ELITE-WEIGHTED SUPERVISED FINE-TUNING FOR GOAL-DIRECTED MOLECULAR OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Goal-directed optimization is essential for steering molecular generators to propose candidates with desired properties. However, it is often implemented with policy-gradient reinforcement learning, which requires a generation-trajectory log-probability whose form depends on the model architecture and generation procedure. This makes an optimizer difficult to reuse across architectures and conditional generative designs. Supervised fine-tuning needs none of that machinery, but its update is driven by a fixed dataset, so the reward never enters the update. We introduce Elite-Weighted Supervised Fine-tuning (EW-SFT), which uses reward to guide elite selection of high-scoring molecules, and updates the model by its own pretraining loss on that set. Because the update consumes only scored molecules and the model's native loss, EW-SFT is a general-purpose formulation for goal-directed molecular optimization. We evaluate EW-SFT applied to three leading molecular generators (SAFE-GPT, an autoregressive model; GenMol, a masked diffusion model; and InVirtuoGen, a discrete flow model) across de novo, motif-extension, and linker-design tasks. Under a fixed budget of 3D shape alignment oracle calls on two kinase reference compounds, we demonstrate EW-SFT consistently outperforms the corresponding native optimizers. Furthermore, it improves goal-directed optimization under a 2D similarity oracle on four held-out references and achieves comparable performance on a sample-efficiency benchmark without a trajectory-level RL formulation. Ablations show that reward information is passed primarily through elite selection, rather than through continuous weighting within the selected set. These results demonstrate that EW-SFT is a unified and effective optimizer across molecular generators, design constraints, references, and oracles.

1 INTRODUCTION

Generative models can propose novel small molecules directly with desired structures and properties rather than screening predefined libraries. Recent general-purpose molecular generators (Noutahi et al., 2024; Lee et al., 2025; Kaech et al., 2026) support both de novo generation (Brown et al., 2019), where a complete molecule is built from scratch, and fragment-constrained generation (Fialková et al., 2021; Tan et al., 2022; Guo et al., 2023; Rossen et al., 2025), where a fixed substructure is retained and only the remainder is designed. Optimizing such a generator toward a design objective is commonly done with policy-gradient reinforcement learning (RL) (Guo & Schwaller, 2024), which requires the log-probability assigned to the generation trajectory of each sampled molecule. Here, an oracle evaluates a completed molecule and returns a scalar reward measuring how well it satisfies the design objective. Policy-gradient methods use this reward, or an advantage derived from it, to weight the gradient of the generation trajectory's log-probability. This requires a policy gradient objective to be specified for each new architecture and generation procedure, resulting in several challenges. For some two-pass generation procedures, the released implementation does not provide a joint policy over the complete generation trajectory, rendering policy-gradient RL impossible. Meanwhile, keeping the policy-gradient update well-behaved requires machinery whose settings have to be tuned (Engstrom et al., 2020; Andrychowicz et al., 2020). An alternative strategy for goal-directed optimization is supervised fine-tuning, which updates the model through the same likelihood-based pretraining objective and needs none of this machinery. However, plain supervised

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

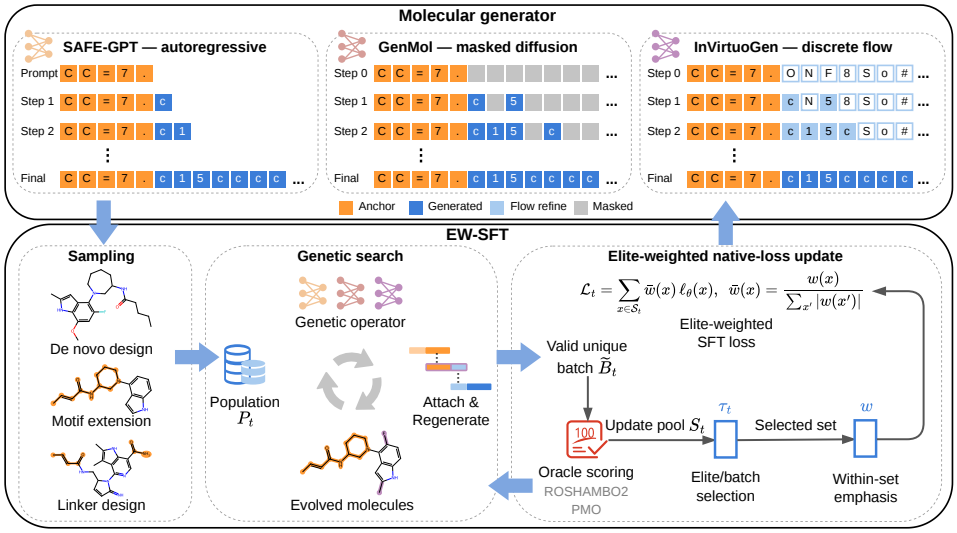


Figure 1: **The workflow of EW-SFT.** Three representative molecular generators first sample under task constraints to initialize a genetic population. Each model then acts as a genetic operator to regenerate new molecules, calling oracle to score and updating the population. Oracle scores determine which genetically evolved molecules are selected for fine-tuning, and the chosen emphasis sets their weights in the native-loss update. Full workflow is run until oracle budget is exhausted.

fine-tuning on a fixed dataset does not, by itself, specify how oracle feedback should change the training examples or their weights during optimization.

To fill this gap, we introduce Elite-Weighted Supervised Fine-tuning (EW-SFT), which lets the reward enter a supervised update through the composition of its training set. Each generator's genetic operator proposes molecules under its own unconditional or fragment-conditioned interface, a rolling *elite* buffer retains the highest-scoring molecules discovered so far, and the model takes one weighted fine-tuning step on the selected set using its native pretraining loss (Figure 1). Prior molecular hill-climbing and GEGL methods also learn from selected high-scoring molecules (Neil et al., 2018; Ahn et al., 2020), but instantiate a specific design choice: elite selection followed by exact-likelihood refitting for de novo generation. EW-SFT places this idea in a broader reward-guided native-loss formulation that separates three decisions: where training molecules come from, which molecules are selected, and how selected molecules are weighted. Across generalist generators with distinct training objectives and constrained generation procedures, we find that once elite selection is applied, continuous *emphasis* can be reduced to a binary indicator. Because EW-SFT operates directly on completed, scored molecules, it requires neither a generation-trajectory log-probability nor related importance ratios, critics, or ratio clipping. It applies across autoregressive, masked diffusion, and discrete flow generators, and also enables optimization for SAFE-GPT two-pass linker generation without defining a joint policy over the two sampling passes.

We evaluate EW-SFT in ligand-based molecular design by optimizing 3D similarity to known reference compounds. Under a fixed oracle budget on two kinase references, EW-SFT improves Top-1k similarity over the corresponding native optimizers in all nine architecture-task combinations. It further improves optimization under a 2D similarity oracle on a fragment-constrained benchmark (Noutahi et al., 2024) previously used only for frozen sampling. Finally, on the de novo Practical Molecular Optimization (PMO) benchmark (Gao et al., 2022), EW-SFT achieves performance comparable to a tuned policy-gradient method that carries more RL machinery. Our contributions are summarized as follows:

- We propose EW-SFT, a general goal-directed update rule that applies across architectures and tasks using only scored molecules and each model's native pretraining loss.
- We show that elite selection, rather than continuous within-set weighting, is the effective channel through which reward reaches the update.

- We show that EW-SFT transfers to fragment-constrained design, where policy-gradient RL is difficult, and report the first goal-directed optimization results for generalist molecular generators under motif and linker constraints.

2 PRELIMINARIES

Molecular generation and optimization. Molecular generative models were traditionally built to perform one task at a time. Simplified Molecular Input Line Entry System (SMILES)-based generators (Weininger, 1988) are often trained for de novo design (Olivecrona et al., 2017; Loeffler et al., 2024), and separate architectures were introduced for scaffold decoration (Fialková et al., 2021) and linker design (Guo et al., 2023; Tan et al., 2022). Sequential Attachment-based Fragment Embedding (SAFE) (Noutahi et al., 2024) represents a molecule as a sequence of attachable fragments, enabling one pretrained generator to support multiple design tasks. Generalist generators built on this representation differ in architecture: SAFE-GPT (Noutahi et al., 2024) is autoregressive, GenMol (Lee et al., 2025) is a masked diffusion model, and InVirtuoGen (Kaech et al., 2026) is a discrete flow model. Their native pretraining objectives differ accordingly: SAFE-GPT uses an exact autoregressive negative log-likelihood, whereas GenMol and InVirtuoGen use likelihood-bound objectives. These models are evaluated by frozen sampling (Noutahi et al., 2024) or by goal-directed optimization under an oracle-call budget (Brown et al., 2019; Gao et al., 2022). Existing optimizers are paired with particular generator architectures and generation procedures and are not directly interchangeable across them.

Policy-gradient RL. Recent molecular optimizers commonly use trajectory-level policy gradients (Loeffler et al., 2024; Guo & Schwaller, 2024; Kaech et al., 2026; Zhu et al., 2026), which scale the gradient of the log-probability assigned to a generation trajectory by its reward or advantage (Williams, 1992). Proximal Policy Optimization (PPO)-style variants additionally use an importance ratio between the current and behavior policies to reuse sampled trajectories (Schulman et al., 2017). For an autoregressive generator, the generation trajectory factorizes over token decisions. For iterative generators (Kaech et al., 2026; Zhu et al., 2026), the trajectory and its probability depend on the architecture. Generation procedures can add dependence: SAFE-GPT linker design samples the two anchor-constrained sequences in separate passes and joins them, so policy-gradient RL requires a joint policy over the complete two-pass generation trajectory. Even when a policy-gradient objective can be formed, keeping its update well-behaved depends heavily on auxiliary design choices. Choices such as value-function clipping and reward scaling can affect final reward more than the training algorithm choice, and their best settings are environment-dependent (Engstrom et al., 2020; Andrychowicz et al., 2020). In molecular optimization, systematic studies of REINFORCE-based methods found that common prior-likelihood regularization need not improve optimization (Thomas et al., 2025). Prior anchors, KL penalties, diversity filters, and replay can improve particular settings but introduce many hyperparameters. Their optima shift across objectives, tuning consumes a finite oracle budget (Gao et al., 2022), and the resulting updates can still collapse or require extensive tuning (Thomas et al., 2022; Guo & Schwaller, 2024).

Supervised fine-tuning. Supervised fine-tuning instead updates a generator on completed molecules under its own native pretraining loss. It therefore does not require a trajectory log-probability or a policy-gradient formulation (Segler et al., 2017; Gupta et al., 2018). But for a given dataset, the update has no oracle feedback. A higher-scoring molecule and a lower-scoring one contribute identically, and where the data comes from is a decision made outside the update (Neil et al., 2018).

Reward-guided supervised fine-tuning. These observations motivate a reward-guided native-loss update. After candidate molecules have been proposed and scored, the shared fine-tuning update should depend only on completed molecules, their oracle scores, and each generator's native pretraining loss, rather than on the generation trajectory that produced them. A weighted native-loss update provides such an interface. Three design decisions then remain — where the training molecules come from, which of them are selected, and how strongly each selected molecule counts. Our ablations test these three roles separately and show that genetic search supplies proposals beyond unguided sampling, elite selection carries the useful reward signal, and continuous within-buffer weighting is unnecessary.

3 METHOD

EW-SFT instantiates the weighted native-loss update identified in Section 2. It adapts to each generator through its native genetic operator G and pretraining loss ℓ_θ while applying the same reward-guided selection and weighting formulation. Algorithm 1 gives the GenMol instantiation with elite selection, the other two backbones differ only in G and ℓ_θ (Appendix B).

3.1 PROBLEM SETTING

Given a pretrained generative model p_θ conditioned on structural constraint C and a budget of N oracle calls to f, our goal is to generate the highest scoring compounds under f within this evaluation budget. Following the taxonomy of Noutahi et al. (2024), we use three generative design tasks: de novo design ($C = \varnothing$), motif extension (C is one fragment that must be retained), and linker design (C is two fragments that must both be retained and joined). We use $\mathcal{M}_C$ to denote the feasible set of molecules that qualify under C. Frozen sampling, which draws N samples from the frozen p_θ and scores them with θ unchanged, is what current generators support natively under constraints.

