FORGETU: LEAVING UNDESIRABLE MOLECULES BEHIND IN DE NOVO DESIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models based on flow matching and diffusion have shown strong potential for 3D de novo molecular design. However, models trained on existing molecular datasets may also learn to generate molecules with toxic or other undesired properties, limiting their reliability and practical utility. Retraining a model from scratch using only retained molecules can address this issue but may be computationally expensive for large 3D molecular datasets. Post-training a pretrained model provides an efficient alternative, but faces two key challenges: suppressing undesired generation while preserving the original generation capability, and preventing the forgotten behavior from being easily recovered through subsequent fine-tuning. In this study, we propose a Reinforcement Learning (RL)-based robust forgetting framework for pretrained 3D molecular flow matching models. RL-based forgetting suppresses undesired generation, while selective distillation preserves generation behavior on retained molecules. We further simulate relearning on undesired molecules and use the resulting loss to further update the model, improving the robustness of forgetting against subsequent fine-tuning. Experiments show that our framework achieves effective and robust forgetting while preserving the original generation capability.

1 INTRODUCTION

Background. In recent years, diffusion and flow matching have been widely used for de novo 3D molecular generation and have shown strong performance across a wide range of molecular design applications Hoogeboom et al. (2022); Xu et al. (2023); Dunn & Koes (2026). These models are typically trained on large general-purpose 3D molecular datasets that may contain undesired molecules, such as toxic or potentially harmful ones, causing the model to learn related generation behaviors Nikitin et al. (2025), as illustrated in Fig. 1. When these models are publicly available and generate such molecules, these molecules may pose safety risks if exploited or misused Urbina et al. (2022). A straightforward solution is to remove these molecules from the training data and retrain the model from scratch, preventing the model from directly learning from these undesired samples Bourtoule et al. (2021). However, this may be computationally expensive for training models on large 3D molecular datasets. Moreover, since generative models can produce novel structures, newly generated molecules can still have undesired properties even if the training data do not contain such molecules Urbina et al. (2022). Therefore, post-training a pretrained model to forget undesired generation behaviors may offer a more practical alternative.

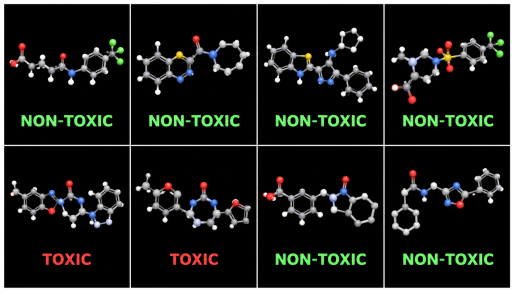

Figure 1: Generated Molecules.

Challenge 1: How to forget properly? A typical strategy is to construct a forget set of undesired samples and maximize their training loss to suppress related generation Kim et al. (2025). However, undesired molecules can be highly diverse. Different undesired properties may arise from different mechanisms, and even the same mechanism can be triggered by molecules with substantially different structures Liebler & Guengerich (2005); Guengerich (2011). For example, molecules that

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.

cause hERG inhibition can still have markedly different structures Perry et al. (2010). Therefore, a finite forget set may not adequately represent the diverse undesired molecules that the model can generate, making it difficult to fully define the range of generation that should be suppressed. In addition, forgetting should preserve the model's original generation capability as much as possible. Because different generation behaviors are realized by the same set of model parameters, updating these parameters to suppress undesired molecules may also affect normal generation unrelated to these undesired properties Cheng et al. (2026). Such collateral changes can degrade the pretrained model's original generation capability, making forgetting come at the cost of reduced model utility Mavrothalassitis et al. (2025). Thus, effectively suppressing diverse undesired generation while preserving the model's original generation capability remains a key challenge.

Challenge 2: How to make forgetting robust? Although a model may no longer generate undesired molecules after forgetting, this effect can be reversed by subsequent fine-tuning Gao et al. (2025). Publicly available generative models can be further fine-tuned to adapt to new data, downstream tasks, or optimization objectives. For example, a general molecular generator may be fine-tuned using target-specific data for inhibitor design Wang et al. (2023). Existing studies have shown that such updates can cause previously suppressed generation behaviors to re-emerge, even when the fine-tuning data are unrelated to the forgotten content Suriyakumar et al. (2026). One possible reason is that forgetting may suppress the model's current generation behavior without fully eliminating the underlying information associated with these behaviors, allowing them to re-emerge after subsequent updates Hu et al. (2025). Moreover, in higher-risk settings, such as adversarial fine-tuning on datasets containing toxic molecules, the model may regain or even increase its propensity to generate toxic molecules Pan et al. (2024). Therefore, how to maintain the forgetting effect under subsequent fine-tuning and prevent it from being easily recovered is another challenge.

Overview. In this study, we propose a Reinforcement Learning (RL)-based forgetting strategy that uses feedback from molecules generated by the current model to optimize a continuous-discrete 3D molecular flow matching model and reduce undesired generation, without relying on a predefined forget set. To preserve unrelated normal generation during forgetting, we introduce selective distillation from the pretrained model, constraining the updated model to remain close to the pretrained model along normal generation trajectories. Furthermore, to prevent forgotten behaviors from being easily recovered in downstream tasks, we simulate relearning on the forgotten model and use the resulting recovery loss to further update the model, making forgetting more robust. We evaluate ForgetU across multiple molecular datasets and forgetting targets, and further assess the preservation of the original generation capability and robustness to subsequent fine-tuning.

Our contributions can be summarized as follows:

- To our knowledge, we introduce the first framework for forgetting undesired 3D molecular generation in continuous-discrete flow matching models.
- We develop an RL-based strategy to forget undesired generation, while using selective distillation to preserve unrelated generation.
- We introduce simulated relearning to improve the robustness of forgetting in downstream tasks.

2 Related Work

Application Domains. Existing forgetting-related studies have mainly focused on image generation Safavigerdini et al. (2026). Forget-Me-Not modifies cross-attention to remove specific identities or image styles Zhang et al. (2024). ESD uses negative guidance to construct a new training objective and erases visual concepts by fine-tuning model weights Gandikota et al. (2023). Ablating Concepts maps the generation distribution of a target concept to an anchor concept to reduce the generation of corresponding image content Kumari et al. (2023). SalUn identifies and updates the model parameters most relevant to forgetting using gradients from forget data, enabling the removal of specified image classes or concepts Fan et al. (2023). UCE uses closed-form parameter editing for concept erasure in images Gandikota et al. (2024). In contrast, **the field of 3D molecular generation remains underexplored.**

RL-based forgetting. RL has been used for forgetting tasks in Large Language Models (LLMs). Quark reduces undesired behaviors through scalar rewards with global KL regularization Lu et al.

(2022). RULE formulates LLM forgetting as RL-based refusal-boundary optimization, encouraging refusal on forget-related queries while preserving responses to permissible inputs Zhang et al. (2025). PURGE uses RL with group-relative policy optimization and intrinsic rewards to penalize forbidden concepts Zaradoukas et al. (2026). These methods mainly target LLMs, where sequential token decisions have tractable action probabilities and are naturally suited to RL. In contrast, flow matching generates samples through iterative state transitions Lipman et al. (2023a), while continuous-discrete flow matching jointly handles continuous coordinates and discrete structures, making RL optimization more challenging Wang et al. (2026). **Currently, RL-based forgetting remains largely unexplored in continuous-discrete flow matching models.** In addition, existing methods mainly preserve the model's original capabilities through global regularization or predefined permissible inputs. The former may conflict with the forgetting objective, while the latter may provide limited coverage of the model's original normal generation distribution.

Robustness of Forgetting. Prior studies have shown that insufficient forgetting can be easily reversed by subsequent fine-tuning Hu et al. (2025). Even a small amount of benign data loosely related to the forgotten content may recover forgotten knowledge Yoon et al. (2026). Similar relearning vulnerabilities have also been observed in diffusion models Suriyakumar et al. (2026). To improve robustness against subsequent updates, existing studies have explored several strategies, including sharpness-aware optimization to reduce susceptibility to relearning attacks Fan et al. (2025) and invariance regularization to maintain forgetting across different downstream fine-tuning settings Wang et al. (2025). These methods mainly introduce auxiliary objectives to slow recovery, whereas we simulate relearning and update the model based on the observed degree of recovery.

3 METHODOLOGY

In this section, we introduce the continuous-discrete flow matching model for 3D molecular generation and describe how its transition likelihoods are constructed. Based on these transition likelihoods, we then present our RL-based robust forgetting framework. The overall architecture is shown in Fig. 2. Additional implementation details, derivations, and proofs are provided in Appendix B.

