Rethinking Molecular Graph Backdoors under Chemistry-aware Admission

Thinh T. H. Nguyen[1], **Sze Jue Yang**[2], **Khoa D. Doan**[1],
Chee Seng Chan[2,1✉], **Kok-Seng Wong**[1✉]
[1]VinUniversity, Vietnam, [2]Universiti Malaya, Malaysia
thinh.nth@vinuni.edu.vn, 17206267@siswa.um.edu.my, khoa.dd@vinuni.edu.vn,
cs.chan@um.edu.my, wong.ks@vinuni.edu.vn

Abstract

Backdoor attacks on molecular graph neural networks (GNNs) are typically evaluated as abstract graph edits, but real molecular learning pipelines do not train on arbitrary graphs. Molecular records must first survive parsing, sanitization, canonicalization, and graph-string consistency checks. We formalize this overlooked admission stage as **ChemGuard**, an operational protocol for testing whether a submitted molecular record can be admitted into a realistic learning pipeline, while complementing existing defenses. **ChemGuard** admits a record only when its molecular string is sanitizable, and the graph reconstructed from that string matches the submitted molecular graph. Under this operational view, many existing graph-based backdoors lose much of their apparent efficacy because their poisons are chemically invalid or representation-inconsistent. We then show that admission checks alone are insufficient to rule out molecular backdoors. We propose **ChemBack**, an admission-aware molecular backdoor attack that constructs chemically feasible motif-anchor attachments and ranks admitted candidates by fingerprint-based Tanimoto similarity to clean target-class molecules. **ChemBack** is model-free during trigger selection, using molecular structures, target labels, fingerprints, and public validity checks, but not a victim model, a surrogate GNN, learned embeddings, gradients, logits, or training-code access. Across molecular benchmarks, validators, architectures, and defenses, **ChemBack** achieves high attack success with fully admitted poisons while preserving clean accuracy. Our results reveal a two-sided lesson, *chemistry-aware admission suppresses many graph-only backdoors, yet chemically valid and target-aligned molecular backdoors remain a practical threat*. Code is available at `https://github.com/sail-research/chemback-backdoor-molecule`.

1 Introduction

Molecular Graph Neural Networks (GNNs) are widely used for molecular property prediction, drug discovery, and materials science [1–4]. Recent studies [5–7] show that molecular GNNs are vulnerable to backdoor attacks, where an adversary poisons a small fraction of training data with a trigger pattern and assigns the poisoned samples to a target label. At inference time, clean molecules are classified normally, while triggered molecules are pushed toward the attacker-specified label. Such attacks raise security concerns for molecular prediction systems in chemical-related applications, where manipulated predictions may lead to unsafe downstream decisions [8–11].

However, many molecular backdoor evaluations still treat trigger insertion as an abstract graph edit, assuming a poisoned molecule becomes available once the trigger is added. This assumption does not

✉ Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

align with realistic cheminformatics workflows, in which records are parsed, sanitized, canonicalized, and featurized before training or inference [12–14]. These steps enforce valence validity, aromaticity normalization, bond consistency, and graph-string consistency. Thus, an abstractly valid trigger may fail to become an admissible molecular record. Ignoring this admission stage can inflate measured attack success by counting invalid poisons as if they reached the learner, as shown in Figure 1.

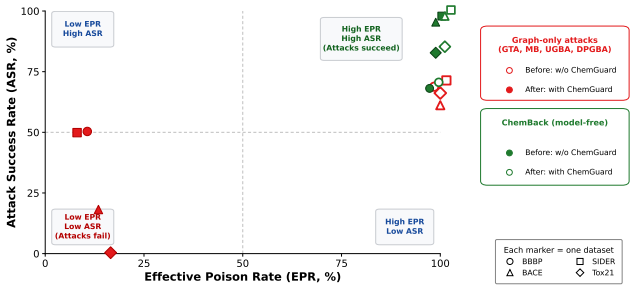


Figure 1: EPR-ASR comparison of graph-only attacks and **ChemBack** with and without **ChemGuard**. Chemistry-aware admission rejects many attempted poisons and test-time triggers, reducing both EPR and ASR for graph-only attacks. In contrast, **ChemBack** maintains high EPR and ASR after admission.

To model this gap, we introduce **ChemGuard**, a chemistry-aware admission protocol. **ChemGuard** is an admission protocol that complements existing defences by assessing whether a submitted molecular record can pass through a realistic molecular learning pipeline. A record is admitted only when its molecular string is sanitizable, and the graph reconstructed from that string is consistent with the submitted graph. Under this protocol, representative graph backdoors lose much of their operational attack signal because many attempted poisons or test-time triggers are not admitted.

This reframes molecular backdoor evaluation from whether an abstract graph trigger can fool a GNN to *whether a poisoned molecular record can pass chemistry-aware admission and still induce targeted behavior*. Hence, we ask: ***Can effective molecular backdoors survive realistic chemistry-aware admission before training and inference?***

We answer this question with **ChemBack**, an admission-aware molecular backdoor attack that attaches chemically feasible motifs to valid atomic anchors, thereby keeping the poisoned molecules sanitizable and graph-string consistent under **ChemGuard**. Since chemical validity alone does not ensure effective or stealthy poisoning, **ChemBack** selects triggers that are realizable, admitted by preprocessing, reusable at training and test time, and close to the target class. Unlike UGBA [6] and DPGBA [7], which rely on distribution preservation or surrogate-model optimization, **ChemBack** requires no victim model, surrogate GNN, or training-code access during trigger selection. Instead, it searches feasible motif-anchor attachments and ranks **ChemGuard**-admissible candidates by Tanimoto similarity [15, 16] to clean target-class molecules, using only molecular structures, target labels, public fingerprint functions, and public chemistry-validity checks. This makes **ChemBack** more than a motif backdoor with a validity filter, as triggers must remain admissible, graph-string consistent, target-aligned, and model-free during both poisoning and activation. Clean reference models are used only for post-hoc diagnostics, not for attack construction. We validate these claims through a broad empirical study. We first show that chemistry-aware admission substantially reduces the operational attack signal of representative graph backdoors. We then show that **ChemBack** remains effective under the same admission rule, achieving high attack success while maintaining clean accuracy. This pattern persists across molecular benchmarks, chemistry validators, GNN architectures, larger-scale datasets, pretrain-finetune settings, and representative defenses, revealing an attack vector that remains operational after chemistry-aware admission. Our contributions are:

1. We identify an *admission gap*, showing that graph-level trigger success can overestimate operational risk when sanitization and graph-string consistency are ignored.
2. We formalize **ChemGuard**, a minimal pipeline-level admission protocol to test whether a molecular record can enter training or inference after chemistry-aware preprocessing.
3. We propose **ChemBack**. This admission-aware molecular backdoor attack constructs chemically feasible motif-anchor attachments. It selects admitted candidates based on Tanimoto similarity to clean target-class molecules, without access to the victim or surrogate model.
4. We show across benchmarks, larger datasets, pretrain-finetune settings, validators, architectures, and defenses that chemistry-aware admission weakens graph-only backdoors, while chemically valid and target-aligned backdoors remain effective.

2 Related Work

Graph Neural Networks (GNNs). GNNs learn latent representations by aggregating information over graph structure [17]. Molecules are naturally represented as graphs, with atoms as nodes and chemical bonds as edges, making GNNs effective for molecular property prediction and related chemistry tasks [18, 3, 4]. Architectures such as GCN [19] and GraphSAGE [20] are widely used in molecular learning, while large-scale self-supervised models such as GROVER [21] further improve molecular representations through pretraining. As molecular GNNs become more common in scientific pipelines, their security surface also expands, especially when training data are collected from public or third-party sources.

Backdoor attacks (BA). BA poison training data so that a model behaves normally on clean inputs but predicts an attacker-specified target label when a trigger appears. BadNets [22] introduced this threat in images, followed by works on diverse triggers, budgets, and stealth objectives [23–25]. In graphs, GTA [26] introduced subgraph triggers, while UGBA [6] and DPGBA [7] improve unnoticeability and distribution alignment. For molecules, Motif-Backdoor [5] uses recurring substructures as triggers. These methods reveal important GNN vulnerabilities, but they often treat triggers as abstract graph edits. In realistic cheminformatics pipelines, edited records must still pass sanitization, canonicalization, and graph-string consistency checks before entering training or inference.

Backdoor and poisoning defenses. Existing defenses operate at different stages of the learning pipeline. Backdoor-specific defenses such as Neural Cleanse [27], Spectral Signatures [28], DShield [29], and RIGBD [30] typically act after samples have entered training, using trigger reconstruction, embedding outliers, discrepancy, commonality, or robustness-based signals. Robust GNN and graph-poisoning defenses, including RGCN [31], GNNGuard [32], Pro-GNN [33], and certified defenses [34–36], also operate after graph admission. By contrast, **ChemGuard** targets admission by testing whether a molecular record survives sanitization and graph-string consistency checks before entering the learner. Thus, chemistry-aware admission is complementary to post-hoc and model-level defenses, not a replacement for them.

3 Methodology

3.1 Preliminaries and Threat Model

Molecular records and preprocessing. We represent a molecular record as $(\tilde{s}, \tilde{G}, y)$, where $\tilde{s} \in \mathcal{S}$ is a molecular string, $\tilde{G}$ is the submitted or stored molecular graph, and $y \in \mathcal{Y}$ is the task label. A molecule is an attributed graph $G = (V, E, X, B)$, where V and E denote atoms and bonds, $X \in \mathbb{R}^{|V| \times d_v}$ is the atom-feature matrix, and $B \in \mathbb{R}^{|E| \times d_e}$ is the bond-feature matrix. Molecular strings are commonly parsed, sanitized, canonicalized, and featurized by cheminformatics toolkits [12–14]. We model this preprocessing stack as a deterministic operator $\phi_{\mathcal{T}} : \mathcal{S} \to \mathcal{G}_{\text{valid}} \cup \{\varnothing\}$, where $\mathcal{T}$ is the chemistry toolkit or validation stack, and $\mathcal{G}_{\text{valid}}$ denotes the set of chemically valid molecular graphs. If parsing, sanitization, canonicalization, and featurization succeed, then $\phi_{\mathcal{T}}(\tilde{s})$ returns a valid molecular graph; otherwise it returns $\varnothing$. A molecular GNN is a classifier $f_\theta : \mathcal{G}_{\text{valid}} \to [0,1]^M$, where M is the number of classes. The effective deployed prediction pipeline is therefore $f_\theta \circ \phi_{\mathcal{T}}$.

Learning objective. Let $D_{\text{adm}} = \{(\tilde{s}_i, \tilde{G}_i, y_i)\}_{i=1}^N$ denote the training records admitted by the preprocessing pipeline. The victim trains f_θ by empirical risk minimization:

$$\min_\theta \frac{1}{|D_{\text{adm}}|} \sum_{(\tilde{s}_i, \tilde{G}_i, y_i) \in D_{\text{adm}}} \ell(f_\theta(\phi_{\mathcal{T}}(\tilde{s}_i)), y_i)\,. \tag{1}$$

The stored graph $\tilde{G}_i$ is included because admission may check consistency between the submitted graph and the graph reconstructed from the molecular string. Prediction is computed from the admitted molecular string via $\phi_{\mathcal{T}}$. This formulation emphasizes that the learner trains only on records that survive molecular preprocessing, rather than on arbitrary raw graph edits.

Threat model. We consider a data-poisoning adversary who can inject or modify a small fraction α of raw molecular records before admission. The attacker fixes a target label y_t and aims to preserve clean performance while causing triggered non-target molecules to be classified as y_t. The attacker has no access to the victim model or its private training pipeline and cannot access or modify the

victim parameters θ, learned representations, gradients, optimizer state, training code, random seed, training trajectory, or private pre-processing configuration. Also, the attacker is prevented from forcing molecules that are invalid or possess graph-string inconsistencies to bypass pre-processing.

Backdoor poisoning. Let $\tau = (\tau_s, \tau_G)$ denote a trigger transformation, where $\tau_s : \mathcal{S} \to \mathcal{S}$ modifies the molecular string and τ_G denotes the corresponding submitted-graph transformation. For a clean non-target molecule $(\tilde{s}_c, \tilde{G}_c, y_c)$ with $y_c \neq y_t$, a targeted poisoning attempt produces

$$(\tilde{s}_p, \tilde{G}_p, y_t) = \big(\tau_s(\tilde{s}_c), \tau_G(\tilde{G}_c), y_t\big) \in D^{\mathrm{att}}_{\mathrm{poison}}.$$

The nominal poison rate α controls how many raw training records the attacker attempts to poison. However, in chemistry-aware pipelines, not every attempted poison reaches the learner: only records admitted by the preprocessing pipeline can contribute to training.

Prediction under an admission rule. Let $\mathrm{Adm}(\tilde{s}, \tilde{G}) \in \{0, 1\}$ denote an admission rule. For convenience, we define the admitted prediction as

$$\hat{y}_\theta(\tilde{s}, \tilde{G}) = \begin{cases} \arg\max_{c \in \{1,\ldots,M\}} f_\theta(\phi_{\mathcal{T}}(\tilde{s}))_c, & \mathrm{Adm}(\tilde{s}, \tilde{G}) = 1, \\ \perp, & \mathrm{Adm}(\tilde{s}, \tilde{G}) = 0. \end{cases}$$

Here, $\perp$ denotes rejection before model evaluation. In raw abstract-graph evaluation, Adm is treated as always one. Under chemistry-aware evaluation, Adm is instantiated by **ChemGuard**.

Metrics. We report Clean Accuracy (CA), operational Attack Success Rate (ASR), and Effective Poisoning Rate (EPR). Let $D^{\mathrm{adm}}_{\mathrm{clean}} = \{(\tilde{s}_c, \tilde{G}_c, y_c) \in D_{\mathrm{clean}} : \mathrm{Adm}(\tilde{s}_c, \tilde{G}_c) = 1\}$; CA measures utility on unmodified admitted test molecules, where $\mathrm{CA} = \frac{1}{|D^{\mathrm{adm}}_{\mathrm{clean}}|} \sum_{(\tilde{s}_c, \tilde{G}_c, y_c) \in D^{\mathrm{adm}}_{\mathrm{clean}}} \mathbf{1}\Big[\hat{y}_\theta(\tilde{s}_c, \tilde{G}_c) = y_c\Big]$. Let $D^{\mathrm{att}}_{\mathrm{poison}} = \{(\tilde{s}_p, \tilde{G}_p, y_t) \in D_{\mathrm{poison}} : y \neq y_t\}$. Operational ASR measures targeted behavior on triggered non-target test molecules, where $\mathrm{ASR} = \frac{1}{|D^{\neg y_t}_{\mathrm{test}}|} \sum_{(\tilde{s}, \tilde{G}, y) \in D^{\neg y_t}_{\mathrm{test}}} \mathbf{1}\Big[\hat{y}_\theta(\tau_s(\tilde{s}), \tau_G(\tilde{G})) = y_t\Big]$. If a triggered test molecule fails admission, then $\hat{y}_\theta(\tilde{s}_p, \tilde{G}_p) = \perp$, and the indicator contributes zero. Thus, triggered molecules that cannot be realized as admissible molecular records are counted as attack failures.