3.2 ELITE-WEIGHTED SUPERVISED FINE-TUNING

We apply EW-SFT to three pretrained molecular generators with distinct generation mechanisms and likelihood-based native pretraining losses ℓ_θ: SAFE-GPT (Noutahi et al., 2024) (autoregressive, next-token cross-entropy), GenMol (Lee et al., 2025) (masked diffusion, Negative Evidence Lower Bound (NELBO) loss (Sahoo et al., 2024)), and InVirtuoGen (Kaech et al., 2026) (discrete flow, time-weighted flow-matching NELBO). EW-SFT replaces their corresponding native optimizers—PPO (Schulman et al., 2017), a genetic algorithm (GA) (Lee et al., 2025), and Genetic-PPO, respectively— with its reward-guided weighted native-loss update.

Algorithm 1 EW-SFT (GenMol).

Require: pretrained p_θ; constraint C; oracle f; budget N; buffer size K; population size $|P|$

$\tilde{B}_0 \sim p_\theta(\cdot \mid C)$; $f(\tilde{B}_0)$; $P_0 \leftarrow \mathrm{Cut}(\tilde{B}_0)$; $E_0 \leftarrow \text{Top-}K(\tilde{B}_0)$; $n \leftarrow |\tilde{B}_0|$ ▷ Cold start

for $t = 1, 2, \ldots$ **while** $n < N$ **do**

  $\tilde{B}_t \leftarrow G(P_{t-1})$ ▷ Attach fragment pairs, re-mask one slot, keep valid and unique

  $f(\tilde{B}_t)$; $n \leftarrow n + |\tilde{B}_t|$

  $P_t \leftarrow \text{Top-}|P|\big(P_{t-1} \cup \mathrm{Cut}(\tilde{B}_t)\big)$; $\mathcal{S}_t \leftarrow \tilde{B}_t \cup E_{t-1}$; $E_t \leftarrow \text{Top-}K(\mathcal{S}_t)$ ▷ Update

  if $E_t \neq E_{t-1}$ **then**

    $\theta_t \leftarrow \theta_{t-1} - \eta \nabla_\theta \sum_{x \in \mathcal{S}_t} \bar{w}(x)\, \ell_\theta(x)$ ▷ Eq. 1 and 2

  end if

end for

return $\bigcup_{t \geq 0} \tilde{B}_t$ ▷ Scored molecules

Genetic search. Each pretrained generator acts as an architecture-specific genetic operator G. It first samples a batch $B_0 \subset \mathcal{M}_C$ to initialize the population P_0. At each subsequent round, G regenerates the batch B_t from the population P_{t-1}. We adapt each model's native de novo genetic mechanism to the fragment-constrained tasks. The population stores high-scoring fragments or molecules, and constrained tokens are never mutated. For each generator, we retain its native mechanism for controlling generated molecule size. The implementation details are in Appendix B. Only the valid and unique molecules in batch $\tilde{B}_t \subset B_t$ are scored, consuming $|\tilde{B}_t|$ of the budget. Molecules not valid for the oracle calculation still consume a call but are excluded from training.

Elite-weighted native-loss update. At round t, let $\mathcal{S}_t$ denote the scored update pool. EW-SFT updates the generator by one gradient step of weighted native-loss fine-tuning over this pool,

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \sum_{x \in \mathcal{S}_t} \bar{w}(x)\, \ell_\theta(x), \qquad \bar{w}(x) = \frac{w(x)}{\sum_{x' \in \mathcal{S}_t} |w(x')|}, \tag{1}$$

where ℓ_θ is the model's own pretraining loss and η is the learning rate. Because every $x \in \mathcal{S}_t$ has already met the structural constraint, the update evaluates each model's native pretraining loss on the

completed molecule x itself. The oracle reaches the update through the weight w, which we write as an elite indicator gating a binary or continuous emphasis:

$$w(x) \;=\; \mathbf{1}[\, f(x) \geq \tau_t \,] \cdot \begin{cases} 1 & \text{binary,} \\ \hat{A}(x) & \text{signed,} \\ \max\big(\hat{A}(x),\, 0\big) & \text{positive,} \end{cases} \qquad \hat{A}(x) = \frac{f(x) - \mu_t}{\sigma_t + \epsilon}, \tag{2}$$

Here, μ_t and σ_t are the mean and standard deviation of f over the selected molecules, and ϵ is a small constant. The selection scheme determines both the update pool $\mathcal{S}_t$ and the selected subset within it. Under batch selection, $\mathcal{S}_t = \tilde{B}_t$ and $\tau_t = -\infty$, so all valid molecules in the current round are selected. Under elite selection, EW-SFT maintains a rolling *elite buffer* E_t containing the K highest-scoring molecules seen through round t, initialized as $E_0 = \text{Top-}K(\tilde{B}_0)$. It sets $\mathcal{S}_t = \tilde{B}_t \cup E_{t-1}$ and $E_t = \text{Top-}K(\mathcal{S}_t)$. The corresponding selection threshold is $\tau_t = \min_{x \in E_t} f(x)$, which selects the elite buffer. τ_t rises monotonically as better molecules are found, and each sample competes against the run's whole history rather than against its own round.

Because the three generators maintain genetic populations at different granularity, we specify E_t by function rather than by implementation, which is a set of K whole molecules, ranked by their own oracle score and refreshed across rounds. GenMol and SAFE-GPT evolve populations of fragments, scored by the mean over the molecules a fragment was cut from, so no complete winner survives a round and E_t must be maintained separately. InVirtuoGen's genetic population instead holds whole molecules ranked by their own f, which already satisfies this definition, and therefore we treat it as that model's elite buffer without adding the second one (Table 12).

Equations 1 and 2 define different update rules through the choice of selection threshold and weighting scheme, which we compare in Section 4.2. We use the elite threshold with binary weighting, denoted elite+binary. In this setting, $\bar{w}(x) = 1/|E_t|$ for molecules in E_t and zero otherwise, so Equation 1 reduces exactly to the mean pretraining loss over E_t, corresponding to one step of standard fine-tuning on the best K molecules found so far. This simple rule is sufficient because the oracle scores determine which molecules are retained in E_t, and the gradient update only learns from those selected molecules. Positive or signed weighting instead assigns different importance to molecules within the elite set. A signed emphasis potentially makes the objective unbounded below, since a negative weight on an unbounded-above loss may let gradient descent reduce the loss by pushing away disfavored molecules rather than toward good ones (Proposition 1 and Figure 7). Since the model is updated only when $E_t \neq E_{t-1}$, a stalled search does not repeatedly fit the same buffer, eliminating the need for an explicit early-stopping rule (Appendix B).

4 Experiments

4.1 3D Shape Similarity-Directed Optimization

Setup. We optimize the 3D shape similarity of generated molecules to a known reference compound under each structural constraint, using ROSHAMBO2 (Atwi et al., 2025), an open-source 3D similarity tool, as the oracle with a budget of 20k calls. We use BMS-986195, a Bruton's tyrosine kinase (BTK) inhibitor (Watterson et al., 2019), and Zasocitinib (TAK-279), a tyrosine kinase 2 (TYK2) inhibitor (Leit et al., 2023), as reference compounds. Table 6 displays the fragment anchors for both. Within each backbone, EW-SFT replaces the native optimizer with all other settings held fixed. Because the goal is to find a high-scoring, novel and diverse pool of molecules, our primary metric is the mean oracle value of the best 1k molecules. Every cell is measured against frozen sampling, drawing the same 20k molecules over 3 runs from the frozen model and scoring them unchanged, which is what current generators support natively under fragment constraints. Because no prior work has optimized these generalist generators under such constraints, we additionally adapt their own de novo optimizers to the constrained interface for motif and linker tasks. REINVENT-family models[1] are per-task references only, since their convergence needs substantially more oracle calls (Wa et al., 2026). Frozen sample quality is characterized in Appendix C.3, and detailed oracle settings and hyperparameters in Appendix B.

[1]REINVENT (Olivecrona et al., 2017; Loeffler et al., 2024) for de novo design, LibINVENT (Fialková et al., 2021) for motif extension, and Link-INVENT (Guo et al., 2023) for linker design

Results. The native optimizers improve Top-1k average 3D shape similarity over frozen sampling, so goal-directed optimization is necessary even when the fixed fragments already supply some structural information (Figure 2 and Table 8). EW-SFT adds significantly beyond that working optimizer across all settings, with Holm-adjusted Wilcoxon signed rank test $p_{\text{SAFE-GPT}} = 9.8 \times 10^{-4}$, $p_{\text{GenMol}} = 4.1 \times 10^{-2}$ and $p_{\text{InVirtuoGen}} = 2.3 \times 10^{-5}$ (Appendix C.2, Top-1k average in Table 1 and AUC Top-10 in Table 9). The advantage is also retained under per-run matching to the early-stopped oracle call consumed by the paired native optimizer (Table 10). This is not merely a remedy for a weak baseline — it reaches $+0.079$ over the strongest InVirtuoGen's Genetic-PPO, against $+0.091$ over the weakest SAFE-GPT's PPO. Neither is the gain concentrated on the easiest task, being nearly flat at $+0.064$ de novo, $+0.051$ motif, and $+0.055$ linker, nor does it follow from one model starting further behind, since the three priors sit within 0.012 of each other on frozen Top-1k. The improvement is also not obtained by drifting toward the reference. EW-SFT is less 2D-similar to the reference than the native optimizer in 9 of the 16 cells while scoring higher in 3D in all 16 (Table 11). Since the objective constrains shape alone, moving away from the reference in 2D while matching it more closely in 3D is consistent with satisfying the shape requirement using chemically distinct molecules rather than simply reconstructing the reference.

Table 1: **Top-1k average scores of 3D shape similarity optimization. Bold** is the best value within each model block (REINVENT excluded). PPO is not evaluated for SAFE-GPT's linker design because its two-pass generation requires a joint policy over the complete generation trajectory.

		BTK			TYK2			
Model	Method	De novo	Motif	Linker	De novo	Motif	Linker	Avg.
REINVENT	REINFORCE	0.530±0.000	0.696±0.010	0.576±0.007	0.416±0.001	0.494±0.008	0.425±0.003	0.523
SAFE-GPT	PPO	0.528±0.000	0.572±0.012	—	0.408±0.001	0.459±0.009	—	0.492
	EW-SFT (ours)	**0.630**±0.023	**0.665**±0.022	**0.563**±0.009	**0.544**±0.025	**0.492**±0.043	**0.446**±0.014	**0.556**
GenMol	GA	0.622±0.008	0.657±0.007	0.624±0.048	0.493±0.009	0.487±0.014	0.521±0.004	0.567
	EW-SFT (ours)	**0.631**±0.019	**0.663**±0.003	**0.650**±0.013	**0.510**±0.040	**0.490**±0.009	**0.532**±0.007	**0.579**
InVirtuoGen	Genetic-PPO	0.723±0.081	0.651±0.009	0.656±0.008	0.565±0.012	0.507±0.021	0.506±0.041	0.602
	EW-SFT (ours)	**0.797**±0.034	**0.769**±0.017	**0.769**±0.048	**0.612**±0.030	**0.559**±0.004	**0.578**±0.020	**0.681**

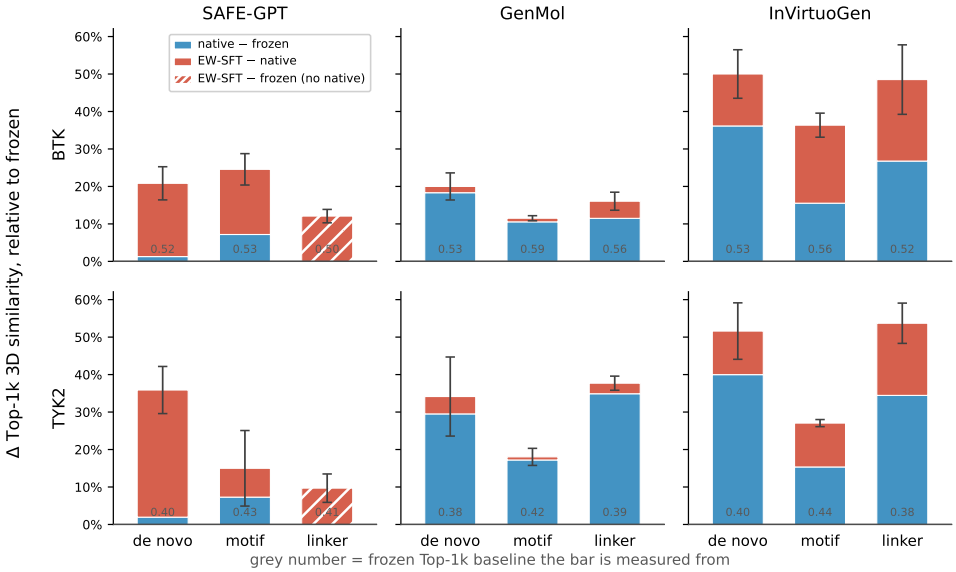


Figure 2: **Relative Top-1k improvements over frozen sampling.** Bars show improvement over frozen sampling from the native optimizer (blue) and the additional EW-SFT gain (red), normalized by the frozen baseline (grey). Hatched bars indicate SAFE-GPT linker tasks without a native PPO result. Absolute gains are in Table 8. EW-SFT provides an additional improvement over the native optimizer overall.