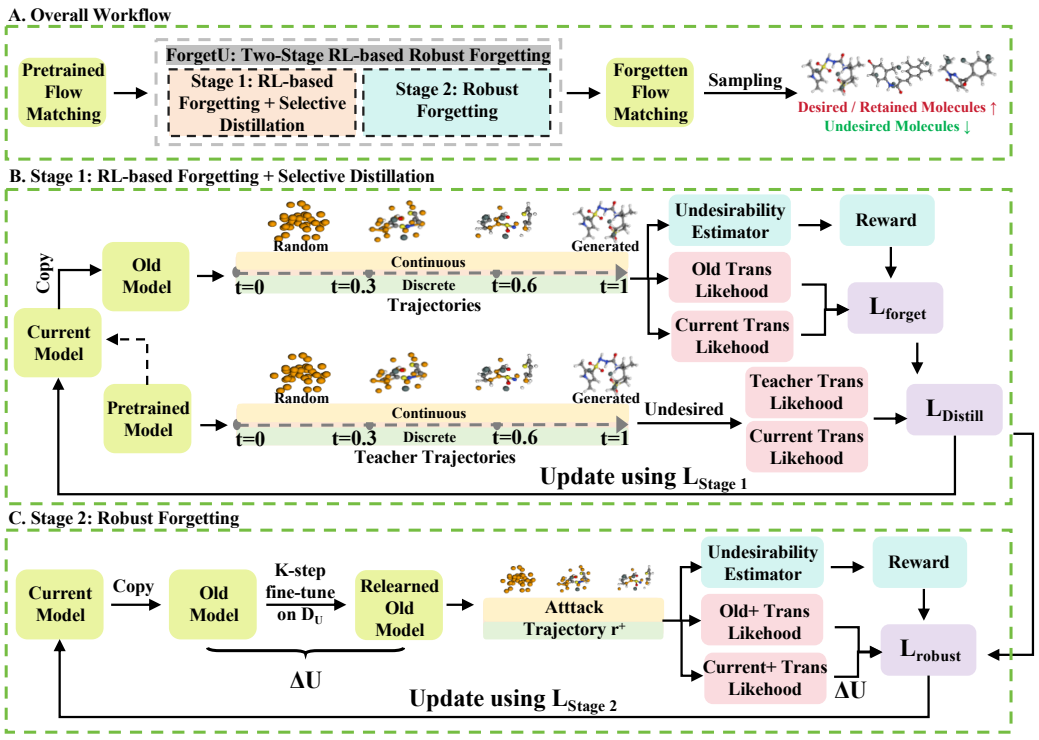

Figure 2: Overall architecture of ForgetU. The two-stage training framework is illustrated, with details provided in Section 3.3.

3.1 CONTINUOUS-DISCRETE FLOW MATCHING

We build on a continuous-discrete flow matching model parameterized by ψ, where the molecular state at time $t \in [0,1]$ is represented as $G_t = (X_t, A_t, C_t, E_t)$, with X_t denoting the atomic coordinates and A_t, C_t, and E_t denoting atom types, formal charges, and bond types, respectively.

For the continuous coordinates, the model predicts the endpoint $\hat{X}_1^\psi$ from the current state G_t and defines the flow field as $v_\psi(G_t, t) = \frac{\alpha_t'}{1-\alpha_t}\left(\hat{X}_1^\psi - X_t\right)$, which determines the coordinate dynamics $\frac{dX_t}{dt} = v_\psi(G_t, t)$, where α_t is the interpolation schedule.

The discrete states A_t, C_t, and E_t are evolved using Campbell continuous-time Markov chain (CTMC) transitions Daley & Vere-Jones (2003). Each discrete variable includes an additional mask state, and generation starts with the discrete variables masked. Given the current molecular state G_t, the model predicts the possible endpoint values of A_1, C_1, and E_1. For a transition from t to $s > t$, a masked variable is revealed with probability $p_{\text{unmask}} = (s-t)\frac{\alpha_t' + \eta\alpha_t}{1-\alpha_t}$, and its value is sampled according to the corresponding endpoint prediction. An already revealed variable can return to the mask state with probability $p_{\text{mask}} = (s-t)\eta$, where η controls the stochasticity of the discrete dynamics. For finite-step sampling, both transition probabilities are clipped to $[0,1]$.

Generation proceeds from an initial noise-and-mask state at $t = 0$ to a complete molecule at $t = 1$. For computation, we discretize the interval as $0 = t_0 < t_1 < \cdots < t_T = 1$ and denote the resulting trajectory as $\tau = (G_0, G_1, \ldots, G_T)$.

3.2 TRANSITION LIKELIHOOD FOR CONTINUOUS-DISCRETE FLOW MATCHING

Transition likelihood measures the probability assigned by the model to each generation step and reflects the likelihood of each state transition under the model. We use ψ to denote the parameters of the model whose transition likelihood is being evaluated.

For the continuous part, the original Flow Matching Ordinary Differential Equation (ODE) Lipman et al. (2023b) is deterministic and does not define a stochastic transition density. To obtain such a density, we use a marginal-preserving Stochastic Differential Equation (SDE) Pitman & Yor (2018) construction associated with the original flow and adopt its discretized Gaussian transition Freidlin & Wentzell (2012). Given $s > t$, the deterministic flow update is $\widetilde{X}_s = X_t + (s-t)v_\psi(G_t, t)$. To ensure numerical stability near the temporal boundaries, we apply a small ϵ-clamping to α_t and α_s. We define the integrated variance as $V_{t,s} = \sigma_x^2\left[\log\frac{\alpha_s}{\alpha_t} - (\alpha_s - \alpha_t)\right]$, where σ_x controls the stochasticity. Using the predicted endpoint $\hat{X}_1^\psi$, we estimate the corresponding score as $S_\psi(G_t, t) = -\frac{X_t - \alpha_t\hat{X}_1^\psi}{(1-\alpha_t)^2}$, which gives the transition mean $\mu_{\psi,t\to s} = \widetilde{X}_s + \frac{1}{2}V_{t,s}S_\psi(G_t, t)$. The resulting continuous transition likelihood is $p_\psi^{\text{cont}}(X_s \mid G_t) = \mathcal{N}(X_s; \mu_{\psi,t\to s}, V_{t,s}I)$.

The discrete states A_t, C_t, and E_t already follow stochastic Campbell CTMC dynamics. Their transition likelihood $p_\psi^{\text{disc}}(A_s, C_s, E_s \mid G_t)$ is computed from the probabilities of the masking and value-sampling operations used during generation. For a masked variable, let $u \in \{0,1\}$ indicate whether it is revealed during the transition and let $\pi_\psi(y \mid G_t, t)$ denote the predicted probability of the sampled value y. Its contribution to the discrete log-likelihood is $\log p_{\text{Bern}}(u; p_{\text{unmask}}) + u\log\pi_\psi(y \mid G_t, t)$. The remasking operation is evaluated analogously using p_{mask}.

The continuous and discrete components are combined to obtain the transition log-likelihood $\log p_\psi(G_s \mid G_t)$, which is subsequently used for RL optimization. All RL rollouts are sampled using the same stochastic transition kernels used to evaluate their transition log-likelihoods.

3.3 RL-BASED ROBUST FORGETTING

This section presents a two-stage RL-based strategy to reduce undesired molecular generation while preserving the pretrained model's original generation capability and improving robustness to downstream recovery. In the first stage (Alg. 1), a binary reward guides RL-based forgetting, while selective distillation preserves retained generation behavior. In the second stage (Alg. 2), we simu-

late relearning on undesired molecules and use the loss after relearning to further update the model, making the forgetting more robust to similar downstream updates.

Reward Design. To clearly distinguish undesired molecules from retained molecules, we design a binary reward. Specifically, for a generated molecule G_T, we define $R(G_T) = -U(G_T)$, where $U(G_T) = 1$ if the molecule is undesired and $U(G_T) = 0$ otherwise. Under this design, the model receives a single and explicit preference signal determined only by whether the generated molecule is undesired. Undesired molecules receive a negative reward, which drives the model to reduce the probability of the generation transitions leading to them, whereas retained molecules receive zero reward and therefore provide no direct forgetting signal. As a result, RL focuses specifically on suppressing undesired generation without introducing additional preferences among retained molecules.

RL-based Forgetting. Given the reward defined above, we view the discretized molecular generation process as a finite-horizon Markov process, where the model parameterizes the transition distribution between consecutive molecular states. At the beginning of each optimization round, we copy the current model ϕ to a frozen old model ϕ_{old}. The old model generates trajectories $\tau = (G_0, \ldots, G_T)$ and records the transition log-probability $\log p_{\phi_{\text{old}}}(G_{t+1} \mid G_t, t)$ at each timestep. The terminal reward $R(G_T)$ is treated as a delayed return and assigned to the transitions along the corresponding trajectory. The current model then evaluates the same transitions and computes the likelihood ratio $\rho_t = \exp\left[\log p_\phi(G_{t+1} \mid G_t, t) - \log p_{\phi_{\text{old}}}(G_{t+1} \mid G_t, t)\right]$. We optimize the current model using a clipped Proximal Policy Optimization (PPO)-style Schulman et al. (2017) objective

$$L_{\text{forget}} = -\mathbb{E}_{\tau,t}\left[\min\left(\rho_t R(G_T), \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon) R(G_T)\right)\right], \tag{1}$$

where ϵ controls the magnitude of each policy update. For trajectories ending in undesired molecules, the negative return encourages the current model to assign lower likelihood than the old model to their transitions, thereby progressively reducing the probability of generation paths leading to undesired molecules. After the PPO updates, the old model is refreshed from the current model and the process is repeated.