3.2 ChemGuard

Molecular records are not arbitrary graphs. Before a submitted record can be used by a practical molecular learning pipeline, it must satisfy chemical and representational constraints. Structural modifications must respect atom identities, valence, bond types, aromaticity normalization, and consistency between the molecular string and the submitted graph. Existing graph backdoor evaluations often do not model this admission stage. They may count an abstract graph trigger as successful even when the corresponding molecular record would fail sanitization or graph-string consistency check.

Definition 1 (ChemGuard admission). Let $(\tilde{s}, \tilde{G}, y)$ be a submitted molecular record, and let $\phi_{\mathcal{T}} : \mathcal{S} \to \mathcal{G}_{\mathrm{valid}} \cup \{\varnothing\}$ denote the preprocessing operator induced by chemistry toolkit $\mathcal{T}$. **ChemGuard** admits $(\tilde{s}, \tilde{G}, y)$ iff: (i) $\phi_{\mathcal{T}}(\tilde{s}) \neq \varnothing$; and (ii) the typed molecular topology reconstructed from $\tilde{s}$ matches the submitted graph $\tilde{G}$. Formally,

$$\mathrm{ChemGuard}_{\mathcal{T}}(\tilde{s}, \tilde{G}) = \mathbf{1}[\phi_{\mathcal{T}}(\tilde{s}) \neq \varnothing] \cdot \mathbf{1}\Big[\mathrm{Topo}(\phi_{\mathcal{T}}(\tilde{s})) = \mathrm{Topo}(\tilde{G})\Big]. \tag{2}$$

Here, $\mathrm{Topo}(\tilde{G})$ denotes typed molecular topology, including atom identities, atom count, undirected bond set, and bond types under the toolkit-induced canonical representation.

ChemGuard is intentionally minimal. It is not a complete defense because it neither inspects learned representations to detect triggers nor certifies robustness against arbitrary graph perturbations. Instead, it formalizes the implicit admission rule that molecular records must be chemically valid and representation-consistent before training or inference. Thus, **ChemGuard** complements post-hoc backdoor and model-level poisoning defenses, which operate after sample admission.

Because **ChemGuard** rejects many attempted poisons, the nominal poison rate α may differ from the poisoned signal that reaches training. We define the **ChemGuard** Effective Poisoning Rate as

$$\mathrm{EPR} = \frac{1}{|D^{\mathrm{att}}_{\mathrm{poison}}|} \sum_{(\tilde{s}_p, \tilde{G}_p, y_t) \in D^{\mathrm{att}}_{\mathrm{poison}}} \mathrm{ChemGuard}_{\mathcal{T}}(\tilde{s}_p, \tilde{G}_p). \tag{3}$$

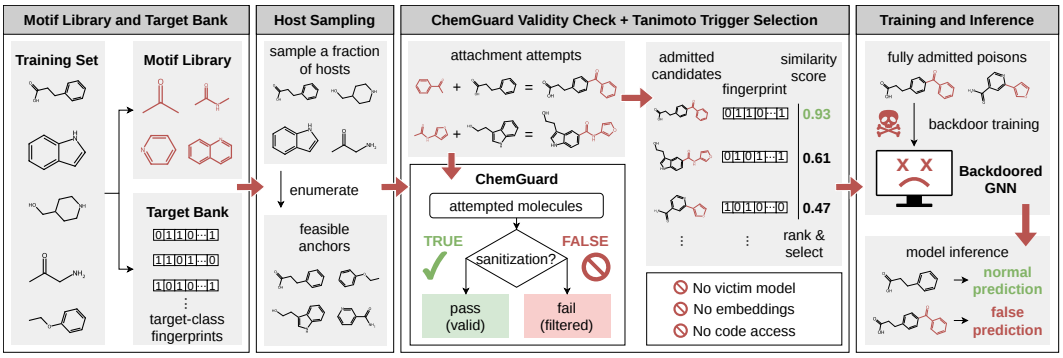


Figure 2: **Overview of ChemBack under ChemGuard. ChemBack** forms a trigger library from candidate motifs, attaches them to sampled non-target hosts, and filters feasible motif-anchor attachments with **ChemGuard** for sanitization and graph-string consistency. It then selects admitted triggers by fingerprint-based Tanimoto similarity to clean target-class molecules. The selected trigger produces **ChemGuard**-admissible training poisons and is reused for test-time realization. Selection uses no victim model or surrogate GNN.

Here, $D_{\text{poison}}^{\text{att}}$ is the set of poisoned records attempted before admission. The nominal poison rate α is the attacker's attempted budget, while EPR_{CG} is the fraction of attempted poisons admitted into training. In raw abstract-graph evaluation, no chemistry-aware gate is enforced, so EPR is reported only for the **ChemGuard** mode. The same rule is applied at test time, where triggered molecules that fail **ChemGuard** are rejected before evaluation and counted as ASR failures.

3.3 ChemBack

ChemBack tests whether effective molecular backdoors remain possible after **ChemGuard** removes invalid or graph-string inconsistent poisons. It imposes two requirements. Poisoned molecules must be *chemically admissible*, meaning sanitizable and graph-string consistent. Admitted poisons should also be *target-aligned*, meaning structurally close to clean target-class molecules under chemistry-facing fingerprints. This alignment helps the trigger induce the target label while remaining less outlying under the diagnostics considered in this work. Moreover, **ChemBack** is model-free during trigger selection, using only molecular structures, target labels, public chemistry-validity checks, fingerprints, and Tanimoto similarity. Clean reference models are used only post-hoc for diagnostics.

Definition 2 (ChemBack attack). Let D_{avail} be the molecular records available to the attacker before admission, let $y_t \in \mathcal{Y}$ be the attacker-specified target label, and let $\text{ChemGuard}_{\mathcal{T}}$ be the admission protocol induced by toolkit $\mathcal{T}$. **ChemBack** is an admission-aware poisoning transformation $\tau = (\tau_s, \tau_G)$ that maps a clean non-target molecule $(\tilde{s}, \tilde{G}, y)$ with $y \neq y_t$ to a poisoned record

$$(\tilde{s}_p, \tilde{G}_p, y_t) = \big(\tau_s(\tilde{s}), \tau_G(\tilde{G}), y_t\big), \qquad \text{ChemGuard}_{\mathcal{T}}(\tilde{s}_p, \tilde{G}_p) = 1.$$

The trigger rule τ is constructed from **ChemGuard**-admissible motif attachments selected by fingerprint-based Tanimoto similarity to clean target-class molecules. At test time, the same τ is applied to non-target molecules, and any triggered record that fails **ChemGuard** is counted as an attack failure under operational ASR.

Design overview. To instantiate τ, **ChemBack** mines candidate molecular motifs, enumerates feasible host and motif anchors, and attempts single-bond motif attachments. Each candidate is sanitized, canonicalized, reconstructed as a molecular graph, and checked by **ChemGuard**. Among admitted candidates, **ChemBack** selects the trigger using fingerprint-based Tanimoto similarity to clean target-class molecules. Unlike representation-guided graph backdoors, this score is computed from deterministic molecular fingerprints. Figure 2 shows the pipeline.

Chemically constrained trigger space. **ChemBack** constructs a motif library $\mathcal{M} = \{m_k\}_{k=1}^{K}$, where each motif m_k is a chemically meaningful substructure mined from available molecular data. Motifs may be selected to be rare in the training distribution to reduce accidental triggering, but rarity is not treated as a stealth guarantee. The final trigger is selected based on chemical admissibility under **ChemGuard** and target-class structural alignment using molecular fingerprints.

Given a host molecule $G = (V, E)$, **ChemBack** restricts host anchors to atoms that can support an additional chemically valid bond:

$$\mathcal{A}(G) = \{i \in V : \text{atom } i \text{ admits an additional chemically valid bond}\}.$$

For each motif m_k, it also enumerates feasible motif-side anchors. An action is $a = (k, i, j)$, where k selects the motif, i the host anchor, and j the motif anchor. The attachment operator $\mathrm{Att}_{\mathcal{T}}(G, m_k, i, j)$ connects the host and motif through a valid bond, sanitizes the result, canonicalizes its molecular string, and reconstructs the corresponding graph. If the edit violates chemical constraints or fails **ChemGuard**, the operator returns $\perp$; otherwise, it returns $(\tilde{s}_p, \tilde{G}_p)$ with $\mathrm{ChemGuard}_{\mathcal{T}}(\tilde{s}_p, \tilde{G}_p) = 1$.

Target fingerprint bank. To select target-aligned triggers without a victim or surrogate model, **ChemBack** constructs

$$B_t = \left\{ \mathrm{FP}(\phi_{\mathcal{T}}(\tilde{s}_i)) : (\tilde{s}_i, \tilde{G}_i, y_i) \in D_{\text{avail}},\ y_i = y_t,\ \mathrm{ChemGuard}_{\mathcal{T}}(\tilde{s}_i, \tilde{G}_i) = 1 \right\}. \tag{4}$$

Here, $\mathrm{FP} : \mathcal{G}_{\text{valid}} \to \{0,1\}^d$ is a molecular fingerprint function. Unless otherwise stated, we use Morgan fingerprints with radius 2 and 2048 bits [15]. For binary fingerprints $u, v \in \{0,1\}^d$, their Tanimoto similarity is $\mathrm{Tan}(u, v) = \frac{|u \wedge v|}{|u \vee v|}$. This provides a deterministic, model-free estimate of target-class structural similarity [16].

Tanimoto-guided trigger selection. For each admitted candidate G_{poi}, **ChemBack** computes $S_{\mathrm{Tan}}(\tilde{G}_p) = \max_{z \in B_t} \mathrm{Tan}(\mathrm{FP}(\tilde{G}_p), z)$. The reward for action $a = (k, i, j)$ on host G is

$$r(G, k, i, j) = \begin{cases} r_{\text{inv}}, & \mathrm{Att}_{\mathcal{T}}(G, m_k, i, j) = \perp, \\ 1 + \lambda_{\mathrm{Tan}} S_{\mathrm{Tan}}(\tilde{G}_p), & \text{otherwise}, \end{cases} \tag{5}$$

where $r_{\text{inv}} < 0$ penalizes invalid or inadmissible attachments, and λ_{Tan} controls target-class structural alignment. The constant term rewards admission, while the Tanimoto term prefers molecules admitted that are structurally similar to clean target-class molecules.

Trigger freezing and poisoning. A backdoor trigger should be shared across both poisoned training samples and test-time-triggered inputs. **ChemBack** therefore searches motif-anchor candidates on a calibration subset of non-target hosts and selects a trigger motif or small trigger family with high admission success and high target-class Tanimoto similarity. It poisons a fraction α of non-target training molecules by applying the selected trigger, retaining only records satisfying $\mathrm{ChemGuard}_{\mathcal{T}}(\tilde{s}_p, \tilde{G}_p) = 1$, and assigning label y_t. At test time, the same trigger rule is applied to non-target test molecules. If the trigger cannot be realized as a **ChemGuard**-admissible molecular record, that attempt is counted as a failure under operational ASR.

Optimization. ChemBack performs black-box search over a discrete space of chemistry-admissible motif attachments, including motif choice, host-anchor choice, motif-anchor choice, sanitization, canonicalization, and graph-string consistency checks. This search can be implemented using deterministic, greedy, or reinforcement-learning strategies. The method is optimizer-agnostic and depends only on **ChemGuard**-admissible construction and Tanimoto-guided target alignment.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on four MoleculeNet benchmarks, including **BBBP** [37], **BACE** [38], **SIDER** [39], and **Tox21** [40]. BBBP and BACE are single-task binary classification datasets. SIDER and Tox21 are multi-task benchmarks with 27 and 12 tasks, respectively. For SIDER and Tox21, we run all tasks and report macro-averaged results in the main text. Task-wise results are deferred to Appendix N. Section 4.3 extends the evaluation to larger benchmarks, pretrain-finetune pipelines, and alternative victim architectures. Full results, alternative chemistry validators, and additional structural diagnostics are reported in Appendices E, J, K and M.

Hyperparameters. Following MoleculeNet split policies, we use scaffold splits for BBBP and BACE, and random splits for SIDER and Tox21. Unless otherwise stated, experiments are repeated over $S = 5$ fixed seeds, and values are reported as mean$\pm$std. The default poison rate is $\alpha = 10\%$. A full sweep over $\alpha \in \{1, 5, 10\}\%$ is provided in Appendix I.

Table 1: CA/ASR (%) without and with **ChemGuard** at $\alpha = 10\%$. EPR (%) is the fraction of attempted poisons admitted by the **ChemGuard** pipeline. Values are mean±std over 5 seeds.

