grad}}$ $<$ **SLIM** holds across all 8 properties at Acc@0.15, with $\mathcal{L}_{\text{grad}}$ contributing +4.4 average improvement. On Mistral, the pattern reveals a striking failure mode: without $\mathcal{L}_{\text{grad}}$, counting properties (HBA, HBD) *degrade below SFT* (-11.0 and -9.2 at Acc@0.15), even though continuous properties improve substantially. This indicates that $\mathcal{L}_{\text{grad}}$ specifically aligns the SAE basis with the gradient structure needed for discrete property control. At Acc@0.65, the gap widens further: SLIM gains +6.5 (Mi) and +3.5 (MG) over the ablated variant, confirming that gradient alignment preserves molecular similarity during steering.

4.5 Multi-Property Steering

To answer **RQ4**, we test whether SLIM's sparse steering directions can be composed for multi-property optimization. Directions combine via vector addition: $\mathbf{d}_{\text{multi}} = \sum_p \alpha_p \cdot \mathbf{d}_{\text{steer}}^{(p)}$. Table 6 evaluates six property pairs on DrugAssist and Mistral, where a candidate is "successful" only if it improves *all* target properties simultaneously with Tanimoto $\geq \tau$.

As shown in Table 6, SLIM improves joint success on 5/6 pairs at both thresholds. The largest gain is on Mi HBA+HBD (+13.6 at Acc@0.15, +6.0 at Acc@0.65), counting properties that are individually difficult but benefit from complementary feature directions. DA logP+MW also shows strong improvement (+9.4/+4.0), as both properties share lipophilicity-related features. The composability of steering directions is a direct consequence of the sparse decomposition: because each prop-

	DrugAssist (Layer 3)						GeLLM3O-Mistral (Layer 14)					
	Acc@0.15			Acc@0.65			Acc@0.15			Acc@0.65		
	SFT	Van.	SLIM	SFT	Van.	SLIM	SFT	Van.	SLIM	SFT	Van.	SLIM
QED	77.0	77.4	**77.6**	60.4	**62.0**	61.8	71.0	97.2	**98.2**	52.2	50.6	**48.6**
DRD2	47.6	47.8	**50.6**	35.8	38.4	**42.6**	52.4	90.6	**90.8**	40.2	45.0	**47.2**
logP	55.8	55.4	**61.2**	43.4	45.8	**51.4**	58.0	91.4	**94.4**	44.0	61.2	**60.2**
MW	41.8	44.0	**46.8**	32.2	35.8	**38.6**	51.4	90.8	**93.0**	37.0	60.2	**63.8**
RotBond	36.4	38.8	**41.2**	27.0	30.4	**33.8**	38.6	63.8	**75.0**	29.6	36.6	**43.6**
SA	67.6	67.2	67.6	54.8	**56.6**	55.6	65.4	**96.2**	95.8	49.8	54.4	**55.4**
HBA	50.6	68.2	**77.4**	38.4	48.2	**56.0**	49.4	**55.4**	54.2	37.0	**31.0**	30.6
HBD	49.6	66.8	**74.2**	37.2	50.4	**55.2**	37.4	39.4	**47.4**	25.4	26.4	**30.0**
Avg	53.3	55.7	**62.1**	41.4	45.9	**49.4**	52.9	78.1	**81.1**	39.4	45.7	**47.4**

Table 4: SAE training objective ablation on DrugAssist and Mistral. Three settings compared: no steering (SFT), steering with vanilla SAE ($\mathcal{L}_{\text{recon}}$ only), and steering with task-oriented SAE (Eq. 3). All use the same gradient direction. Bold = best per row.

	GeLLM3O-Mistral (Layer 14)						MolGen-Large (Layer 5)					
	Acc@0.15			Acc@0.65			Acc@0.15			Acc@0.65		
	SFT	w/o $\mathcal{L}_{\text{grad}}$	SLIM	SFT	w/o $\mathcal{L}_{\text{grad}}$	SLIM	SFT	w/o $\mathcal{L}_{\text{grad}}$	SLIM	SFT	w/o $\mathcal{L}_{\text{grad}}$	SLIM
QED	71.0	96.4	**98.2**	52.2	47.6	**48.6**	42.4	74.8	**76.2**	19.4	34.0	**36.6**
DRD2	52.4	88.8	**90.8**	40.2	44.6	**47.2**	37.2	79.0	**83.6**	23.0	31.2	**33.4**
logP	58.0	89.6	**94.4**	44.0	55.8	**60.2**	46.2	78.0	**86.4**	27.2	37.8	**46.2**
MW	51.4	87.4	**93.0**	37.0	58.2	**63.8**	48.2	83.6	**87.0**	22.6	43.8	**46.4**
RotBond	38.6	58.0	**75.0**	29.6	30.4	**43.6**	43.2	75.2	**85.6**	18.0	25.4	**35.4**
SA	65.4	95.2	**95.8**	49.8	53.8	**55.4**	44.4	46.8	**49.4**	25.2	30.4	**31.2**
HBA	49.4	38.4	**54.2**	37.0	20.4	**30.6**	42.6	52.0	**55.0**	23.0	30.2	**32.4**
HBD	37.4	28.2	**47.4**	25.4	16.6	**30.0**	33.8	39.6	**40.8**	16.8	18.8	**18.2**
Avg	52.9	72.8	**81.1**	39.4	40.9	**47.4**	42.3	66.1	**70.5**	21.9	31.5	**35.0**

Table 5: Gradient alignment loss ablation on Mistral and MolGen. w/o $\mathcal{L}_{\text{grad}}$ removes gradient alignment while retaining $\mathcal{L}_{\text{recon}}$, $\mathcal{L}_{\text{contrast}}$, $\mathcal{L}_{\text{sup}}$. Bold = best per row.

erty's direction activates a distinct subset of SAE features (Table 7), adding directions produces minimal interference.

5 Case Study: Feature Interpretability

A key advantage of SLIM over black-box steering is that each SAE feature encodes a specific molecular property, not just a structural motif. Table 7 quantifies this: for each property, the top Importance Gate feature strongly discriminates molecules by property value.

For six of eight properties, the best Importance Gate feature achieves $|\rho| \geq 0.52$ (Table 7). Feature F31553 ($\rho = +0.93$) encodes molecular weight with a 251 Da gap between high- and low-activation quartiles. Feature F5418 ($\rho = -0.85$) captures molecular complexity, cleanly separating drug-like (QED=0.77) from non-drug-like (QED=0.27) molecules. For counting properties (HBD) and binary properties (DRD2), single-feature correlations are weaker ($\rho \approx 0.32$), indicating multi-feature encoding, consistent with the composite steering directions that successfully im-

		Acc@0.15		Acc@0.65	
Model	**Properties**	SFT	+SLIM	SFT	+SLIM
DA	logP+MW	42.2	**51.6**	38.8	**42.8**
DA	QED+SA	63.8	**64.4**	54.2	54.2
DA	DRD2+SA	16.6	**17.4**	13.4	**15.0**
Mi	HBA+HBD	16.4	**30.0**	11.4	**17.4**
Mi	QED+SA	91.4	**93.4**	75.8	**77.4**
Mi	DRD2+SA	73.4	**73.6**	51.0	**51.4**

Table 6: Multi-property steering via direction addition on DrugAssist and Mistral. Joint success = candidate improves *all* target properties simultaneously with Tanimoto $\geq \tau$.

Feat.	Prop.	Top-25%	Bot-25%	Δ	ρ
F31553	MW	574.0	323.1	+250.9	+0.93
F5418	QED	0.27	0.77	−0.50	−0.85
F17369	RotBond	10.60	4.93	+5.67	+0.68
F31602	logP	6.25	2.63	+3.62	+0.65
F18632	HBA	6.66	3.45	+3.21	+0.63
F17369	SA	3.97	3.01	+0.96	+0.52
F4291	HBD	2.07	0.98	+1.10	+0.32
F7906	DRD2	0.13	0.08	+0.06	+0.32

Table 7: SAE feature interpretability (DrugAssist, Layer 3, 50K ZINC molecules). For each property, we report the best feature among the top-50 Importance Gate features ranked by $|\rho|$. Top/Bot-25% = mean property value of molecules with highest/lowest 25% feature activation. ρ = Spearman rank correlation between feature activation and property value.

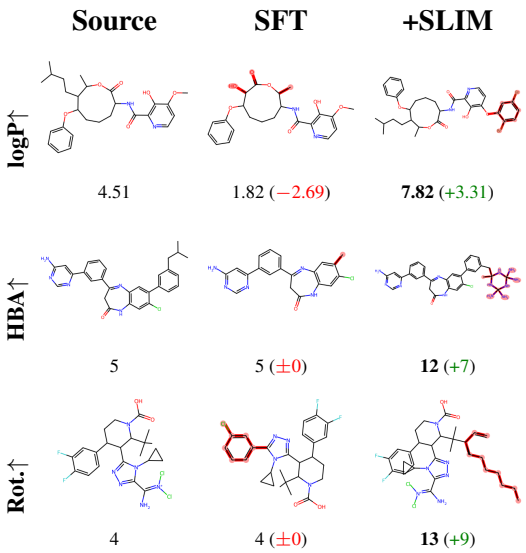


Figure 2: Molecular editing examples on DrugAssist. Each row shows source, SFT output, and SLIM output for one property. Numbers below molecules indicate property values and changes (Δ). Tanimoto similarities (SFT / SLIM): logP 0.65 / 0.79, HBA 0.69 / 0.75, RotBond 0.67 / 0.72.

prove these properties (Table 1).

To confirm that features are not merely correlational, we steer using individual SAE decoder columns ($\alpha = 2.0$, 500 test molecules). Single-feature steering produces targeted molecular changes: F3714 (logP-relevant) adds aromatic groups (+3.0 avg logP), F28089 (HBD-relevant) introduces NH groups (+1–2 donors), with 13–96 contrast pairs per feature where SFT fails but single-feature steering succeeds. Full molecule visualizations are in Appendix B.

Figure 2 shows representative cases where SFT fails but SLIM succeeds, each reflecting a distinct failure mode of the SFT editor.

Property-reversing edit (logP). The source molecule (logP=4.51) contains an isopentyl chain contributing to hydrophobicity. SFT removes this chain and replaces it with a hydroxyl group, *decreasing* logP to 1.82, a structurally valid but property-reversing edit. SLIM instead replaces the methoxypyridine ring with a dibromophenoxy moiety, adding halogenated aromatic bulk that raises logP to 7.82 (+3.31) with higher structural similarity (Tanimoto 0.79 vs. 0.65).

Ineffective edit (HBA). SFT removes a side chain but leaves acceptor count unchanged (5→5). SLIM introduces a guanidinium-like group with multiple nitrogen and oxygen atoms, adding 7 new acceptor sites (5→12) by activating SAE features associated with nitrogen-rich heterocycles.

Overly conservative edit (RotBond). SFT modifies a substituent on the triazole ring but does not introduce any additional rotatable bonds (4→4). SLIM instead extends the molecule with a flexible alkenyl-alkyl chain, adding 9 rotatable bonds (4→13, Tanimoto 0.72), overcoming SFT's tendency toward rigid-scaffold-preserving edits.

These cases share a common pattern: SFT produces valid edits but lacks a mechanism to ensure they align with the target property. SLIM provides this directional signal through property-relevant SAE features.

6 Conclusion

We presented SLIM, a plug-and-play framework that decomposes the dense hidden states of molecular editing LLMs into sparse, property-aligned features, enabling training-free property control at inference time without modifying model parameters. Our task-oriented SAE, trained with contrastive, predictive, and gradient-alignment objectives alongside per-property Importance Gates, produces a sparse basis that is causally aligned with target properties rather than merely correlated. Experiments on the MolEditRL benchmark across four architectures and eight properties show that SLIM consistently improves property-directed editing accuracy, with gains of up to 42.4 percentage points. Case studies further reveal that the learned SAE features encode interpretable chemical semantics and that steering corrects specific failure modes of SFT editors, including property-reversing edits, ineffective modifications, and overly conservative structural changes.

Limitations

SLIM has several limitations. The steering strength and intervention layer require per-property and per-model tuning, adding setup cost before deployment. The SAE training pipeline introduces additional computational overhead beyond the base SFT model, though inference-time cost is negligible. Finally, our evaluation relies on computable property oracles, and the framework has not yet been validated on experimental assay endpoints.

References

G. Richard Bickerton, Gaia V. Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L. Hopkins. 2012. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4(2):90–98.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, and 5 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *International Conference on Learning Representations*. ArXiv:2309.08600.

Vishal Dey, Xiao Hu, and Xia Ning. 2025. GeLLMO: Generalizing large language models for multi-property molecule optimization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. ArXiv:2502.13398.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Transformer Circuits Thread*. ArXiv:2209.10652.

Peter Ertl and Ansgar Schuffenhauer. 2009. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of Cheminformatics*, 1(1):8.

Yin Fang, Ningyu Zhang, Zhuo Chen, Lingbing Guo, Xiaohui Fan, and Huajun Chen. 2024. Domain-agnostic molecular generation with chemical feedback. In *International Conference on Learning Representations*. ArXiv:2301.11259.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

James P. Hughes, Stephen Rees, S. Barrett Kalindjian, and Karen L. Philpott. 2011. Principles of early drug discovery. *British Journal of Pharmacology*, 162(6):1239–1249.