Algorithm 1: Stage 1: RL-based Forgetting and Selective Distillation

Input: Pretrained model θ; undesired criterion U; PPO clipping parameter ϵ; distillation weight λ_{distill}; number of optimization steps K_{opt}
Output: Initial forgotten model ϕ

$\phi \leftarrow \theta$;
Freeze θ;
for *each training round* **do**
  $\phi_{\text{old}} \leftarrow \phi$; freeze ϕ_{old};
  /* RL-based Forgetting */;
  $\tau = (G_0, \ldots, G_T) \sim p_{\phi_{\text{old}}}$;
  $R(G_T) \leftarrow -U(G_T)$;
  /* Selective Distillation */;
  $\mathcal{T}^\theta \sim p_\theta$;
  $\mathcal{T}_{\text{ret}} \leftarrow \{\tau^\theta \in \mathcal{T}^\theta : U(G_T^\theta) = 0\}$;
  for $k = 1, \ldots, K_{\text{opt}}$ **do**
    for $t = 0, \ldots, T-1$ **do**
      $\rho_t \leftarrow \frac{p_\phi(G_{t+1} \mid G_t, t)}{p_{\phi_{\text{old}}}(G_{t+1} \mid G_t, t)}$;
    Compute L_{forget} using Eq. 1;
    Compute L_{distill} on $\mathcal{T}_{\text{ret}}$ using Eq. 2;
    $L_{\text{stage1}} \leftarrow L_{\text{forget}} + \lambda_{\text{distill}} L_{\text{distill}}$;
    $\phi \leftarrow \text{Update}(\phi, L_{\text{stage1}})$;
return ϕ

Selective Distillation. Although retained molecules receive zero reward and therefore do not directly contribute to the forgetting update, the model parameters are shared across different generation trajectories. Consequently, updates that suppress undesired generation may also indirectly alter transition predictions along retained trajectories. To preserve the original generation capability, we selectively distill the pretrained model along retained generation trajectories. Specifically, we freeze the pretrained model θ and sample trajectories $\tau^\theta = (G_0^\theta, \ldots, G_T^\theta)$. Trajectories whose terminal molecules satisfy $U(G_T^\theta) = 0$ are retained for distillation. For each retained intermediate state G_t^θ, we feed the same state and timestep into both the pretrained model and the current model ϕ, and match their continuous and discrete transition predictions using Kullback-Leibler (KL) divergence Kullback & Leibler (1951). We define the distillation objective as

$$L_{\text{distill}} = \mathbb{E}_{\tau^\theta,t}\left[D_{\text{KL}}\left(p_\theta(\cdot \mid G_t^\theta, t) \| p_\phi(\cdot \mid G_t^\theta, t)\right)\right]. \tag{2}$$

By applying this matching only to retained trajectories, the current model is encouraged to preserve the pretrained generation behavior on retained trajectories.

Robust Forgetting. Although the first stage suppresses undesired generation, the forgotten behavior may still be recovered through subsequent fine-tuning. To make the forgotten model more robust to such recovery, we simulate relearning after the initial forgetting stage. Let D_U denote a set of undesired molecules. At the beginning of each training round, we copy the current model ϕ to ϕ_{old} and fine-tune ϕ_{old} on D_U for K steps using the original flow matching objective, obtaining a relearned model ϕ^+_{old}. We define the resulting parameter displacement as $\Delta_U = \phi^+_{\text{old}} - \phi_{\text{old}}$, which represents the model change induced by simulated relearning. We then apply the same displacement to the current model and define $\phi^+ = \phi + \text{sg}(\Delta_U)$, where sg keeps the simulated relearning update fixed while the current model remains trainable. The relearned old model ϕ^+_{old} generates trajectories $\tau^+ = (G^+_0, \ldots, G^+_T)$, while the relearned current model ϕ^+ evaluates the same transitions. For each timestep, we compute

$$\tilde{\rho}_t = \exp\Big[\log p_{\phi^+}(G^+_{t+1} \mid G^+_t, t) - \log p_{\phi^+_{\text{old}}}(G^+_{t+1} \mid G^+_t, t)\Big], \tag{3}$$

and use the same reward to define the PPO-style objective

$$L_{\text{robust}} = -\mathbb{E}_{\tau^+,t}\Big[\min\Big(\tilde{\rho}_t R(G^+_T), \text{clip}(\tilde{\rho}_t, 1-\epsilon, 1+\epsilon)\, R(G^+_T)\Big)\Big]. \tag{4}$$

Because ϕ^+ depends directly on ϕ while Δ_U is fixed during this update, minimizing L_{robust} updates ϕ based on its behavior after simulated relearning. In this way, the model is trained to remain forgotten even after similar downstream updates.

Algorithm 2: Stage 2: Robust Forgetting

Input: Initial forgotten model ϕ; pretrained model θ; undesired criterion U; undesired molecule set D_U; relearning steps K; relearning learning rate η_U; PPO clipping parameter ϵ; λ_{distill}; λ_{robust}; optimization steps K_{opt}

Output: Final model ϕ

Freeze θ;
for *each training round* **do**
  $\phi_{\text{old}} \leftarrow \phi$; freeze ϕ_{old};
  /* Simulated Relearning */;
  $\phi^+_{\text{old}} \leftarrow \phi_{\text{old}}$;
  for $k = 1, \ldots, K$ **do**
    $\phi^+_{\text{old}} \leftarrow \phi^+_{\text{old}} - \eta_U \nabla_{\phi^+_{\text{old}}} L_{\text{FM}}(\phi^+_{\text{old}}; D_U)$;
  Freeze ϕ^+_{old};
  $\Delta_U \leftarrow \phi^+_{\text{old}} - \phi_{\text{old}}$;
  /* Robust Forgetting */;
  $\tau^+ = (G^+_0, \ldots, G^+_T) \sim p_{\phi^+_{\text{old}}}$;
  $R(G^+_T) \leftarrow -U(G^+_T)$;
  for $j = 1, \ldots, K_{\text{opt}}$ **do**
    $\phi^+ \leftarrow \phi + \text{sg}(\Delta_U)$;
    for $t = 0, \ldots, T-1$ **do**
      $\tilde{\rho}_t \leftarrow \frac{p_{\phi^+}(G^+_{t+1} \mid G^+_t, t)}{p_{\phi^+_{\text{old}}}(G^+_{t+1} \mid G^+_t, t)}$;
    Compute L_{robust} using Eq. 4;
    Compute L_{forget} and L_{distill} following Alg. 1;
    $L_{\text{stage2}} \leftarrow L_{\text{forget}} + \lambda_{\text{distill}} L_{\text{distill}} + \lambda_{\text{robust}} L_{\text{robust}}$;
    $\phi \leftarrow \text{Update}(\phi, L_{\text{stage2}})$;
return ϕ

Overall Two-Stage Optimization. We use two stages because relearning is only meaningful after forgetting has been established. Otherwise, fine-tuning on undesired molecules would reinforce existing undesired behavior rather than recover forgotten behavior. In the first stage, RL-based forgetting and selective distillation are jointly optimized as $L_{\text{stage1}} = L_{\text{forget}} + \lambda_{\text{distill}} L_{\text{distill}}$, where λ_{distill} controls the strength of the distillation objective. This stage first reduces undesired generation while preserving retained generation behavior, producing the initial forgotten model. In the second stage, we further introduce the robust forgetting objective while continuing the first-stage objectives, giving $L_{\text{stage2}} = L_{\text{forget}} + \lambda_{\text{distill}} L_{\text{distill}} + \lambda_{\text{robust}} L_{\text{robust}}$, where λ_{robust} controls the strength of robust forgetting. Continuing L_{forget} and L_{distill} during the second stage maintains the achieved forgetting and preserved generation behavior while the model is further trained to resist subsequent relearning.

4 Experiments

4.1 Experimental Settings

In this section, we describe the task setup, baselines, ablation study design, and computational resources. Details of the datasets, such as their sources, descriptions, and preprocessing, as well as hyperparameter selection and performance metrics, are provided in Appendix C.

Task setup. Our primary task is to suppress the generation of toxic molecules, which we evaluate on the large-scale drug-like molecular dataset GEOM-Drug Axelrod & Gómez-Bombarelli (2022). Toxicity is identified using ADMET-AI Swanson et al. (2023), considering molecules predicted to exhibit acute toxicity, clinical toxicity, or carcinogenicity as undesired. To further examine whether forgetting preserves task-relevant generation capability, we additionally apply toxicity forgetting to a model trained for drug design Chen et al. (2026) and evaluate changes in its drug-relevant molecular properties after forgetting. We also evaluate whether ForgetU generalizes across different forgetting targets and molecular datasets. Specifically, we consider poor permeability Swanson et al. (2023) as another undesired molecular property on GEOM-Drug. Moreover, we evaluate ForgetU on QM9 Ramakrishnan et al. (2014), which contains substantially smaller molecules, and define fluorine-containing molecules as the forgetting target.

Baselines. We design baselines from a few complementary perspectives. Pretrained denotes the original model without forgetting, while Retained represents the model retrained from scratch using only desired molecules. For classical forget-set-based unlearning, we consider GA, which applies gradient ascent on undesired molecules, and GA+Retain, which additionally trains on retained molecules to preserve normal generation. As an alternative RL method, we use REINFORCE with the same reward design as ForgetU. We further consider Global KL, which applies global KL regularization toward the pretrained model to preserve its original generation behavior. For robustness to relearning, we compare with Smoothness, which uses sharpness-aware optimization to obtain a smoother forgetting solution and reduce recovery under subsequent fine-tuning.

Ablation Study. We conduct ablation studies to examine the contribution of the main components of ForgetU. We compare the full framework with: (1) Forgetting only, which uses only the RL-based forgetting objective; (2) single-stage training, which introduces the robust forgetting objective from the beginning rather than after the initial forgetting stage; (3) w/o selective distillation, which removes the distillation objective used to preserve retained generation behavior; (4) w/o robust forgetting, which removes the second-stage robust forgetting objective; (5) w/o continuous, where only the discrete generation components are optimized during forgetting; and (6) w/o discrete, where only the continuous coordinate component is optimized.

Evaluation. We evaluate ForgetU from three aspects: forgetting effectiveness, preservation of the original generation capability, and robustness to relearning. Forgetting effectiveness is measured by the Undesired generation Rate (UR), defined as the proportion of generated molecules satisfying $U(G) = 1$. To evaluate preservation, we report validity, uniqueness, novelty, and Fréchet ChemNet Distance (FCD) Preuer et al. (2018), together with task-relevant molecular properties when applicable. For robustness evaluation, we divide toxic molecules from the original pretraining data into two disjoint subsets. D_U^{train} is used only for simulated relearning during robust forgetting, while D_U^{test} is kept unseen during ForgetU training and used for held-out relearning evaluation. We further evaluate cross-source robustness using toxic molecules from an external dataset. We report Recovery@K at different fine-tuning steps to evaluate the robustness of forgetting.