4.2 Ablation Study

Setup. The mechanism of EW-SFT consists of three axes. **Source** determines whether training samples come from the genetic operator G or directly from the pretrained model p_θ. **Selection** determines whether fine-tuning uses the current round's batch or the elite buffer E_t. **Emphasis** is layered on top of selection and determines how strongly each selected sample contributes. Crossing the last two gives the EW-SFT variants in Table 2. Two settings are left out — signed emphases on the elite buffer, which Proposition 1 rules ill-posed, and the batch-selection setting for InVirtuoGen, whose genetic population is already an elite buffer in the sense defined in Section 3.2. Table 12 in Appendix C.8 gives the resulting settings and the reasoning behind each exclusion.

Table 2: **Ablation results.** Top-1k mean±std 3D similarity over 3 runs at 20k oracle calls, for each model's native optimizer, the EW-SFT variants and the two standalone settings (GA only and SFT only). The best result in each task is marked in **bold**. SAFE-GPT PPO cannot run linker design. The last column is the mean difference from that model's native optimizer over its cells.

	Setting			BTK		TYK2		
Model	Source	Selection	Emphasis	De novo	Linker	De novo	Linker	$\overline{\Delta}_{\text{var-nat}}$
SAFE-GPT	Native (PPO)			0.528±0.000	—	0.408±0.001	—	—
	EW-SFT (ours)							
	G	elite	binary	**0.630**±0.023	0.563±0.009	**0.544**±0.025	0.446±0.014	+**0.119**
	G	elite	positive	0.593±0.013	0.563±0.011	0.507±0.031	0.455±0.014	+0.082
	G	batch	positive	0.620±0.029	0.559±0.005	0.491±0.011	**0.459**±0.027	+0.088
	G	batch	signed	0.616±0.023	**0.566**±0.002	0.496±0.027	0.458±0.011	+0.088
	Standalone							
	G	—	—	0.607±0.029	0.564±0.005	0.501±0.008	0.438±0.008	+0.086
	p_θ	elite	binary	0.524±0.001	0.553±0.005	0.399±0.001	0.421±0.002	−0.006
GenMol	Native (GA)			0.622±0.008	0.624±0.048	0.493±0.009	0.521±0.004	—
	EW-SFT (ours)							
	G	elite	binary	0.631±0.019	0.650±0.013	0.510±0.040	**0.532**±0.007	+0.016
	G	elite	positive	0.638±0.018	**0.664**±0.014	0.505±0.029	0.529±0.007	+**0.019**
	G	batch	positive	0.631±0.020	0.635±0.045	**0.513**±0.033	0.520±0.023	+0.010
	G	batch	signed	**0.644**±0.007	0.635±0.025	0.513±0.027	0.516±0.014	+0.012
	Standalone							
	G	—	—	0.622±0.008	0.624±0.048	0.493±0.009	0.521±0.004	+0.000
	p_θ	elite	binary	0.531±0.000	0.520±0.001	0.385±0.000	0.405±0.002	−0.105
InVirtuoGen	Native (Genetic-PPO)			0.723±0.081	0.656±0.008	0.565±0.012	0.506±0.041	—
	EW-SFT (ours)							
	G	elite	binary	**0.797**±0.034	**0.769**±0.048	**0.612**±0.030	**0.578**±0.020	+**0.076**
	G	elite	positive	0.778±0.023	0.725±0.008	0.597±0.021	0.547±0.039	+0.049
	G	elite	signed	0.710±0.039	0.661±0.048	0.543±0.048	0.550±0.037	+0.003
	Standalone							
	G	—	—	0.749±0.025	0.700±0.058	0.576±0.050	0.556±0.057	+0.032
	p_θ	elite	binary	0.532±0.001	0.565±0.003	0.409±0.002	0.411±0.003	−0.134

Source of training samples. In two standalone cases, capturing training samples from the genetic operator alone improves on the native optimizer, which aligns with the claim that genetic algorithm is already a strong baseline in this domain (Jensen, 2019; Tripp & Hernández-Lobato, 2023). Directly using the policy-generated samples fails to enhance the model. SFT-only holds the elite buffer and the binary emphasis fixed, it is therefore essentially distillation. In our setting, the generator sharpens around elites drawn from its own output but does not reliably introduce improvements beyond its prior. The genetic operator is what breaks that closed loop by mutating or refining the samples for EW-SFT to further update.

Reward guidance. Given that source, elite selection into a rolling buffer is sufficient to deliver the reward to the update. As a molecule enters the buffer only by outscoring the minimum there, the elites are the best seen so far and intrinsically carry the reward signal, so a continuous emphasis provides no consistent additional benefit. When fine-tuning uses the full current batch without elite filtering, a positive emphasis does improve the policy model and mildly surpasses the GA alone, because the emphasis is then the only channel through which the reward can act. Applying a positive emphasis on top of the elite buffer is redundant and weakens the update by only using the positive-weighted elites. The signed emphasis reduces 0.073 average improvement on InVirtuoGen against

the same source under a binary emphasis and has the diverging issue as shown in Proposition 1 (Appendix C.9). Overall, the two components are complementary: genetic search supplies improved candidates beyond direct sampling from the current generator, while elite native-loss fine-tuning transfers selected discoveries back into the model parameters. We report EW-SFT with the binary emphasis over the elite buffer based on the genetic operator throughout.

Quality and diversity tradeoff. Figure 3 exhibits the quality and diversity tradeoff of the Top-1k samples across each ablation setting. Higher-scoring settings tend to be less diverse (per-cell Pearson r averaging -0.52, negative in 11 of 12 cells, $p = 5 \times 10^{-4}$), but EW-SFT with elite+binary is the only setting that sits significantly above what that tradeoff predicts ($p = 4 \times 10^{-4}$), while the native optimizer is significantly below it. Specifically, Figure 8 shows that the moving-average batch diversity under EW-SFT falls early and then settles onto a plateau rather than continuing to decline, holding near 0.65–0.68 on SAFE-GPT and GenMol and 0.77–0.79 on InVirtuoGen. The batch maximum keeps improving over that same stretch on SAFE-GPT TYK2 and InVirtuoGen BTK de novo design, where the native optimizer has either flattened or stopped under its own novelty rule at roughly 8k calls (Figures 4–6). A search whose diversity levels off while its best molecules keep improving is consistent with continued exploration rather than complete collapse.

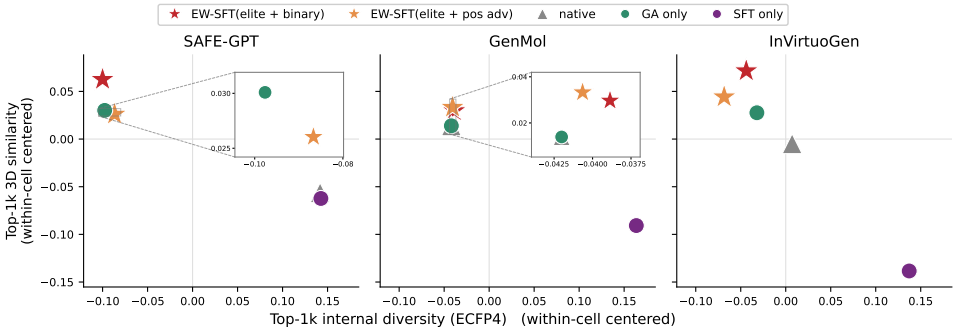


Figure 3: **EW-SFT gains score while staying diverse.** Each point is a model's one ablation setting, averaged over the task cells in which every setting ran (SAFE-GPT's linker cells are excluded). Both axes are centered within a model. GenMol's native optimizer is its GA, so those two markers coincide. EW-SFT improves over the native optimizer on score while staying close to the genetic algorithm on diversity.

4.3 Generalization to Held-out References and PMO

Fragment-constrained benchmark. We select four reference drugs (Table 6) from the fragment-constrained benchmark (Noutahi et al., 2024), and use 2D ECFP4 Tanimoto similarity (Rogers & Hahn, 2010) to the reference compound as the oracle. We first follow the common practice of fragment-constrained evaluation by frozen sampling with the best backbone, InVirtuoGen, under fragment anchor conditions (Noutahi et al., 2024; Lee et al., 2025; Kaech et al., 2026). To our knowledge, we are the first to adapt the native de novo optimizer to this benchmark's reference compounds and find that it substantially improves the average performance by $+0.135$ (adjusted $p_{\text{nat-frz}} = 2.4 \times 10^{-7}$) (Table 3). Our best EW-SFT arm (genetic source, elite selection, and binary emphasis) further improves $+0.086$ on top, winning 7 of 8 tasks (adjusted $p_{\text{ours-nat}} = 6.4 \times 10^{-5}$).

Table 3: **Top-1k average scores of 2D ECFP similarity on the held-out references.** The results are mean±std over 3 runs at 20k oracle calls. The best result for each task is marked in bold. The four drug compounds and their fragment anchors are in Table 6.

	Eliglustat		Erlotinib		Futibatinib		Spirapril		
Method	motif	linker	motif	linker	motif	linker	motif	linker	Avg.
Frozen (Top-1k)	0.372±0.001	0.433±0.002	0.263±0.001	0.331±0.001	0.331±0.000	0.425±0.001	0.483±0.001	0.522±0.002	0.395
Genetic-PPO	**0.611**±0.021	0.645±0.005	0.411±0.060	0.448±0.008	0.441±0.040	0.491±0.013	0.562±0.016	0.630±0.009	0.530
EW-SFT	0.588±0.066	**0.740**±0.091	**0.453**±0.085	**0.540**±0.016	**0.520**±0.057	**0.605**±0.072	**0.743**±0.033	**0.743**±0.056	**0.616**

Practical Molecular Optimization. PMO (Gao et al., 2022) evaluates sample-efficient de novo optimization under a 10k oracle-call budget. We further evaluate elite+binary EW-SFT with InVirtuoGen. Table 4 shows comparable aggregate performance to Genetic-PPO under both prescreening (18.863 vs. 18.992, adjusted $p = 0.69$) and cold-start (16.385 vs. 16.674, adjusted $p = 0.87$). The similar aggregate scores and nonsignificant paired differences indicate that comparable sample-efficient optimization can be achieved without a trajectory-level RL formulation. Genetic-PPO uses stored behavior-policy generation-trajectory log-probabilities, importance ratios, ratio clipping, negative advantage scaling, and experience replay. In contrast, EW-SFT uses a native-loss step on the elite set and requires no generation-trajectory log-probability. Per oracle, EW-SFT takes the best result on 10 of the 23 prescreening oracles against Genetic-PPO's 9, and on 8 against its 10 under cold-start.