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

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

José Jiménez-Luna, Francesca Grisoni, and Gisbert Schneider. 2020. Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, 2(10):573–584.

Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alán Aspuru-Guzik. 2020. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024.

Greg Landrum and 1 others. RDKit: Open-source cheminformatics. `https://www.rdkit.org`.

Seul Lee, Jaehyeong Jo, and Sung Ju Hwang. 2023. Exploring chemical space with score-based out-of-distribution generation. In *Proceedings of the 40th International Conference on Machine Learning*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Llama Team, AI @ Meta. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Hannes H. Loeffler, Jiazhen He, Alessandro Tibo, Jon Paul Janet, Alexey Voronov, Lewis H. Mervin, and Ola Engkvist. 2024. REINVENT 4: Modern AI-driven generative molecule design. *Journal of Cheminformatics*, 16:20.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *PMLR*, pages 39643–39666.

Qizhi Pei, Wei Zhang, Jinhua Zhu, Kehan Wu, Kaiyuan Gao, Lijun Wu, Yingce Xia, and Rui Yan. 2023. BioT5: Enriching cross-modal integration in biology with chemical knowledge and natural language associations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1102–1123.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. 2024. Improving dictionary learning with gated sparse autoencoders. *arXiv preprint arXiv:2404.16014*.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522.

David Rogers and Mathew Hahn. 2010. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754.

Lee Sharkey, Dan Braun, and Beren Millidge. 2022. Taking features out of superposition with sparse autoencoders. Alignment Forum.

Teague Sterling and John J. Irwin. 2015. ZINC 15 – ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 55(11):2324–2337.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Ulisse Mini, and Monte MacDiarmid. 2024. Activation addition: Steering language models without optimization.

Geyan Ye, Xibao Cai, Houtim Lai, Xing Wang, Junhong Huang, Longyue Wang, Wei Liu, and Xiangxiang Zeng. 2025. DrugAssist: A large language model for molecule optimization. *Briefings in Bioinformatics*, 26(1):bbae693. ArXiv:2401.10334.

Mingxu Zhang, Dazhong Shen, and Ying Sun. 2025a. AtomDisc: An atom-level tokenizer that boosts molecular LLMs and reveals structure–property associations. *arXiv preprint arXiv:2512.03080*.

Mingxu Zhang, Dazhong Shen, Qi Zhang, and Ying Sun. 2025b. ChemATP: A training-free chemical reasoning framework for large language models. *arXiv preprint arXiv:2512.19240*.

Yuanxin Zhuang, Dazhong Shen, and Ying Sun. 2025. MolEditRL: Structure-preserving molecular editing via discrete diffusion and reinforcement learning. *arXiv preprint arXiv:2505.20131*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2025. Representation engineering: A top-down approach to AI transparency. In *International Conference on Learning Representations*. ArXiv:2310.01405.

A Experimental Settings

Evaluation Metrics. We follow the MolEditRL benchmark protocol (Zhuang et al., 2025). For each test molecule x and property p, we generate $n = 5$ candidate molecules $\{x'_1, \ldots, x'_n\}$ via nucleus sampling. A candidate x'_i is considered *successful* if it satisfies three conditions: (1) it is a chemically valid SMILES string (verified by RDKit (Landrum et al.) parsing), (2) it improves the target property relative to the input (oracle(x'_i) > oracle(x) for increase tasks, or < for decrease tasks), and (3) its Tanimoto similarity to the input molecule meets or exceeds the threshold: Tanimoto$(x, x'_i) \geq \tau$. Tanimoto similarity is computed on Morgan fingerprints (Rogers and Hahn, 2010) with radius 2 and 2048 bits. We report Acc@τ: the percentage of test molecules for which at least one candidate is successful. We evaluate at two thresholds: $\tau = 0.15$ (permissive, allowing larger structural changes) and $\tau = 0.65$ (strict, requiring high structural preservation).

Test Set. We use the 500-molecule test set from the DrugAssist evaluation suite, which is a standardized subset also adopted by MolEditRL. These molecules span diverse drug-like chemical space from the DrugAssist training distribution.

Properties and Oracles. Table 8 describes all eight molecular properties and their computational oracles.

Models. Table 9 summarizes the four base models evaluated.

Hyperparameters. Table 10 lists all key hyperparameters for each component.

Steering Strength Selection. The steering strength α is tuned per property by evaluating on a held-out 100-molecule validation split. We search over $\alpha \in \{0.01, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 1.0, 1.5, 2.0, 5.0, 10.0\}$ and select the α maximizing Acc@0.15 on the validation set. Optimal values vary by model and property type: composite properties (QED, SA) prefer smaller strengths ($\alpha \leq 0.5$), while counting properties (HBA, HBD, RotBond) benefit from larger strengths ($\alpha \geq 1.0$). MolGen (BART, $d = 1024$) requires $\sim$10$\times$ larger α than the 4096-dimensional causal LMs. Table 11 lists the selected values.

Property	Oracle	Dir.
QED	RDKit QED (Bickerton et al., 2012) — quantitative estimate of drug-likeness	↑
DRD2	Random Forest classifier on Morgan FP (radius 2, 2048 bits)	↑
logP	RDKit Crippen MolLogP — octanol-water partition coefficient	↑
MW	RDKit MolWt — molecular weight in Daltons	↑
RotBond	RDKit NumRotatableBonds	↑
SA	SA scorer (Ertl and Schuffenhauer, 2009) — synthetic accessibility (lower = easier)	↓
HBA	RDKit NumHAcceptors — hydrogen bond acceptor count	↑
HBD	RDKit NumHDonors — hydrogen bond donor count	↑

Table 8: Molecular properties and oracles. Dir. indicates optimization direction. All oracles are deterministic and computed from molecular structure.

Model	Architecture	Layer l^*	d
DrugAssist (DA)	LLaMA-2-7B	3	4096
GeLLM3O-L3 (L3)	LLaMA-3.1-8B	8	4096
GeLLM3O-Mi (Mi)	Mistral-7B	14	4096
MolGen (MG)	BART-Large	5	1024

Table 9: Models evaluated. Layer l^* is selected by ridge probing (§3.1). d = hidden dimension. SAE expansion = 8$\times$.

Contrastive Loss Details. The group contrastive loss $\mathcal{L}_{\text{contrast}}$ operates per property. For each property p, molecules in a training batch are ranked by their oracle-computed property value. The top 25% molecules form the positive set $\mathcal{P}$ and the bottom 25% form the negative set $\mathcal{N}$. Each sample's importance-gated code $\mathbf{z}_p = \mathbf{w}_p \odot \mathbf{z}$ is L2-normalized, and the group contrastive loss with temperature $\tau_c = 0.07$ is:

$$\mathcal{L}^{(p)}_{\text{contrast}} = -\frac{1}{|\mathcal{P}|+|\mathcal{N}|} \sum_i \log \frac{\sum_{j \in \mathcal{G}(i), j \neq i} \exp(\text{sim}_{ij}/\tau_c)}{\sum_{k \neq i} \exp(\text{sim}_{ik}/\tau_c)} \tag{8}$$

where sim_{ij} is the cosine similarity between L2-normalized gated codes and $\mathcal{G}(i) \in \{\mathcal{P}, \mathcal{N}\}$ is the group containing sample i. The total contrastive loss averages over all properties: $\mathcal{L}_{\text{contrast}} = \frac{1}{|P|} \sum_p \mathcal{L}^{(p)}_{\text{contrast}}$.

SFT (LoRA (Hu et al., 2022))		*Task-Oriented SAE*		*Gradient Direction*	
LoRA rank r	16	Architecture	Gated SAE	Precision	fp32
LoRA alpha α_{lora}	32	Expansion factor	$8\times$ ($d \to 8d$)	Pairs per property	5,000
Target modules	q_proj, v_proj	Learning rate	1e-3 (Adam)	Token aggregation	Mean all positions
Learning rate	2e-4 (cosine)	Batch size	256	Per-example norm	L2 before avg
Batch size	128	Epochs	50		
Epochs	3	Molecules	50K (ZINC)	*Inference*	
Optimizer	AdamW	Decoder norm	Unit-norm cols	Candidates/molecule	5
Precision	bf16	λ_c (contrastive)	0.1	Temperature	0.8
		λ_s (supervised)	0.1	Top-p (nucleus)	0.95
		λ_{sp} (sparsity)	1e-3	Max new tokens	256
		λ_g (gradient)	0.5	Thresholds τ	0.15, 0.65
		Top-k for steering	200		

Table 10: Full hyperparameter settings. Steering strength α is tuned per property on a held-out validation set; optimal values are listed in Table 11.

Property	DA	L3	Mi	MG
QED	0.07	1.5	0.5	5.0
DRD2	0.7	0.7	0.3	10.0
logP	0.7	0.7	0.5	10.0
MW	0.5	1.5	0.3	10.0
RotBond	1.0	1.0	1.0	10.0
SA	0.2	0.1	0.1	5.0
HBA	2.0	1.0	1.0	10.0
HBD	2.0	2.0	1.0	10.0

Table 11: Per-property steering strength α selected on validation set. Composite properties (QED, SA) use small α; counting properties (HBA, HBD) use large α. MolGen ($d = 1024$) requires larger α than 4096-dim models.

Property	Prompt Description
QED	"drug-likeness (QED)"
DRD2	"DRD2 binding activity"
logP	"lipophilicity (logP)"
MW	"molecular weight"
RotBond	"number of rotatable bonds"
SA	"synthetic accessibility"
HBA	"number of hydrogen bond acceptors"
HBD	"number of hydrogen bond donors"

Table 12: Property descriptions used in instruction prompts.

Supervised Loss Details. The supervised loss $\mathcal{L}_{\text{sup}}$ uses a per-property 2-layer MLP (hidden dim 256, GELU activation) that takes the importance-gated code $\mathbf{z}_p$ as input and predicts the normalized property value. The loss is MSE averaged over properties: $\mathcal{L}_{\text{sup}} = \frac{1}{|P|}\sum_p \text{MSE}(\hat{y}_p, y_p)$.

Instruction Templates. For causal LMs (DrugAssist, LLaMA-3, Mistral), we use the LLaMA-2 chat format:

```
<s>[INST] Modify this molecule to
{increase/decrease} its {property
description}:  {SMILES} [/INST]
```

Property descriptions use full natural language names to match SFT training prompts:

For MolGen (BART encoder-decoder), the input is the source SMILES directly, and the model generates the target SMILES via beam search (beam width 5).

Computational Infrastructure. All experiments were conducted on NVIDIA A100 80GB GPUs. SFT training uses 2 GPUs with DeepSpeed ZeRO-2. SAE training and evaluation use a single GPU. Gradient direction computation requires fp32 precision and takes approximately 2 hours per model per property on a single A100. SAE training converges in approximately 30 minutes. Inference-time steering adds negligible overhead (<1% latency increase) as it only involves a single vector addition per forward pass.

B Case Studies

B.1 Single-Feature Steering

Tables 13–16 show molecular editing examples using *single SAE feature* steering, demonstrating that individual features capture interpretable chemical semantics. All cases use the DrugAssist model with Layer-3 SAE steering. The instruction prompt for each property group is shown above the table; `{SMILES}` is replaced with the source molecule at inference time.

B.2 SLIM Steering Examples

Figures 5–10 show molecular editing examples using the full SLIM steering pipeline on DrugAssist. Red-highlighted regions indicate structural modifi-

`[INST] Modify this molecule to increase its lipophilicity (logP): {SMILES} [/INST]`

SAE Feature F3714, semantics: phenyl 94%, halogen 79%, logP↑165%

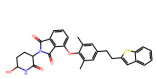
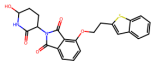
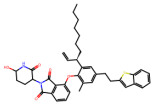
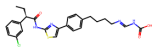
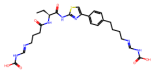
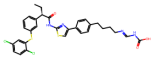
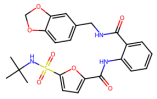
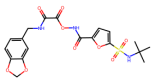
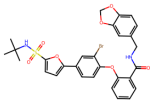
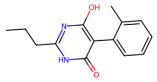
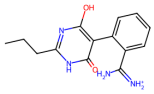
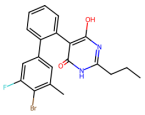
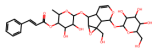
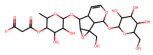
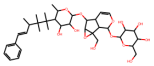
Source	SFT	+SLIM	Property / Similarity
			SFT: 2.72 (Δ-2.57, ✗) Sim 0.72 SLIM: 8.61 (Δ+3.32, ✓) Sim 0.75
			SFT: 3.34 (Δ-2.88, ✗) Sim 0.68 SLIM: 9.02 (Δ+2.81, ✓) Sim 0.81
			SFT: 0.59 (Δ-2.68, ✗) Sim 0.65 SLIM: 6.24 (Δ+2.97, ✓) Sim 0.62
			SFT: -0.44 (Δ-2.84, ✗) Sim 0.68 SLIM: 4.97 (Δ+2.57, ✓) Sim 0.64
			SFT: -6.05 (Δ-3.57, ✗) Sim 0.69 SLIM: 0.92 (Δ+3.39, ✓) Sim 0.67

Table 13: Case study: steering logP with Feature F3714. Source logP values: 5.29, 6.21, 3.27, 2.40, -2.47. SFT *decreases* logP in every case; SLIM steering *increases* it by +2.6 to +3.4.

cations relative to the source molecule. Each figure shows cases where SFT fails to improve the target property but SLIM succeeds. For each case, we report the property value, change (Δ), and Tanimoto similarity (Sim) to the source molecule.