Computational Resources. Our experiments were conducted using eight NVIDIA A100 GPUs, each with 80 GB of memory.

4.2 Results and Discussion

This section presents the main results and key ablation studies. Additional details and results are provided in Appendix D.

4.2.1 Main Results

As shown in Table 1, all forgetting methods reduce undesired generation relative to the pretrained model, but differ in their ability to preserve the original generation capability and resist subsequent recovery. Classical forget-set methods such as GA reduce undesired generation but noticeably affect generation quality, while GA+Retain improves preservation by incorporating retained molecules. REINFORCE further confirms that reward-based optimization can suppress undesired generation, but leads to larger changes in validity and FCD. Global KL better preserves the pretrained generation behavior, but also constrains forgetting, resulting in higher undesired generation rates than our method. Smoothness reduces recovery under subsequent fine-tuning, showing that robustness-aware optimization can make forgotten behavior harder to recover. In comparison, our method reduces the

undesired generation rate from 8.74% to 1.43% for toxicity and from 10.56% to 1.83% for permeability, while its validity, uniqueness, novelty, and FCD remain close to those of the pretrained model. It also shows the lowest recovery after subsequent fine-tuning on both tasks. These results show that our method can effectively suppress undesired generation while preserving the original generation capability and improving the robustness of forgetting.

Table 1: Baseline performance vs. our method. Best and Second-best results are highlighted.

Method	Validity (%) ↑	Uniqueness (%) ↑	Novelty (%) ↑	FCD ↓	UR (%) ↓	Recovery@20 (%) ↓
Toxic						
Pretrained	98.90 ± 0.26	100.00 ± 0.00	100.00 ± 0.00	–	8.74 ± 0.54	–
Retained	98.82 ± 0.31	100.00 ± 0.00	100.00 ± 0.00	5.38 ± 0.20	1.62 ± 0.11	–
GA	96.91 ± 0.58	99.95 ± 0.03	99.93 ± 0.04	6.47 ± 0.27	3.84 ± 0.34	5.18 ± 0.57
GA+Retain	98.19 ± 0.40	99.98 ± 0.02	99.96 ± 0.03	5.61 ± 0.22	3.63 ± 0.29	3.98 ± 0.34
REINFORCE	94.96 ± 0.46	99.97 ± 0.02	99.15 ± 0.03	7.73 ± 0.24	2.87 ± 0.27	3.91 ± 0.45
Global KL	98.74 ± 0.29	100.00 ± 0.00	100.00 ± 0.00	5.35 ± 0.19	1.64 ± 0.28	3.62 ± 0.42
Smoothness	98.51 ± 0.32	99.99 ± 0.01	99.97 ± 0.02	5.72 ± 0.20	1.71 ± 0.23	1.87 ± 0.14
Ours	**98.92 ± 0.30**	**100.00 ± 0.00**	**100.00 ± 0.00**	**5.31 ± 0.19**	**1.43 ± 0.05**	**1.52 ± 0.15**
Permeability						
Pretrained	98.90 ± 0.26	100.00 ± 0.00	100.00 ± 0.00	–	10.56 ± 0.99	–
Retained	98.78 ± 0.32	100.00 ± 0.00	100.00 ± 0.00	5.53 ± 0.21	2.13 ± 0.32	–
GA	95.97 ± 0.63	99.94 ± 0.04	99.92 ± 0.04	6.62 ± 0.29	3.47 ± 0.45	6.87 ± 0.64
GA+Retain	96.08 ± 0.42	100.00 ± 0.00	99.97 ± 0.03	5.69 ± 0.23	2.83 ± 0.37	4.91 ± 0.54
REINFORCE	94.24 ± 0.49	99.97 ± 0.03	99.95 ± 0.03	8.81 ± 0.25	2.42 ± 0.34	4.35 ± 0.50
Global KL	98.57 ± 0.31	100.00 ± 0.00	100.00 ± 0.00	5.20 ± 0.23	2.71 ± 0.36	3.98 ± 0.46
Smoothness	98.46 ± 0.34	99.99 ± 0.01	99.97 ± 0.02	5.35 ± 0.21	2.27 ± 0.21	3.01 ± 0.29
Ours	**98.82 ± 0.31**	99.99 ± 0.01	**100.00 ± 0.00**	**5.26 ± 0.14**	**1.83 ± 0.19**	**2.49 ± 0.22**

Note: All results are evaluated based on 1000 generated molecules and are averaged over three runs with different random seeds.

We further evaluate ForgetU using 10,000 generated molecules to examine whether its performance remains stable at a larger generation scale. As shown in Table 2, the results are highly consistent with those obtained using 1,000 generated molecules. For toxicity, the undesired generation rate decreases from 9.02% to 1.70%, while for permeability it decreases from 10.76% to 2.13%. Validity, uniqueness, and novelty remain nearly unchanged, and Recovery@20 remains low at 1.94% and 2.69% for the two tasks, respectively. These results further confirm that the forgetting, preservation, and robustness performance of ForgetU remains stable under large-scale generation.

Table 2: Large-scale generation results.

Method	Validity (%) ↑	Uniqueness (%) ↑	Novelty (%) ↑	FCD ↓	UR (%) ↓	Recovery@20 (%) ↓
Toxic						
Pretrained	98.90	99.91	100.00	–	9.02	–
Ours	**98.90**	**99.88**	**100.00**	**5.35**	**1.70**	**1.94**
Permeability						
Pretrained	98.90	99.91	100.00	–	10.76	–
Ours	**98.85**	**99.86**	**100.00**	**5.37**	**2.13**	**2.69**

Note: The results are evaluated based on 10,000 generated molecules.

We also evaluate ForgetU on a model trained for drug design to examine whether forgetting preserves its task-relevant molecular properties. As shown in Fig. 3, the distributions of QED, SAS, and binding affinity remain highly similar before and after forgetting, with only minor changes in their overall shapes and central tendencies. These results indicate that ForgetU largely preserves the drug-relevant generation capability of the pretrained model while suppressing undesired generation.

4.2.2 Ablation Study

Ablation results in Table 3 further demonstrate the effectiveness of each component. For forgetting effectiveness, the undesired generation rate is the primary indicator. Forgetting Only already reduces this rate to 1.50% and 1.85% across two tasks, showing that the RL-based forgetting objective is effective by itself. However, its lower validity and higher FCD indicate degradation of the original generation capability. Similarly, removing selective distillation leaves the undesired generation rate nearly unchanged but reduces validity and increases FCD, confirming that selective distillation

mainly serves to preserve the pretrained generation behavior. Removing robust forgetting has little effect on the initial undesired generation rate or generation quality, while recovery after fine-tuning increases from 1.52% to 5.48% for toxicity and from 2.49% to 4.98% for permeability, showing its contribution to robustness. Single-stage optimization also results in higher undesired generation and recovery, supporting the use of staged optimization after forgetting has first been established. Finally, optimizing only the continuous or only the discrete component results in higher undesired generation and recovery on both tasks, indicating that neither component alone is sufficient and that joint continuous-discrete optimization is necessary. Overall, the ablation results show that the forgetting objective, selective distillation, and robust forgetting mainly contribute to forgetting effectiveness, preservation, and robustness, respectively, while their combination provides a better balance across these objectives.

Table 3: Ablation study. Best and Second-best results are highlighted.

Method	Validity (%) ↑	Uniqueness (%) ↑	Novelty (%) ↑	FCD ↓	UR (%) ↓	Recovery@20 (%) ↓
Toxic						
Forgetting Only	96.84 ± 0.32	99.95 ± 0.03	99.93 ± 0.04	8.34 ± 0.25	1.50 ± 0.03	5.86 ± 0.29
Single-stage	98.63 ± 0.23	99.99 ± 0.01	99.98 ± 0.02	5.47 ± 0.20	5.58 ± 0.39	6.88 ± 0.18
w/o Selective Distillation	96.97 ± 0.47	99.96 ± 0.03	99.92 ± 0.04	6.21 ± 0.24	1.45 ± 0.07	1.66 ± 0.12
w/o Robust Forgetting	98.81 ± 0.29	100.00 ± 0.00	100.00 ± 0.00	5.38 ± 0.09	1.47 ± 0.06	5.48 ± 0.29
w/o Continuous	98.84 ± 0.29	100.00 ± 0.00	100.00 ± 0.00	5.43 ± 0.10	6.77 ± 0.09	7.76 ± 0.17
w/o Discrete	98.54 ± 0.34	100.00 ± 0.00	100.00 ± 0.00	5.69 ± 0.11	4.98 ± 0.23	6.01 ± 0.25
Ours	$\mathbf{98.92 \pm 0.30}$	$\mathbf{100.00 \pm 0.00}$	$\mathbf{100.00 \pm 0.00}$	$\mathbf{5.31 \pm 0.19}$	$\mathbf{1.43 \pm 0.05}$	$\mathbf{1.52 \pm 0.15}$
Permeability						
Forgetting Only	96.24 ± 0.46	99.95 ± 0.03	99.92 ± 0.03	8.76 ± 0.72	1.85 ± 0.11	4.62 ± 0.39
Single-stage	98.52 ± 0.15	99.99 ± 0.01	99.98 ± 0.01	5.44 ± 0.18	6.01 ± 0.20	7.95 ± 0.25
w/o Selective Distillation	97.03 ± 0.51	99.95 ± 0.03	99.91 ± 0.04	6.48 ± 0.26	1.86 ± 0.07	2.06 ± 0.17
w/o Robust Forgetting	98.75 ± 0.36	99.98 ± 0.01	100.00 ± 0.00	5.41 ± 0.12	1.86 ± 0.09	4.98 ± 0.23
w/o Continuous	98.85 ± 0.23	100.00 ± 0.00	100.00 ± 0.00	5.40 ± 0.17	8.75 ± 0.14	10.23 ± 0.34
w/o Discrete	98.71 ± 0.28	100.00 ± 0.00	100.00 ± 0.00	5.61 ± 0.16	6.07 ± 0.41	8.94 ± 0.38
Ours	$\mathbf{98.82 \pm 0.31}$	$\mathbf{99.99 \pm 0.01}$	$\mathbf{100.00 \pm 0.00}$	$\mathbf{5.26 \pm 0.14}$	$\mathbf{1.83 \pm 0.19}$	$\mathbf{2.49 \pm 0.22}$

Note: All results are evaluated based on 1000 generated molecules and are averaged over three runs with different random seeds.