Table 4: **PMO benchmark.** AUC Top-10 scores (mean±std over 3 runs) under the two initialization settings used by InVirtuoGen, with a 10k oracle-call budget per run. *With prescreening* initializes the population with the 40 highest-scoring ZINC250k molecules under the task oracle; the 250k × 23 prescreening evaluations are excluded from this budget. *Cold-start* initializes from the generator alone. **Bold** indicates the best mean within each setting.

	w/ Prescreening				Cold-Start (w/o Prescreening)				
Oracle	InVirtuoGen EW-SFT (ours)	InVirtuoGen Genetic-PPO	GenMol	f-RAG	InVirtuoGen EW-SFT (ours)	InVirtuoGen Genetic-PPO	Gen. GFN	Mol GA	REINVENT
albuterol_similarity	0.973±0.015	0.975±0.016	0.937±0.010	**0.977**±0.002	**0.966**±0.011	0.950±0.017	0.949±0.010	0.896±0.035	0.882±0.006
amlodipine_mpo	0.812±0.023	**0.836**±0.031	0.810±0.012	0.749±0.019	0.740±0.089	0.733±0.043	**0.761**±0.019	0.688±0.039	0.635±0.035
celecoxib_rediscovery	0.832±0.018	**0.839**±0.013	0.826±0.018	0.778±0.007	**0.819**±0.026	0.798±0.028	0.802±0.029	0.567±0.083	0.713±0.067
deco_hop	**0.981**±0.007	0.968±0.012	0.960±0.010	0.936±0.011	0.714±0.068	**0.748**±0.109	0.733±0.109	0.649±0.025	0.666±0.044
drd2	**0.995**±0.000	**0.995**±0.000	**0.995**±0.000	0.992±0.000	0.984±0.001	**0.985**±0.002	0.974±0.006	0.936±0.016	0.945±0.007
fexofenadine_mpo	0.897±0.035	**0.904**±0.000	0.894±0.028	0.856±0.016	**0.863**±0.031	0.845±0.016	0.856±0.039	0.825±0.019	0.784±0.006
gsk3b	**0.989**±0.002	0.988±0.001	0.986±0.003	0.969±0.003	0.914±0.045	**0.952**±0.016	0.881±0.042	0.843±0.039	0.865±0.043
isomers_c7h8n2o2	**0.991**±0.001	0.988±0.002	0.942±0.004	0.955±0.008	0.964±0.013	0.968±0.005	**0.969**±0.003	0.878±0.026	0.852±0.036
isomers_c9h10n2o2pf2cl	0.892±0.030	**0.898**±0.018	0.833±0.014	0.850±0.005	0.878±0.025	0.874±0.013	**0.897**±0.007	0.865±0.012	0.642±0.054
jnk3	0.900±0.044	0.898±0.031	**0.906**±0.023	0.904±0.004	**0.841**±0.021	0.825±0.016	0.764±0.069	0.702±0.123	0.783±0.023
median1	0.385±0.011	0.386±0.003	**0.398**±0.000	0.340±0.007	0.354±0.016	0.342±0.008	**0.379**±0.010	0.257±0.009	0.356±0.009
median2	0.376±0.008	**0.377**±0.006	0.359±0.004	0.323±0.005	**0.310**±0.013	0.288±0.008	0.294±0.007	0.301±0.021	0.276±0.008
mestranol_similarity	0.987±0.008	**0.991**±0.002	0.982±0.000	0.671±0.021	**0.830**±0.053	0.797±0.033	0.708±0.057	0.591±0.053	0.618±0.048
osimertinib_mpo	**0.882**±0.017	0.881±0.012	0.876±0.008	0.866±0.009	0.859±0.025	**0.870**±0.005	0.860±0.008	0.844±0.015	0.837±0.009
perindopril_mpo	**0.758**±0.012	0.753±0.019	0.718±0.012	0.681±0.017	0.619±0.035	**0.645**±0.032	0.595±0.014	0.547±0.022	0.537±0.016
qed	**0.944**±0.000	0.943±0.000	0.942±0.000	0.939±0.001	**0.942**±0.001	**0.942**±0.000	**0.942**±0.000	0.941±0.001	0.941±0.000
ranolazine_mpo	**0.858**±0.014	0.854±0.012	0.821±0.011	0.820±0.016	**0.852**±0.004	0.848±0.010	0.819±0.018	0.804±0.011	0.760±0.009
scaffold_hop	0.652±0.011	**0.711**±0.081	0.628±0.008	0.576±0.014	0.597±0.043	0.589±0.021	**0.615**±0.100	0.527±0.025	0.560±0.019
sitagliptin_mpo	**0.763**±0.036	0.743±0.022	0.584±0.034	0.601±0.011	0.624±0.055	**0.709**±0.029	0.634±0.039	0.582±0.040	0.021±0.003
thiothixene_rediscovery	0.629±0.011	0.652±0.024	**0.692**±0.123	0.584±0.009	0.621±0.048	**0.625**±0.014	0.583±0.034	0.519±0.041	0.534±0.013
troglitazone_rediscovery	0.850±0.015	0.853±0.003	**0.867**±0.022	0.448±0.017	0.554±0.052	**0.595**±0.053	0.511±0.054	0.427±0.031	0.441±0.032
valsartan_smarts	0.886±0.010	**0.935**±0.012	0.822±0.042	0.627±0.058	0.000±0.000	**0.210**±0.297	0.135±0.271	0.000±0.000	0.178±0.358
zaleplon_mpo	**0.632**±0.008	0.624±0.040	0.584±0.011	0.486±0.004	0.539±0.006	0.536±0.006	**0.552**±0.033	0.519±0.029	0.358±0.062
Sum	18.863	**18.992**	18.362	16.928	16.385	**16.674**	16.213	14.708	14.184

5 CONCLUSION

Goal-directed molecular optimization often relies on policy-gradient RL, whose generation-trajectory log-probabilities depend on how a particular generator and generation procedure produce a molecule. We show that oracle feedback can instead guide native-loss fine-tuning through selection: genetically evolved, high-scoring molecules are retained in a rolling elite buffer, and the generator is updated on them using its own pretraining loss. EW-SFT is a unified optimizer that adapts to each generator through its native proposal mechanism and pretraining loss. Across autoregressive, masked diffusion, and discrete flow generators, and across de novo, motif-extension, and linker-design tasks, it significantly improves on the native optimizer's Top-1k 3D shape similarity, generalizes to held-out references under a 2D oracle, and achieves comparable performance to Genetic-PPO on PMO without requiring its policy-gradient machinery. Across three architectures, three tasks, six references, both 3D and 2D objectives, and 23 further oracles on an external benchmark, we demonstrate that EW-SFT is a general and effective method for goal-directed molecular optimization.

AI USE STATEMENT

In this work, we used generative AI tools to improve the readability and grammar of the manuscript, and to assist with debugging implementation scripts. We did not use generative AI tools to generate datasets, develop the conceptual framework, or design the experimental methodology. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

EW-SFT is intended to improve computational molecular design. Generated molecules should be evaluated under appropriate institutional, safety, and regulatory oversight. Improvements under 3D-shape and 2D-similarity oracles are computational evidence of optimization quality and do not by themselves establish binding, selectivity, synthesis feasibility, safety, or biological activity. Any downstream use should therefore include appropriate experimental validation before biological or therapeutic claims are made.

REPRODUCIBILITY STATEMENT

To reproduce the results, Appendix B details the generators' checkpoint version, the design and hyperparameter settings of genetic operators, optimizers, and the oracle. Appendix C.11 shows the computational resources and runtime for our experiments. All the data and source code will be open-sourced after peer-review.

REFERENCES

Sungsoo Ahn, Junsu Kim, Hankook Lee, and Jinwoo Shin. Guiding deep molecular optimization with genetic exploration, 2020. URL https://arxiv.org/abs/2007.04897.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphael Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters in on-policy reinforcement learning? a large-scale empirical study, 2020. URL https://arxiv.org/abs/2006.05990.

Rasha Atwi, Stephen Farr, Ye Wang, Adam Antoszewski, and Simone Sciabola. Roshambo2: Accelerating molecular alignment for large chemical libraries with gpu optimization and algorithmic advances. *Journal of Chemical Information and Modeling*, 65(19):9842–9849, 2025. doi: 10.1021/acs.jcim.5c01322. URL https://doi.org/10.1021/acs.jcim.5c01322. PMID: 40990368.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces, 2023. URL https://arxiv.org/abs/2107.03006.

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, March 2019. ISSN 1549-960X. doi: 10.1021/acs.jcim.8b00839. URL http://dx.doi.org/10.1021/acs.jcim.8b00839.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design, 2024. URL https://arxiv.org/abs/2402.04997.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on ppo and trpo, 2020. URL https://arxiv.org/abs/2005.12729.

Vendy Fialková, Jiaxi Zhao, Kostas Papadopoulos, Ola Engkvist, Esben Jannik Bjerrum, Thierry Kogej, and Atanas Patronov. Libinvent: Reaction-based generative scaffold decoration for in silico library design. *Journal of Chemical Information and Modeling*, 62(9):2046–2063, 08 2021.

ISSN 1549-9596. doi: 10.1021/acs.jcim.1c00469. URL https://doi.org/10.1021/acs.jcim.1c00469.

Wenhao Gao, Tianfan Fu, Jimeng Sun, and Connor W. Coley. Sample efficiency matters: A benchmark for practical molecular optimization. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=yCZRdI0Y7G.

Jeff Guo and Philippe Schwaller. Augmented memory: Sample-efficient generative molecular design with reinforcement learning. *JACS Au*, 4(6):2160–2172, 04 2024. ISSN 2691-3704. doi: 10.1021/jacsau.4c00066. URL https://doi.org/10.1021/jacsau.4c00066.

Jeff Guo, Franziska Knuth, Christian Margreitter, Jon Paul Janet, Kostas Papadopoulos, Ola Engkvist, and Atanas Patronov. Link-invent: generative linker design with reinforcement learning. *Digital Discovery*, 2(2):392–408, 04 2023. ISSN 2635-098X. doi: 10.1039/d2dd00115b. URL https://doi.org/10.1039/d2dd00115b.

Anvita Gupta, Alex T. Müller, Berend J. H. Huisman, Jens A. Fuchs, Petra Schneider, and Gisbert Schneider. Generative recurrent networks for de novo drug design. *Molecular Informatics*, 37(1-2):1700111, 2018. doi: https://doi.org/10.1002/minf.201700111. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/minf.201700111.

Jan H. Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical Science*, 10(12):3567–3572, 03 2019. ISSN 2041-6520. doi: 10.1039/c8sc05372c. URL https://doi.org/10.1039/c8sc05372c.

Benno Kaech, Luis Wyss, Karsten Borgwardt, and Gianvito Grasso. Refine drugs, don't complete them: Uniform-source discrete flows for fragment-based drug discovery. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Qdu92a5DiM.

Seul Lee, Karsten Kreis, Srimukh Prasad Veccham, Meng Liu, Danny Reidenbach, Yuxing Peng, Saee Gopal Paliwal, Weili Nie, and Arash Vahdat. Genmol: A drug discovery generalist with discrete diffusion. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=KM7pXWG1xj.

Silvana Leit, Jeremy Greenwood, Samantha Carriero, Sayan Mondal, Robert Abel, Mark Ashwell, Heather Blanchette, Nicholas A. Boyles, Mark Cartwright, Alan Collis, Shulu Feng, Phani Ghanakota, Geraldine C. Harriman, Vinayak Hosagrahara, Neelu Kaila, Rosanna Kapeller, Salma B. Rafi, Donna L. Romero, Paul M. Tarantino, Jignesh Timaniya, Angela V. Toms, Ronald T. Wester, William Westlin, Bhaskar Srivastava, Wenyan Miao, Peter Tummino, Joshua J. McElwee, Scott D. Edmondson, and Craig E. Masse. Discovery of a potent and selective tyrosine kinase 2 inhibitor: Tak-279. *Journal of Medicinal Chemistry*, 66(15):10473–10496, 07 2023. ISSN 0022-2623. doi: 10.1021/acs.jmedchem.3c00600. URL https://doi.org/10.1021/acs.jmedchem.3c00600.