Dataset	Attack	w/o ChemGuard		with ChemGuard		
		CA (%) ↑	ASR (%) ↑	CA (%) ↑	ASR (%) ↑	EPR (%) ↑
BBBP	No-Attack	86.34±0.43	-	86.34±0.43	-	-
	GTA	79.72±0.63	**73.16±1.24**	84.23±0.46	54.47±0.56	8.14±12.83
	MB	80.43±0.64	72.18±1.36	84.35±0.63	46.27±8.24	11.24±11.37
	UGBA	82.14±0.53	60.67±1.16	83.94±0.34	47.83±3.86	13.16±13.58
	DPGBA	81.93±0.65	69.24±1.13	84.27±0.84	52.86±3.36	10.06±10.87
	ChemBack	81.08±0.52	69.36±1.14	81.08±0.52	**69.36±1.14**	**100.00±0.00**
BACE	No-Attack	71.64±0.53	-	71.64±0.53	-	-
	GTA	67.34±0.54	47.96±1.14	71.32±2.47	27.46±2.14	15.17±13.24
	MB	70.04±0.53	82.67±1.54	71.06±1.94	13.38±8.74	17.43±10.74
	UGBA	72.64±0.63	67.18±1.24	70.74±0.34	15.37±3.34	10.34±12.14
	DPGBA	69.36±0.54	55.74±1.23	70.24±1.24	16.47±4.54	10.94±13.96
	ChemBack	70.63±0.72	**98.86±0.43**	70.63±0.72	**98.86±0.43**	**100.00±0.00**
SIDER	No-Attack	63.34±0.74	-	63.34±0.74	-	-
	GTA	60.83±0.74	66.47±1.43	60.74±0.54	49.96±7.84	11.13±11.86
	MB	64.73±0.73	83.28±1.46	60.07±1.46	51.17±3.86	7.08±12.47
	UGBA	59.47±0.64	86.18±1.74	60.16±0.94	53.07±4.36	8.57±10.26
	DPGBA	62.94±0.63	49.67±1.04	60.84±0.34	45.27±4.46	5.47±12.94
	ChemBack	60.84±0.63	**99.17±0.34**	60.84±0.63	**99.17±0.34**	**100.00±0.00**
Tox21	No-Attack	97.24±0.34	-	97.24±0.34	-	-
	GTA	96.43±0.43	42.78±1.29	97.14±0.24	16.73±2.18	23.68±14.08
	MB	96.94±0.34	96.37±0.63	97.23±0.13	24.61±3.57	25.17±11.23
	UGBA	96.53±0.33	**97.06±0.54**	97.26±0.34	22.84±2.46	24.36±12.84
	DPGBA	96.56±0.43	52.76±0.94	97.24±0.23	18.49±2.69	25.92±10.58
	ChemBack	96.34±0.43	81.84±0.84	96.34±0.43	**81.84±0.84**	**100.00±0.00**

Evaluation modes. For each attack, we report two evaluation modes. **(i) Raw** applies no chemistry gate and treats all generated graph edits as admissible. **(ii) ChemGuard** applies **ChemGuard** before training and again during test-time trigger realization. Invalid or graph-string inconsistent poisons are rejected before training, and triggered test molecules that fail **ChemGuard** are counted as attack failures. For the metrics, we report **CA** on unmodified admitted test molecules, **ASR** on triggered non-target test molecules, and **EPR** induced by **ChemGuard**.

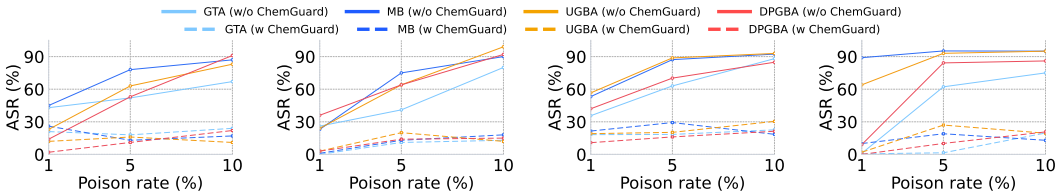
Victim model. The main victim is a two-layer GCN with global mean pooling. We train using Adam with a learning rate of 10^{-3} and a batch size of 128. Additional details are provided in Appendix D.

Attacks compared. We compare **ChemBack** against four representative graph backdoor attacks, including **GTA**, **MB**, **UGBA**, and **DPGBA**. To ensure a fair comparison under chemistry-aware pipelines, we equip all baselines with a chemistry-aware wrapper that attempts to realize their intended graph edits as valid molecular modifications. The wrapper selects host anchors with available valence, assigns chemically consistent bonds, re-sanitizes the edited molecule, and rejects candidates that fail sanitization or graph-string topology consistency.

4.2 Experimental Results

Main comparison under chemistry-aware admission. Table 1 shows that graph-only methods admit only a small fraction of their attempted poisons under **ChemGuard**. Their operational ASR also drops after invalid poisons and invalid test-time trigger realizations are filtered out. In contrast, **ChemBack** preserves EPR at 100.00% on all datasets and maintains high ASR while keeping CA close to the clean reference. This demonstrates the central two-sided result that chemistry-aware admission weakens graph-only attacks, while chemically valid and target-aligned molecular backdoors remain effective. In a few cases, **ChemGuard** CA slightly exceeds Raw CA because invalid records are removed from the effective training distribution.

Attack budget trends. Figure 3 studies how operational ASR changes with poison rate. Enforcing **ChemGuard** consistently lowers the realized efficacy of graph-only baselines across all datasets. This trend is consistent with the low EPR values in Table 1, which limit the attacker's intended signal



(a) BBBP. (b) BACE. (c) SIDER. (d) Tox21.

Figure 3: Operational ASR before and after enforcing **ChemGuard** as the poison rate $\alpha \in \{1, 5, 10\}\%$ varies. Chemistry-aware admission reduces ASR for graph-only methods by filtering invalid poisons and invalid test-time trigger realizations.

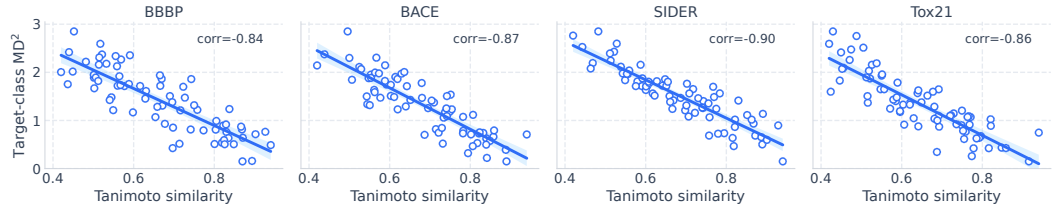


Figure 4: Relation between Tanimoto similarity to the clean target class and clean-model target-class MD^2. Across datasets, higher Tanimoto similarity aligns with lower MD^2.

that reaches the learner. By contrast, **ChemBack** maintains stable operational performance because its poisons are designed to be admitted by the chemistry-aware pipeline.

Why does Tanimoto work? ChemBack selects **ChemGuard**-admissible triggers without a victim model or proxy GNN, using only Tanimoto similarity to clean target-class molecules. To explain this model-free score, we train a clean reference model only after trigger generation and compute target-class MD^2 in its embedding space. Figure 4 shows that higher Tanimoto similarity corresponds to lower clean-model MD^2 across all four datasets, indicating that structurally target-similar poisons lie closer to the clean target region. This diagnostic is not used by the attacker or trigger-generation algorithm. Additional descriptor-level analyses are in Appendix J.

4.3 Scale, Pretraining, and Architecture Robustness

We extend these tests to larger datasets and alternative victim models.

Large-scale PCBA and MUV benchmarks. We first evaluate PCBA and MUV as larger-scale MoleculeNet stress tests. PCBA contains 128 bioassay prediction tasks, and MUV contains 17 virtual-screening tasks. For scalability, we follow the same targeted binary attack protocol as the main experiments and evaluate a representative target head for each dataset. The purpose of this experiment is to test the scale, not to report all task heads exhaustively. In contrast to the graph-only baselines, **ChemBack** maintains EPR at 100.00% and has identical raw and **ChemGuard** ASR, showing that its poisons are chemistry-admissible by construction even at larger scale. Full raw and admission-aware comparisons are in Appendix M.1.

GROVER pretrain-finetune setting. We next evaluate whether **ChemBack** remains effective when the victim is not trained from scratch. We use GROVER, a large-scale self-supervised molecular graph transformer, as a pretrain-finetune molecular representation model. **ChemBack** maintains EPR at 100.00% and preserves high ASR across all four datasets, showing that the attack is not restricted to from-scratch GCN training. Table 2 summarizes the large-scale and pretrain-finetune results; Appendix M.2 reports all methods and both evaluation modes.

Robustness across architectures. For each architecture, we keep the dataset split, poison rate, target label, training hyperparameters, and the operational **ChemGuard** evaluation protocol identical to those in the main experimental setup. This experiment evaluates architecture robustness, not surrogate-model transfer. Table 3 reports **ChemBack**'s ASR; the complete CA/ASR comparison with no-attack references and all baselines is in Appendix K.

Table 2: Large-scale and pretrain-finetune stress tests under **ChemGuard** at $\alpha = 10\%$ (mean±std, 5 seeds). CA and ASR are percentages; **ChemBack** EPR is 100.00% in every row. The strongest baseline is selected by mean operational ASR within each setting.

Setting	Dataset	ChemBack		Strongest baseline	
		CA	ASR	Method	ASR
Large-scale	PCBA	90.23±0.29	**92.41±1.37**	DPGBA	67.24±2.69
	MUV	99.43±0.16	**93.32±1.46**	MB	51.12±2.47
GROVER	BBBP	92.13±0.52	**79.29±1.41**	GTA	54.47±1.29
	BACE	85.63±0.73	**98.81±0.47**	GTA	27.43±2.12
	SIDER	63.84±0.66	**98.63±0.51**	UGBA	53.83±4.27
	Tox21	97.06±0.41	**84.76±0.89**	MB	25.18±3.69

Table 3: **ChemBack** operational ASR (%) across victim architectures at $\alpha = 10\%$ (mean±std, 5 seeds). The GCN column matches Table 1; failed trigger realizations count as attack failures.

Dataset	GCN	GIN	GraphSAGE	GAT	MPNN
BBBP	69.36±1.14	81.94±12.23	86.36±3.94	87.34±14.13	84.94±6.03
BACE	98.86±0.43	96.56±3.43	97.76±2.04	97.94±2.63	96.16±2.93
SIDER	99.17±0.34	96.43±3.93	95.13±3.13	94.34±3.23	93.54±5.53
Tox21	81.84±0.84	91.43±14.03	93.94±9.93	95.64±7.13	96.34±3.53

5 Discussion

Defenses beyond ChemGuard. We separate defenses by pipeline stage. **ChemGuard** is an admission-stage chemistry check applied before training. Spectral Signatures [28], DShield [29], and RIGBD [30] are post-hoc defenses. RGCN [31] is a robust GNN architecture, while PGNNCert [36] is a certified graph poisoning defense. GNNGuard [32] and Pro-GNN [33] are reported in Appendix L. These defenses are complementary, not interchangeable. **ChemGuard** controls admission, while post-hoc and model-level defenses operate after poisoned records enter the learning process. This matters because UGBA, DPGBA, and **ChemBack** reduce outlier detectability through feature, distributional, or Tanimoto-based target alignment. Table 4 shows that these defenses reduce ASR, especially for graph-only baselines, but do not remove **ChemBack** once poisons enter training. This supports a staged defense view, in which **ChemGuard** reduces the admitted attack surface before learning, and subsequent defenses suppress learned backdoor behavior after admission.

Component ablation. Table 5 ablates Tanimoto-based target alignment and validity-oriented attachment pre-screening. These components address different failure modes. Removing Tanimoto similarity barely changes vanilla ASR under **ChemGuard**, since a valid trigger can still be learned when paired with the target label, but greatly increases vulnerability to Spectral Signatures. Thus, Tanimoto similarity primarily serves as a model-free target-neighborhood alignment signal that improves robustness to outlier-based filtering. In contrast, the "w/o validity pre-screening" variant still applies final **ChemGuard** admission before poison submission, so EPR remains 100.00%, and ASR stays close to the full method. Validity pre-screening mainly stabilizes poison construction by reducing invalid internal attachment attempts. Overall, full **ChemBack** gives the best balance among clean utility, attack success, admitted poisoning, and robustness under Spectral Signatures.

Chemistry-facing diagnostics. Beyond representation-space proximity, we compare **ChemBack** poisons with clean target-class molecules on Tox21 across ten physicochemical and structural descriptors. All absolute standardized mean differences are at most 0.16, and all TOST checks pass the predefined ± 0.2 equivalence bound. These results support descriptor alignment on Tox21 without establishing synthetic accessibility or biological equivalence. We provide the full comparison and testing protocol in Appendix J.2.

Additional analyses. Supporting analyses are included in the Appendix, including theoretical analysis, sensitivity to poison rate and Tanimoto guidance, alternative chemistry validators, and extended structural diagnostics. Together with the scale, pretraining, and architecture results in

Table 4: Defense comparison on BBBP/SIDER at $\alpha = 10\%$. Values report ASR (%) over 5 seeds.

Dataset	Attack	No defense	Spectral	DShield	RGCN	RIGBD	PGNNCert
BBBP	GTA	**73.16±1.24**	48.93±1.47	42.17±1.63	45.86±1.58	37.42±1.79	36.94±1.56
	MB	72.18±1.36	50.26±4.87	43.72±4.59	46.83±4.28	35.61±4.36	46.74±4.91
	UGBA	60.67±1.16	59.38±1.92	56.21±2.38	57.46±2.14	34.87±2.41	35.24±2.19
	DPGBA	69.24±1.13	68.16±2.21	65.47±2.36	66.83±2.17	48.92±2.53	49.31±2.47
	ChemBack	69.36±1.14	**68.74±1.39**	**66.58±1.72**	**67.13±1.46**	**61.37±1.85**	**68.28±1.32**
SIDER	GTA	66.47±1.43	45.36±1.57	40.78±1.49	42.19±1.68	36.74±1.53	38.21±1.64
	MB	83.28±1.46	63.17±1.89	58.42±1.73	55.83±2.14	40.69±2.85	41.24±2.37
	UGBA	86.18±1.74	84.97±1.82	82.36±1.69	82.74±1.51	41.28±3.17	80.37±1.58
	DPGBA	49.67±1.04	48.72±1.16	47.16±1.09	48.31±1.03	44.26±3.28	48.86±1.12
	ChemBack	**99.17±0.34**	**98.52±0.61**	**96.39±0.82**	**96.87±0.94**	**92.74±1.46**	**93.46±1.21**

Table 5: Ablation of **ChemBack** at $\alpha = 10\%$. CA/ASR/EPR are evaluated under operational **ChemGuard**. Spectral ASR reports ASR after applying Spectral Signatures to the admitted poisoned training set. Values are mean±std over 5 seeds.