5 Conclusion

In this work, we introduced ForgetU, a two-stage RL-based robust forgetting framework for pretrained continuous-discrete flow matching models in 3D molecular generation. Across multiple molecular datasets and forgetting targets, our results show that ForgetU can effectively reduce undesired generation while preserving the original generation capability and making forgotten behaviors more difficult to recover through subsequent fine-tuning.

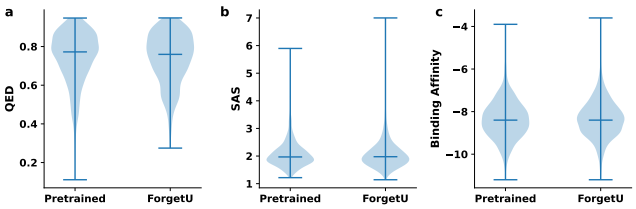

Figure 3: Preservation of drug-relevant molecular properties after forgetting. Distributions of (a) QED, (b) SAS, and (c) binding affinity for 1,000 molecules generated by the pretrained model and ForgetU.

Significance: ForgetU enables pretrained molecular generative models to suppress undesired generation without full retraining, providing a practical approach to reducing toxic, harmful, or otherwise restricted molecular outputs and supporting safer and more responsible model release.

Limitations and Future Work: The current study focuses on continuous-discrete flow matching models for 3D molecular generation. A natural direction for future work is to extend the proposed framework to other molecular generative models, such as diffusion models.

AI USE STATEMENT

In this work, we used generative AI tools to polish the manuscript and improve its English expression, grammar, and readability. Generative AI was not involved in the research process or the formation of scientific conclusions. All AI-assisted edits were reviewed by the authors.

REPRODUCIBILITY STATEMENT

To improve the reproducibility of this work, we describe the proposed method and experimental settings in the main text and Appendix C, and provide implementation details, dataset settings, hyperparameters, baseline settings, and evaluation procedures. The main experiments are repeated with multiple random seeds, and the results are reported. The code will be made publicly available upon acceptance.

REFERENCES

Simon Axelrod and Rafael Gómez-Bombarelli. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9:185, 2022. doi: 10.1038/s41597-022-01288-4.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 141–159, 2021. doi: 10.1109/SP40001.2021.00019.

Lianghong Chen, Yan Yi Li, Gen Zhou, Yuxi Long, Ganlin Feng, Mike Domaratzki, and Pingzhao Hu. Temporal selective exploration for reinforcement learning-guided continuous-discrete flow matching in 3D molecular design. In *Advances in Neural Information Processing Systems*, 2026.

Jingpu Cheng, Ping Liu, Qianxiao Li, and Chi Zhang. Machine unlearning under retain–forget entanglement. In *International Conference on Learning Representations*, 2026.

D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes, Volume I: Elementary Theory and Methods*. Springer, 2 edition, 2003.

Ian Dunn and David R. Koes. Flowmol3: flow matching for 3d de novo small-molecule generation. *Digital Discovery*, 5(5):2052–2066, 2026. doi: 10.1039/D5DD00363F.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *arXiv preprint arXiv:2310.12508*, 2023. doi: 10.48550/arXiv.2310.12508. Accepted by ICLR 2024 as a Spotlight paper.

Chongyu Fan, Jinghan Jia, Yihua Zhang, Anil Ramakrishna, Mingyi Hong, and Sijia Liu. Towards llm unlearning resilient to relearning attacks: A sharpness-aware minimization perspective and beyond. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 15762–15778. PMLR, 2025.

Mark I. Freidlin and Alexander D. Wentzell. *Random Perturbations of Dynamical Systems*. Grundlehren der Mathematischen Wissenschaften. Springer, 2012. URL https://doi.org/10.1007/978-3-642-25847-3.

Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2426–2436, 2023.

Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified concept editing in diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5111–5120, 2024.

Hongcheng Gao, Tianyu Pang, Chao Du, Taihang Hu, Zhijie Deng, and Min Lin. Meta-unlearning on diffusion models: Preventing relearning unlearned concepts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2131–2141, 2025.

F. Peter Guengerich. Mechanisms of drug toxicity and relevance to pharmaceutical development. *Drug Metabolism and Pharmacokinetics*, 26(1):3–14, 2011. doi: 10.2133/dmpk.DMPK-10-RV-062.

Emiel Hoogeboom, Víctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8867–8887. PMLR, 2022.

Shengyuan Hu, Yiwei Fu, Zhiwei Steven Wu, and Virginia Smith. Unlearning or obfuscating? jogging the memory of unlearned llms via benign relearning. In *International Conference on Learning Representations*, 2025.

Hoki Kim, Keonwoo Kim, Sungwon Chae, and Sangwon Yoon. Unlearning-aware minimization. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-3138.

Solomon Kullback and Richard A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. doi: 10.1214/aoms/1177729694.

Nupur Kumari, Bingliang Zhang, Sheng-Yu Wang, Eli Shechtman, Richard Zhang, and Jun-Yan Zhu. Ablating concepts in text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22691–22702, 2023.

Daniel C. Liebler and F. Peter Guengerich. Elucidating mechanisms of drug-induced toxicity. *Nature Reviews Drug Discovery*, 4:410–420, 2005. doi: 10.1038/nrd1720.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2023a. doi: 10.48550/arXiv.2210.02747.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023b. URL https://arxiv.org/abs/2210.02747.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. Quark: Controllable text generation with reinforced unlearning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-2001.

Ioannis Mavrothalassitis, Pol Puigdemont, Noam Levi, and Volkan Cevher. Ascent fails to forget. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1510.

Filipp Nikitin, Ian Dunn, David Ryan Koes, and Olexandr Isayev. Geom-drugs revisited: toward more chemically accurate benchmarks for 3d molecule generation. *Digital Discovery*, 4(11):3282–3291, 2025. doi: 10.1039/D5DD00206K.

Jiadong Pan, Hongcheng Gao, Zongyu Wu, Taihang Hu, Li Su, Qingming Huang, and Liang Li. Leveraging catastrophic forgetting to develop safe diffusion models against malicious finetuning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3658.

Matthew Perry, Michael Sanguinetti, and John Mitcheson. Revealing the structural basis of action of herg potassium channel activators and blockers. *The Journal of Physiology*, 588(17):3157–3167, 2010. doi: 10.1113/jphysiol.2010.194670.

Jim Pitman and Marc Yor. A guide to brownian motion and related stochastic processes. *arXiv preprint arXiv:1802.09679*, 2018. URL https://doi.org/10.48550/arXiv.1802.09679.

Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter Klambauer. Fréchet chemnet distance: A metric for generative models for molecules in drug discovery. *Journal of Chemical Information and Modeling*, 58(9):1736–1741, 2018. doi: 10.1021/acs.jcim.8b00234.

Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1:140022, 2014. doi: 10.1038/sdata.2014.22.

Kaveh Safavigerdini, Juan Mogollon, Amirreza Daghighi, and Kannappan Palaniappan. Machine unlearning in computer vision: Survey of methods from discriminative to generative foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 8538–8549, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://doi.org/10.48550/arXiv.1707.06347.

Vinith Menon Suriyakumar, Rohan Alur, Ayush Sekhari, Manish Raghavan, and Ashia C. Wilson. Unstable unlearning: The hidden risk of concept resurgence in diffusion models. *Transactions on Machine Learning Research*, 2026.

Kyle Swanson, Parker Walther, Jeremy Leitz, Souhrid Mukherjee, Joseph C. Wu, Rabindra V. Shivnaraine, and James Zou. Admet-ai: A machine learning admet platform for evaluation of large-scale chemical libraries. *bioRxiv*, 2023. URL https://doi.org/10.1101/2023.12.28.573531.

Fabio Urbina, Filippa Lentzos, Cédric Invernizzi, and Sean Ekins. Dual use of artificial-intelligence-powered drug discovery. *Nature Machine Intelligence*, 4(3):189–191, 2022. doi: 10.1038/s42256-022-00465-9.

Changsheng Wang, Yihua Zhang, Jinghan Jia, Parikshit Ram, Dennis Wei, Yuguang Yao, Soumyadeep Pal, Nathalie Baracaldo, and Sijia Liu. Invariance makes llm unlearning resilient even to unanticipated downstream fine-tuning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 65464–65479. PMLR, 2025.

Kunyu Wang, Jon Paul Janet, and Alessandro Tibo. Controllable molecular generation with fine-tuned flow-matching model. *Communications Chemistry*, 9:291, 2026. doi: 10.1038/s42004-026-02188-z.