Hannes H. Loeffler, Jiazhen He, Alessandro Tibo, Jon Paul Janet, Alexey Voronov, Lewis H. Mervin, and Ola Engkvist. Reinvent 4: Modern ai–driven generative molecule design. *Journal of Cheminformatics*, 16(1):20, Feb 2024. ISSN 1758-2946. doi: 10.1186/s13321-024-00812-5. URL https://doi.org/10.1186/s13321-024-00812-5.

Krzysztof Maziarz, Henry Jackson-Flux, Pashmina Cameron, Finton Sirockin, Nadine Schneider, Nikolaus Stiefl, Marwin Segler, and Marc Brockschmidt. Learning to extend molecular scaffolds with structural motifs, 2024. URL https://arxiv.org/abs/2103.03864.

Daniel Neil, Marwin Segler, Laura Guasch, Mohamed Ahmed, Dean Plumbley, Matthew Sellwood, and Nathan Brown. Exploring deep recurrent models with reinforcement learning for molecule design, 2018. URL https://openreview.net/forum?id=HkcTe-bR-.

Emmanuel Noutahi, Cristian Gabellini, Michael Craig, Jonathan S. C. Lim, and Prudencio Tossou. Gotta be safe: a new framework for molecular design. *Digital Discovery*, 3(4):796–804, 04 2024. ISSN 2635-098X. doi: 10.1039/d4dd00019f. URL https://doi.org/10.1039/d4dd00019f.

Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9(1):48, Sep 2017. ISSN 1758-2946. doi: 10.1186/s13321-017-0235-x. URL https://doi.org/10.1186/s13321-017-0235-x.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019. URL https://arxiv.org/abs/1910.00177.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 04 2010. ISSN 1549-9596. doi: 10.1021/ci100050t. URL https://doi.org/10.1021/ci100050t.

Luke Rossen, Finton Sirockin, Nadine Schneider, and Francesca Grisoni. Scaffold hopping with generative reinforcement learning. *Journal of Chemical Information and Modeling*, 65(13):6513–6525, 06 2025. ISSN 1549-9596. doi: 10.1021/acs.jcim.5c00029. URL https://doi.org/10.1021/acs.jcim.5c00029.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models, 2024. URL https://arxiv.org/abs/2406.07524.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating focussed molecule libraries for drug discovery with recurrent neural networks, 2017. URL https://arxiv.org/abs/1701.01329.

Teague Sterling and John J. Irwin. Zinc 15 – ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 55(11):2324–2337, 10 2015. ISSN 1549-9596. doi: 10.1021/acs.jcim.5b00559. URL https://doi.org/10.1021/acs.jcim.5b00559.

Youhai Tan, Lingxue Dai, Weifeng Huang, Yinfeng Guo, Shuangjia Zheng, Jinping Lei, Hongming Chen, and Yuedong Yang. Drlinker: Deep reinforcement learning for optimization in fragment linking design. *Journal of Chemical Information and Modeling*, 62(23):5907–5917, 11 2022. ISSN 1549-9596. doi: 10.1021/acs.jcim.2c00982. URL https://doi.org/10.1021/acs.jcim.2c00982.

Morgan Thomas, Noel M. O'Boyle, Andreas Bender, and Chris de Graaf. Augmented hill-climb increases reinforcement learning efficiency for language-based de novo molecule generation. *Journal of Cheminformatics*, 14(1):68, Oct 2022. ISSN 1758-2946. doi: 10.1186/s13321-022-00646-z. URL https://doi.org/10.1186/s13321-022-00646-z.

Morgan Thomas, Albert Bou, Jose Carlos Gómez-Tamayo, Gary Tresadern, Mazen Ahmad, and Gianni De Fabritiis. Reinforce-ing chemical language models for drug discovery. *Journal of Chemical Information and Modeling*, 65(23):12752–12763, 11 2025. ISSN 1549-9596. doi: 10.1021/acs.jcim.5c02053. URL https://doi.org/10.1021/acs.jcim.5c02053.

Austin Tripp and José Miguel Hernández-Lobato. Genetic algorithms are strong baselines for molecule generation, 2023. URL https://arxiv.org/abs/2310.09267.

Shiyun Wa, Yifei Wang, Simone Sciabola, and Ye Wang. Advancing ligand-based virtual screening and molecular generation with pretrained molecular embedding distance, 2026. URL https://arxiv.org/abs/2604.24474.

Scott H. Watterson, Qingjie Liu, Myra Beaudoin Bertrand, Douglas G. Batt, Ling Li, Mark A. Pattoli, Stacey Skala, Lihong Cheng, Mary T. Obermeier, Robin Moore, Zheng Yang, Rodney Vickery, Paul A. Elzinga, Lorell Discenza, Celia D'Arienzo, Kathleen M. Gillooly, Tracy L. Taylor, Claudine Pulicicchio, Yifan Zhang, Elizabeth Heimrich, Kim W. McIntyre, Qian Ruan, Richard A. Westhouse, Ian M. Catlett, Naiyu Zheng, Charu Chaudhry, Jun Dai, Michael A. Galella, Andrew J. Tebben, Matt Pokross, Jianqing Li, Rulin Zhao, Daniel Smith, Richard Rampulla, Alban Allentoff, Michael A. Wallace, Arvind Mathur, Luisa Salter-Cid, John E.

Macor, Percy H. Carter, Aberra Fura, James R. Burke, and Joseph A. Tino. Discovery of branebrutinib (bms-986195): A strategy for identifying a highly potent and selective covalent inhibitor providing rapid in vivo inactivation of bruton's tyrosine kinase (btk), 2019. URL https://doi.org/10.1021/acs.jmedchem.9b00167. PMID: 30893553.

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. doi: 10.1021/ci00057a005. URL https://doi.org/10.1021/ci00057a005.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992. ISSN 1573-0565. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Shuchen Xue, Chongjian Ge, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning rl with pretraining in diffusion models, 2025. URL https://arxiv.org/abs/2509.25050.

Soojung Yang, Doyeong Hwang, Seul Lee, Seongok Ryu, and Sung Ju Hwang. Hit and lead discovery with explorative rl and fragment-based molecule generation, 2021. URL https://arxiv.org/abs/2110.01219.

Baoheng Zhu, Deyu Bo, Delvin Ce Zhang, and Xiao Wang. Graph-grpo: Training graph flow models with reinforcement learning, 2026. URL https://arxiv.org/abs/2603.10395.

A Related Work

Recent molecular generative models. The REINVENT family (Olivecrona et al., 2017; Loeffler et al., 2024) established the recurrent Simplified Molecular Input Line Entry System (SMILES) (Weininger, 1988) decoder trained with policy gradients for goal-directed design, and its per-task variants LibINVENT (Fialková et al., 2021) and Link-INVENT (Guo et al., 2023) remain the reference implementations for constrained generation. Later work moved toward a single generalist model. SAFE-GPT (Noutahi et al., 2024) introduced the Sequential Attachment-based Fragment Embedding (SAFE), a fragment-based representation that enables universal molecular design, and trained a GPT-2 backbone on it. GenMol (Lee et al., 2025) applied a masked diffusion language model (Austin et al., 2023; Sahoo et al., 2024) on the same representation, decoupling next-token prediction from SAFE's order-agnostic structure. InVirtuoGen (Kaech et al., 2026) argues that generation is refinement rather than completion, and learns a discrete flow model (Campbell et al., 2024) transporting a uniform source over the vocabulary to the data distribution. Graph-GRPO (Zhu et al., 2026) works on graphs instead of sequences, deriving analytic transition probabilities for graph flow models to form PPO-style (Schulman et al., 2017) importance ratios, illustrating the architecture-specific policy formulation required by policy-gradient RL.

Tasks and evaluations in molecular generation. Frozen sampling is typically used as the first evaluation step for de novo and fragment-constrained generations, where samples generated by the pretrained model are scored for validity, uniqueness, diversity, and quality. SAFE (Noutahi et al., 2024) collected ten existing drugs and built a benchmark for fragment-constrained frozen sampling. Another task is goal-directed optimization, where GuacaMol (Brown et al., 2019) first standardized the objectives and PMO (Gao et al., 2022) fixed 10k calls over 23 oracles. Each architecture used in this paper (Noutahi et al., 2024; Lee et al., 2025; Kaech et al., 2026) has been tuned in this regime. However, Graph-GRPO's oracle calls spent on fine-tuning each task are not counted, and the oracle budget applies only to the evaluation run that follows. Although these models achieved SOTA scores, the tasks are de novo, so whether their optimizers are effective under fragment constraints is untested.

Optimization under a fixed substructure has been done before. LibINVENT (Fialková et al., 2021) decorates a fixed scaffold with an RL-trained recurrent decorator under a 3D shape objective, and Link-INVENT (Guo et al., 2023) generates linkers between fixed warheads with docking in the loop. DRlinker (Tan et al., 2022) rewards linker design for high 3D similarity at low 2D similarity with a transformer sequence-to-sequence policy. MoLeR (Maziarz et al., 2024) decodes a molecular graph motif by motif and optimizes by latent-space search, and FREED (Yang et al., 2021) attaches fragments to a growing graph under PPO. Rossen et al. (2025) argues that confining generation to a predefined substructure limits the exploration of the chemical space. These are per-task architectures with specific optimizers, whether one update rule can apply to all of them is the question this paper takes up.

Likelihood updates as a substitute for policy gradients. Many molecular optimizations use policy-gradient RL, which needs an architecture-specific adaptation. Many studies keep that update stable by adding machinery around it: a prior-likelihood anchor (Olivecrona et al., 2017), diversity filters and replay memory (Loeffler et al., 2024), an augmented likelihood (Thomas et al., 2022), and experience replay with SMILES augmentation (Guo & Schwaller, 2024). Each component improves sample efficiency but adds its own hyperparameters to tune (Gao et al., 2022; Thomas et al., 2025). We take the opposite direction, removing the trajectory-level formulation rather than stabilizing it. Advantage-weighted regression (Peng et al., 2019) turns the reward into a per-sample weight and fits the model to its own samples by weighted maximum likelihood, bypassing the importance ratio entirely. But the exponentiated advantage has a temperature to tune, and it needs a value function estimated alongside the policy. Advantage-weighted matching (Xue et al., 2025) removes the value function by weighting a diffusion model's own pretraining loss, on continuous latents with an advantage weight. In the molecular domain, hill-climbing selects the top-scoring samples a model has generated and refits the model to them by maximum likelihood (Neil et al., 2018), and GEGL (Ahn et al., 2020) adds an external graph-based genetic expert improvement that mutates the contents of a max-reward queue, training an LSTM to imitate that queue by unweighted maximum likelihood. Both take the reward through selection rather than through an importance ratio, assume an architecture whose exact likelihood is available, and are confined to de novo design. Whether such

an update transfers across architectures whose training objectives are likelihood bounds rather than exact likelihoods, whether it extends to two-pass constrained generation procedures that require a dedicated joint-policy formulation, and whether weighting is needed once selection is in place, remain open questions. Rather than assuming elite refitting as the update rule, EW-SFT treats selection and within-set weighting as separate design choices and tests them directly.

B Implementation Details

Generators. All three generative baselines used in this paper are pretrained. GenMol is the released `model_v1` checkpoint (Lee et al., 2025). InVirtuoGen is the released `invirtuo_gen` checkpoint (Kaech et al., 2026). SAFE-GPT is loaded from HuggingFace `https://huggingface.co/datamol-io/safe-gpt`.