GeLLM3O-Mistral

GeLLM3O-LLaMA3

MolGen-Large

`[INST] Modify this molecule to increase its drug-likeness (QED): {SMILES} [/INST]`

SAE Feature F4829, semantics: compact heterocyclic scaffolds, MW∼200, QED↑, MW↓95%

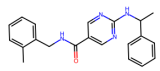
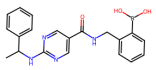
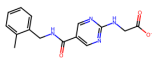
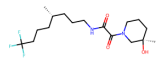
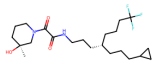
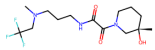
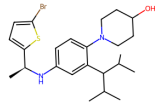
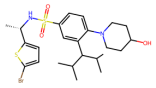
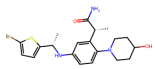
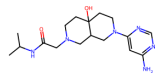
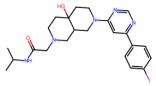
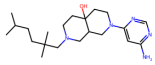
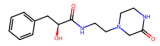
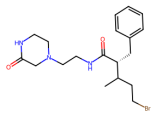
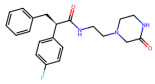
Source	SFT	+SLIM	Property / Similarity
			SFT: 0.46 (Δ-0.25, ✗) Sim 0.76 SLIM: 0.77 (Δ+0.06, ✓) Sim 0.62
			SFT: 0.38 (Δ-0.16, ✗) Sim 0.75 SLIM: 0.56 (Δ+0.02, ✓) Sim 0.64
			SFT: 0.44 (Δ-0.01, ✗) Sim 0.69 SLIM: 0.61 (Δ+0.17, ✓) Sim 0.76
			SFT: 0.57 (Δ-0.12, ✗) Sim 0.71 SLIM: 0.80 (Δ+0.10, ✓) Sim 0.65
			SFT: 0.61 (Δ-0.03, ✗) Sim 0.68 SLIM: 0.78 (Δ+0.14, ✓) Sim 0.72

Table 14: Case study: steering QED with Feature F4829. Source QED values: 0.71, 0.54, 0.45, 0.69, 0.64. SFT *decreases* QED in every case; SLIM steering *increases* it by +0.02 to +0.17.

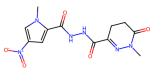
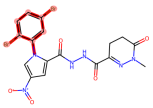
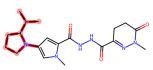
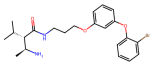
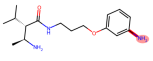
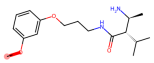
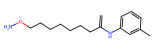
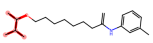
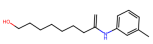
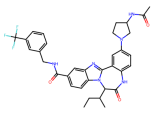
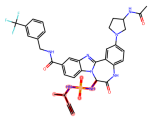
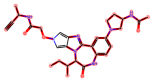
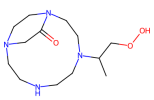
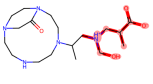
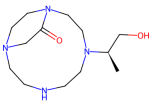
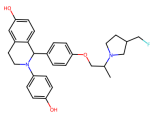
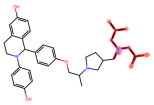
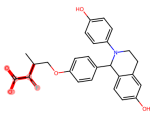
Source	SFT	+SLIM	Source	SFT	+SLIM
					
0.56	0.45 (-0.11) Sim 0.68 **0.75** (+0.19) Sim 0.54		0.51	0.50 (-0.01) Sim 0.67 **0.69** (+0.17) Sim 0.70	
					
0.49	0.46 (-0.04) Sim 0.70 **0.64** (+0.15) Sim 0.71		0.24	0.17 (-0.07) Sim 0.75 **0.37** (+0.14) Sim 0.56	
					
0.51	0.38 (-0.13) Sim 0.57 **0.63** (+0.13) Sim 0.74		0.48	0.30 (-0.18) Sim 0.71 **0.57** (+0.09) Sim 0.62	

Figure 3: SLIM steering examples: QED↑ (DrugAssist). Red highlights indicate structural modifications. Each case shows SFT failing while SLIM succeeds.

```
[INST] Modify this molecule to increase its number of hydrogen bond donors:
{SMILES} [/INST]
```

SAE Feature F28089, semantics: NH 72%, phenyl only 40%, low logP

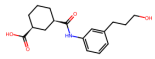
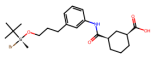
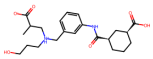
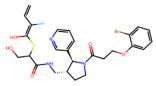
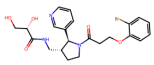
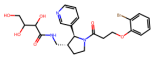
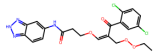
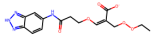
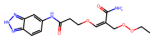
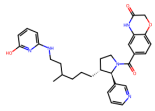
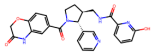
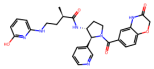
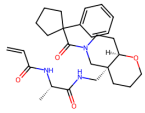
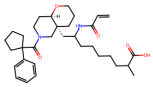
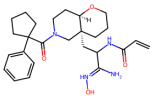
Source	SFT	+SLIM	Property / Similarity
			SFT: 2 (Δ-1, ✗) Sim 0.65 SLIM: 4 (Δ+1, ✓) Sim 0.64
			SFT: 3 (Δ0, ✗) Sim 0.71 SLIM: 4 (Δ+1, ✓) Sim 0.69
			SFT: 2 (Δ0, ✗) Sim 0.71 SLIM: 3 (Δ+1, ✓) Sim 0.71
			SFT: 3 (Δ0, ✗) Sim 0.66 SLIM: 4 (Δ+1, ✓) Sim 0.77
			SFT: 2 (Δ0, ✗) Sim 0.68 SLIM: 4 (Δ+2, ✓) Sim 0.67

Table 15: Case study: steering H-bond donors with Feature F28089. Source HBD values: 3, 3, 2, 3, 2. SFT fails to increase donors (unchanged or decreased); SLIM steering adds +1 to +2.

`[INST] Modify this molecule to increase its number of hydrogen bond acceptors: {SMILES} [/INST]`

SAE Feature F18533, semantics: piperidine 33%, ester 26%, phenyl 84%

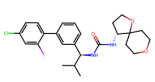
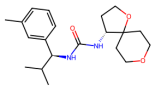
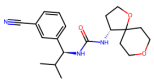
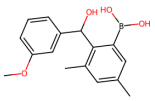
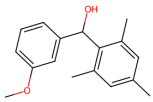
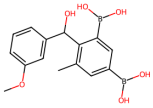
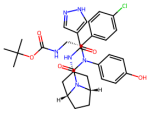
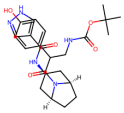
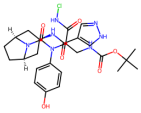
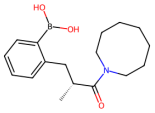
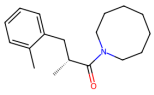
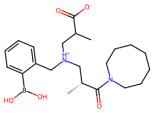
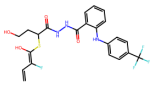
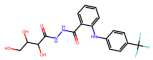
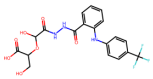
Source	SFT	+SLIM	Property / Similarity
			SFT: 3 (Δ0, ✗) Sim 0.67 SLIM: 4 (Δ+1, ✓) Sim 0.63
			SFT: 2 (Δ-2, ✗) Sim 0.79 SLIM: 6 (Δ+2, ✓) Sim 0.79
			SFT: 6 (Δ0, ✗) Sim 0.68 SLIM: 7 (Δ+1, ✓) Sim 0.75
			SFT: 1 (Δ-2, ✗) Sim 0.72 SLIM: 5 (Δ+2, ✓) Sim 0.63
			SFT: 6 (Δ0, ✗) Sim 0.56 SLIM: 7 (Δ+1, ✓) Sim 0.54

Table 16: Case study: steering H-bond acceptors with Feature F18533. Source HBA values: 3, 4, 6, 3, 6. SFT fails to increase acceptors (unchanged or decreased); SLIM steering adds +1 to +2.

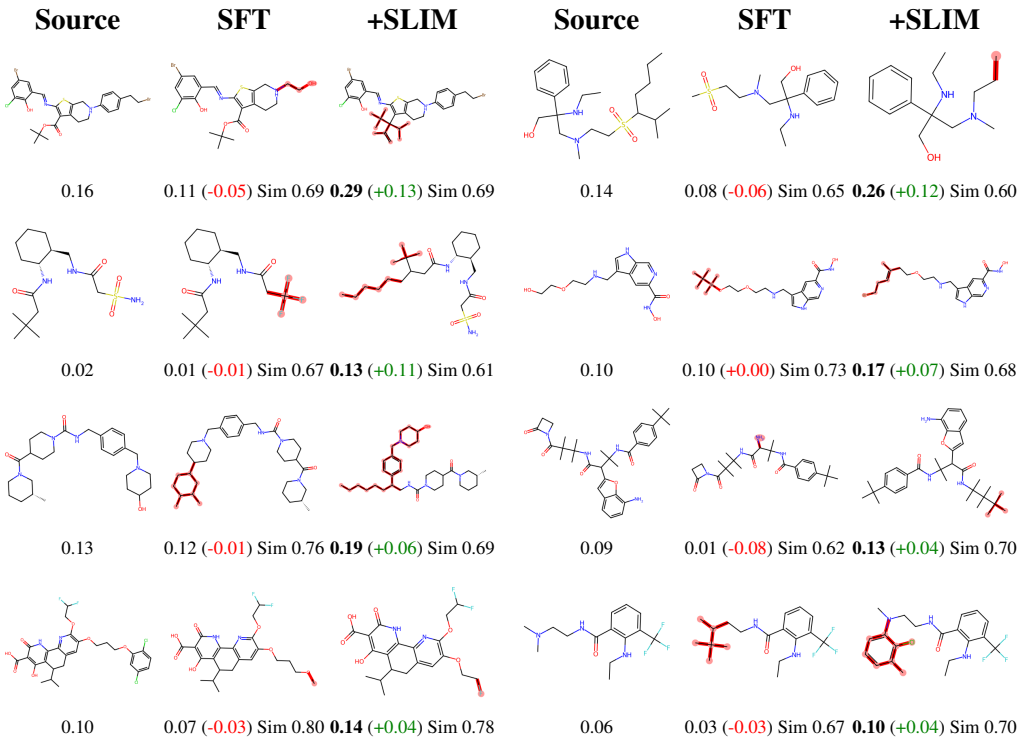


Figure 4: SLIM steering examples: DRD2↑ (DrugAssist). Red highlights indicate structural modifications. Each case shows SFT failing while SLIM succeeds.

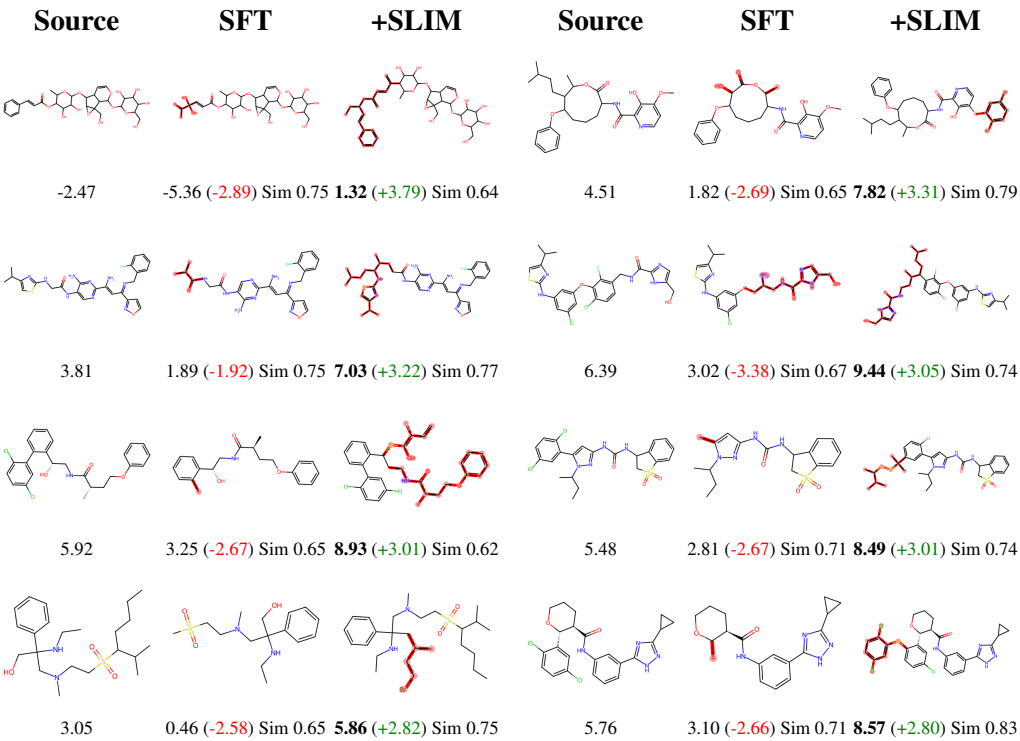


Figure 5: SLIM steering examples: logP↑ (DrugAssist). Red highlights indicate structural modifications. Each case shows SFT failing while SLIM succeeds.