Ye Wang, Honggang Zhao, Simone Sciabola, and Wenlu Wang. cmolgpt: A conditional generative pre-trained transformer for target-specific de novo molecular generation. *Molecules*, 28(11):4430, 2023. doi: 10.3390/molecules28114430.

Minkai Xu, Alexander S. Powers, Ron O. Dror, Stefano Ermon, and Jure Leskovec. Geometric latent diffusion models for 3d molecule generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38592–38610. PMLR, 2023.

Sangyeon Yoon, Hyesoo Hong, Wonje Jeung, and Albert No. Rethinking benign relearning: Syntax as the hidden driver of unlearning failures. In *International Conference on Learning Representations*, 2026.

Efstratios Zaradoukas, Bardh Prenkaj, and Gjergji Kasneci. Reinforcement unlearning via group relative policy optimization. In *International Conference on Learning Representations*, 2026.

Chenlong Zhang, Zhuoran Jin, Hongbang Yuan, Jiaheng Wei, Tong Zhou, Kang Liu, Jun Zhao, and Yubo Chen. Rule: Reinforcement unlearning achieves forget-retain pareto optimality. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1315.

Gong Zhang, Kai Wang, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Forget-me-not: Learning to forget in text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1755–1764, 2024.

APPENDIX

CONTENTS

A Design Choices and Rationale **14**

A.1 Choice of Optimization Framework 14

B Additional Methodology Details **14**

B.1 Continuous-Discrete Flow Matching Construction 14

B.2 Discrete CTMC Transition Derivation 15

B.3 Marginal Preservation of the Stochastic Flow 15

C Additional Experimental Details **17**

C.1 Data Sources and Descriptions 17

C.2 Data Preprocessing and Splitting 17

C.3 Baseline Implementation 18

C.4 Hyperparameters for Flow Matching Model Training 18

C.5 Forgetting Hyperparameters 18

C.6 Performance Metrics 18

D Additional Experiments, Results and Discussions **20**

D.1 Robustness of forgetting under different K 20

D.2 Generalization Experiments on QM9 20

A Design Choices and Rationale

A.1 Choice of Optimization Framework

We adopt a PPO-style RL framework because our forgetting objective is defined by an absolute binary reward: undesired trajectories receive a negative reward, while retained trajectories receive zero reward. This reward design is more naturally aligned with PPO than with group-relative optimization.

GRPO is primarily designed for settings where multiple outputs are generated under a shared input condition, such as a prompt or question, and their advantages are determined by relative rewards within each group. Although it can in principle be adapted to unconditional generation by treating samples as belonging to the same group, such grouping is artificial in our setting because independently generated molecular trajectories do not share a meaningful conditioning context. More importantly, group-wise reward normalization would make the optimization signal depend on the rewards of other trajectories in the same group. Consequently, the effective advantage of a molecule could vary with group composition rather than being determined solely by whether the molecule is undesired.

In contrast, our PPO-style formulation directly preserves the semantics of the binary forgetting reward. Each trajectory is optimized according to its own terminal reward, while the clipped likelihood ratio constrains the change from the old model during each update. This avoids introducing relative preferences among independently generated molecules and keeps the optimization signal directly aligned with the undesired criterion. Therefore, PPO provides a simpler and better-aligned optimization framework for our unconditional molecular forgetting setting.

B Additional Methodology Details

B.1 Continuous-Discrete Flow Matching Construction

The main text describes the continuous-discrete generation dynamics. Here, we provide the corresponding training construction and define the original flow matching objective $\mathcal{L}_{\mathrm{FM}}$ used for pretraining and simulated relearning.

Let $G_1 = (X_1, A_1, C_1, E_1)$ denote a molecule sampled from the data distribution and let $G_0 = (X_0, A_0, C_0, E_0)$ denote an initial state sampled from the prior, where X_0 follows an isotropic Gaussian distribution and the discrete variables are initialized with the mask state. We sample $t \sim \mathcal{U}(0, 1)$ and construct the intermediate molecular state G_t using the corresponding continuous and discrete probability paths.

For the continuous coordinates, the interpolation is defined as $X_t = (1 - \alpha_t)X_0 + \alpha_t X_1$, where α_t is the interpolation schedule used in the main text. The model predicts the terminal coordinates $\hat{X}_1^{\psi}$ from (G_t, t). The continuous training objective is $\mathcal{L}_X = \mathbb{E}\Big[w(t)\|\hat{X}_1^{\psi}(G_t, t) - X_1\|^2\Big]$, where $w(t)$ controls the relative contribution of different timesteps.

For each discrete modality $m \in \{A, C, E\}$, the conditional probability path for an individual variable moves probability mass from the mask state M toward its terminal value m_1 according to $q_t(m_t \mid m_1) = (1 - \alpha_t)\delta(m_t = M) + \alpha_t\delta(m_t = m_1)$. The model predicts the corresponding endpoint distribution from (G_t, t). Training uses the categorical likelihood of the terminal value for variables that remain masked at time t. Specifically, for modality m, we use $\mathcal{L}_m = \mathbb{E}\big[-\sum_{i\in\mathcal{I}_m} \mathbf{1}(m_{t,i} = M) \log \pi_{\psi}(m_{1,i} \mid G_t, t)\big]$, where $\mathcal{I}_m$ indexes the variables associated with modality m.

The complete continuous-discrete flow matching objective is $\mathcal{L}_{\mathrm{FM}} = \lambda_X\mathcal{L}_X + \lambda_A\mathcal{L}_A + \lambda_C\mathcal{L}_C + \lambda_E\mathcal{L}_E$, where the coefficients balance the contributions of the four molecular modalities. Thus, one training iteration samples (G_0, G_1) and t, constructs G_t, predicts the terminal continuous and discrete states, and jointly optimizes all modalities using $\mathcal{L}_{\mathrm{FM}}$. The same objective is used for the K-step simulated relearning procedure in Stage 2, with the undesired molecule set D_U serving as the training data.

B.2 Discrete CTMC Transition Derivation

The main text gives the discrete transition likelihood in compact form. Here, we further describe the stochastic transition decisions used during generation and their replay during RL optimization.

Consider an individual discrete variable in one of the modalities A, C, or E. If the variable is masked at time t, its transition from t to $s > t$ consists of two stochastic decisions. First, an unmasking indicator $u \in \{0, 1\}$ is sampled according to the unmasking probability p_{unmask} defined in Section 3.1. If $u = 1$, an endpoint value y is subsequently sampled from the model prediction $\pi_\psi(y \mid G_t, t)$. The corresponding transition log-likelihood is $\log p_{\text{Bern}}(u; p_{\text{unmask}}) + u \log \pi_\psi(y \mid G_t, t)$. When $u = 0$, the variable remains masked and no endpoint value is sampled.

If the variable is already unmasked at time t, let $v \in \{0, 1\}$ denote whether it is remasked during the transition. Since the remasking decision follows the probability p_{mask} defined in Section 3.1, its contribution to the transition log-likelihood is $\log p_{\text{Bern}}(v; p_{\text{mask}})$. When no remasking occurs, the current categorical value is retained.

At the final transition to $t = 1$, remasking is disabled. Under the linear interpolation schedule used in our experiments, the unmasking probability reaches one after clipping, so all remaining masked variables are revealed. The transition likelihood is evaluated using the same final-step rule.

The discrete transition log-likelihood is obtained by accumulating these contributions over the atom types, formal charges, and bond types. Together with the continuous Gaussian transition likelihood described in Section 3.2, this gives the complete transition log-likelihood $\log p_\psi(G_s \mid G_t)$ used for RL optimization.

During trajectory generation, the sampled discrete transition decisions are recorded, including the masking or unmasking outcomes and, when unmasking occurs, the sampled endpoint values. During PPO replay, these same transition outcomes are evaluated under both models rather than resampled. In Stage 1, trajectories generated by ϕ_{old} are therefore evaluated using both $p_{\phi_{\text{old}}}(G_{t+1} \mid G_t, t)$ and $p_\phi(G_{t+1} \mid G_t, t)$. Similarly, in Stage 2, transitions generated by ϕ^+_{old} are evaluated using both $p_{\phi^+_{\text{old}}}(G^+_{t+1} \mid G^+_t, t)$ and $p_{\phi^+}(G^+_{t+1} \mid G^+_t, t)$. Replaying the same sampled transition decisions ensures that the likelihood ratios compare the two models on identical stochastic transitions.

B.3 Marginal Preservation of the Stochastic Flow

In Section 3.2, we convert the deterministic continuous flow into a stochastic process in order to define transition likelihoods for RL-based robust forgetting. Here, we show that the corresponding continuous-time SDE preserves the marginal distribution of the original flow.

Marginal-preserving continuous dynamics. Let $p_{\psi,t}(G)$ denote the marginal distribution induced by the original continuous-discrete flow at time t, where $G = (X, A, C, E)$. Since the discrete variables follow the same CTMC dynamics in both the original and stochastic processes, it is sufficient to show that introducing stochasticity into the continuous coordinates does not change the continuous probability flux.

Let $\mathcal{Q}^*_t$ denote the forward operator associated with the discrete CTMC transitions. The joint marginal distribution of the original process satisfies

$$\frac{\partial p_{\psi,t}(G)}{\partial t} = -\nabla_X \cdot \left(v_\psi(G, t)\, p_{\psi,t}(G)\right) + \mathcal{Q}^*_t p_{\psi,t}(G). \tag{B.1}$$

Consider the stochastic continuous dynamics

$$dX_t = \left[v_\psi(G_t, t) + \frac{1}{2} g_t^2 S_\psi(G_t, t)\right] dt + g_t\, dW_t, \tag{B.2}$$

where

$$S_\psi(G_t, t) = \nabla_{X_t} \log p_{\psi,t}(G_t). \tag{B.3}$$

The discrete CTMC dynamics remain unchanged.