Native optimizers. SAFE-GPT's native optimizer is PPO with an adaptive KL penalty and no clipping. The paper states this setting but does not release the implementation, so we reimplemented it from the paper together with the adaptive-penalty variant of Schulman et al. (2017). We keep a batch size of 100 and a learning rate of 1×10^{-5} under AdamW, and take the rest from that variant. The penalty is $\beta \hat{D}_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}})$ against the frozen pretrained policy rather than against the previous iterate, estimated with the k_3 estimator and averaged per action token so that the target does not scale with molecule length. The target is 0.03 per token and β starts at 0.1. After each step β doubles when the measured divergence exceeds the target by more than $1.5\times$ and halves when it falls below the target divided by 1.5, clamped to $[10^{-4}, 10]$. Advantages are the batch-standardized oracle scores. Each step accumulates over micro-batches of 16 and clips gradients at 1.0.

We use GenMol's fragment-remasking GA from its PMO setup. We use the pretrained model under constraints to sample molecules and initialize a population of 100 fragments, each scored by the mean score of the molecules that contain it. The length of each cold-start molecule is drawn from the ZINC250k length distribution (Sterling & Irwin, 2015). Each subsequent round samples 128 parents and remasks one fragment region of each.

InVirtuoGen's is Genetic-PPO with the default configuration for PMO (learning rate 1×10^{-5}, 50 timesteps, 5 reinforce steps per round, clipping $\epsilon = 0.2$, experience replay size 300, population size 100, genetic prompter and mutation on, no prescreening).

Genetic operators. Each model's population content decides what its operator can select, and its generation interface decides how the selected material is regenerated (Table 5). GenMol and SAFE-GPT sample 1,000 molecules de novo and 500 under fragment constraints as a cold start, both charged to the oracle budget, and keep a ranked set of 100 cut fragments, each scored by the mean oracle score of the molecules it was cut from, so their operators select fragments and the model has to assemble a molecule from them. InVirtuoGen samples 110 molecules de novo and 300 under constraints, since the anchor filter discards most of the latter, and keeps the 100 highest-scoring whole molecules. Its genetic prompter vocabulary retains the top 10 of them, so its operator recombines complete molecules.

GenMol and SAFE-GPT both build a parent by attaching fragments and then have the model regenerates part of it. In de novo design, both attach two population fragments into one molecule. GenMol remasks one randomly chosen fragment region of the parent with 5 to 15 mask tokens. SAFE-GPT re-cuts the joined molecule and uses the largest open piece as the prompt to finish regeneration. Under fragment constraints, a single population seed is joined to the fixed anchor, so both operators become mutations of that seed. GenMol remasks a region as before and only keeps molecules that contain the anchor. SAFE-GPT re-cuts as in de novo but keeps the largest open piece that still contains the anchor.

InVirtuoGen recombines two population molecules at the fragment level, keeping all but the last fragment of the first and the last fragment of the second, and the flow model samples a completion conditioned on that token prompt. Under fragment constraints, the prompter is switched off and the anchor alone forms the prompt, with its tokens written back at every denoising step. InVirtuoGen also applies the atom-level graph-based genetic algorithm of Jensen (2019) to the highest-scoring molecules found so far, keeping mutants that retain the anchor. That operator is the only chemical variation among the three models, the others resampling their own generative distribution.

Table 5: **Genetic operator per model.**

		Attachment or recombination		
Model	Population	De novo	Fragment-constrained	Regeneration
SAFE-GPT	100 scored fragments	attach pair, re-cut	attach anchor + one seed, re-cut keeping the anchor	complete the open prefix
GenMol	100 scored fragments	attach fragment pair	attach anchor + one seed	remask a random region, 5–15 tokens
InVirtuoGen	10 scored molecules	fragment-level prompt crossover	no crossover; the anchor alone is the prompt	prompt-conditioned flow sampling, anchor forced

EW-SFT. For three models, the update is AdamW at learning rate 1×10^{-5} with gradient clipping at 1.0. Each round samples 128 molecules (GenMol, SAFE-GPT) or 100 (InVirtuoGen, whose offspring count its own GA sets), scores them, and takes one optimizer step in micro-batches of 16 in rounds where the elite buffer changes. GenMol and SAFE-GPT use a rolling buffer of capacity $K = 64$ to be the elite buffer, because their populations are fragment-level and retain no complete winner. InVirtuoGen realizes it through its own genetic prompter vocabulary, which already holds whole molecules ranked by their oracle score, so no second buffer is added (Table 12).

Oracle. The 3D shape similarity of ROSHAMBO2 (Atwi et al., 2025) is computed by Tanimoto Combination at its default equal weighting of shape and color, against a single reference conformer per reference compound. A molecule is scored by enumerating up to 4 stereoisomers and 25 ETKDGv3 conformers each and taking its best overlap with the reference. We fix the random seed to 42 to ensure reproducibility. Conformer generation dominates the computing cost, using multiple CPU cores to compute in parallel can accelerate it (Table 13). A molecule whose conformation embedding or alignment fails returns -1, it still consumes the oracle budget, but the reported metrics drop it. Oracle calls are charged once per unique canonical SMILES. A repeat is queried from cache and costs no budget, so the budget counts distinct molecules evaluated.

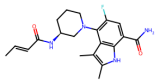
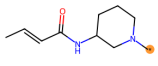
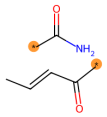
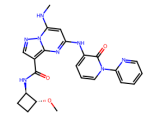
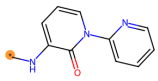
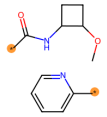
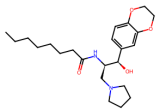
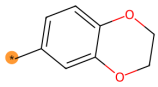
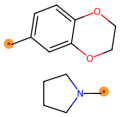
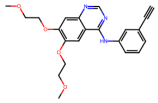
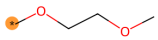
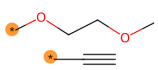
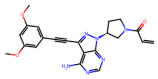
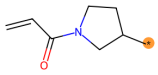
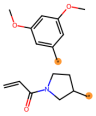
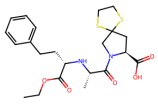
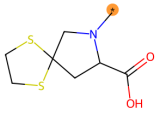
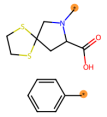
Early stopping and budget accounting. Each backbone applies the same stopping rule to its native optimizer and to EW-SFT. InVirtuoGen's rule is inherited unchanged from the released implementation. It counts rounds in which fewer than 5% of generated molecules are new, discounts them by rounds in which at least half are, and stops when that count reaches 10. For SAFE-GPT, both arms stop after 30 consecutive rounds that add no new molecules. GenMol's GA and EW-SFT need no such rule, since a population that keeps turning over always emits new molecules. The rule is therefore identical within each backbone, and only the native optimizers trigger it, every EW-SFT run spends the full budget. A run that stops early is held flat at its final top-10 mean, following the *mol_opt* convention, which neither rewards nor penalizes stopping.

C Additional Experimental Results

C.1 Fragment-constrained Anchors

The four fragment-constrained-benchmark reference compounds were originally collected by Noutahi et al. (2024), and their anchors are obtained from `https://github.com/NVIDIA-BioNeMo/genmol/blob/main/data/fragments.csv`.

Table 6: **Fragment-constrained anchors.** A motif-extension anchor has one attachment point, the model grows the rest. Linker-design anchors contain two fixed end fragments, the model designs the connecting core. Attachment points ($*$) are highlighted in orange. Top panel: the two kinases used in the main experiments. Bottom panel: the four fragment-constrained-benchmark references used for the held-out evaluation of Section 4.3.

Reference	Molecule	Motif anchor	Linker anchor
BTK inhibitor			
TYK2 inhibitor			
Eliglustat			
Erlotinib			
Futibatinib			
Spirapril			

C.2 Hypothesis Testing

Unless stated otherwise, each p-value we report is a Wilcoxon signed-rank test on paired differences, one-sided where the claim is directional (EW-SFT improves on the native optimizer, or the native optimizer improves on frozen sampling) and two-sided for the nondirectional PMO comparison. The reported p-values are Holm-adjusted within a family of hypotheses.

We pair hypothesis tests by seed, with the two arms sharing an initial population and other random states. Table 1 pairs on (reference, task, seed), giving $n = 18$ per model and $n = 12$ for SAFE-GPT, because we do not evaluate PPO for its two-pass linker generation. Table 3 pairs the same way over four references and two tasks, $n = 24$. The PMO comparison pairs on the oracle, $n = 23$, because the competitor columns there are per-oracle means transcribed from the InVirtuoGen paper (Kaech et al., 2026), and their individual runs are unavailable, so a seed-level pairing is not defined.

The quality–diversity claims in Section 4.2 are the exception, since they concern a correlation rather than a paired comparison. For each of the 12 cells (three models $\times$ two references $\times$ {de novo, linker}) we compute the Pearson correlation between Top-1k internal diversity and Top-1k score over the four optimizing arms and three seeds, $n = 12$ points per cell, and test the resulting 12 correlations against zero with a one-sample t-test. The SFT-only arm is excluded because it is a single extreme point (lowest score, highest diversity), and including it raises the average correlation to -0.88.

C.3 FROZEN GENERATION

The frozen baseline of Section 4.1 is additionally characterized by the fraction of chemically valid and single connected molecules (validity), fraction of distinct canonical SMILES among the valid (uniqueness), and pairwise ECFP4 Tanimoto distance (Rogers & Hahn, 2010) over 1000 random molecules (diversity) (Noutahi et al., 2024).

Table 7 shows all four frozen models produce near-perfectly valid and unique (> 0.926) samples with high diversity (> 0.870) for de novo design. Under fragment constraints, however, validity is largely preserved while uniqueness and diversity drop sharply. These constraints make generation more challenging by restricting the feasible design space to a narrow region of chemical space. This degradation is both anchor- and model-specific. On linker design, both SAFE-GPT and InVirtuoGen generate fewer valid compounds on BTK task (0.461, 0.181) than for TYK2 (0.998, 0.256). InVirtuoGen fails differently due to its hard write-back for fragment anchor tokens (Kaech et al., 2026), which may generate disconnected molecules, yielding only 0.083 validity of BTK motif extension. These results show that frozen sampling remains reliable in the unconstrained setting but cannot transfer robustly to constrained design.

Table 7: **Frozen sampling results.** The results are over 3 runs, higher is better.

		BTK			TYK2		
Model	Task	Valid.	Uniq.	Div.	Valid.	Uniq.	Div.
REINVENT	De novo	0.992±.001	1.000±.000	0.870±.001	0.992±.001	1.000±.000	0.870±.001
	Motif	1.000±.000	0.345±.001	0.569±.001	1.000±.000	0.697±.005	0.587±.001
	Linker	1.000±.000	0.830±.001	0.764±.002	1.000±.000	0.440±.003	0.591±.002
SAFE-GPT	De novo	0.926±.001	1.000±.000	0.879±.001	0.926±.001	1.000±.000	0.879±.001
	Motif	0.967±.001	0.206±.001	0.503±.006	0.946±.001	0.253±.005	0.517±.001
	Linker	0.461±.050	0.617±.007	0.740±.005	0.998±.003	0.500±.098	0.565±.014
GenMol	De novo	0.989±.001	0.928±.001	0.894±.001	0.989±.001	0.928±.001	0.894±.001
	Motif	0.989±.001	0.241±.003	0.558±.001	0.985±.001	0.195±.001	0.533±.002
	Linker	0.998±.000	0.543±.002	0.715±.001	0.999±.000	0.316±.002	0.541±.001
InVirtuoGen	De novo	0.978±.000	1.000±.000	0.881±.001	0.978±.000	1.000±.000	0.881±.001
	Motif	0.083±.004	0.852±.010	0.566±.001	0.853±.002	0.578±.005	0.535±.000
	Linker	0.181±.002	0.872±.003	0.762±.001	0.256±.003	0.693±.008	0.548±.001

C.4 GAIN DECOMPOSITION IN ABSOLUTE UNITS

Figure 2 reports each term as a fraction of the cell's frozen baseline. Table 8 shows the decomposition in raw Top-1k absolute gain.