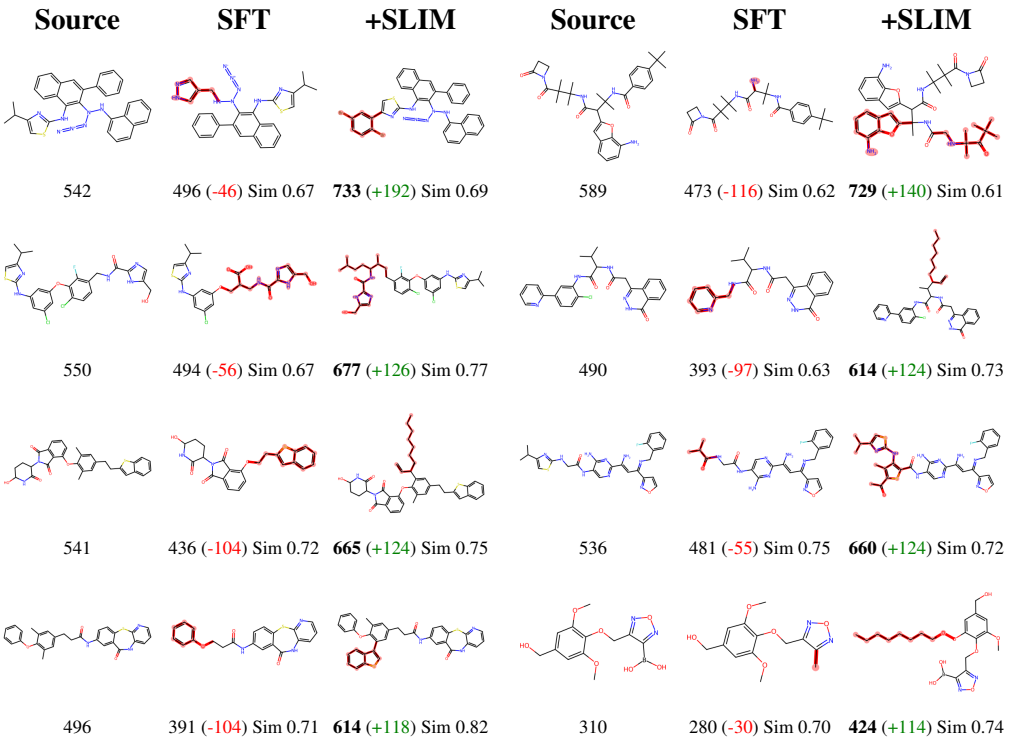


Figure 6: SLIM steering examples: MW↑ (DrugAssist). Red highlights indicate structural modifications. Each case shows SFT failing while SLIM succeeds.

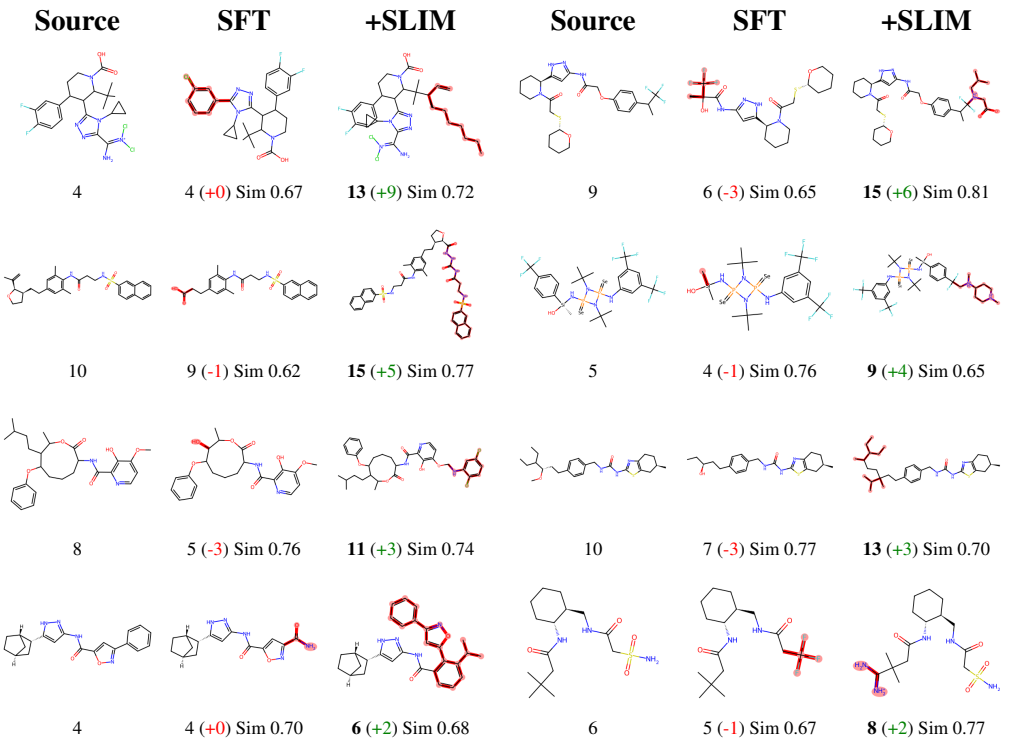


Figure 7: SLIM steering examples: RotBond↑ (DrugAssist). Red highlights indicate structural modifications. Each case shows SFT failing while SLIM succeeds.

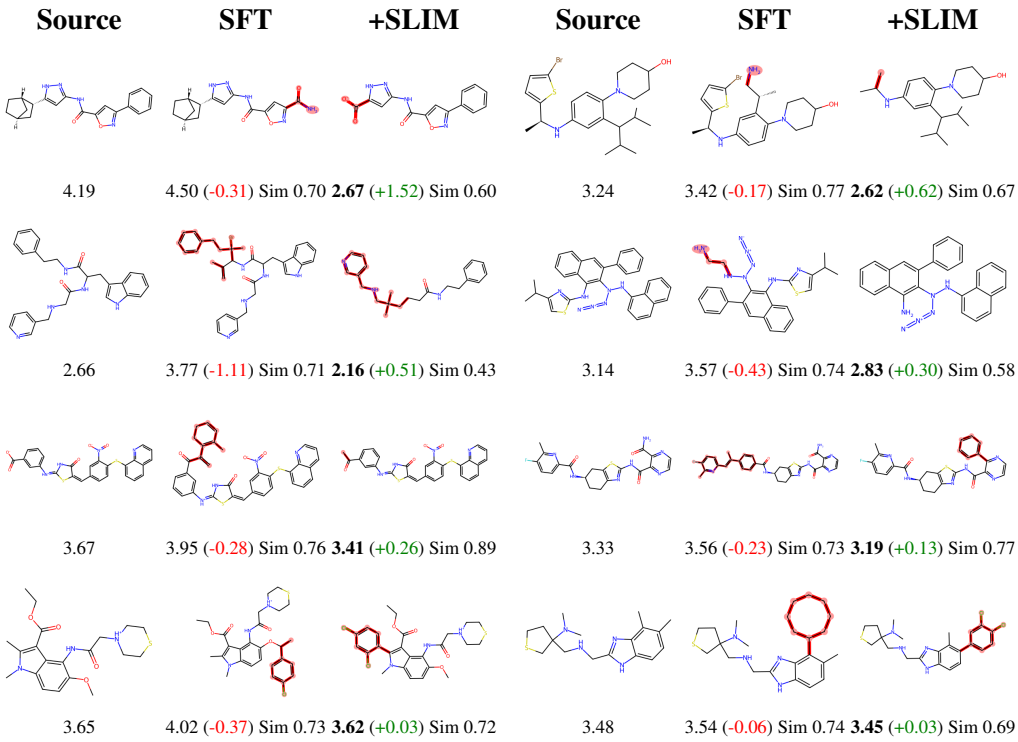


Figure 8: SLIM steering examples: SA↓ (DrugAssist). Red highlights indicate structural modifications. Each case shows SFT failing while SLIM succeeds.

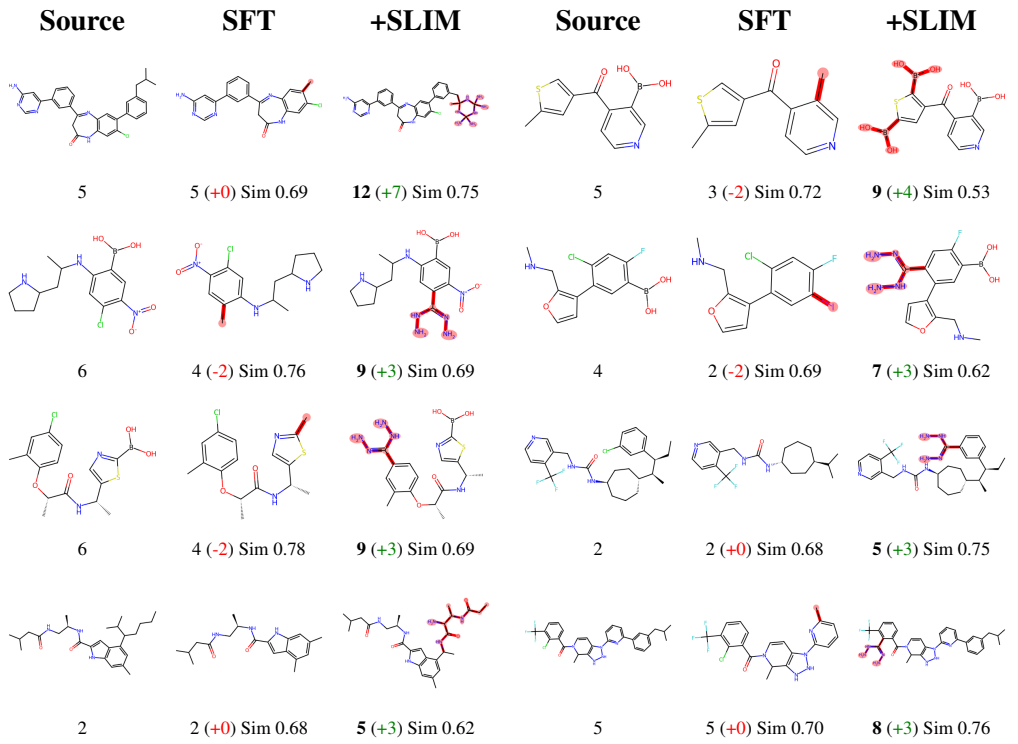


Figure 9: SLIM steering examples: HBA↑ (DrugAssist). Red highlights indicate structural modifications. Each case shows SFT failing while SLIM succeeds.

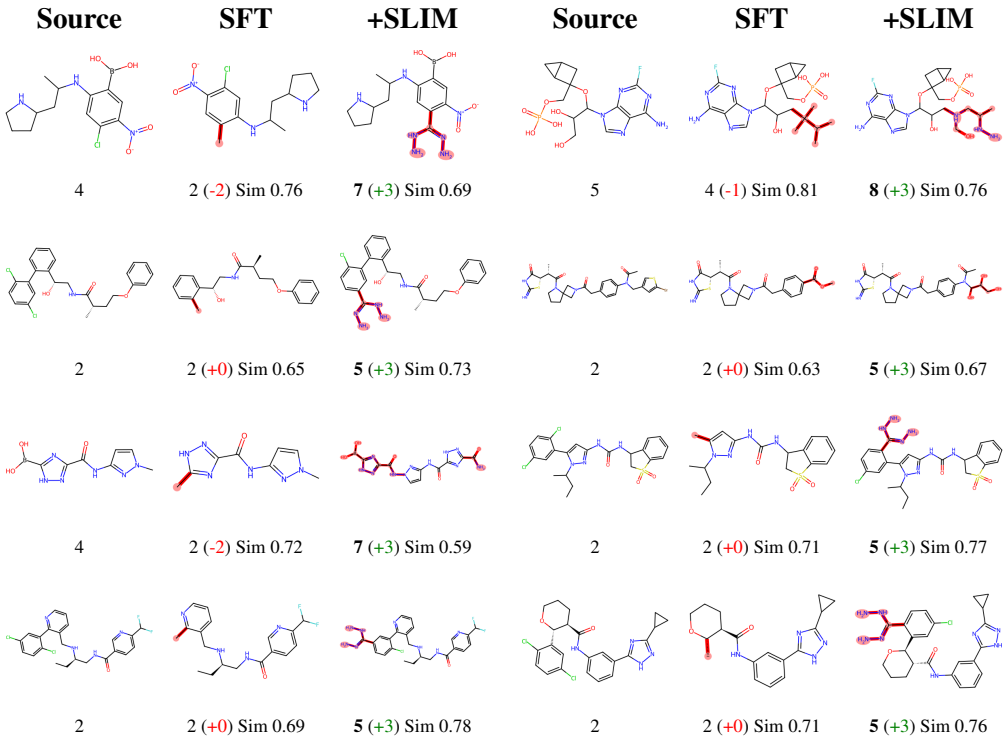


Figure 10: SLIM steering examples: HBD↑ (DrugAssist). Red highlights indicate structural modifications. Each case shows SFT failing while SLIM succeeds.