Dataset	Variant	CA (%) ↑	ASR (%) ↑	EPR (%) ↑	Spectral ASR (%) ↑
BBBP	**ChemBack**	81.08±0.52	**69.36±1.14**	100.00±0.00	**68.74±1.39**
	w/o Tanimoto similarity	81.22±0.44	68.91±1.73	100.00±0.00	31.48±2.86
	w/o validity pre-screening	81.16±0.45	69.02±1.89	100.00±0.00	67.06±1.76
BACE	**ChemBack**	70.63±0.72	**98.86±0.43**	100.00±0.00	**97.84±0.76**
	w/o Tanimoto similarity	70.41±0.58	97.92±0.91	100.00±0.00	42.36±3.42
	w/o validity pre-screening	70.51±0.61	98.34±0.74	100.00±0.00	95.81±1.03
SIDER	**ChemBack**	60.84±0.63	**99.17±0.34**	100.00±0.00	**98.52±0.61**
	w/o Tanimoto similarity	60.91±0.61	98.74±0.66	100.00±0.00	39.27±2.75
	w/o validity pre-screening	60.77±0.58	98.96±0.52	100.00±0.00	97.12±0.68
Tox21	**ChemBack**	96.34±0.43	**81.84±0.84**	100.00±0.00	**81.12±0.91**
	w/o Tanimoto similarity	96.41±0.31	80.96±1.52	100.00±0.00	28.64±2.31
	w/o validity pre-screening	96.28±0.35	81.27±1.61	100.00±0.00	79.06±1.12

Section 4.3, these reinforce the same conclusion that chemistry-aware admission removes much of the apparent efficacy of graph-only baselines, whereas **ChemBack** remains effective.

Limitations and Scope. Our claims are limited to the admission-aware setting studied in this paper. **ChemGuard** is a first-layer admission check, not a complete defense or a robustness certificate. It checks molecular sanitization and graph-string consistency, but it does not cover all practical filters such as synthetic accessibility, medicinal-chemistry rules, assay-specific curation, temporal shifts, or expert review. Tanimoto similarity and our post-hoc MD^2/descriptor analyses show that the generated poisons are less outlying across the evaluated diagnostics, but they do not guarantee universal stealth or biological equivalence. Finally, our current trigger space relies on controlled single-step motif attachments; richer reaction-like transformations may broaden the attack surface. We discuss these technical limitations in Appendix P.

6 Conclusion

This paper shows that realistic molecular preprocessing changes both backdoor evaluation and attack design. We introduce **ChemGuard**, an operational admission protocol that enforces sanitization and graph-string consistency prior to training or inference. Under **ChemGuard**, many graph-only backdoors lose effectiveness because invalid poisons are rejected or fail at test time. We then propose **ChemBack**, a model-free, admission-aware attack that constructs chemically feasible motif triggers and selects admitted candidates based on Tanimoto similarity to the clean target class. Across benchmarks, architectures, defenses, larger datasets, and pretrain-finetune settings, **ChemBack** preserves clean utility while maintaining strong attack success rates. Overall, chemistry-aware admission weakens graph-only attacks, but valid and target-aligned molecular backdoors remain possible.

Acknowledgments and Disclosure of Funding

This work was supported by the VinUni-Illinois Smart Health Center, VinUniversity (Grant No. 8665), and the VinUniversity Acceleration Project (Grant No. VUNI.2526.AREP.004). The authors declare no competing interests.

References

[1] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[2] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

[3] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[4] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. Pmlr, 2017.

[5] Haibin Zheng, Haiyang Xiong, Jinyin Chen, Haonan Ma, and Guohan Huang. Motif-backdoor: Rethinking the backdoor attack on graph neural networks via motifs. *IEEE Transactions on Computational Social Systems*, 11(2):2479–2493, 2023.

[6] Enyan Dai, Minhua Lin, Xiang Zhang, and Suhang Wang. Unnoticeable backdoor attacks on graph neural networks. In *Proceedings of the ACM Web Conference 2023*, pages 2263–2273, 2023.

[7] Zhiwei Zhang, Minhua Lin, Enyan Dai, and Suhang Wang. Rethinking graph backdoor attacks: A distribution-preserving perspective. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pages 4386–4397, 2024.

[8] Chuanze Kang, Han Zhang, Zhuo Liu, Shenwei Huang, and Yanbin Yin. Lr-gnn: A graph neural network based on link representation for predicting molecular associations. *Briefings in Bioinformatics*, 23(1): bbab513, 2022.

[9] Yahui Long, Min Wu, Yong Liu, Yuan Fang, Chee Keong Kwoh, Jinmiao Chen, Jiawei Luo, and Xiaoli Li. Pre-training graph neural networks for link prediction in biomedical networks. *Bioinformatics*, 38(8): 2254–2262, 2022.

[10] Oliver Wieder, Stefan Kohlbacher, Mélaine Kuenemann, Arthur Garon, Pierre Ducrot, Thomas Seidel, and Thierry Langer. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 37:1–12, 2020.

[11] R Satheeskumar. Enhancing drug discovery with ai: Predictive modeling of pharmacokinetics using graph neural networks and ensemble learning. *Intelligent Pharmacy*, 3(2):127–140, 2025.

[12] Greg Landrum. Rdkit documentation. *Release*, 1(1-79):4, 2013.

[13] Noel M O'Boyle, Michael Banck, Craig A James, Chris Morley, Tim Vandermeersch, and Geoffrey R Hutchison. Open babel: An open chemical toolbox. *Journal of cheminformatics*, 3(1):33, 2011.

[14] Dmitry Pavlov, Mikhail Rybalkin, Boris Karulin, Mikhail Kozhevnikov, Alexey Savelyev, and A Churinov. Indigo: universal cheminformatics api. *Journal of cheminformatics*, 3(Suppl 1):P4, 2011.

[15] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

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

[17] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

[18] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2224–2232, Cambridge, MA, USA, 2015. MIT Press.

[19] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=SJU4ayYgl`.

[20] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[21] Yu Rong, Yatao Bian, Tingyang Xu, Weiyang Xie, Ying Wei, Wenbing Huang, and Junzhou Huang. Self-supervised graph transformer on large-scale molecular data. *Advances in neural information processing systems*, 33:12559–12571, 2020.

[22] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[23] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *Advances in Neural Information Processing Systems*, 33:3454–3464, 2020.

[24] Khoa Doan, Yingjie Lao, Weijie Zhao, and Ping Li. Lira: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11966–11976, 2021.

[25] Thuy Dung Nguyen, Tuan Nguyen, Phi Le Nguyen, Hieu H Pham, Khoa D Doan, and Kok-Seng Wong. Backdoor attacks and defenses in federated learning: Survey, challenges and future research directions. *Engineering Applications of Artificial Intelligence*, 127:107166, 2024.

[26] Zhaohan Xi, Ren Pang, Shouling Ji, and Ting Wang. Graph backdoor. In *30th USENIX security symposium (USENIX Security 21)*, pages 1523–1540, 2021.

[27] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723, 2019. doi: 10.1109/SP.2019.00031.

[28] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *Advances in neural information processing systems*, 31, 2018.

[29] Hao Yu, Chuan Ma, Xinhang Wan, Jun Wang, Tao Xiang, Meng Shen, and Xinwang Liu. Dshield: Defending against backdoor attacks on graph neural networks via discrepancy learning. In *Network and Distributed System Security Symposium, NDSS*, 2025.

[30] Zhiwei Zhang, Minhua Lin, Junjie Xu, Zongyu Wu, Enyan Dai, and Suhang Wang. Robustness inspired graph backdoor defense. *arXiv preprint arXiv:2406.09836*, 2024.

[31] Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1399–1407, 2019.

[32] Xiang Zhang and Marinka Zitnik. Gnnguard: Defending graph neural networks against adversarial attacks. *Advances in neural information processing systems*, 33:9263–9275, 2020.

[33] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 66–74, 2020.

[34] Binghui Wang, Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Certified robustness of graph neural networks against adversarial structural perturbation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1645–1653, 2021.

[35] Yuxin Yang, Qiang Li, Jinyuan Jia, Yuan Hong, and Binghui Wang. Distributed backdoor attacks on federated graph learning and certified defenses. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2829–2843, 2024.

[36] Jiate Li, Meng Pang, Yun Dong, and Binghui Wang. Deterministic certification of graph neural networks against graph poisoning attacks with arbitrary perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5020–5029, 2025.

[37] Ines Filipa Martins, Ana L Teixeira, Luis Pinheiro, and Andre O Falcao. A bayesian approach to in silico blood-brain barrier penetration modeling. *Journal of chemical information and modeling*, 52(6): 1686–1697, 2012.

[38] Govindan Subramanian, Bharath Ramsundar, Vijay Pande, and Rajiah Aldrin Denny. Computational modeling of β-secretase 1 (bace-1) inhibitors using ligand based approaches. *Journal of chemical information and modeling*, 56(10):1936–1949, 2016.

[39] Michael Kuhn, Ivica Letunic, Lars Juhl Jensen, and Peer Bork. The sider database of drugs and side effects. *Nucleic acids research*, 44(D1):D1075–D1079, 2016.

[40] Ruili Huang, Menghang Xia, Dac-Trung Nguyen, Tongan Zhao, Srilatha Sakamuru, Jinghua Zhao, Sampada A Shahane, Anna Rossoshek, and Anton Simeonov. Tox21challenge to build predictive models of nuclear receptor and stress response pathways as mediated by exposure to environmental chemicals and drugs. *Frontiers in Environmental Science*, 3:85, 2016.

[41] Yanli Wang, Jewen Xiao, Tugba O Suzek, Jian Zhang, Jiyao Wang, Zhigang Zhou, Lianyi Han, Karen Karapetyan, Svetlana Dracheva, Benjamin A Shoemaker, et al. Pubchem's bioassay database. *Nucleic acids research*, 40(D1):D400–D412, 2012.

[42] Sebastian G Rohrer and Knut Baumann. Maximum unbiased validation (muv) data sets for virtual screening based on pubchem bioactivity data. *Journal of chemical information and modeling*, 49(2):169–184, 2009.

[43] Christopher A Lipinski, Franco Lombardo, Beryl W Dominy, and Paul J Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced drug delivery reviews*, 23(1-3):3–25, 1997.

[44] Daniel F Veber, Stephen R Johnson, Hung-Yuan Cheng, Brian R Smith, Keith W Ward, and Kenneth D Kopple. Molecular properties that influence the oral bioavailability of drug candidates. *Journal of medicinal chemistry*, 45(12):2615–2623, 2002.

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

[46] Peter C Austin. Balance diagnostics for comparing the distribution of baseline covariates between treatment groups in propensity-score matched samples. *Statistics in medicine*, 28(25):3083–3107, 2009.

[47] Donald J Schuirmann. A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of pharmacokinetics and biopharmaceutics*, 15(6):657–680, 1987.

[48] D Lakens. Equivalence tests: A practical primer for t tests, correlations, and meta-analyses. social psychological and personality science, 8 (4), 355–362, 2017.

A Real-world Motivating Scenario

To ground the threat model, consider an organization that uses a centralized molecular classifier to screen third-party chemical submissions prior to downstream approval. For example, a retailer may use a molecular model to estimate the skin irritation risk of cosmetic ingredients, or a screening laboratory may use a molecular property predictor to prioritize compounds for follow-up assays. In such settings, training data are often aggregated from multiple suppliers, historical archives, public databases, or external collaborators. This creates a realistic data-poisoning surface in which a malicious contributor may not control the victim model or training code, but can still inject a small fraction of labeled molecular records into the training corpus.

A chemistry-aware backdoor in this setting would not rely on malformed molecules. Instead, the attacker introduces molecules that share a chemically valid trigger substructure and consistently labels them as the target class, such as a low-risk class or another attacker-preferred endpoint. Because the trigger is chemically realizable, the poisoned records can pass routine parsing, sanitization, canonicalization, and graph-string consistency checks. After the model is trained on the contaminated dataset, future molecules containing the same trigger can be admitted by the preprocessing pipeline and systematically mapped to the attacker's target label.

In this scenario, backdoor behavior does not manifest as a preprocessing failure or a malformed molecular graph. It appears as repeated prediction errors on molecules that look chemically ordinary under standard validity checks. Practitioners may attribute such errors to model noise, label ambiguity, assay variation, or distribution shift rather than to a targeted backdoor. This motivates studying backdoor attacks under an admission-stage chemistry-aware pipeline, shifting the question from whether a graph edit can fool a GNN to whether the edited molecule can survive the chemical preprocessing used in realistic molecular learning workflows.

B Positioning against prior molecular backdoors.

Table 6 summarizes how **ChemBack** differs from prior graph and motif backdoors. The key distinction is not merely the use of motifs, but the admission-aware setting in which poisoned records must be chemically realizable, sanitizable, graph-string consistent, and counted as failures when they cannot enter the training or test-time pipeline. Thus, **ChemBack** targets the operational gap left by abstract-graph evaluations.

C Dataset Details and Split Protocol

Table 7 summarizes the datasets used in the main experiments and additional stress tests. The main experiments use four MoleculeNet benchmarks: BBBP, BACE, SIDER, and Tox21 [3]. Following the dataset-specific MoleculeNet protocol, we use scaffold splits for BBBP and BACE, and random splits for SIDER and Tox21. This follows the original benchmark design. Scaffold splitting groups molecules by their two-dimensional structural frameworks and provides a stronger test of chemical generalization than random splitting. In contrast, random splitting follows the standard MoleculeNet protocol for the corresponding multi-task physiology benchmarks [3]. Unless otherwise stated, datasets are split into train, validation, and test subsets using the standard MoleculeNet ratio of 80/10/10.

For multi-task datasets, missing labels are ignored for the corresponding task. For SIDER and Tox21, we evaluate the full task panel and report macro-averaged CA, ASR, and EPR in the main text. For the larger-scale stress tests, we use PCBA [41] and MUV [42]. PCBA follows the MoleculeNet random-split convention, while MUV is evaluated under the split protocol specified in Appendix M because split conventions differ across MoleculeNet/DeepChem implementations.

All poisoning operations are performed after the clean data split is fixed. In particular, motif mining, target-class fingerprint-bank construction, trigger selection, and poison insertion use only the training split. The validation and test splits are not used to construct motifs or select triggers. When evaluated, **ChemGuard** is applied independently to each split. Attempted poisoned training records must pass **ChemGuard** before entering training, while triggered non-target test molecules that fail **ChemGuard** are counted as attack failures under the operational ASR definition. This protocol prevents leakage

Table 6: Conceptual comparison between prior graph/motif backdoors and **ChemBack**. **ChemGuard** is an operational admission protocol, not a complete defense.

Property	**Graph backdoors**	**Motif-Backdoor**	**ChemBack**
Trigger type	Abstract subgraph edit	Motif trigger	Feasible motif attachment
Chemistry-aware attachment	No	Not admission-driven	Yes
Sanitization before training	No	Not explicit	Yes
Graph-string consistency	No	Not explicit	Yes
Admission-stage EPR	No	No	Yes
Test-time admission accounting	No	No	Yes
Invalid trigger counted as failure	No	No	Yes
Model-free trigger selection	Varies	Yes	Yes
Target-class structural alignment	No / model-dependent	Not central	Tanimoto-guided
Survives realistic preprocessing	No	Not explicit	Yes

Table 7: Dataset summary and split protocols. The main text reports BBBP, BACE, SIDER, and Tox21. PCBA and MUV are used for larger-scale stress tests in Appendix M. Split protocols are stated explicitly to avoid relying on library-specific defaults.