Table 8: **Optimization improvement over different settings.** The results are Top-1k mean 3D similarity over 3 runs. $\overline{\Delta}_{\mathrm{nat-frz}}$ indicates what the model's native optimizer adds over frozen sampling, $\overline{\Delta}_{\mathrm{ours-nat}}$ is the EW-SFT-to-native improvement, and $\overline{\Delta}_{\mathrm{ours-frz}}$ shows what EW-SFT adds on the frozen sampling. SAFE-GPT linker has no native PPO arm because we do not implement the joint policy required for its two-pass generation.

		BTK			TYK2			
Model	Term	De novo	Motif	Linker	De novo	Motif	Linker	Avg.
SAFE-GPT	Frozen (Top-1k)	0.521	0.534	0.502	0.400	0.428	0.406	0.465
	$\overline{\Delta}_{\mathrm{nat-frz}}$	+0.007	+0.038	—	+0.008	+0.031	—	+0.021
	$\overline{\Delta}_{\mathrm{ours-nat}}$	+0.102	+0.093	—	+0.136	+0.033	—	+0.091
	$\overline{\Delta}_{\mathrm{ours-frz}}$	+0.109	+0.131	+0.061	+0.144	+0.064	+0.039	+0.091
GenMol	Frozen (Top-1k)	0.525	0.594	0.560	0.381	0.415	0.386	0.477
	$\overline{\Delta}_{\mathrm{nat-frz}}$	+0.096	+0.063	+0.064	+0.112	+0.071	+0.135	+0.090
	$\overline{\Delta}_{\mathrm{ours-nat}}$	+0.009	+0.006	+0.026	+0.018	+0.003	+0.011	+0.012
	$\overline{\Delta}_{\mathrm{ours-frz}}$	+0.105	+0.068	+0.090	+0.130	+0.075	+0.146	+0.102
InVirtuoGen	Frozen (Top-1k)	0.531	0.564	0.518	0.404	0.440	0.376	0.472
	$\overline{\Delta}_{\mathrm{nat-frz}}$	+0.192	+0.088	+0.139	+0.161	+0.068	+0.130	+0.129
	$\overline{\Delta}_{\mathrm{ours-nat}}$	+0.074	+0.117	+0.113	+0.047	+0.052	+0.072	+0.079
	$\overline{\Delta}_{\mathrm{ours-frz}}$	+0.266	+0.205	+0.251	+0.208	+0.119	+0.202	+0.209

C.5 Optimization Convergence Curves and AUC

Figures 4- 6 illustrate the running Top-10 mean versus oracle calls for each model's native optimizer and its EW-SFT replacement. Table 9 shows the corresponding AUC Top-10 results.

Table 9: **AUC Top-10 scores of 3D shape similarity optimization.** The results are mean±std over 3 runs, with 20k oracle calls. SAFE-GPT and InVirtuoGen arms inherit their models' native early-stopping. Following the *mol_opt* convention, a run that ends early contributes its final Top-10 mean over the remaining calls, which neither rewards nor penalizes stopping early. The final area under the running Top-10 curve is normalized by the full budget.

		BTK			TYK2			
Model	Method	De novo	Motif	Linker	De novo	Motif	Linker	Avg.
REINVENT	REINFORCE	0.583±0.004	0.741±0.011	0.626±0.013	0.454±0.006	0.530±0.013	0.470±0.001	0.567
SAFE-GPT	PPO	0.581±0.005	0.676±0.016	—	0.447±0.004	0.525±0.005	—	0.557
	EW-SFT (ours)	**0.641**±0.022	**0.695**±0.017	**0.618**±0.010	**0.557**±0.026	**0.528**±0.034	**0.501**±0.015	**0.590**
GenMol	GA	0.645±0.011	0.711±0.005	0.641±0.044	0.508±0.002	0.514±0.013	0.530±0.007	0.592
	EW-SFT (ours)	**0.660**±0.018	**0.711**±0.005	**0.658**±0.009	**0.519**±0.037	**0.518**±0.006	**0.542**±0.011	**0.602**
InVirtuoGen	Genetic-PPO	0.758±0.080	0.778±0.011	0.734±0.015	0.601±0.012	0.561±0.010	0.564±0.054	0.666
	EW-SFT (ours)	**0.807**±0.028	**0.799**±0.024	**0.776**±0.026	**0.612**±0.025	**0.584**±0.016	**0.595**±0.013	**0.695**

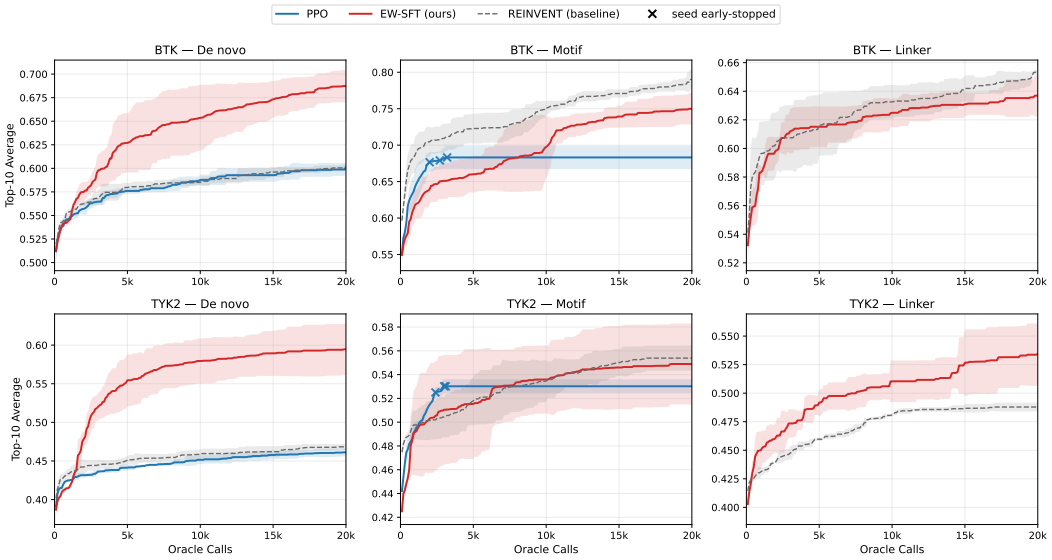


Figure 4: **Optimization convergence for SAFE-GPT.** SAFE-GPT's native PPO with adaptive KL penalty vs. EW-SFT (ours), with REINVENT overlaid in grey as an external baseline. An × marks where a run stopped, the curves are then held flat, so the blue curve is truncated rather than plateauing. EW-SFT runs the full budget in every panel. The linker column has no PPO curve as discussed in Section 2.

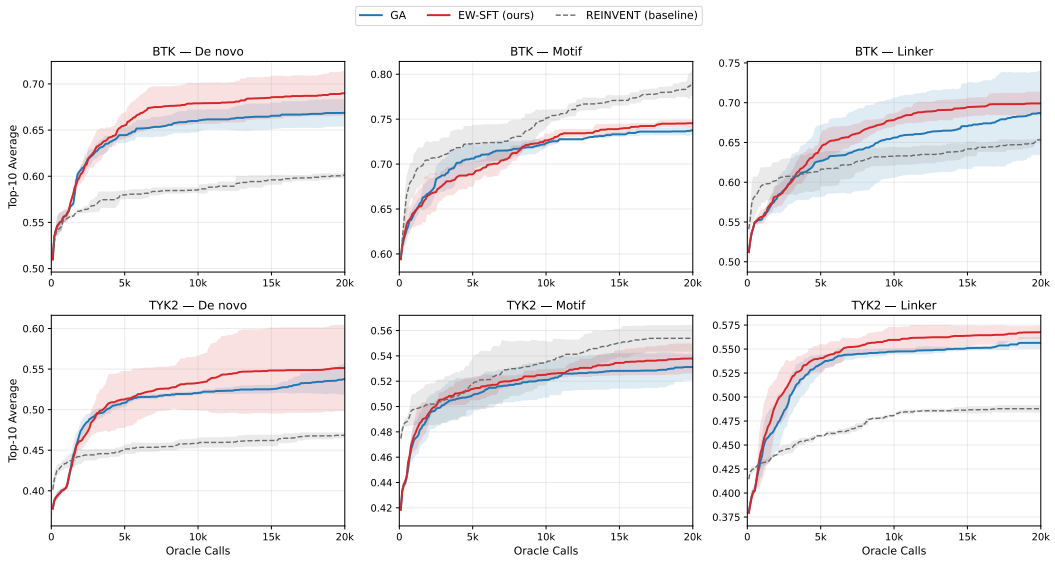


Figure 5: **Optimization convergence for GenMol.** GenMol's native fragment-remasking GA vs. EW-SFT (ours), with REINVENT overlaid in grey as an external baseline. Neither GenMol optimizer early-stops, both curves run to the full budget.

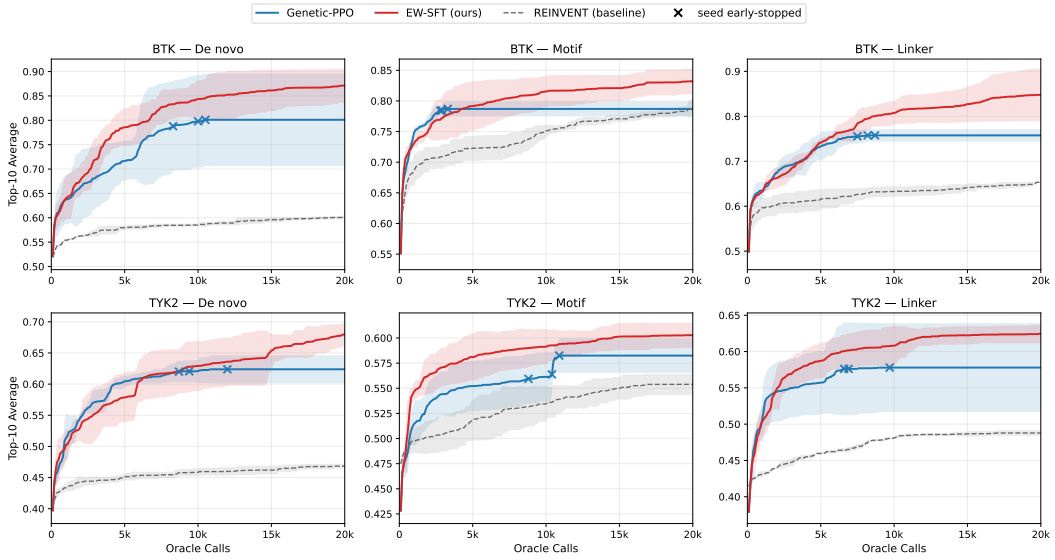


Figure 6: **Optimization convergence for InVirtuoGen.** InVirtuoGen's native Genetic-PPO vs. EW-SFT (ours), with REINVENT overlaid in grey as an external baseline. An $\times$ marks where a run early-stopped. Genetic-PPO exhausts its novelty budget and stops early, whereas EW-SFT keeps producing novel candidates through the full budget.

C.6 Comparison at Matched Oracle Calls

To separate performance at matched query counts from the ability to sustain optimization over the full oracle budget, we additionally compare EW-SFT at the oracle-call count consumed by each native optimizer before early stopping. For each seed, we truncate the paired EW-SFT trajectory at

that call index and recompute Top-1k over the resulting prefix. Table 10 therefore compares the two optimizers at the same oracle-call count without relying on the flat-hold convention of Table 9.

Table 10: **Top-1k average scores at matched native oracle-call counts.** Mean±std over 3 runs. *Native calls (K)* reports the number of oracle calls consumed by each native optimizer, averaged over seeds. For each seed, *EW-SFT, matched* is recomputed from the EW-SFT prefix ending at that seed's native call count; matching is therefore performed per seed before averaging. Cells at 20.0 reproduce the corresponding full-budget result. **Bold** marks the better value between the native optimizer and its matched EW-SFT result. *EW-SFT, full 20k* is shown for reference and is not included in the matched comparison. SAFE-GPT linker tasks appear only in this row because PPO has no corresponding linker run. Row averages use only the cells available in that row.