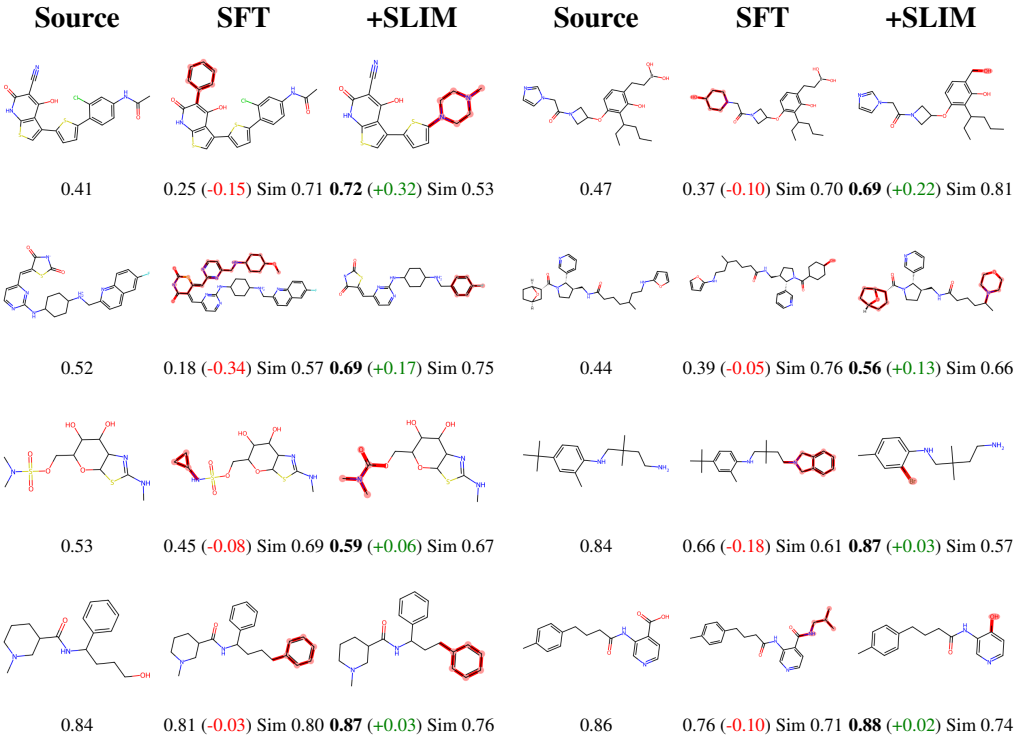


Figure 11: SLIM steering examples: QED↑ (GeLLM3O-Mistral).

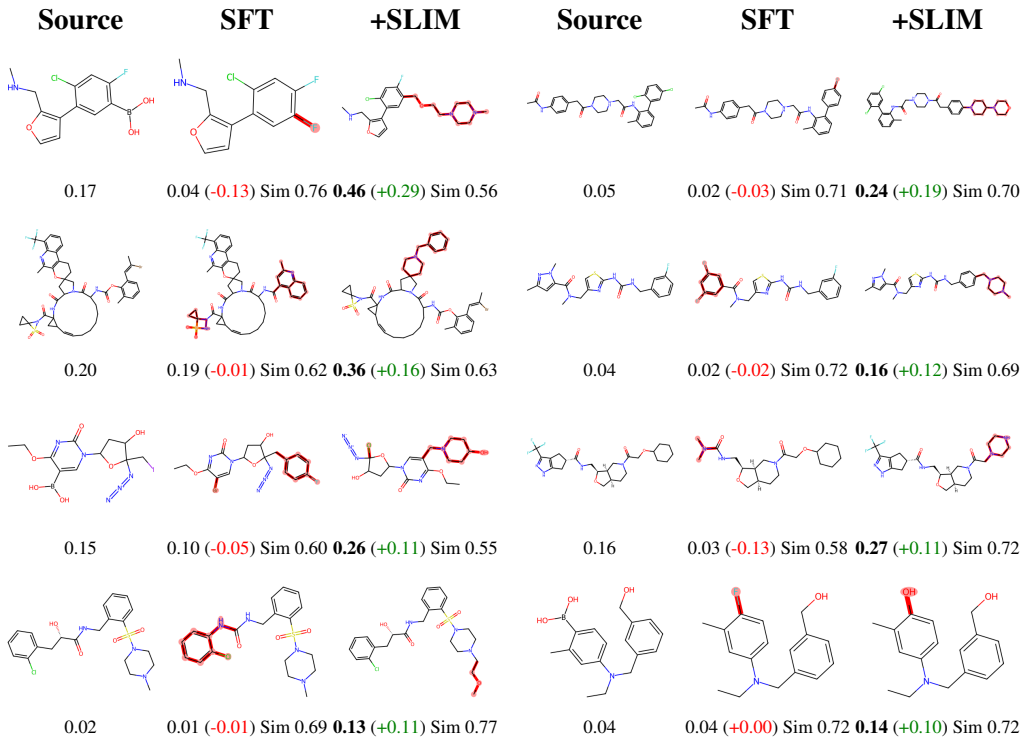


Figure 12: SLIM steering examples: DRD2↑ (GeLLM3O-Mistral).

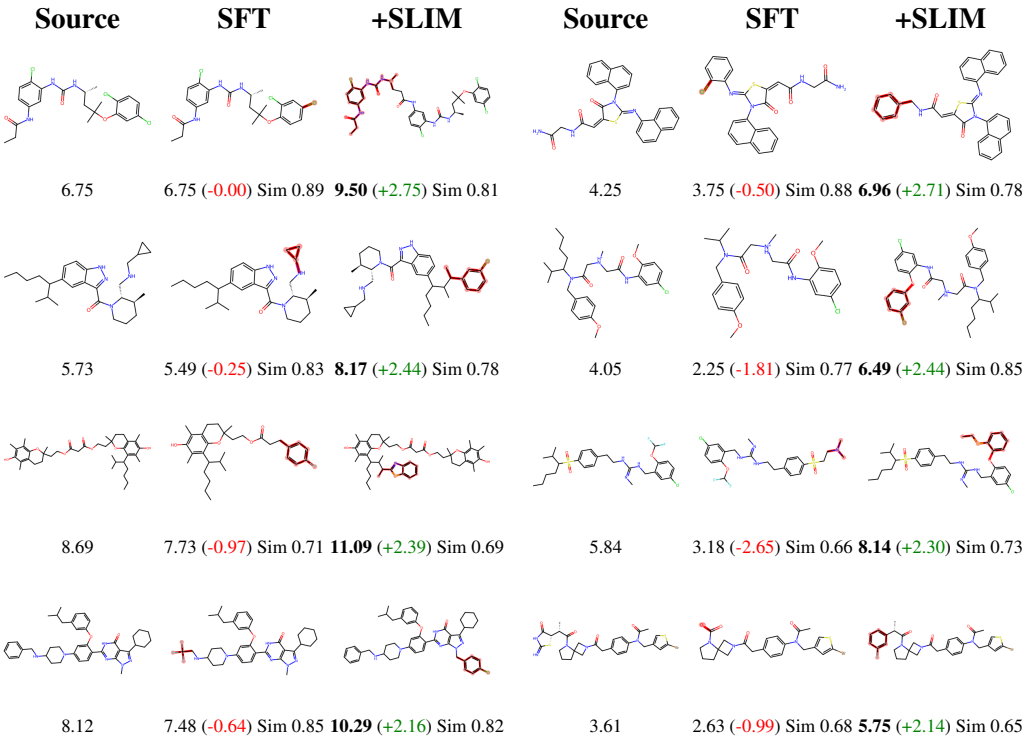


Figure 13: SLIM steering examples: logP↑ (GeLLM3O-Mistral).

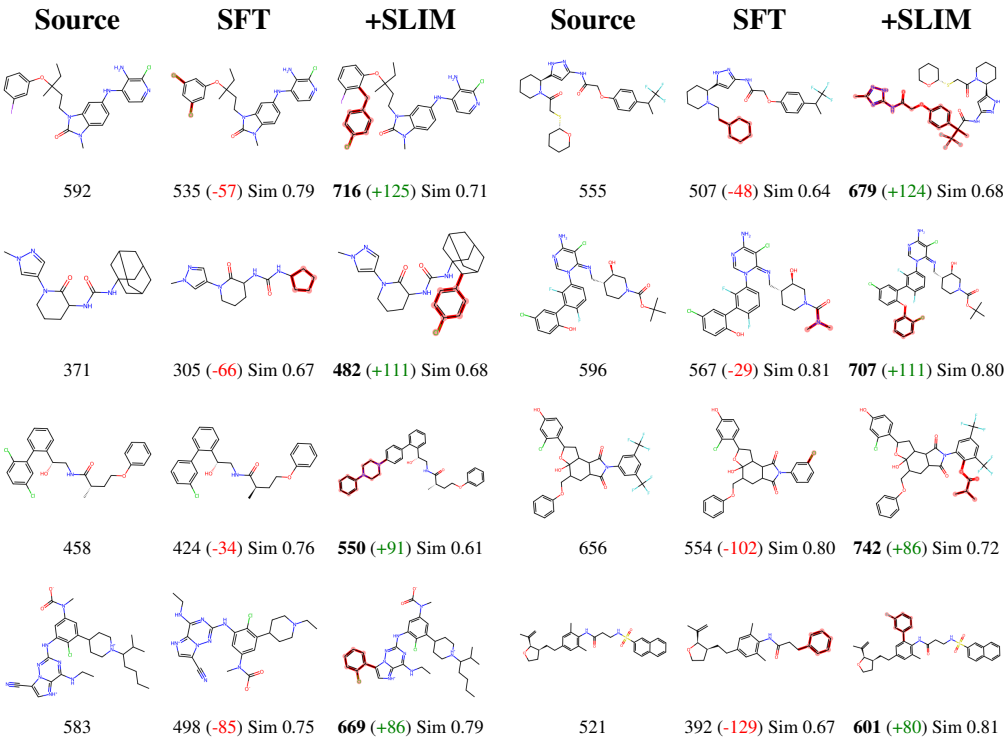


Figure 14: SLIM steering examples: MW↑ (GeLLM3O-Mistral).

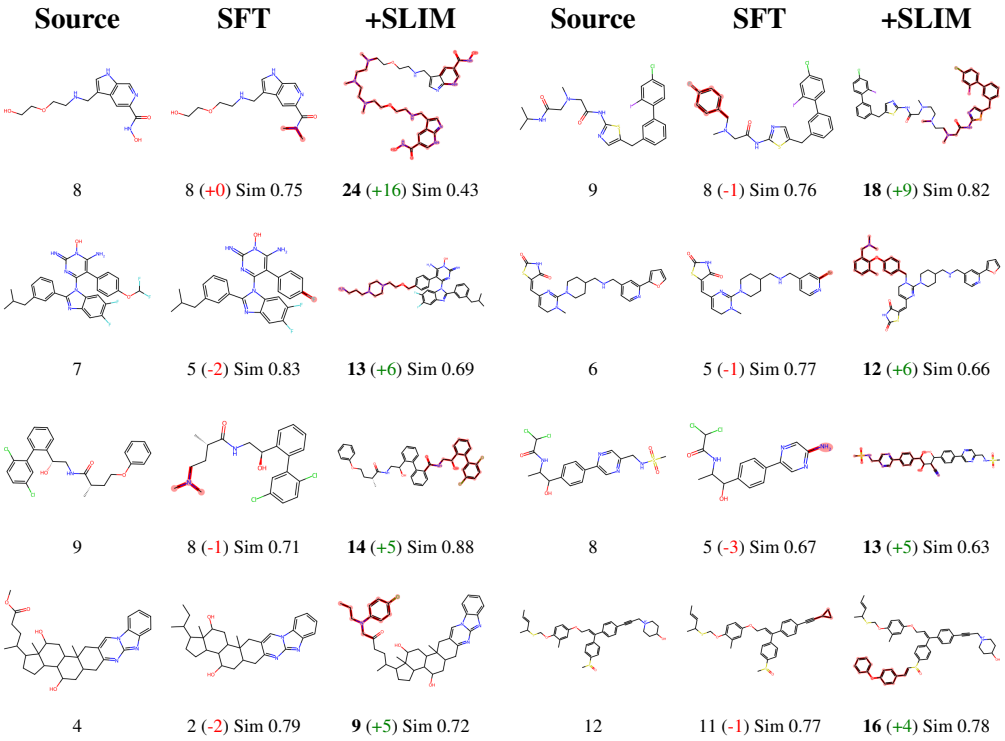


Figure 15: SLIM steering examples: RotBond↑ (GeLLM3O-Mistral).

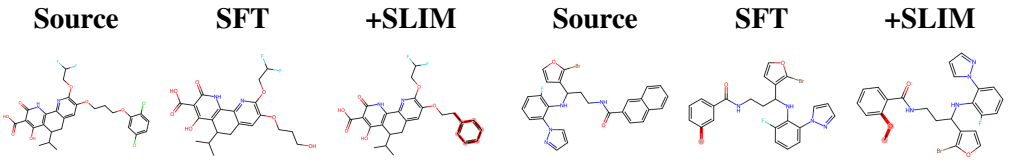


Figure 16: SLIM steering examples: SA↓ (GeLLM3O-Mistral).