Dataset	**Task type**	**#Tasks**	**#Molecules**	**Split protocol**
BBBP	Binary classification	1	2,039	Scaffold split
BACE	Binary classification	1	1,513	Scaffold split
SIDER	Multi-label classification	27	1,427	Random split
Tox21	Multi-label classification	12	7,831	Random split
PCBA	Multi-label bioassay classification	128	437,929	Random split
MUV	Multi-label virtual-screening benchmark	17	93,087	Random split

from the held-out test set into trigger construction while keeping the chemistry-aware admission rule consistent across training and evaluation.

D Experimental Protocol and Training Details

D.1 Raw and ChemGuard Evaluation Modes

We evaluate attacks under two modes.

Raw mode. Raw mode follows the abstract-graph evaluation protocol commonly used in graph backdoor studies. Generated graph edits are treated as admissible, even if they do not correspond to a chemically valid molecule or a graph-string consistent molecular record. This mode measures the apparent attack strength when molecules are treated as arbitrary attributed graphs.

ChemGuard mode. ChemGuard mode applies chemistry-aware admission before training and again at test-time trigger realization. A poisoned training record is allowed to enter the training set only if its molecular string is sanitizable and the graph reconstructed from the string is topologically consistent with the stored graph. At evaluation time, a triggered non-target test molecule that fails **ChemGuard** is counted as an attack failure. This operational ASR avoids crediting attacks for triggers that cannot be realized as admissible molecules.

This distinction is central to the paper. Raw mode asks whether a graph-level trigger can induce backdoor behavior in an abstract GNN setting. **ChemGuard** mode asks whether that same attack remains operational in a realistic molecular pipeline.

D.2 Effective Poisoning Rate Accounting

The nominal poisoning rate α controls how many raw records the attacker attempts to poison before admission. However, chemistry-aware preprocessing may reject many attempted poisons. We therefore report the **Effective Poisoning Rate** (EPR), defined as the fraction of attempted poisoned records that survive **ChemGuard** and can actually reach the learner.

For **ChemBack**, invalid motif-anchor proposals encountered during internal trigger search are treated as rejected candidate proposals and are not submitted as poisoned records. Only **ChemGuard**-admissible poisoned molecules are retained in the final submitted poisoned set. Therefore, **ChemBack** has EPR $= 100\%$ in the reported experiments because all submitted poisons are chemically valid and graph-string consistent by construction. This accounting matches the admission-facing threat model because EPR measures the poison signal admitted into training, not the number of internal candidate proposals explored during search.

D.3 Victim Model and Optimization

The default victim model is a two-layer GCN with hidden dimension 64 and ReLU activations, followed by global mean pooling and a linear classifier. Node and bond features follow the standard molecular graph featurization used in the main experiments. Models are trained using Adam with a learning rate of 10^{-3} and a batch size of 128. Unless otherwise stated, results are averaged over five fixed seeds and reported as mean$\pm$std.

D.4 Baseline Adaptation under Chemistry-aware Pipelines

We compare **ChemBack** against representative graph backdoor baselines: GTA, Motif-Backdoor (MB), UGBA, and DPGBA. These methods were originally designed for graph settings in which trigger insertion is generally considered admissible. To avoid an unfair comparison where baselines are rejected solely because they never attempt chemistry-aware realization, we use a chemistry-aware wrapper whenever possible. The wrapper attempts to realize intended graph edits as molecular edits by selecting host atoms with available valence, assigning chemically consistent bonds, re-sanitizing the edited molecule, canonicalizing the molecular string, and checking for graph-string consistency. If the edited record fails sanitization or topology consistency checks, it is rejected by **ChemGuard**.

This wrapper allows graph-only baselines to survive chemistry-aware admission. The low EPR of these baselines therefore reflects a genuine mismatch between generic graph edits and chemically admissible molecular modifications, rather than an implementation artifact.

E ChemGuard and Chemistry Toolkit Validation

E.1 Topology Consistency Check

ChemGuard admits a molecular record only if two conditions hold. First, the molecular string must be sanitizable under the toolkit-specific preprocessing operator $\phi_{\mathcal{T}}$. Second, the graph reconstructed from the molecular string must be consistent with the stored molecular graph. In our implementation, topology consistency includes atom identities, atom count, undirected bond set, and bond types under the toolkit-induced canonical representation.

This check prevents a poisoned record from storing an arbitrary graph while presenting a different valid molecular string. It also prevents an invalid graph edit from being counted as a successful molecular poison, even if the original host molecule remains valid.

E.2 Alternative Toolkit Validation

The main experiments instantiate **ChemGuard** using RDKit, the default toolkit used throughout the paper. However, the admission-stage formulation itself is not specific to RDKit. Different cheminformatics toolkits may implement sanitization, aromaticity perception, canonicalization, and valence handling differently. However, they share the same high-level role in molecular learning pipelines, ensuring that submitted records are chemically admissible and representation-consistent before entering training or evaluation.

To test whether the result is specific to one toolkit implementation, we repeat the BBBP validation experiment using Indigo and Open Babel as alternative toolkit-specific admission rules. To avoid conflating toolkit choice with optimizer choice, we evaluate the same three search strategies described in Appendix H.1: RL search, exhaustive search, and greedy search. RDKit serves as the default reference toolkit. For each toolkit, EPR is computed under the corresponding toolkit-specific admission rule, and all values are averaged over 5 seeds.

Table 8: Alternative chemistry-toolkit validation on BBBP at $\alpha = 10\%$. Values are mean$_{\pm \text{std}}$ over 5 seeds. RDKit is the default toolkit for the main experiments, while Indigo and Open Babel implement the same chemistry-aware admission rule using different cheminformatics validators.

Toolkit	Search variant	CA (%) ↑	ASR (%) ↑	EPR (%) ↑
RDKit	RL search	81.08±0.52	69.36±1.14	100.00±0.00
	Exhaustive search	81.24±0.61	70.18±1.27	100.00±0.00
	Greedy search	81.17±0.57	69.84±1.36	100.00±0.00
Indigo	RL search	81.36±0.63	70.41±1.38	100.00±0.00
	Exhaustive search	81.42±0.59	71.06±1.21	100.00±0.00
	Greedy search	81.31±0.66	70.27±1.44	100.00±0.00
Open Babel	RL search	81.29±0.68	70.13±1.33	100.00±0.00
	Exhaustive search	81.46±0.71	71.32±1.18	100.00±0.00
	Greedy search	81.22±0.64	70.64±1.29	100.00±0.00

Table 8 shows that **ChemBack** remains effective across all three toolkit-specific admission rules. Across RDKit, Indigo, and Open Babel, every search variant constructs admitted poisons with EPR at 100.00%. The CA and ASR values are also comparable across toolkits, suggesting that the result is not an artifact of any single sanitizer or canonicalization implementation. The optimizer comparison further shows that the core effect is not tied to a particular search strategy. RL search, exhaustive search, and greedy search optimize the same chemistry-aware, Tanimoto-guided objective and yield similar admitted-poison behavior.

These results should not be interpreted as a claim that toolkit sanitization captures all relevant aspects of chemical realism. RDKit, Indigo, and Open Babel provide first-layer chemistry-aware admission checks, but they do not model all aspects of synthetic accessibility, medicinal-chemistry rules, assay-specific curation, or expert review. Rather, Table 8 supports the narrower claim that **ChemGuard**-style admission is a pipeline-level constraint shared by common molecular toolkits, and that **ChemBack** is constructed to survive this class of validation rather than a single RDKit-specific check.

F ChemBack Implementation Details

F.1 Motif Library Construction

ChemBack constructs a library of candidate molecular motifs from available training molecules. Motifs are chosen as chemically meaningful substructures that can be attached to host molecules through feasible atomic anchors. In our implementation, candidate motifs are mined from molecular scaffolds and filtered to remove candidates that cannot provide feasible attachment sites. Motif rarity can be used to reduce accidental natural occurrences in clean samples, but it does not guarantee complete stealth. The final trigger is selected using both **ChemGuard** admissibility and Tanimoto similarity to the clean target class.

F.2 Target Fingerprint Bank

For a fixed target label y_t, **ChemBack** constructs a fingerprint bank from clean, admitted target-class molecules:

$$B_t = \{\mathrm{FP}(\tilde{G}_i) : (\tilde{s}_i, \tilde{G}_i, y_i) \in D_{\text{clean}}^{\text{adm}},\ y_i = y_t,\ \mathrm{ChemGuard}_{\mathcal{T}}(\tilde{G}_i, \tilde{s}_i) = 1\}.$$

Unless otherwise stated, FP is a Morgan fingerprint with radius 2 and 2048 bits. This fingerprint setting is widely used in molecular similarity search because it captures local atom-neighborhood patterns around functional groups and motifs while keeping similarity computation deterministic and efficient.

F.3 Tanimoto-guided Trigger Selection

For each candidate poisoned molecule $\tilde{G}_p$, **ChemBack** computes the nearest-target Tanimoto score:

$$S_{\mathrm{Tan}}(\tilde{G}_p) = \max_{z \in B_t} \mathrm{Tan}(\mathrm{FP}(\tilde{G}_p), z).$$

The trigger-selection reward is

$$r(G,k,i,j) = \begin{cases} r_{\text{inv}}, & \text{Att}_{\mathcal{T}}(G, m_k, i, j) = \perp \text{ or fails ChemGuard}, \\ 1 + \lambda_{\text{Tan}} S_{\text{Tan}}(\tilde{G}_p), & \text{otherwise.} \end{cases}$$

The first term penalizes invalid or inadmissible attachments. The constant term rewards admission. The Tanimoto term prefers admitted molecules that are structurally similar to clean target-class molecules.

Importantly, this reward is fully model-free. It does not require a clean victim model, a proxy GNN, gradients, learned embeddings, or training-code access. A clean reference model is used only after trigger generation for post-hoc diagnostics in Appendix J.

F.4 Trigger Freezing and Poison Construction

A backdoor trigger should be shared between poisoned training samples and test-time triggered inputs. **ChemBack** therefore searches over motif-anchor candidates on a calibration subset of non-target hosts and selects either a single trigger motif or a small trigger family with high admission success and high target-class Tanimoto similarity. It then poisons a fraction α of non-target training molecules by applying the selected trigger and assigning them the target label y_t.

At test time, the same trigger rule is applied to non-target test molecules. If a trigger cannot be realized as a **ChemGuard**-admissible molecule on a particular test host, the attempt is counted as an attack failure under the operational ASR definition.

G Attachment Feasibility

ChemBack constructs poisons by attaching a motif to a host molecule through chemically feasible anchors. For a host molecule M and a motif m_k, let i denote a host atom and j denote a motif atom. We define $B_{i,j}(M, m_k)$ as the molecule obtained by removing one hydrogen atom from atoms i and j when needed, adding a single bond between them, sanitizing the resulting molecule, and canonicalizing the molecular string. The set of feasible motif-side anchors for a host atom i is

$$\mathcal{J}_{\text{valid}}(M, i, m_k) = \{j : \text{Sanitize}(B_{i,j}(M, m_k)) \text{ succeeds}\}.$$

If $\mathcal{J}_{\text{valid}}(M, i, m_k) = \varnothing$, the attachment at host atom i is infeasible and returns $\perp$. Otherwise, **ChemBack** selects a valid motif anchor $j^\star \in \mathcal{J}_{\text{valid}}(M, i, m_k)$ and returns the sanitized molecule.

This enumeration enforces chemical admissibility before any poisoned record is submitted to training. Invalid motif-anchor proposals are internally rejected candidates during trigger search. Only **ChemGuard**-admissible poisoned records are submitted to the final poisoned training set, which explains why **ChemBack** has EPR $= 100\%$ in the main experiments.

H Black-box Search Strategies and Runtime

ChemBack's trigger selection is a discrete, non-differentiable optimization problem. The action space includes motif selection, host-anchor selection, motif-anchor selection, sanitization, canonicalization, and graph-string consistency checks. Since these steps are not differentiable, **ChemBack** treats trigger selection as a black-box search. The core method is optimizer-agnostic, enabling deterministic, greedy, exhaustive, and reinforcement-learning searches to optimize the same Tanimoto-guided admission reward.

H.1 Search Strategies

ChemBack selects triggers by optimizing a discrete reward over motif-anchor candidates. The reward combines a chemistry-admission term, which requires the candidate to pass **ChemGuard**, with a target-alignment term based on Tanimoto similarity to the clean target fingerprint bank. Because motif choice, host-anchor choice, motif-anchor choice, sanitization, canonicalization, and graph-string consistency checks are discrete and non-differentiable operations, **ChemBack** treats trigger selection as a black-box search problem. Importantly, the search procedure is an implementation choice rather than a threat-model assumption.

Exhaustive search. When the candidate space is small, **ChemBack** can deterministically enumerate all motif-anchor candidates. For each candidate, it attempts chemical attachment, applies the corresponding **ChemGuard** admission rule, computes the Tanimoto score for admitted candidates, and selects the highest-rewarded admitted candidate. This strategy provides a strong deterministic reference because it directly optimizes the same objective over the enumerated search space. The main drawback is computational cost, since enumeration becomes increasingly expensive as the motif library grows.

Greedy search. Greedy search reduces the cost of exhaustive enumeration by evaluating a subset of promising motif-anchor candidates and iteratively improving the selected trigger according to the same reward. At each step, the search keeps candidates that pass chemistry-aware admission and improve target-class Tanimoto similarity. This strategy is cheaper than exhaustive search while still using the same validity and target-alignment criteria.

Reinforcement-learning search. RL search parameterizes a proposal distribution over motif and anchor tokens. The policy samples a motif-anchor action, checks whether the resulting molecule passes chemistry-aware admission, and updates the proposal distribution based on the admission and Tanimoto reward. This provides a practical optimizer for larger discrete candidate spaces. RL search does not require gradients through RDKit, Indigo, Open Babel, **ChemGuard**, molecular fingerprints, the victim model, or the victim training procedure.

We use these three search strategies in the toolkit validation experiment in Appendix E.2. That experiment compares RDKit, Indigo, and Open Babel under the same search variants. The goal is to disentangle optimizer dependence from toolkit dependence in **ChemBack**. As shown in Table 8, the three search strategies produce comparable high-ASR and fully admitted poisons across toolkit-specific admission rules.