		BTK			TYK2			
Model	Method	De novo	Motif	Linker	De novo	Motif	Linker	Avg.
SAFE-GPT	*Native calls (K)*	*20.0*	*2.6*	—	*20.0*	*2.8*	—	—
	PPO	0.528±0.000	**0.572**±0.012	—	0.408±0.001	**0.459**±0.009	—	0.492
	EW-SFT, matched	**0.630**±0.023	0.520±0.008	—	**0.544**±0.025	0.429±0.037	—	0.531
	EW-SFT, full 20k	0.630±0.023	0.665±0.022	0.563±0.009	0.544±0.025	0.492±0.043	0.446±0.014	0.556
GenMol	*Native calls (K)*	*20.0*	*20.0*	*20.0*	*20.0*	*20.0*	*20.0*	—
	GA	0.622±0.008	0.657±0.007	0.624±0.048	0.493±0.009	0.487±0.014	0.521±0.004	0.567
	EW-SFT, matched	**0.631**±0.019	**0.663**±0.003	**0.650**±0.013	**0.510**±0.040	**0.490**±0.009	**0.532**±0.007	0.579
	EW-SFT, full 20k	0.631±0.019	0.663±0.003	0.650±0.013	0.510±0.040	0.490±0.009	0.532±0.007	0.579
InVirtuoGen	*Native calls (K)*	*9.5*	*2.9*	*8.1*	*10.0*	*10.0*	*7.7*	—
	Genetic-PPO	0.723±0.081	**0.651**±0.009	0.656±0.008	0.565±0.012	0.507±0.021	0.506±0.041	0.602
	EW-SFT, matched	**0.742**±0.045	0.639±0.007	**0.665**±0.027	**0.570**±0.038	**0.535**±0.013	**0.525**±0.034	0.613
	EW-SFT, full 20k	0.797±0.034	0.769±0.017	0.769±0.048	0.612±0.030	0.559±0.004	0.578±0.020	0.681

Even under this matched-call comparison, EW-SFT improves the overall average for all three backbones. This analysis deliberately removes the benefit of continuing the search after the native optimizer has stopped. In the primary fixed-budget evaluation, SAFE-GPT and InVirtuoGen apply the same backbone-specific early-stopping rule to both optimization methods. Their native optimizers trigger this rule, whereas EW-SFT does not and therefore continues to use the available oracle budget. GenMol does not early-stop in either arm. Thus, the additional calls in Table 1 reflect a difference in optimizer behavior rather than a difference in the assigned oracle budget.

C.7 2D Novelty

On the same Top-1k molecules scored by 3D shape similarity against each of the two kinase references in Table 1, we also compute the ECFP4 Tanimoto similarity of each molecule to the reference (Table 11), to show whether the optimizer improves the 3D score by rediscovering the reference compound.

Table 11: **2D similarity to the reference.** The results are ECFP4 Tanimoto similarity over the Top-1k by 3D score (mean ± std over 3 runs). A molecule that has high 3D shape similarity while low 2D similarity to reference means it is shape-matched but chemically distinct. The motif and linker columns have higher scores because the provided anchor is a fragment of the reference and already supplies some similarity. Underlined EW-SFT cell is less similar to the reference than the model's native optimizer (9 of 16 cells). Lower is the desirable direction here.

		BTK			TYK2		
Model	Method	De novo	Motif	Linker	De novo	Motif	Linker
SAFE-GPT	PPO	0.125±0.001	0.394±0.009	—	0.139±0.002	0.361±0.002	—
	EW-SFT (ours)	0.159±0.085	0.438±0.018	0.207±0.005	0.153±0.017	0.310±0.011	0.328±0.005
GenMol	GA	0.133±0.020	0.411±0.013	0.229±0.061	0.138±0.014	0.347±0.012	0.350±0.007
	EW-SFT (ours)	0.157±0.062	0.405±0.006	0.190±0.018	0.157±0.025	0.347±0.023	0.341±0.006
InVirtuoGen	Genetic-PPO	0.170±0.097	0.372±0.016	0.163±0.028	0.148±0.032	0.238±0.029	0.193±0.044
	EW-SFT (ours)	0.249±0.073	0.279±0.075	0.254±0.087	0.103±0.029	0.229±0.063	0.186±0.052

C.8 Ablation Settings

Table 12 defines the six ablation settings compared in Section 4.2 Table 2. As genetic search+elite+binary is the best setting for EW-SFT, as a comparison, the SFT-only setting is also based on elite+binary but removing the genetic search and sampling straight from the pretrained model. Proposition 1 shows the signed emphasis is ill-posed and it empirically does not perform well, so we exclude the elite+signed setting. The batch-binary is also excluded because the reward signal needs to enter somewhere. The elite buffer differs by architecture, but every model is run at exactly one molecule-level elite stage. GenMol and SAFE-GPT evolve fragment populations with no complete winners, so for them the buffer has to be added. In contrast, InVirtuoGen runs on the round batch, because its genetic prompter vocabulary already keeps whole molecules ranked by their own oracle score, which is what E_t is defined to be. Its $\tau_t = -\infty$ rows are therefore its elite rows, and adding our buffer on top would compose a second selection stage.

Table 12: **Ablation settings.** Two factors of Equation 2: the threshold τ_t determines the selected set, and the emphasis re-weights the selected molecules. Elite is the rolling buffer of capacity $K = 64$ with $\tau_t = \min_{x' \in E_t} f(x')$. $\tau_t = -\infty$ selects the whole valid round batch. GA is the genetic search of Section 3.

Setting	GA	τ_t (threshold)	Emphasis
EW-SFT (ours)			
elite + binary	✓	elite min	1
elite + positive	✓	elite min	$\max(\hat{A}, 0)$
batch + positive	✓	$-\infty$	$\max(\hat{A}, 0)$
batch + signed	✓	$-\infty$	$\hat{A}$
Standalone			
GA only	✓	—	—
SFT only	—	elite min	1

C.9 Why the emphasis is not signed

A variant sets the emphasis to a batch-normalized advantage $\hat{A}$, which is negative for below-average samples. It is not merely worse empirically, but also unbounded. Gradient descent can therefore reduce the unbounded-below loss by pushing away disfavored samples rather than toward the good ones. Figure 7 shows the failure mode. This update rule's property applies to all three models.

Proposition 1 (Signed emphasis makes the objective unbounded below)**.** *Fix a batch $\tilde{B}_t$ under signed emphasis, and suppose that $w(x^-) = \hat{A}(x^-) < 0$ for some $x^- \in \tilde{B}_t$. If the native loss $\ell_\theta(x^-)$ is non-negative and unbounded above, then the weighted objective $\sum_{x \in \tilde{B}_t} w(x)\, \ell_\theta(x)$ is unbounded below. Holding the remaining terms finite, letting $\ell_\theta(x^-) \to \infty$ gives $w(x^-)\ell_\theta(x^-) \to -\infty$. Restricting to $w \geq 0$ makes every term a non-negative multiple of a non-negative loss, so the objective is bounded below by* 0.

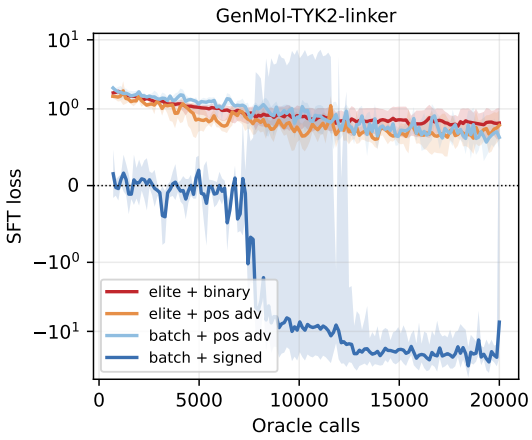


Figure 7: **Signed emphasis drives the SFT loss below zero.** The signed per-round training loss crosses zero and keeps descending, every other setting stays bounded below.

C.10 Exploration and Exploitation Under the Elite Update

Figure 8 shows the exploration and exploitation dynamics of three representative cases. EW-SFT's gain is associated with a longer productive search and is not accompanied by a significant reduction in diversity.

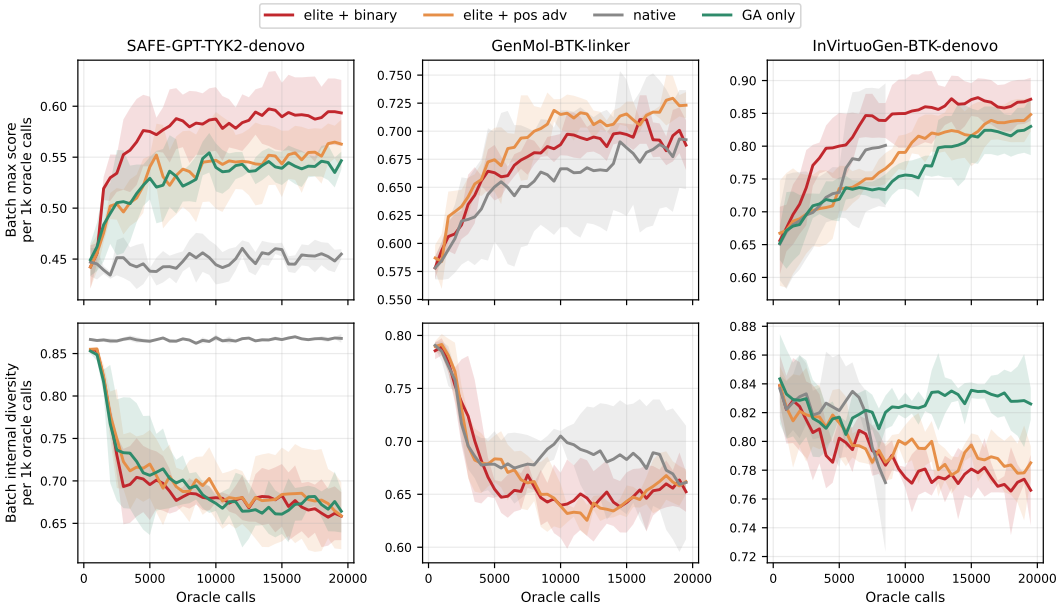


Figure 8: **The exploration and exploitation dynamics of three optimization cases.** Each curve's point is computed within a rolling window of 1000 oracle calls at a stride of 500, over three seeds. Internal diversity is the pairwise ECFP4 Tanimoto distance within a window. InVirtuoGen's native Genetic-PPO curve early-stopped at ≈8k calls. GenMol's native optimizer is GA itself, so its grey curve is also its GA-only baseline.

C.11 Runtime

Wall-clock hours reflect the budget each arm actually consumed, since the native optimizers frequently terminate before exhausting the budget (Appendix B).

Table 13: **Cost per run.** Wall-clock hours (h), thousands of oracle calls spent (K), and their ratio (ms/call), as means over both references and three seeds on one A100 with 15 CPU. SAFE-GPT PPO has no linker entry because it is not evaluated on that task (Table 1).

		De novo			Motif			Linker		
Model	Method	h	calls (K)	ms/call	h	calls (K)	ms/call	h	calls (K)	ms/call
SAFE-GPT	Native	1.5	20.0	278	1.1	2.7	1452	—	—	—
	EW-SFT (ours)	3.8	20.0	682	4.9	20.0	883	5.9	20.0	1056
GenMol	Native	4.1	20.0	743	7.8	20.0	1411	8.8	20.0	1578
	EW-SFT (ours)	5.4	20.0	964	7.7	20.0	1394	15.9	20.0	2868
InVirtuoGen	Native	7.1	9.8	2618	3.0	6.5	1658	4.7	7.9	2151
	EW-SFT (ours)	7.4	20.0	1325	7.4	20.0	1327	9.5	20.0	1703