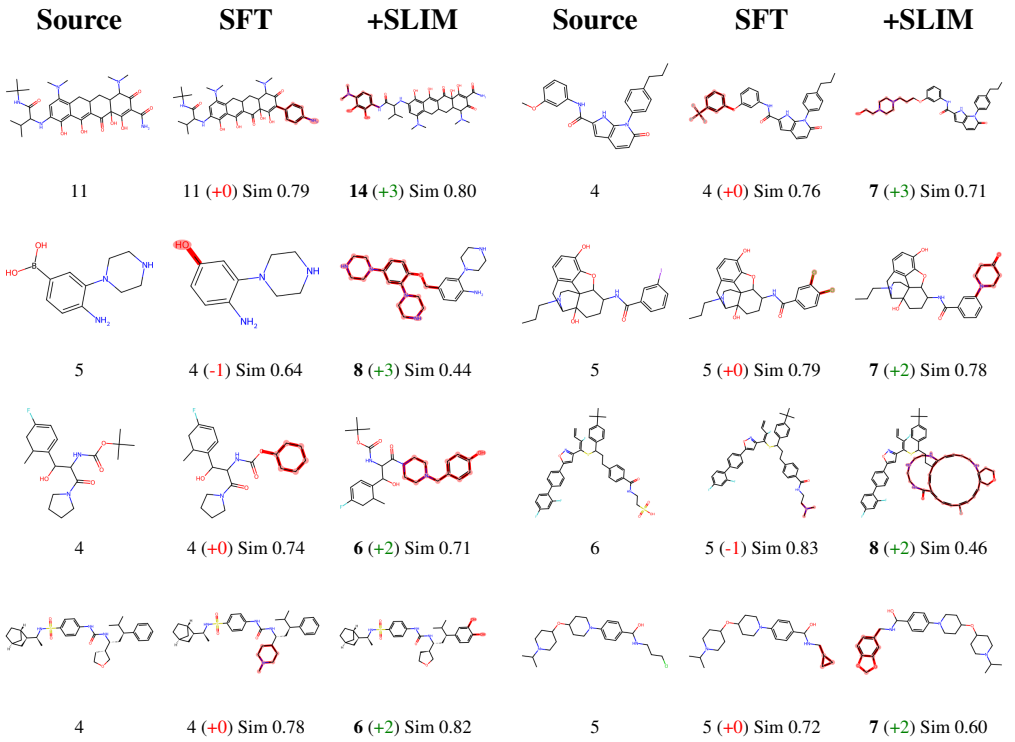


Figure 17: SLIM steering examples: HBA↑ (GeLLM3O-Mistral).

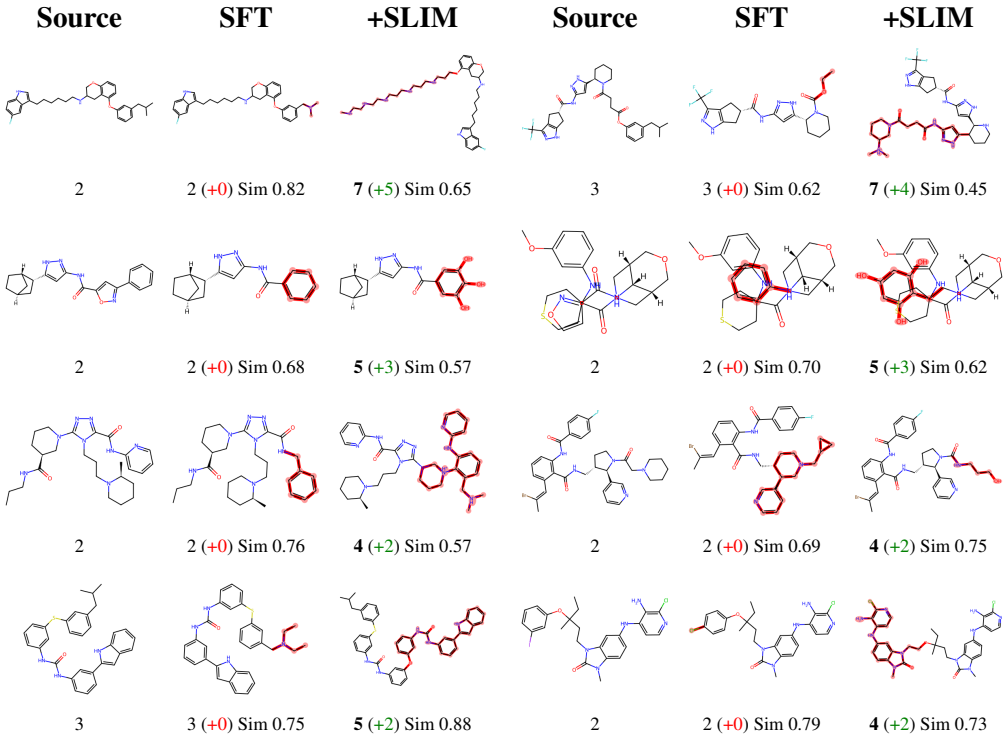


Figure 18: SLIM steering examples: HBD↑ (GeLLM3O-Mistral).

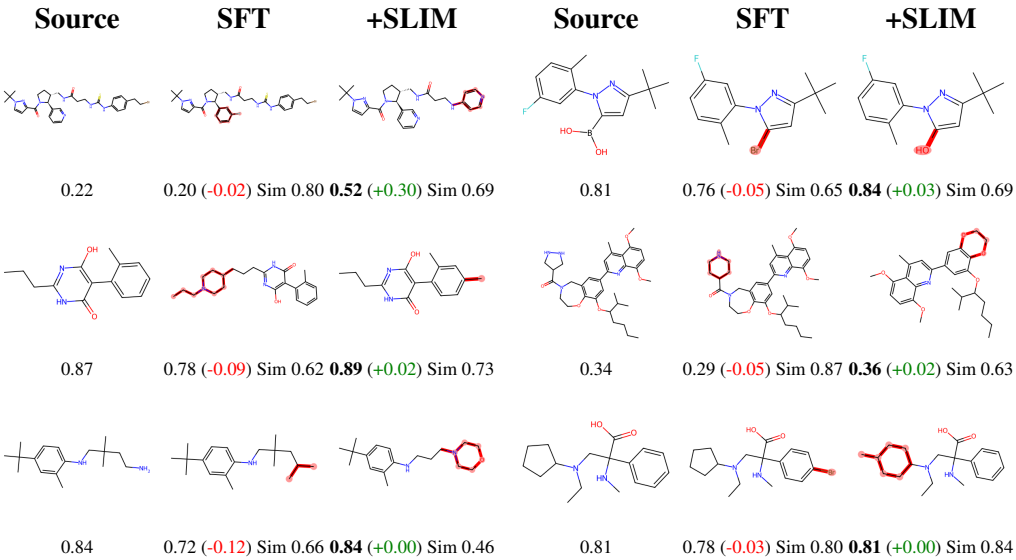


Figure 19: SLIM steering examples: QED↑ (GeLLM3O-LLaMA3).

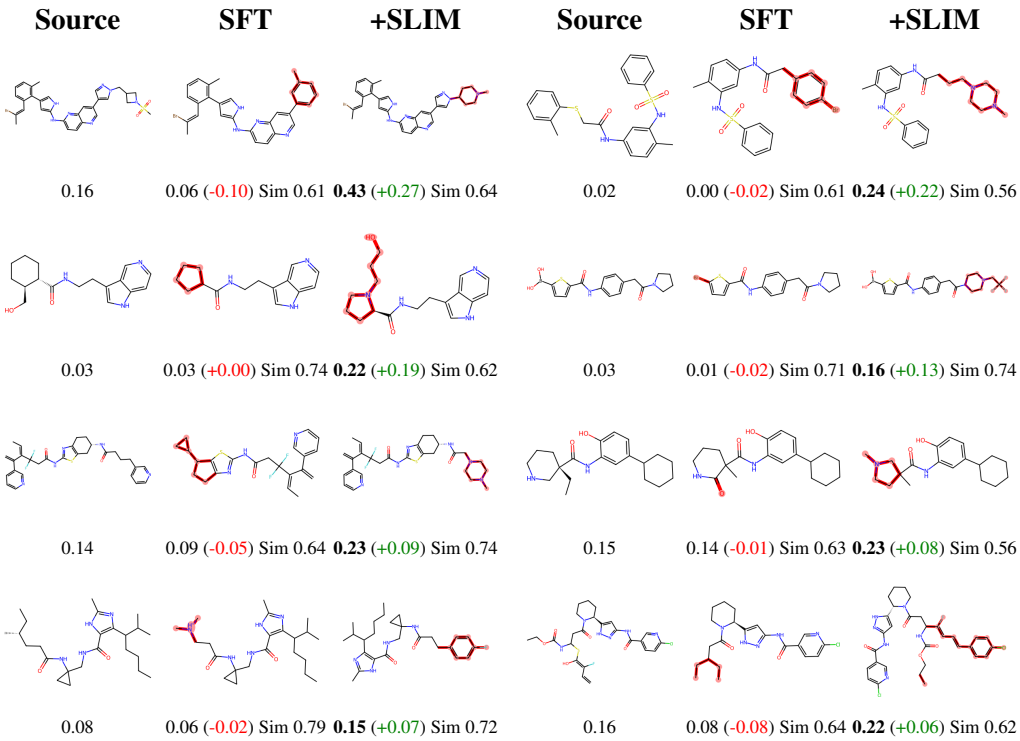


Figure 20: SLIM steering examples: DRD2↑ (GeLLM3O-LLaMA3).

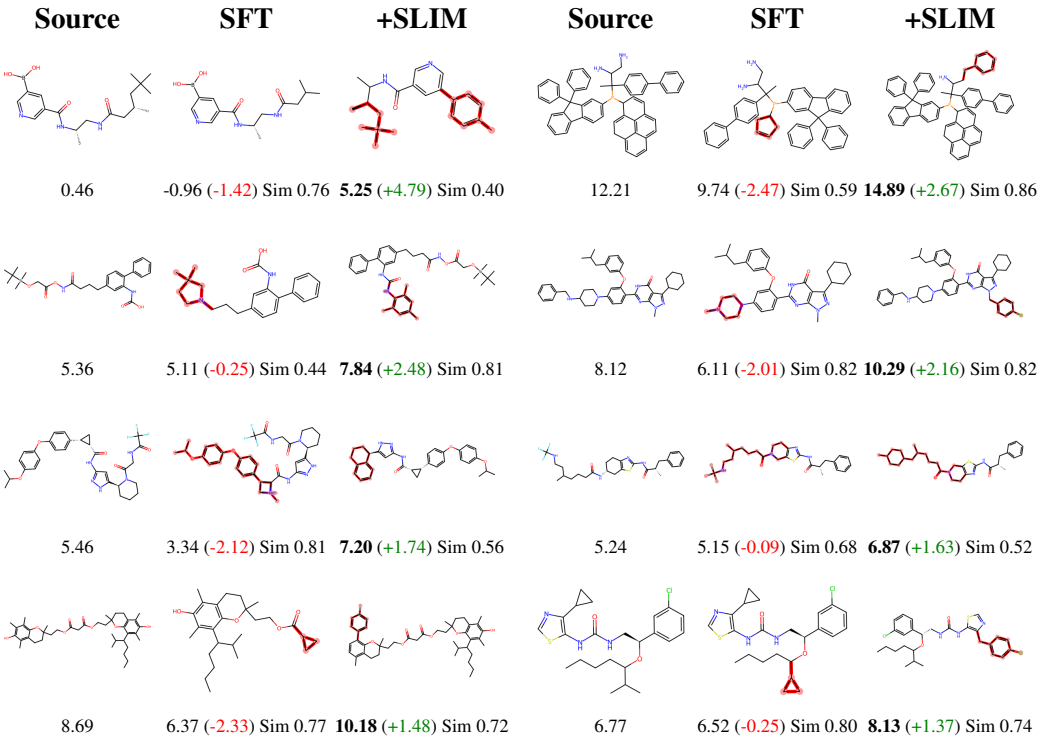


Figure 21: SLIM steering examples: logP↑ (GeLLM3O-LLaMA3).