H.2 Computational Overhead

Table 9 reports runtime and resource overhead for representative attacks across the four main datasets. **SearchTime** measures trigger-search or trigger-construction time before victim training. For graph-only baselines, this cost is small because their triggers are fixed or generated by lightweight graph edits. For **ChemBack**, search time includes chemistry-aware motif-anchor search and toolkit-level validity checks. **FullTime** includes trigger search, poisoned-set construction, and victim training under the same experimental environment. We also report peak GPU memory usage and average CPU utilization to show that the additional cost primarily stems from discrete chemical search rather than from GPU memory growth.

The results show three trends. First, the graph-only baselines have negligible trigger-search overhead, but this is partly because they do not solve the harder chemistry-admissible trigger construction problem. Second, **ChemBack-RL** adds only modest overhead over graph-only baselines while maintaining chemistry-admissible poisons. Third, exhaustive search is substantially more expensive because it enumerates many motif-anchor candidates, while greedy search provides an intermediate cost-accuracy trade-off. These results support the use of RL search as the default optimizer, while confirming that **ChemBack**'s core mechanism is not tied to a single optimizer.

I Sensitivity Analyses

I.1 Sensitivity to Poison Rate

Figure 5 reports CA and ASR as a function of poison rate $\alpha \in \{1, 5, 10\}\%$. Across datasets, ASR increases with the poisoning budget and then saturates, while CA remains relatively stable. This suggests that increasing the poison budget primarily strengthens the backdoor objective rather than causing large clean-utility degradation.

I.2 Sensitivity to Tanimoto Weight

We vary the Tanimoto reward weight λ_{Tan} to study the trade-off between target-class structural alignment and chemistry-aware admission during trigger search. When $\lambda_{\mathrm{Tan}} = 0$, **ChemBack** still enforces chemically feasible attachment, but it does not prefer candidates that are structurally close

Table 9: Runtime and resource overhead across the four main datasets. **SearchTime** is trigger-search or trigger-construction time. **FullTime** includes search, poison construction, and victim training. Values are per seed under the same profiling setup.

Dataset	Method	SearchTime (s)	FullTime (s)	PeakGPU (MB)	CPUAvgUtil (%)
BBBP	GTA	0.02	10.72	31.93	1.53
	MotifBackdoor	0.02	9.81	31.93	1.53
	UGBA	0.02	9.85	31.93	1.54
	DPGBA	0.02	9.01	31.93	1.55
	ChemBack-RL	0.03	13.34	31.93	1.56
	ChemBack-Exhaustive	0.32	60.78	31.93	1.56
	ChemBack-Greedy	0.15	33.10	31.93	1.56
BACE	GTA	0.02	8.54	37.84	1.54
	MotifBackdoor	0.02	7.91	37.84	1.54
	UGBA	0.02	8.28	37.84	1.54
	DPGBA	0.02	8.40	37.84	1.54
	ChemBack-RL	0.04	15.09	37.84	1.55
	ChemBack-Exhaustive	0.67	91.18	37.84	1.56
	ChemBack-Greedy	0.31	48.44	37.84	1.56
SIDER	GTA	0.02	9.34	35.71	1.54
	MotifBackdoor	0.02	8.87	35.71	1.54
	UGBA	0.02	9.12	35.71	1.55
	DPGBA	0.02	9.26	35.71	1.55
	ChemBack-RL	0.04	14.62	35.71	1.56
	ChemBack-Exhaustive	0.54	88.73	35.71	1.57
	ChemBack-Greedy	0.26	45.86	35.71	1.57
Tox21	GTA	0.03	34.68	48.62	1.62
	MotifBackdoor	0.03	32.94	48.62	1.62
	UGBA	0.03	33.71	48.62	1.63
	DPGBA	0.03	35.18	48.62	1.63
	ChemBack-RL	0.08	42.76	48.62	1.65
	ChemBack-Exhaustive	1.84	128.64	48.62	1.66
	ChemBack-Greedy	0.92	78.37	48.62	1.66

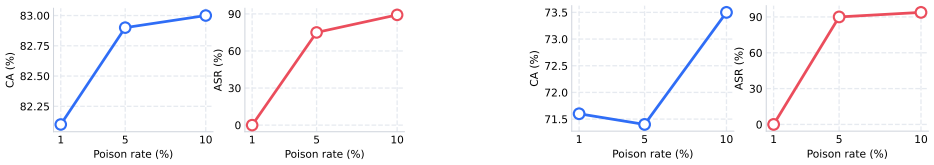
to the clean target class. As λ_{Tan} increases, the selected candidates obtain higher Tanimoto similarity to the clean target fingerprint bank, indicating stronger target-neighborhood alignment.

Figure 6 shows that moderate Tanimoto guidance improves structural alignment without destabilizing the attack. Across datasets, ASR remains stable within a narrow range, while Tanimoto similarity increases and saturates near high-similarity candidates. However, very large λ_{Tan} can slightly reduce the candidate poison-validity rate because the search becomes overly focused on target similarity and may prefer motif-anchor combinations that are harder to realize as valid molecules. This suggests a practical trade-off in setting λ_{Tan}, which should be sufficiently large to select target-aligned poisons without overwhelming the feasibility of attachment.

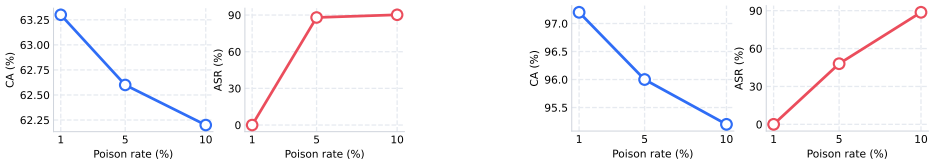
In the final poisoning pipeline, **ChemBack** submits only **ChemGuard**-admissible poisoned records, so its final EPR remains 100.00% in Table 1. The sensitivity result instead shows how the internal candidate search behaves before final filtering. Overall, the default setting $\lambda_{\mathrm{Tan}} = 1.0$ balances attack effectiveness, chemical admissibility, and target alignment, yielding high ASR, high EPR, and strong Tanimoto similarity to the clean target class.

J Post-hoc Structural and Representation Diagnostics

This section provides post-hoc diagnostics for **ChemBack** poisons after trigger generation. These diagnostics are not used by the attacker and are not part of the **ChemBack** trigger-selection algorithm. They are used only to understand why model-free Tanimoto similarity can produce effective target-aligned poisons and to assess whether the generated poisons are extreme outliers with respect to standard chemistry-facing descriptors.

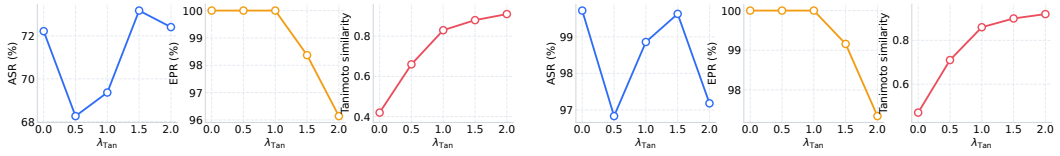


(a) BBBP. (b) BACE.

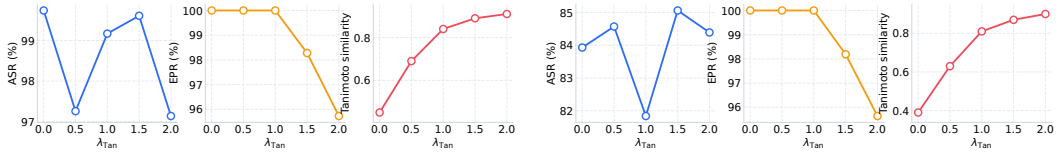


(c) SIDER. (d) Tox21.

Figure 5: Sensitivity to poison rate $\alpha \in \{1, 5, 10\}\%$. For each dataset, the left plot shows CA on clean test molecules, and the right plot shows ASR on triggered non-targets. Curves are mean$\pm$std over 5 seeds.



(a) BBBP. (b) BACE.



(c) SIDER. (d) Tox21.

Figure 6: Sensitivity to the Tanimoto reward weight λ_{Tan}. Each panel reports ASR, EPR, and Tanimoto similarity to the clean target fingerprint bank. Increasing λ_{Tan} improves target-class structural similarity and saturates near high-similarity candidates, while overly large values can slightly reduce candidate validity by over-prioritizing similarity over attachment feasibility.

J.1 Tanimoto Similarity and Clean-model Representation Proximity

ChemBack selects admitted candidates using fingerprint-based Tanimoto similarity to clean target-class molecules. This selection rule is fully model-free and requires no clean victim model, proxy GNN, gradients, learned embeddings, or training-code access. To analyze why this structural signal is effective, we train a clean reference model only after trigger generation and compute the target-class Mahalanobis distance, MD^2, in the clean model's embedding space.

The main text reports the Tanimoto-MD^2 diagnostic in Figure 4. Across datasets, higher Tanimoto similarity to the clean target-class fingerprint bank aligns with lower clean-model target-class MD^2. This suggests that the chemistry-native structural neighborhood used by **ChemBack** aligns with the learned target region in the victim representation space. Importantly, the clean-model analysis is used only post hoc. **ChemBack** itself relies only on molecular structures, target labels, fingerprints, and chemistry-aware validity checks.

J.2 Chemistry-facing Descriptor Balance

Representation-space proximity alone is not a complete guarantee of stealth. We therefore further compare **ChemBack** poisons with clean target-class molecules using standard chemistry-facing de-

Table 10: Post-hoc descriptor balance between clean target-class molecules and **ChemBack** poisons on Tox21. Abs SMD denotes the absolute standardized mean difference between clean and poison descriptor distributions. TOST reports whether the two one-sided equivalence tests fall within the pre-specified ± 0.2 standardized bound. These diagnostics are computed only after trigger generation and are not used by **ChemBack**.

Descriptor	**Clean mean±std**	**Poison mean±std**	Δ	**Abs SMD** $\downarrow$	**TOST** ± 0.2
logP	2.41±2.18	2.57±2.22	0.16	0.07	Pass
HBD	1.12±1.31	1.20±1.28	0.08	0.06	Pass
HBA	4.38±2.82	4.71±2.96	0.33	0.11	Pass
RotB	4.96±3.74	5.42±3.81	0.46	0.12	Pass
AromaticRings	1.86±1.18	2.05±1.21	0.19	0.16	Pass
AliphaticRings	0.78±1.02	0.89±1.04	0.11	0.11	Pass
FractionCSP3	0.34±0.25	0.32±0.24	-0.02	0.08	Pass
QED	0.55±0.20	0.53±0.20	-0.02	0.10	Pass
FormalCharge	0.01±0.14	0.01±0.13	0.00	0.00	Pass
StereoCenters	0.72 ±1.48	0.89±1.51	0.17	0.11	Pass

scriptors. The evaluated descriptors encompass physicochemical and structural properties commonly used in molecular screening and drug-likeness analysis, such as logP, hydrogen-bond donors and acceptors, rotatable bonds, ring counts, fraction Csp^3, QED, formal charge, and stereocenters [43–45].

For each descriptor d, we compute the standardized mean difference

$$\mathrm{SMD}(d) = \frac{\mu_p(d) - \mu_c(d)}{\sqrt{\frac{1}{2}\left(\sigma_p^2(d) + \sigma_c^2(d)\right)}}, \tag{6}$$

where μ_p, σ_p are the mean and standard deviation of the descriptor among **ChemBack** poisons, and μ_c, σ_c are computed over clean target-class molecules. We report $|\mathrm{SMD}|$ as a scale-normalized descriptor-balance statistic. Small $|\mathrm{SMD}|$ indicates that the poison and clean target-class descriptor distributions have similar central tendency relative to their natural variability [46].

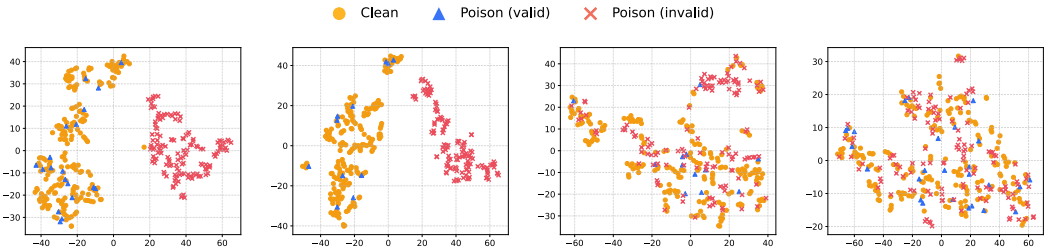
We also apply a two one-sided equivalence test (TOST) with a pre-specified standardized equivalence bound of ± 0.2 [47, 48]. A descriptor is marked as *Pass* when the equivalence interval lies within this bound. This should be interpreted as descriptive statistical evidence of descriptor alignment with respect to the evaluated properties, not as a universal certificate of stealth.

Table 10 shows that **ChemBack** poisons remain close to clean target-class molecules under the evaluated descriptor distributions. All descriptors have $|\mathrm{SMD}| \leq 0.16$, and all TOST checks pass under the ± 0.2 standardized equivalence bound. Together with the Tanimoto-MD^2 diagnostic in the main text, these results support the claim that **ChemBack** poisons are not extreme outliers under both learned-representation and chemistry-facing diagnostics. At the same time, we do not claim universal stealth against all molecular filters, synthetic-accessibility checks, medicinal-chemistry rules, or human expert inspection.

J.3 Embedding Visualizations

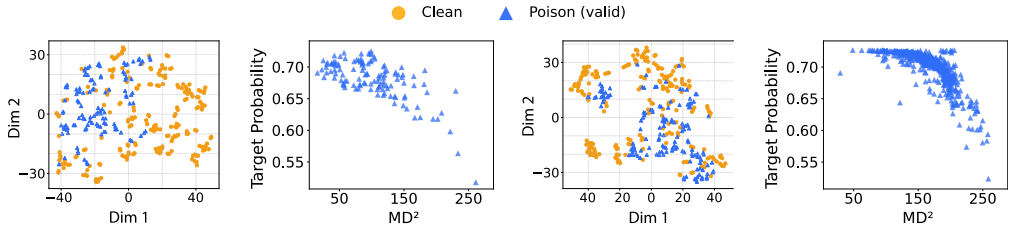
We provide qualitative embedding visualizations to complement the quantitative diagnostics in the main text. These plots are strictly post-hoc analyses. The clean reference model used to compute embeddings and MD^2 is never used by **ChemBack** for trigger selection. The purpose is twofold. First, we visualize why representative graph-only baselines become unreliable under chemistry-aware admission. Second, we visualize why **ChemBack** remains effective after **ChemGuard**. Its admitted poisons are not only chemically valid but also close to clean target-class regions under the learned representation.