Let $q_{\psi,t}(G)$ denote the joint marginal density induced by this stochastic process. Its forward equation is

$$\frac{\partial q_{\psi,t}(G)}{\partial t} = -\nabla_X \cdot \left[\left(v_\psi + \frac{1}{2} g_t^2 S_\psi \right) q_{\psi,t} \right] + \frac{1}{2} g_t^2 \Delta_X q_{\psi,t} + \mathcal{Q}_t^* q_{\psi,t}. \tag{B.4}$$

We now verify that $p_{\psi,t}$ also satisfies Eq. equation B.4. From Eq. equation B.3,

$$S_\psi(G,t) p_{\psi,t}(G) = \nabla_X p_{\psi,t}(G), \tag{B.5}$$

and therefore

$$\nabla_X \cdot (S_\psi p_{\psi,t}) = \Delta_X p_{\psi,t}. \tag{B.6}$$

Substituting $q_{\psi,t} = p_{\psi,t}$ into Eq. equation B.4 and applying Eq. equation B.6 gives

$$\frac{\partial p_{\psi,t}}{\partial t} = -\nabla_X \cdot (v_\psi p_{\psi,t}) - \frac{1}{2} g_t^2 \nabla_X \cdot (S_\psi p_{\psi,t}) + \frac{1}{2} g_t^2 \Delta_X p_{\psi,t} + \mathcal{Q}_t^* p_{\psi,t} \tag{B.7}$$

$$= -\nabla_X \cdot (v_\psi p_{\psi,t}) + \mathcal{Q}_t^* p_{\psi,t}, \tag{B.8}$$

because the two diffusion terms cancel exactly. This is identical to the forward equation of the original process in Eq. equation B.1. Therefore, provided that the two processes start from the same initial distribution and the corresponding forward equation admits a unique solution,

$$q_{\psi,t}(G) = p_{\psi,t}(G), \qquad \text{for all } t \text{ with } \alpha_t \in (0,1). \tag{B.9}$$

Thus, Eq. equation B.2 introduces stochastic trajectories while preserving the time-dependent marginals of the original continuous-discrete flow.

Score induced by endpoint prediction. We next show how the score used in Section 3.2 follows from the endpoint parameterization of the flow model. For the continuous interpolation,

$$X_t = \alpha_t X_1 + (1-\alpha_t) X_0, \qquad X_0 \sim \mathcal{N}(0, I), \tag{B.10}$$

the conditional distribution of X_t given X_1 is

$$p(X_t \mid X_1) = \mathcal{N}\left(\alpha_t X_1, (1-\alpha_t)^2 I\right). \tag{B.11}$$

Hence,

$$\nabla_{X_t} \log p(X_t \mid X_1) = -\frac{X_t - \alpha_t X_1}{(1-\alpha_t)^2}. \tag{B.12}$$

Let $D_t = (A_t, C_t, E_t)$ denote the discrete components of the molecular state. When the model is consistent with the interpolation probability path, $p_{\psi,t}$ is induced by $q_t(G_t \mid G_1)$, which factorizes as

$$q_t(G_t \mid G_1) = q_t^X(X_t \mid X_1)\, q_t^D(D_t \mid D_1), \tag{B.13}$$

where q_t^D does not depend on X_t. Using the conditional-score identity,

$$\nabla_{X_t} \log p_{\psi,t}(G_t) = \mathbb{E}\left[\nabla_{X_t} \log q_t(G_t \mid G_1) \mid G_t\right] \tag{B.14}$$

$$= \mathbb{E}\left[\nabla_{X_t} \log q_t^X(X_t \mid X_1) \mid G_t\right] \tag{B.15}$$

$$= -\frac{X_t - \alpha_t \mathbb{E}[X_1 \mid G_t]}{(1-\alpha_t)^2}. \tag{B.16}$$

Therefore, under this probability-path consistency condition, when the endpoint predictor satisfies

$$\hat{X}_1^\psi(G_t, t) = \mathbb{E}[X_1 \mid G_t], \tag{B.17}$$

the score estimator used in the main text satisfies

$$S_\psi(G_t, t) = -\frac{X_t - \alpha_t \hat{X}_1^\psi}{(1-\alpha_t)^2} = \nabla_{X_t} \log p_{\psi,t}(G_t). \tag{B.18}$$

Hence, the endpoint parameterization provides exactly the score required by the marginal-preserving SDE when the endpoint predictor is exact.

Connection to the finite-step transition. For the stochasticity schedule used in our transition construction, define

$$g_t^2 = \sigma_x^2 \alpha_t' \frac{1-\alpha_t}{\alpha_t}. \tag{B.19}$$

The integrated diffusion variance between t and $s > t$ is

$$\int_t^s g_u^2 \, du = \sigma_x^2 \int_t^s \alpha_u' \frac{1-\alpha_u}{\alpha_u} \, du \tag{B.20}$$

$$= \sigma_x^2 \int_{\alpha_t}^{\alpha_s} \left(\frac{1}{\alpha} - 1 \right) d\alpha \tag{B.21}$$

$$= \sigma_x^2 \left[\log \frac{\alpha_s}{\alpha_t} - (\alpha_s - \alpha_t) \right] \tag{B.22}$$

$$= V_{t,s}, \tag{B.23}$$

which is exactly the integrated variance defined in Section 3.2.

Using this integrated variance and freezing the drift and score over the interval $[t, s]$ gives the finite-step approximation

$$X_s \approx X_t + (s-t) v_\psi(G_t, t) + \frac{1}{2} V_{t,s} S_\psi(G_t, t) + \sqrt{V_{t,s}} \, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{B.24}$$

With the deterministic flow update $\widetilde{X}_s = X_t + (s-t) v_\psi(G_t, t)$, this yields

$$\mu_{\psi, t \to s} = \widetilde{X}_s + \frac{1}{2} V_{t,s} S_\psi(G_t, t), \tag{B.25}$$

and consequently the Gaussian transition used in the main text,

$$p_\psi(X_s \mid G_t) = \mathcal{N}\left(X_s; \mu_{\psi, t \to s}, V_{t,s} I\right). \tag{B.26}$$

Conclusion. The continuous-time SDE in Eq. equation B.2 is exactly marginal-preserving when S_ψ equals the marginal score. Under the endpoint parameterization, this score is obtained from $\hat{X}_1^\psi$ in the ideal prediction limit. The transition likelihood used in our method is the finite-step discretization of this marginal-preserving process. Therefore, marginal preservation is exact for the underlying continuous-time construction, while the implemented Gaussian transition inherits it up to endpoint-prediction and numerical-discretization error.

C Additional Experimental Details

C.1 Data Sources and Descriptions

Table C.1 provides the data sources and brief descriptions of the QM9 and GEOM Drugs datasets.

C.2 Data Preprocessing and Splitting

Processing. We use the same preprocessing procedure for QM9 and GEOM-Drugs to obtain valid 3D molecular structures for pretraining the flow matching models. For each molecule, we extract the atomic coordinates, atom types, formal charges, and bond information available in the raw data. Molecules that cannot be successfully parsed or processed are removed. This includes molecules with missing or invalid 3D coordinates, chemically invalid structures, or processing failures encountered during data cleaning. After preprocessing, the resulting datasets contain only chemically valid and well-defined molecular structures.

For GEOM-Drugs, a single molecule may be associated with multiple 3D conformations corresponding to different low-energy geometries of the same molecular graph. We treat these conformations as individual samples because they capture structural variation while preserving the same chemical identity. When multiple conformations are available for a molecule, only a subset may be retained to control the overall size of the dataset. This allows the model to learn a distribution over 3D molecular geometries rather than representing each molecule using only a single fixed

Table C.1: Key characteristics of the QM9 and GEOM Drugs datasets.

Characteristic	QM9	GEOM Drugs
Dataset description	Three-dimensional small organic molecules derived from the GDB-17 molecular space	Large-scale three-dimensional dataset of drug-like molecules
Dataset size	About 134K molecules	About 304K molecules
Available properties	Molecular energies, dipole moments, polarizabilities, and additional physicochemical properties	N/A
Included atom types	C, N, O, F, H	C, H, N, O, F, P, S, Cl, Br, I
Valency and charge states	Mostly simple and well-defined valency patterns	More diverse valency and formal charge configurations across atom types
Molecular size	Small molecules with at most 9 heavy atoms	Larger molecules with greater structural complexity
Dataset complexity	Relatively compact and structurally simple	More heterogeneous and structurally diverse
Data sources	`http://deepchem.io.s3-website-us-west-1.amazonaws.com/datasets/gdb9.tar.gz`	`https://bits.csb.pitt.edu/files/geom_raw/`

conformation. For QM9, molecules are directly loaded from the original SDF files, and explicit hydrogen atoms are retained. Molecules marked as uncharacterized in the official QM9 filtering list are removed before further processing.

Splitting. We randomly shuffle the datasets at the molecule level and divide them into training, validation, and test sets using an 80%, 10%, and 10% split, respectively. For GEOM-Drugs, all conformations corresponding to the same molecule are assigned to the same subset to prevent information leakage across the training, validation, and test sets.