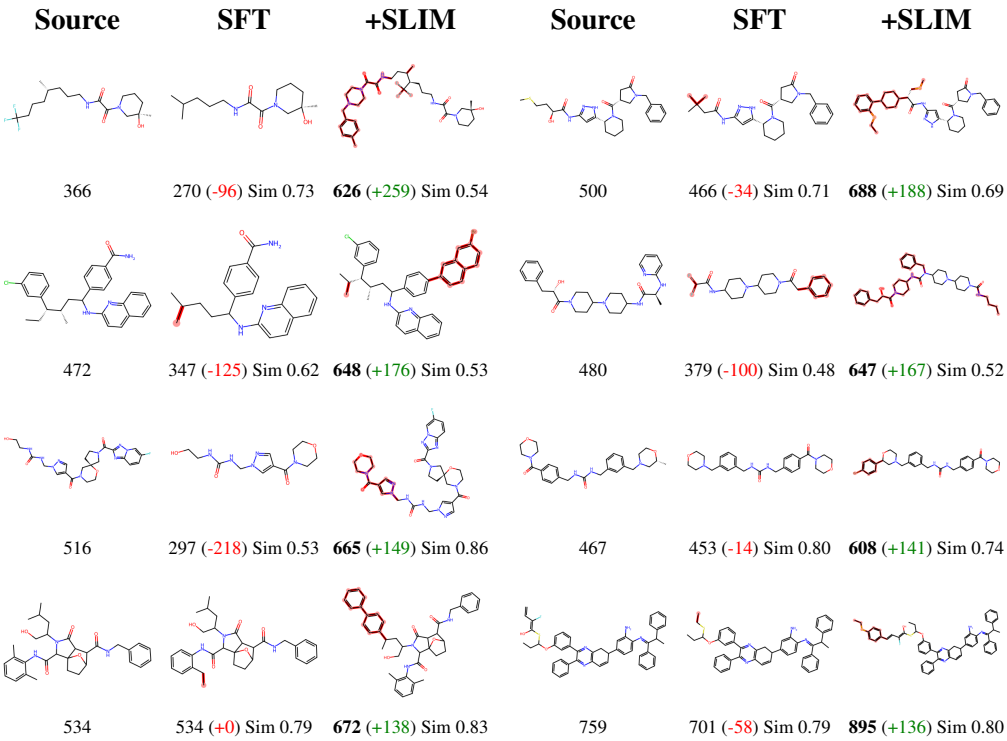


Figure 22: SLIM steering examples: MW↑ (GeLLM3O-LLaMA3).

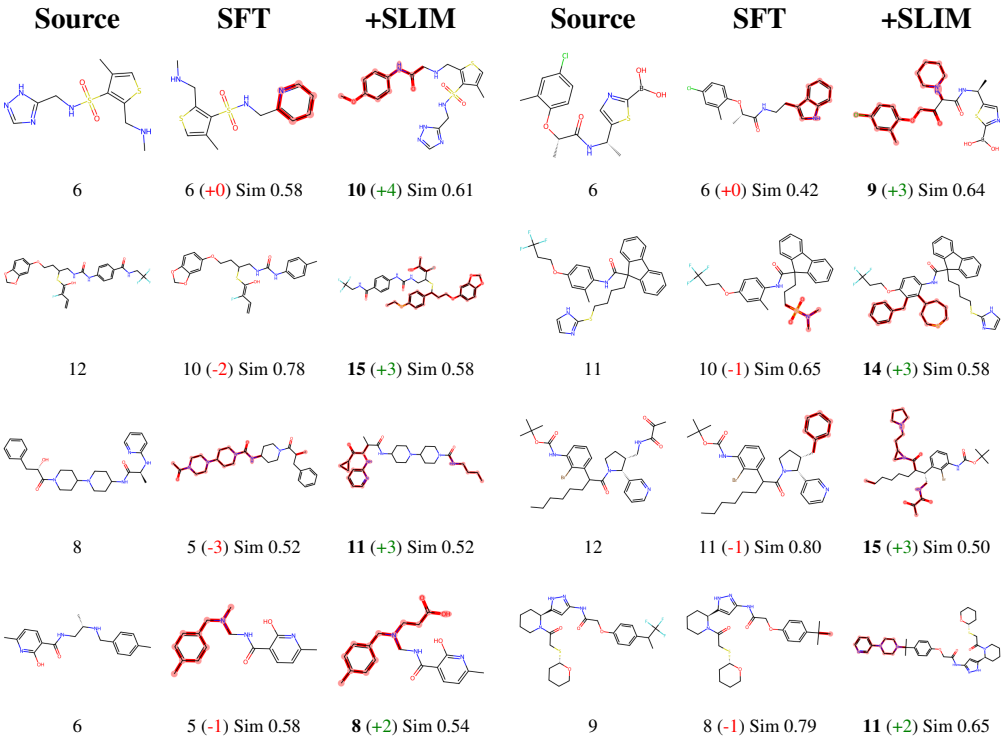


Figure 23: SLIM steering examples: RotBond↑ (GeLLM3O-LLaMA3).

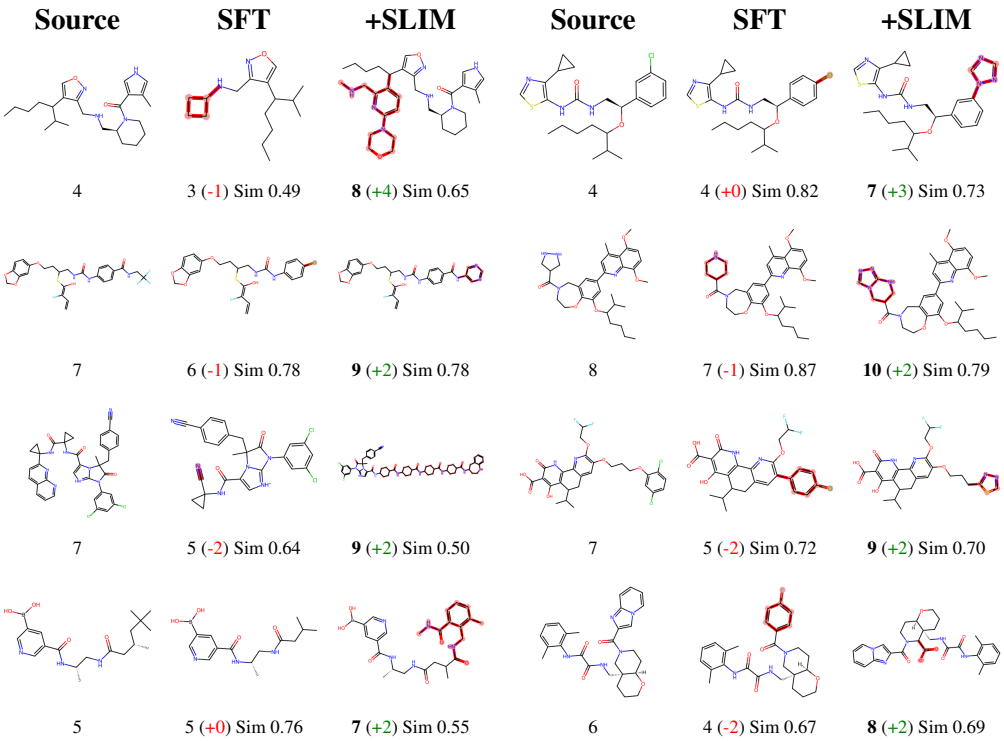


Figure 24: SLIM steering examples: HBA↑ (GeLLM3O-LLaMA3).

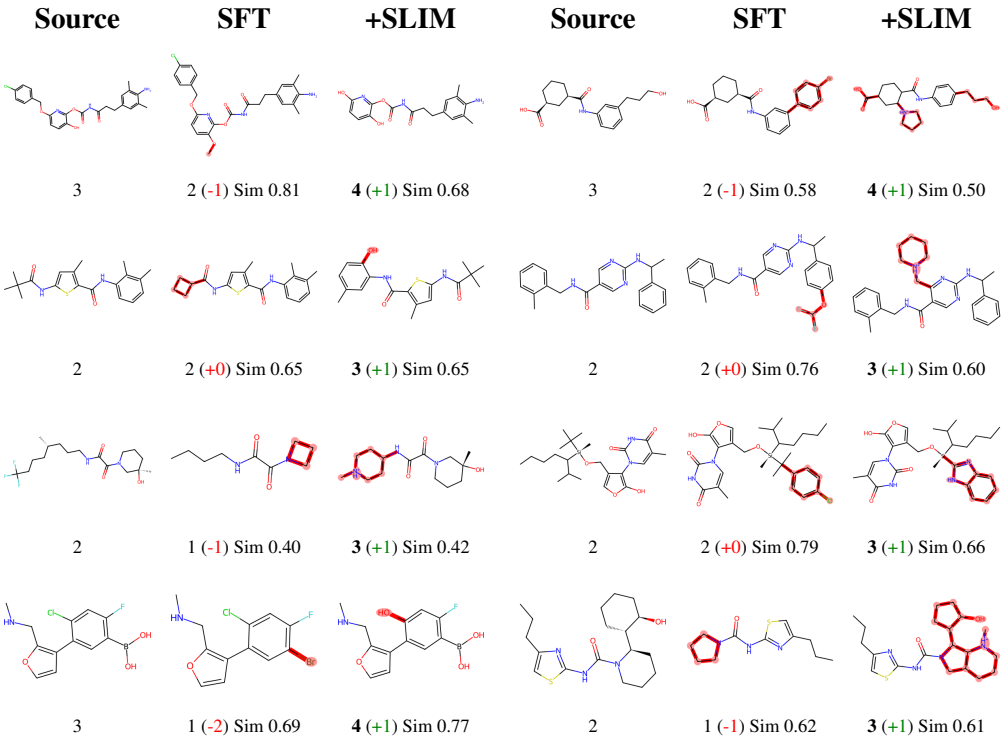


Figure 25: SLIM steering examples: HBD↑ (GeLLM3O-LLaMA3).

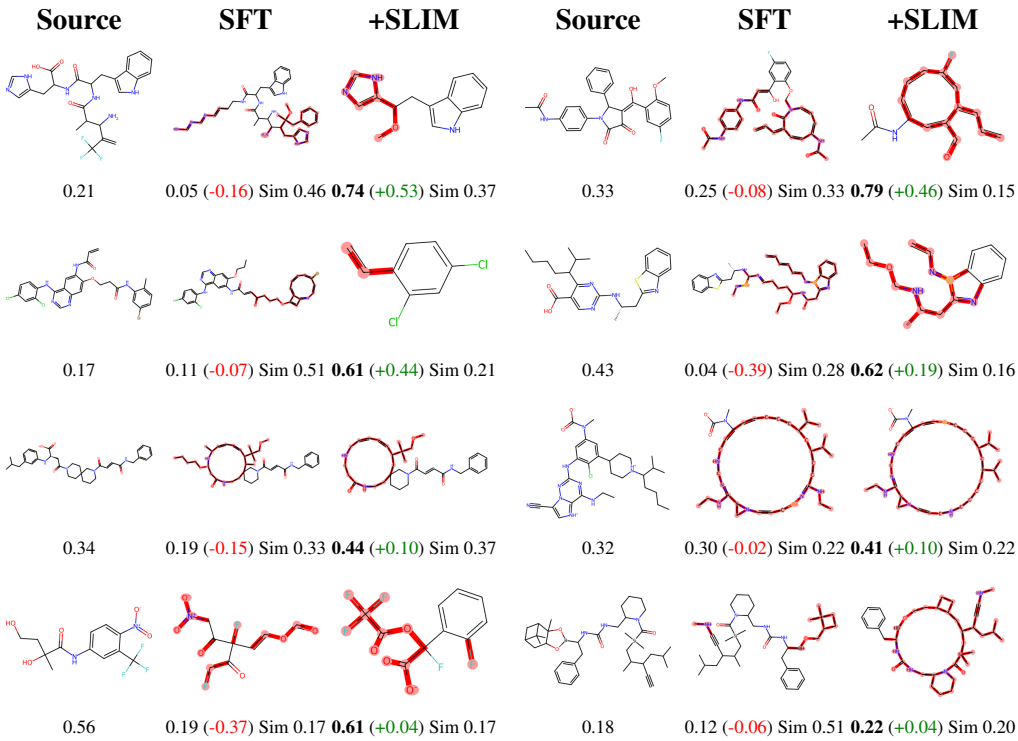


Figure 26: SLIM steering examples: QED↑ (MolGen-Large).

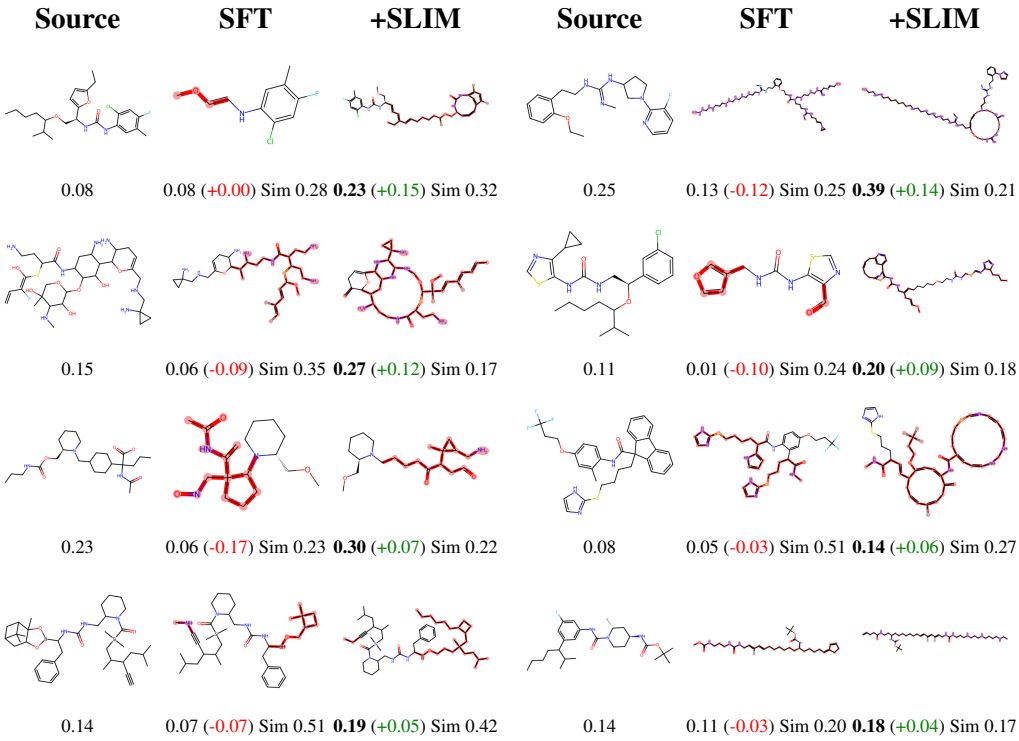


Figure 27: SLIM steering examples: DRD2↑ (MolGen-Large).