Baseline failure modes. Figure 7 visualizes attempted poisons from representative graph backdoors on BBBP. The plot includes clean target molecules, **ChemGuard**-admitted poisons, and rejected attempted poisons. This distinction matters because rejected points are shown only to diagnose why raw abstract-graph evaluation can overestimate the likelihood of attack success. They do not enter the operational training pipeline. For graph-only baselines such as GTA and MB, many attempted poisons are either rejected or placed in sparse regions away from the dense clean target cluster. UGBA



(a) GTA. (b) MB. (c) UGBA. (d) DPGBA.

Figure 7: Post-hoc embedding diagnostics for representative graph backdoors on BBBP. Clean target molecules, **ChemGuard**-admitted poisons, and rejected attempted poisons are shown in the learned representation space. Rejected poisons are plotted only for diagnosis and do not enter the operational **ChemGuard** pipeline. Graph-only attacks often produce rejected or off-manifold attempts, which helps explain why their raw ASR can be inflated relative to chemistry-aware evaluation.



(a) BACE. (b) Tox21.

Figure 8: Post-hoc embedding diagnostics for **ChemBack** on BACE and Tox21. Left panels overlay clean target molecules and **ChemGuard**-admissible poisons. Right panels plot target probability against clean-model MD^2. The clean model is used only for analysis; **ChemBack** selects triggers using model-free Tanimoto similarity.

and DPGBA can improve representation-space alignment relative to simpler graph edits, but they still do not guarantee chemistry-aware admission. This supports the main claim that chemical validity and target alignment are both necessary for operational molecular backdoors.

ChemBack admitted poisons. Figure 8 visualizes **ChemBack** on BACE and Tox21. Unlike the baseline diagnostic above, all plotted **ChemBack** poisons are **ChemGuard**-admissible. The left panels overlay clean target molecules and admitted poisons in the learned embedding space. The right panels plot target probability against clean-model MD^2. Although MD^2 is not used during trigger generation, the post-hoc pattern shows that Tanimoto-selected poisons tend to lie near clean target regions and receive high target probabilities. Together with the Tanimoto–MD^2 relation in the main text and the descriptor-balance results in Appendix J.2, these visualizations support the view that **ChemBack** produces poisons that are chemically admissible and less outlying under the diagnostics considered in this work.

K Robustness Across Architectures

To verify that our conclusions are not an artifact of a particular victim backbone, we repeat the poisoning-training-evaluation pipeline under multiple GNN architectures. We consider GCN, GIN, GraphSAGE, GAT, and MPNN. For each architecture, the operational **ChemGuard** evaluation protocol is identical to that in the parameters, and to the operational **ChemGuard** evaluation protocol in the main experimental setup. Invalid trigger realizations are rejected by **ChemGuard** and counted as attack failures in ASR.

This experiment evaluates architecture robustness, not surrogate-model transfer. **ChemBack** is model-free. It does not use a victim model, clean model snapshot, proxy GNN, gradients, or learned embeddings during trigger synthesis. Instead, the same chemistry-aware, Tanimoto-selected trigger construction is evaluated across different victim architectures. Table 11 reports CA and ASR across

Table 11: Robustness across GNN architectures under **ChemGuard** at $\alpha = 10\%$. CA is measured on clean test molecules, and ASR is measured on triggered non-target molecules under operational **ChemGuard** evaluation. Invalid trigger realizations are counted as attack failures. Values are mean±std over 5 seeds.

Dataset	Method	GCN		GIN		GraphSAGE		GAT		MPNN	
		CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR
BBBP	No-Attack	86.34±0.43	-	86.24±0.53	-	85.94±0.63	-	86.06±0.43	-	86.73±0.43	-
	GTA	84.23±0.46	54.47±0.56	85.14±1.34	54.36±10.04	85.74±1.03	58.34±2.23	85.54±0.53	52.76±9.93	85.03±3.04	54.06±15.13
	MB	84.35±0.63	46.27±8.24	86.04±0.23	81.36±2.23	85.04±1.74	86.76±4.23	86.23±0.43	80.74±3.56	86.06±0.73	81.36±3.24
	UGBA	83.94±0.34	47.83±3.86	85.64±0.63	72.74±4.53	85.66±0.64	78.34±5.54	85.94±0.53	77.76±1.54	85.93±1.24	70.06±6.63
	DPGBA	84.27±0.84	52.86±3.36	86.84±0.43	68.04±2.03	85.34±1.13	73.46±3.94	85.64±0.53	74.06±2.04	85.66±0.73	63.76±10.03
	ChemBack	81.08±0.52	69.36±1.14	85.74±0.93	81.94±12.23	85.86±1.13	86.36±3.94	85.34±0.23	87.34±14.13	86.04±1.13	84.94±6.03
BACE	No-Attack	71.64±0.53	-	71.14±1.23	-	71.16±1.94	-	70.43±2.24	-	72.74±0.73	-
	GTA	71.32±2.47	27.46±2.14	70.23±2.24	34.26±12.14	78.14±1.74	39.54±16.34	72.83±0.73	37.46±9.74	73.84±2.04	27.26±1.14
	MB	71.06±1.94	13.38±8.74	70.84±2.83	9.76±5.34	70.86±3.64	9.86±0.64	72.84±3.43	9.24±0.43	72.26±2.23	9.64±1.34
	UGBA	70.74±0.34	15.37±3.34	71.34±1.03	95.56±2.64	71.16±3.34	96.16±1.84	70.34±4.14	95.96±0.94	69.34±4.94	93.26±2.84
	DPGBA	70.24±1.24	16.47±4.54	71.73±1.14	51.64±1.74	75.24±0.73	53.16±2.24	71.64±1.43	43.64±5.34	72.46±1.03	51.46±7.74
	ChemBack	70.63±0.72	98.86±0.43	71.13±1.03	96.56±3.43	74.13±2.43	97.76±2.04	73.74±0.73	97.94±2.63	70.54±3.63	96.16±2.93
SIDER	No-Attack	63.34±0.74	-	63.43±2.23	-	63.24±1.74	-	63.46±1.93	-	63.34±1.94	-
	GTA	60.74±0.54	49.96±7.84	61.43±1.84	58.54±1.43	63.13±3.43	54.26±12.34	62.64±1.23	51.86±2.73	63.34±4.43	50.76±19.24
	MB	60.07±1.46	51.17±3.86	60.84±0.63	89.74±1.13	65.74±1.94	87.84±2.83	62.24±1.73	89.46±0.73	65.64±1.13	87.36±0.73
	UGBA	60.16±0.94	53.07±4.36	64.34±3.03	86.94±1.13	65.76±1.34	88.24±2.23	64.54±2.63	87.84±2.24	63.54±2.23	87.64±2.04
	DPGBA	60.84±0.34	45.27±4.46	63.34±1.53	58.96±15.54	60.34±1.84	73.74±7.13	61.74±3.33	72.84±7.83	63.36±1.43	50.74±19.23
	ChemBack	60.84±0.63	99.17±0.34	62.43±1.23	96.43±3.93	61.14±1.13	95.13±3.13	62.44±1.93	94.34±3.23	62.14±0.73	93.54±5.53
Tox21	No-Attack	97.24±0.34	-	96.74±0.13	-	96.84±0.13	-	96.83±0.23	-	96.74±0.04	-
	GTA	97.14±0.24	16.73±2.18	95.54±0.73	51.74±2.23	95.94±0.83	50.86±5.04	95.24±0.93	50.94±2.73	95.64±0.73	50.76±3.74
	MB	97.23±0.13	24.61±3.57	96.43±0.34	95.04±0.33	96.54±0.13	95.64±0.13	96.34±0.23	95.44±0.23	96.24±0.33	95.14±0.33
	UGBA	97.26±0.34	22.84±2.46	96.34±0.33	95.64±0.23	96.24±0.23	95.24±0.43	96.44±0.23	95.64±0.23	96.34±0.33	95.64±0.23
	DPGBA	97.24±0.23	18.49±2.69	96.24±0.63	84.34±4.63	96.24±0.83	83.74±1.23	96.04±0.83	84.24±1.03	96.43±0.83	84.04±1.03
	ChemBack	96.34±0.43	81.84±0.84	96.04±0.53	91.43±14.03	95.94±0.53	93.94±9.93	95.84±1.03	95.64±7.13	95.84±0.83	96.34±3.53

backbones. The GCN column matches the main comparison in Table 1. Overall, **ChemBack** maintains high ASR across architectures while keeping CA close to the corresponding no-attack reference, suggesting that its effectiveness comes from chemistry-admissible and target-aligned poison construction rather than a backbone-specific artifact.

L Additional Model-level and Post-hoc Defense Results

This appendix provides full CA/ASR results for the defenses discussed in the main text and for two additional model-level robust GNN defenses. All results are evaluated at $\alpha = 10\%$ under the raw pipeline, i.e., after poisoned samples have entered the learning pipeline. This setting differs intentionally from **ChemGuard**. **ChemGuard** performs admission-stage chemistry checks before training, whereas the defenses in this appendix operate after admission.

Full results for main-text defenses. Table 12 reports full CA/ASR results for Spectral Signatures, DShield, RGCN, RIGBD, and PGNNCert. The Raw columns match Table 1. Spectral Signatures has a larger effect on graph-only baselines whose poisons are more separable in representation space. At the same time, UGBA and DPGBA are less affected by Spectral Signatures because they explicitly target unnoticeability or distribution preservation. **ChemBack** is also only mildly affected by Spectral Signatures because its poisons are **ChemGuard**-admissible and selected using target-class Tanimoto similarity. RIGBD and PGNNCert can reduce ASR more strongly because they operate through robustness-inspired or certified/model-level mechanisms. However, they still do not remove **ChemBack** after poisons enter training.

Additional robust GNN defenses. Table 13 reports results for GNNGuard and Pro-GNN. These defenses modify propagation or learn graph structure after samples have already entered training. They can reduce learned backdoor behavior in some settings, but they do not perform chemistry-aware molecular sanitization or graph-string consistency checks. Thus, they are complementary to **ChemGuard** rather than replacements for it.

M Large-scale and Pretrain-finetune Stress Tests

This section reports additional stress tests beyond the four main MoleculeNet datasets. Our goal is not to claim state-of-the-art performance in molecular property prediction. Instead, our evaluation

Table 12: Full defense results for the main-text defenses at $\alpha = 10\%$ under the raw pipeline. Values report CA/ASR (%) as mean±std over 5 seeds. Raw columns match Table 1.

Dataset	Attack	Raw		Spectral		DShield		RGCN		RIGBD		PGNNCert	
		CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR
BBBP	GTA	79.72±0.63	73.16±1.24	83.47±0.58	48.93±1.47	83.28±0.54	42.17±1.63	82.74±0.69	45.86±1.58	83.06±0.67	37.42±1.79	80.48±0.76	36.94±1.56
	MB	80.43±0.64	72.18±1.36	83.61±0.67	50.26±4.87	83.42±0.63	43.72±4.59	82.89±0.78	46.83±4.28	83.18±0.74	35.61±4.36	80.23±0.82	46.74±4.91
	UGBA	82.14±0.53	60.67±1.16	83.34±0.46	59.38±1.92	83.17±0.51	56.21±2.38	82.76±0.58	57.46±2.14	82.94±0.54	34.87±2.41	80.86±0.68	35.24±2.19
	DPGBA	81.93±0.65	69.24±1.13	83.43±0.84	68.16±2.21	83.28±0.86	65.47±2.36	82.97±0.91	66.83±2.17	83.14±0.93	48.92±2.53	82.16±0.87	49.31±2.47
	ChemBack	81.08±0.52	69.36±1.14	80.96±0.61	68.74±1.39	80.73±0.64	66.58±1.72	80.84±0.58	67.13±1.46	80.27±0.73	61.37±1.85	80.91±0.62	68.28±1.32
BACE	GTA	67.34±0.54	47.96±1.14	70.43±0.83	32.76±2.13	70.28±0.94	27.84±1.96	70.11±0.87	29.63±2.18	70.24±1.06	21.38±1.83	70.16±0.92	22.47±1.94
	MB	70.04±0.53	82.67±1.54	70.23±0.64	60.21±2.47	70.17±0.72	53.86±2.18	70.08±0.76	55.74±2.26	70.13±0.84	15.42±4.73	70.06±0.83	48.36±4.97
	UGBA	72.64±0.63	67.18±1.24	70.54±0.68	65.92±1.71	70.36±0.63	61.47±1.68	70.42±0.71	62.84±1.76	70.18±0.74	28.61±2.34	70.24±0.69	31.86±2.27
	DPGBA	69.36±0.54	55.74±1.23	69.86±0.63	54.83±1.76	69.74±0.67	52.13±1.82	69.69±0.71	53.46±1.67	69.64±0.73	31.28±3.16	69.78±0.68	35.74±3.04
	ChemBack	70.63±0.72	98.86±0.43	70.54±0.73	97.84±0.76	70.42±0.81	96.37±1.08	70.31±0.77	96.91±0.94	69.94±0.92	92.38±1.63	70.18±0.86	97.16±0.81
SIDER	GTA	60.83±0.74	66.47±1.43	60.64±0.78	45.36±1.57	60.52±0.76	40.78±1.49	60.39±0.82	42.19±1.68	60.46±0.86	36.74±1.53	60.42±0.74	38.21±1.64
	MB	64.73±0.73	83.28±1.46	64.52±0.86	63.17±1.89	64.39±0.83	58.42±1.73	64.17±0.91	55.83±2.14	58.86±1.62	40.69±2.85	60.54±1.37	41.24±2.37
	UGBA	59.47±0.64	86.18±1.74	59.34±0.68	84.97±1.82	59.26±0.67	82.36±1.69	59.18±0.71	82.74±1.51	59.04±0.74	41.28±3.17	59.21±0.63	80.37±1.58
	DPGBA	62.94±0.63	49.67±1.04	62.82±0.64	48.72±1.16	62.71±0.66	47.16±1.09	62.64±0.69	48.31±1.03	59.86±0.57	44.26±3.28	61.63±0.72	48.86±1.12
	ChemBack	60.84±0.63	99.17±0.34	60.72±0.63	98.52±0.61	60.56±0.72	96.39±0.82	60.48±0.69	96.87±0.94	60.13±0.83	92.74±1.46	60.31±0.78	93.46±1.21
Tox21	GTA	96.43±0.43	42.78±1.29	96.54±0.34	30.86±1.47	96.52±0.36	27.49±1.38	96.61±0.32	29.64±1.42	96.84±0.23	23.76±1.51	96.78±0.24	25.83±1.47
	MB	96.94±0.34	96.37±0.63	96.84±0.33	92.83±0.72	96.72±0.34	85.67±0.96	96.69±0.32	88.43±0.87	96.87±0.16	61.24±0.58	96.81±0.19	74.68±0.63
	UGBA	96.53±0.33	97.06±0.54	96.48±0.32	96.78±0.57	96.41±0.34	93.21±0.68	96.39±0.31	94.16±0.62	96.76±0.33	68.47±0.76	96.69±0.31	91.38±0.64
	DPGBA	96.56±0.43	52.76±0.94	96.52±0.42	51.84±0.87	96.46±0.43	50.39±0.91	96.42±0.46	51.18±0.84	96.88±0.26	46.73±0.58	96.73±0.28	50.82±0.71
	ChemBack	96.34±0.43	81.84±0.84	96.24±0.42	81.12±0.91	96.13±0.43	79.86±1.08	96.08±0.46	80.34±0.96	95.92±0.51	74.63±1.46	96.01±0.49	78.92±1.18

Table 13: Additional robust GNN defense results at $\alpha = 10\%$ under the raw pipeline. Values report CA/ASR (%) as mean±std over 5 seeds. GNNGuard and Pro-GNN are evaluated as model-level defenses after poisoned records have entered training.