C.3 Baseline Implementation

All baselines use the same continuous-discrete flow matching backbone, undesired criterion, data splits, and evaluation protocol as ForgetU. Post-training baselines are initialized from the same pretrained checkpoint, and comparable optimization budgets are used whenever applicable. For GA+Retain, retained molecules are sampled from the desired portion of the original training data. For REINFORCE, the same binary reward as ForgetU is used. Global KL uses the same PPO-based forgetting objective as ForgetU, with selective distillation replaced by global KL regularization. Smoothness is initialized from the same Stage-1 forgotten model as the robust forgetting stage and uses sharpness-aware optimization instead of simulated relearning.

C.4 Hyperparameters for Flow Matching Model Training

We separately pretrain flow matching models on the GEOM-Drugs and QM9 datasets. The corresponding training configurations are summarized in Table C.2.

C.5 Forgetting Hyperparameters

The hyperparameter search space and the final selected configurations are summarized in Tables C.4 and C.3, respectively.

C.6 Performance Metrics

We provide detailed definitions of the key evaluation metrics used in our experiments below.

Validity. Validity measures the proportion of generated samples that can be successfully converted into chemically valid molecules. Generated molecular graphs are reconstructed as RDKit molecules

Table C.2: Hyperparameters used to train the flow matching models.

Parameter	GEOM-Drugs	QM9
Training configuration		
Batch size	12	128
Number of workers	6	8
Maximum epochs	20	20
Learning rate	1×10^{-4}	2.5×10^{-4}
Weight decay	1×10^{-12}	1×10^{-12}
Gradient accumulation	1	1
Maximum edges	50,000	200,000
Model architecture		
Hidden scalar dimension	256	256
Edge feature dimension	128	128
Vector channels	32	16
Number of molecule updates	6	8
Message GVP layers	3	3
Update GVP layers	3	3
Maximum RBF distance	10	14
RBF dimension	32	32
Time embedding dimension	64	64
Data processing		
Distortion probability	0.2	0.2
Fake atom probability	0.3	0.3
Self conditioning	True	True

Table C.3: Hyperparameter configurations used for ForgetU post-training.

Hyperparameter	GEOM-Drugs	QM9
Rollout batch size	64	128
Sampling steps	250	250
SDE noise scale	0.05	0.05
Post-training learning rate	1×10^{-5}	2×10^{-5}
PPO clipping parameter	5×10^{-4}	3×10^{-4}
Gradient clipping norm	0.5	0.5
Selective distillation weight	0.1	0.1
Robust forgetting weight	1.0	1.0
Simulated relearning steps	10	10
Simulated relearning learning rate	1×10^{-4}	2.5×10^{-4}

and processed using standard sanitization procedures. Let $\mathcal{M}_{\text{gen}}$ denote all generated samples and $\mathcal{M}_{\text{valid}}$ denote the subset that can be successfully reconstructed and sanitized. Validity is computed as $\text{Validity} = |\mathcal{M}_{\text{valid}}|/|\mathcal{M}_{\text{gen}}| \times 100\%$. A higher value indicates that the model produces a larger proportion of chemically valid molecular structures.

Uniqueness. Uniqueness measures the diversity of valid generated molecules. Duplicate molecules are identified using canonical SMILES. Let $\mathcal{M}_{\text{unique}}$ denote the set of distinct molecules in $\mathcal{M}_{\text{valid}}$. We compute $\text{Uniqueness} = |\mathcal{M}_{\text{unique}}|/|\mathcal{M}_{\text{valid}}| \times 100\%$. A higher uniqueness indicates less duplication among generated molecules.

Novelty. Novelty measures the proportion of unique valid molecules that are not contained in the original pretraining dataset. Molecular identity is compared using canonical SMILES. Let $\mathcal{M}_{\text{train}}$

Table C.4: Hyperparameter search space for ForgetU post-training.

Hyperparameter	Search Values
Rollout batch size	16, 32, 64, 128
Sampling steps	100, 150, 200, 250
SDE noise scale	0.01, 0.03, 0.05, 0.08, 0.1
Post-training learning rate	5×10^{-6}, 1×10^{-5}, 2×10^{-5}, 5×10^{-5}, 1×10^{-4}
PPO clipping parameter	1×10^{-4}, 2×10^{-4}, 3×10^{-4}, 5×10^{-4}, 7×10^{-4}, 1×10^{-3}
Gradient clipping norm	0.1, 0.3, 0.5, 1.0
Selective distillation weight	0.01, 0.05, 0.1, 0.2, 0.5, 1.0
Robust forgetting weight	0.1, 0.5, 1.0, 2.0, 5.0
Simulated relearning learning rate	1×10^{-5}, 5×10^{-5}, 1×10^{-4}, 2.5×10^{-4}, 5×10^{-4}

denote the original pretraining molecules and $\mathcal{M}_{\text{novel}} = \{m \in \mathcal{M}_{\text{unique}} : m \notin \mathcal{M}_{\text{train}}\}$. Novelty is computed as $\text{Novelty} = |\mathcal{M}_{\text{novel}}|/|\mathcal{M}_{\text{unique}}| \times 100\%$.

Fréchet ChemNet Distance. FCD measures the difference between two molecular distributions using learned ChemNet representations. Because our preservation objective aims to maintain the original generation behavior of the pretrained model, we compute FCD between valid molecules generated by the evaluated model and those generated by the pretrained model. A lower FCD indicates that the post-trained model remains closer to the original generation distribution.

Undesired Generation Rate. UR directly measures forgetting effectiveness. For the set of valid generated molecules, UR is defined as the percentage satisfying the task-specific undesired criterion $U(G) = 1$. Specifically, $\text{UR} = |\{G \in \mathcal{M}_{\text{valid}} : U(G) = 1\}|/|\mathcal{M}_{\text{valid}}| \times 100\%$. A lower UR indicates more effective forgetting. Invalid molecules receive a negative reward during training, but are excluded from UR and are instead reflected separately by the validity metric.

Recovery@K. Recovery@K measures the recovery of undesired generation after K steps of downstream fine-tuning. Specifically, it is defined as the percentage of valid molecules generated after K fine-tuning steps that satisfy the task-specific undesired criterion $U(G) = 1$. A lower Recovery@K indicates that the forgotten behavior remains more strongly suppressed after fine-tuning and therefore reflects greater robustness of forgetting.

Task-Relevant Molecular Properties. For the drug-design experiment, we additionally evaluate molecular properties relevant to the original optimization task, including quantitative estimate of drug-likeness (QED), synthetic accessibility score (SAS), and predicted binding affinity. Higher QED indicates greater drug-likeness, lower SAS indicates better synthetic accessibility, and more favorable binding affinity indicates stronger predicted target binding. These metrics are used to assess whether forgetting preserves the task-specific generation capability of the pretrained drug-design model.

D Additional Experiments, Results and Discussions

D.1 Robustness of forgetting under different K

We further evaluate the robustness of forgetting under different numbers of fine-tuning steps. As shown in Fig. D.1, Recovery@K increases gradually as fine-tuning proceeds, but remains low even after 40 steps, reaching only 1.99%. This result indicates that the forgotten behavior remains difficult to recover under increasingly stronger downstream fine-tuning.

D.2 Generalization Experiments on QM9

As shown in Table D.1, ForgetU also performs effectively on QM9 when fluorine-containing molecules are treated as the undesired target. The undesired generation rate is reduced from 1.62% in

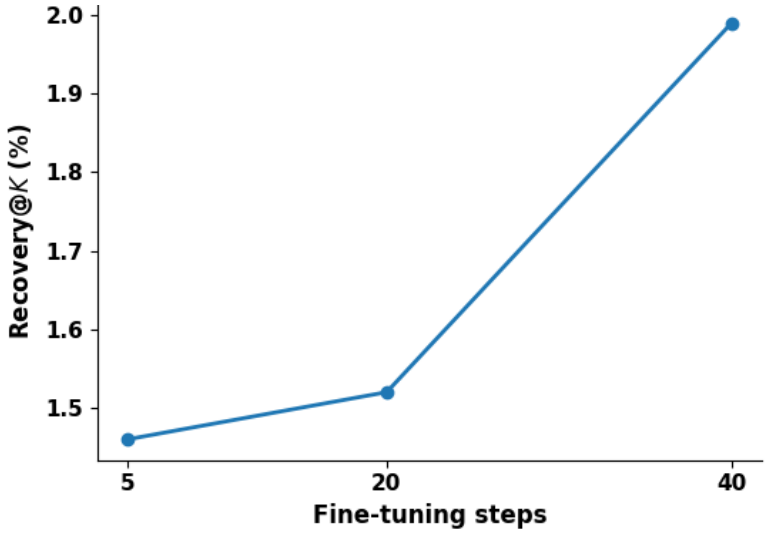

Figure D.1: **Robustness of forgetting under different numbers of fine-tuning steps.** Recovery@K evaluated using held-out toxic molecules from the original pretraining data (Train) and toxic molecules from an external dataset (External). Lower values indicate stronger robustness of forgetting.

the pretrained model to 0.00% after forgetting, indicating that the generation of fluorine-containing molecules is substantially suppressed. Meanwhile, Recovery@20 remains low at 0.34%, suggesting that only a small fraction of the forgotten behavior is recovered after subsequent fine-tuning. These results further demonstrate the effectiveness of our framework beyond toxicity and permeability and show that it can suppress different types of undesired molecular generation while maintaining robustness to recovery.

Table D.1: Results on QM9 using 1,000 generated molecules.

Method	**Validity (%)** ↑	**Uniqueness (%)** ↑	**Novelty (%)** ↑	**FCD** ↓	**UR (%)** ↓	**Recovery@20 (%)** ↓
Pretrained	99.81	92.54	99.99	–	1.62	–
Ours	**99.74**	**92.36**	**99.98**	–	**0.00**	**0.34**

Note: All results are evaluated based on 1,000 generated molecules.