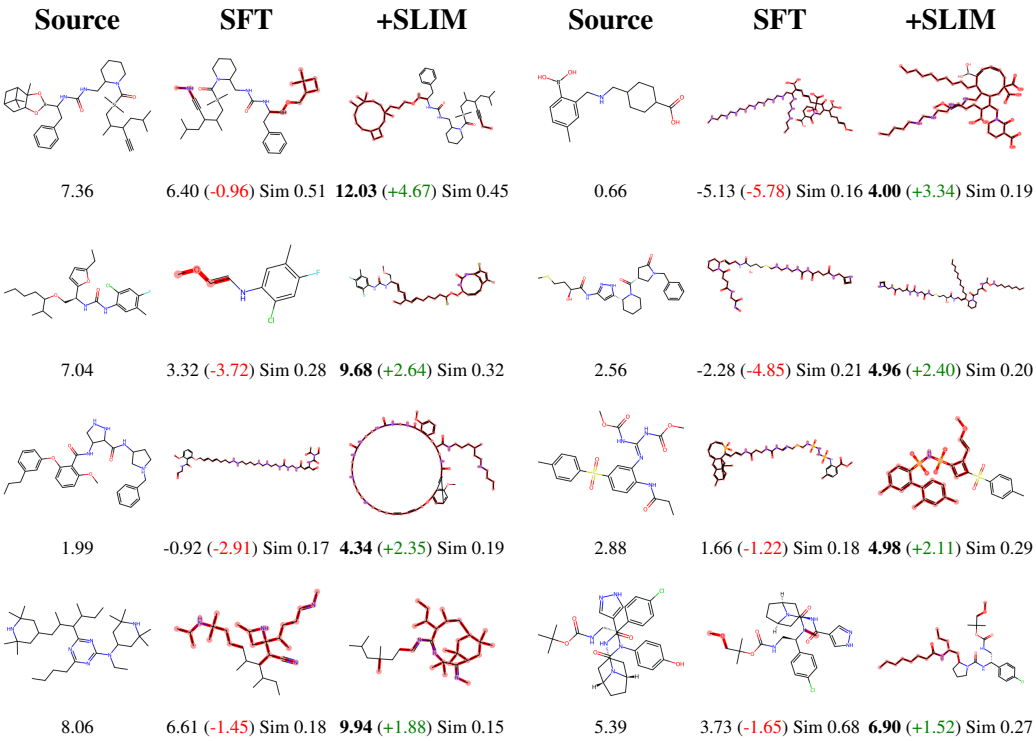


Figure 28: SLIM steering examples: logP↑ (MolGen-Large).

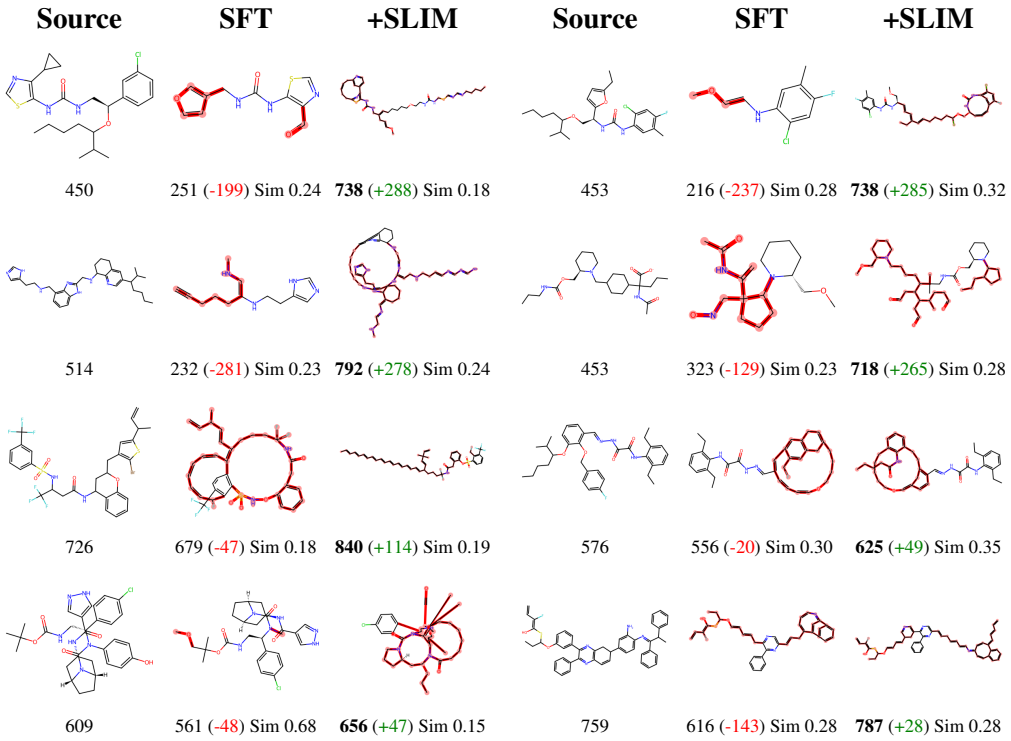


Figure 29: SLIM steering examples: MW↑ (MolGen-Large).

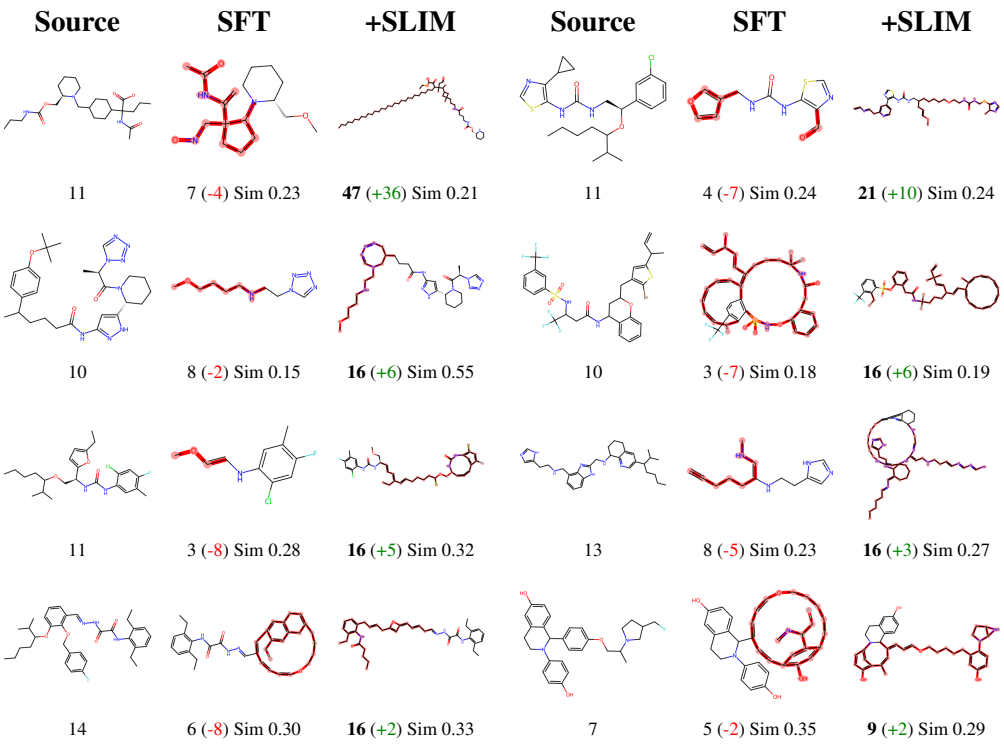


Figure 30: SLIM steering examples: RotBond↑ (MolGen-Large).

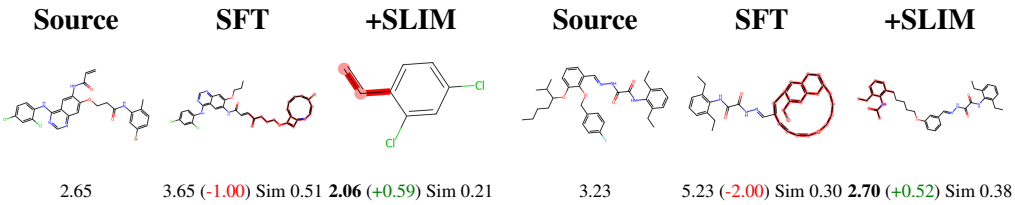


Figure 31: SLIM steering examples: SA↓ (MolGen-Large).

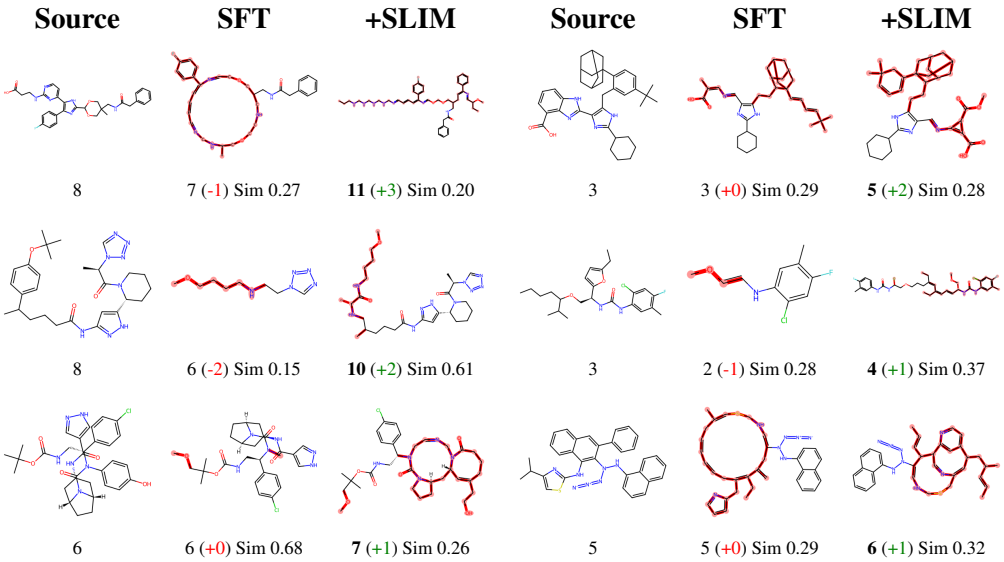


Figure 32: SLIM steering examples: HBA↑ (MolGen-Large).

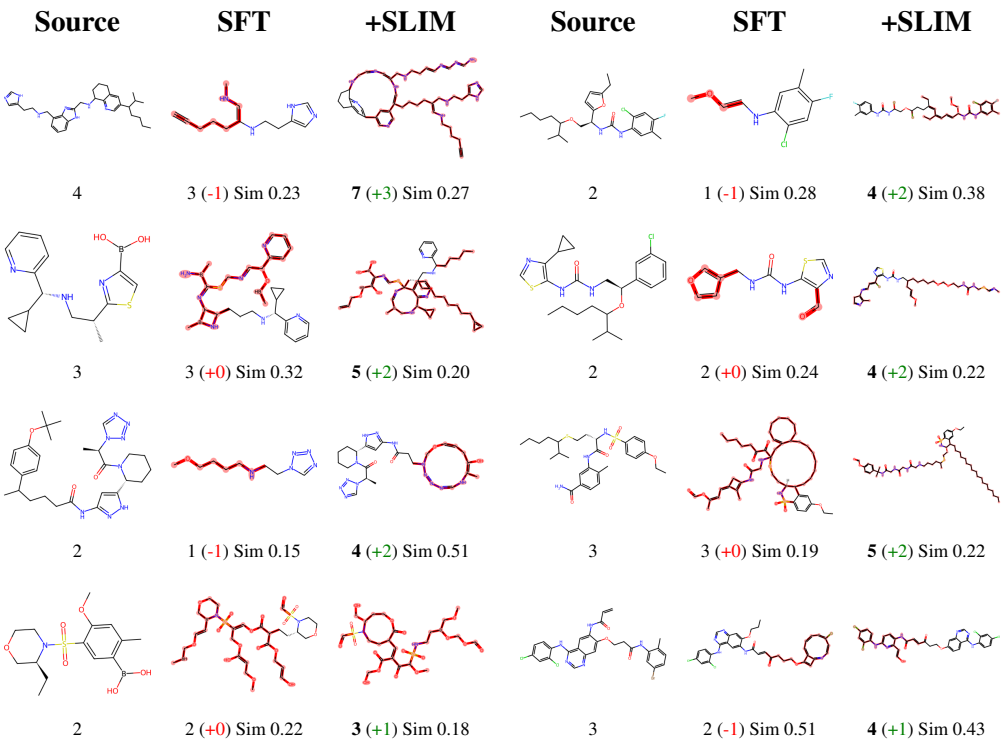


Figure 33: SLIM steering examples: HBD↑ (MolGen-Large).