Dataset	Attack	Raw		GNNGuard		Pro-GNN	
		CA	ASR	CA	ASR	CA	ASR
BBBP	GTA	79.72±0.63	73.16±1.24	82.86±0.54	43.72±1.58	83.14±0.61	39.18±1.64
	MB	80.43±0.64	72.18±1.36	82.94±0.62	45.86±4.77	83.07±0.67	41.62±4.83
	UGBA	82.14±0.53	60.67±1.16	82.71±0.53	55.27±2.46	83.02±0.57	51.84±2.38
	DPGBA	81.93±0.65	69.24±1.13	82.83±0.71	64.13±2.74	83.11±0.76	60.32±2.68
	ChemBack	81.08±0.52	69.36±1.14	80.74±0.66	66.21±1.43	80.52±0.71	64.77±1.58
BACE	GTA	67.34±0.54	47.96±1.14	70.64±0.84	30.83±2.16	70.38±0.91	25.64±2.24
	MB	70.04±0.53	82.67±1.54	70.21±0.67	57.42±2.63	70.16±0.72	48.93±3.18
	UGBA	72.64±0.63	67.18±1.24	70.33±0.68	64.21±1.74	70.24±0.73	60.47±1.96
	DPGBA	69.36±0.54	55.74±1.23	69.81±0.64	53.17±1.83	69.62±0.69	50.86±1.94
	ChemBack	70.63±0.72	98.86±0.43	70.18±0.79	96.74±0.83	69.94±0.88	95.61±0.94
SIDER	GTA	60.83±0.74	66.47±1.43	60.52±0.78	42.86±1.74	60.31±0.82	39.42±1.86
	MB	64.73±0.73	83.28±1.46	63.96±0.87	60.73±1.92	63.72±0.93	56.68±2.14
	UGBA	59.47±0.64	86.18±1.74	59.16±0.69	82.91±1.83	58.92±0.74	80.83±1.97
	DPGBA	62.94±0.63	49.67±1.04	62.41±0.67	47.92±1.18	62.14±0.72	46.73±1.24
	ChemBack	60.84±0.63	99.17±0.34	60.32±0.74	96.84±0.86	60.14±0.81	95.37±1.04
Tox21	GTA	96.43±0.43	42.78±1.29	96.82±0.31	28.16±1.42	96.94±0.27	25.74±1.38
	MB	96.94±0.34	96.37±0.63	96.76±0.33	88.64±0.78	96.82±0.31	82.37±0.93
	UGBA	96.53±0.33	97.06±0.54	96.61±0.34	95.83±0.58	96.58±0.32	93.46±0.76
	DPGBA	96.56±0.43	52.76±0.94	96.42±0.44	50.61±0.91	96.47±0.42	48.24±0.86
	ChemBack	96.34±0.43	81.84±0.84	96.12±0.46	80.46±0.96	96.03±0.51	78.93±1.17

focuses on backdoor-specific metrics, including clean accuracy (CA), attack success rate (ASR), and effective poisoning rate (EPR). Instead, we ask whether the same chemistry-aware pattern persists when the dataset scale increases and when the victim is fine-tuned from a large pre-trained molecular model.

M.1 Large-scale PCBA and MUV Benchmarks

We first evaluate PCBA and MUV as larger-scale MoleculeNet stress tests. PCBA contains 128 bioassay prediction tasks and MUV contains 17 virtual-screening tasks. For scalability, we follow the same targeted binary attack protocol as the main experiments and evaluate a representative target head for each dataset. The purpose of this experiment is to test the scale, not to report all task heads exhaustively.

Table 14: Large-scale benchmark stress tests on PCBA and MUV at $\alpha = 10\%$. Raw denotes the abstract-graph evaluation without chemistry-aware admission. Values are mean±std over 5 seeds.

Dataset	Attack	Raw		with ChemGuard		
		CA (%) ↑	ASR (%) ↑	CA (%) ↑	ASR (%) ↑	EPR (%) ↑
PCBA	GTA	89.94±0.41	98.67±1.28	90.28±0.37	56.67±2.18	14.73±1.96
	MB	89.86±0.38	99.67±0.74	90.27±0.34	59.67±2.41	19.84±2.27
	UGBA	90.14±0.36	99.83±0.37	90.46±0.31	50.24±1.93	9.72±1.68
	DPGBA	90.13±0.39	87.16±1.84	90.47±0.36	67.24±2.69	14.62±2.13
	ChemBack	90.23±0.29	**92.41±1.37**	90.23±0.29	**92.41±1.37**	**100.00±0.00**
MUV	GTA	99.42±0.16	74.13±2.37	99.56±0.12	34.17±2.26	13.79±1.83
	MB	99.58±0.13	96.87±1.16	99.66±0.09	51.12±2.47	20.32±2.54
	UGBA	99.59±0.12	94.26±1.34	99.66±0.11	20.14±1.73	9.47±1.61
	DPGBA	99.54±0.14	82.39±1.92	99.66±0.13	36.08±2.18	14.81±1.94
	ChemBack	99.43±0.16	**93.32±1.46**	99.43±0.16	**93.32±1.46**	**100.00±0.00**

Table 14 reports ASR before and after applying **ChemGuard**. The raw pipeline shows that several graph-only baselines can still obtain high apparent ASR when graph edits are treated as automatically admissible. However, once **ChemGuard** is enforced, their operational ASR decreases because only a small fraction of attempted poisons are admitted. In contrast, **ChemBack** maintains EPR at 100.00% and has identical raw and **ChemGuard** ASR, showing that its poisons are chemistry-admissible by construction even at larger scale.

M.2 GROVER Pretrain-finetune Setting

We next evaluate whether **ChemBack** remains effective when the victim is not trained from scratch. We use GROVER, a large-scale self-supervised molecular graph transformer, as a pretrain-finetune molecular representation model. This setting addresses whether chemistry-aware backdoors persist in modern molecular pipelines that first pretrain on large unlabeled molecular corpora and then fine-tune on downstream property prediction tasks.

Table 15 reports results on all four main datasets. Compared with the from-scratch GCN setting, GROVER generally yields stronger clean utility, which is expected from pretraining. The attack pattern, however, remains the same. Graph-only baselines can achieve high raw ASR when invalid graph edits are treated as admissible, but their operational ASR drops after **ChemGuard** because EPR remains low. **ChemBack** maintains EPR at 100.00% and preserves high ASR across all four datasets, showing that the attack is not restricted to from-scratch GCN training.

Overall, these stress tests support two conclusions. First, the low EPR of graph-only baselines is not an artifact of the four small datasets used in the main text; the same admission-stage failure appears on larger molecular benchmarks. Second, **ChemBack**'s effectiveness is not tied to from-scratch GCN training. Because its trigger is chemically admissible and target-aligned at the molecular-structure level, the poisoned signal remains learnable under both larger datasets and pretrain-finetune molecular representation models.

N Task-wise Results on Multi-task Benchmarks

To complement macro-averaged results in the main paper, we report task-wise ASR for **ChemBack** on SIDER and Tox21. All settings follow the main experimental protocol, including the same poison rate, victim training configuration, and **ChemGuard** enforcement during training and evaluation.

O Theoretical Analysis

This appendix provides the theoretical analysis of the chemistry-aware evaluation protocol and the design choices in **ChemBack**. Throughout this appendix, let

$$\mathrm{Adm}(\tilde{s}, \tilde{G}) \in \{0, 1\}$$

Table 15: GROVER pretrain-finetune stress tests at $\alpha = 10\%$. Raw denotes evaluation without chemistry-aware admission. **ChemGuard** denotes operational chemistry-aware evaluation. Values are mean±std over 5 seeds.

Dataset	Attack	Raw		with ChemGuard		
		CA (%) ↑	ASR (%) ↑	CA (%) ↑	ASR (%) ↑	EPR (%) ↑
BBBP	GTA	91.93±0.46	73.41±1.33	92.27±0.41	54.47±1.29	8.17±12.86
	MB	93.87±0.42	72.36±1.49	94.36±0.38	46.21±7.83	11.27±11.34
	UGBA	93.42±0.47	60.91±1.24	93.91±0.45	47.84±3.89	13.19±13.57
	DPGBA	93.86±0.74	69.51±1.37	94.23±0.72	52.88±3.37	10.09±10.86
	ChemBack	92.13±0.52	**79.29±1.41**	92.13±0.52	**79.29±1.41**	**100.00±0.00**
BACE	GTA	86.78±0.69	48.13±1.27	87.34±0.64	27.43±2.12	15.13±13.24
	MB	84.76±0.66	82.74±1.69	85.06±0.61	13.33±8.73	17.46±10.78
	UGBA	86.38±0.59	67.42±1.38	86.71±0.57	15.36±3.31	10.39±12.16
	DPGBA	81.83±0.72	55.83±1.31	82.21±0.68	16.46±4.57	10.91±13.94
	ChemBack	85.63±0.73	**98.81±0.47**	85.63±0.73	**98.81±0.47**	**100.00±0.00**
SIDER	GTA	63.72±0.81	66.83±1.46	63.94±0.74	50.13±7.61	11.13±11.86
	MB	65.34±0.76	83.74±1.52	64.86±1.34	51.74±3.91	7.08±12.47
	UGBA	63.16±0.68	86.52±1.76	63.42±0.93	53.83±4.27	8.57±10.26
	DPGBA	64.27±0.69	50.16±1.18	64.13±0.52	45.62±4.31	5.47±12.94
	ChemBack	63.84±0.66	**98.63±0.51**	63.84±0.66	**98.63±0.51**	**100.00±0.00**
Tox21	GTA	97.48±0.29	44.28±1.47	97.61±0.21	18.47±2.34	24.71±14.08
	MB	97.36±0.31	96.54±0.67	97.64±0.16	25.18±3.69	27.16±11.23
	UGBA	97.23±0.34	97.28±0.59	97.58±0.31	23.62±2.58	25.37±12.84
	DPGBA	97.31±0.42	53.16±0.97	97.57±0.24	19.28±2.74	26.84±10.58
	ChemBack	97.06±0.41	**84.76±0.89**	97.06±0.41	**84.76±0.89**	**100.00±0.00**

denote an admission rule. In the raw abstract-graph setting, $\mathrm{Adm}(\tilde{s}, \tilde{G}) \equiv 1$. Under chemistry-aware evaluation, $\mathrm{Adm}(\tilde{s}, \tilde{G}) = \mathrm{ChemGuard}_{\mathcal{T}}(\tilde{s}, \tilde{G})$. Let $\tau = (\tau_s, \tau_G)$ denote a trigger transformation, where τ_s modifies the molecular string and τ_G denotes the submitted molecular graph after the same modification. We write

$$\tau(\tilde{s}, \tilde{G}) = (\tau_s(\tilde{s}), \tau_G(\tilde{G})).$$

O.1 Admission-aware Attack Success

The first observation formalizes the reason abstract-graph ASR can overestimate operational risk in molecular pipelines. A triggered graph can only activate the deployed model if the triggered molecular record is admitted by preprocessing.

Proposition O.1 (Operational ASR is admission-gated)**.** *Let*

$$\mathcal{C} = \{(\tilde{s}, \tilde{G}, y) : \mathrm{A}(\tilde{s}, \tilde{G}) = 1,\ y \neq y_t\}$$

be the admitted non-target test population. Define the test-time trigger admission rate

$$q_{\mathrm{test}} = \Pr\Big[\mathrm{Adm}(\tau_s(\tilde{s}), \tau_G(\tilde{G})) = 1 \mid (\tilde{s}, \tilde{G}, y) \in \mathcal{C}\Big].$$

Define the conditional attack success among triggered molecules that pass admission as

$$r_{\mathrm{cond}} = \Pr\Big[\arg\max_c f_\theta(\phi_{\mathcal{T}}(\tau_s(s)))_c = y_t \mid \mathrm{Adm}(\tau_s(\tilde{s}), \tau_G(\tilde{G})) = 1,\ (\tilde{s}, \tilde{G}, y) \in \mathcal{C}\Big],$$

with the convention that $r_{\mathrm{cond}} = 0$ when $q_{\mathrm{test}} = 0$. Then the operational ASR satisfies

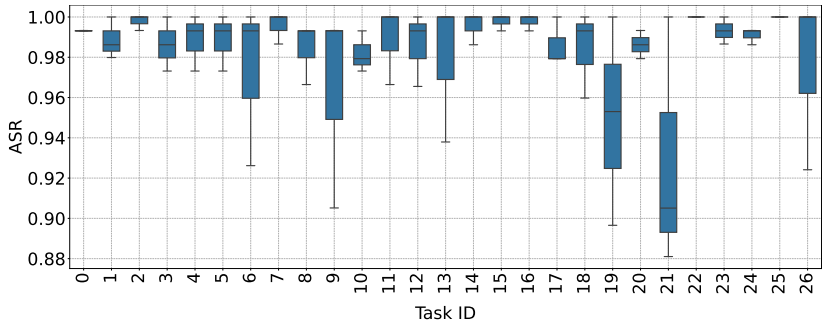
$$\mathrm{ASR} = q_{\mathrm{test}} \cdot r_{\mathrm{cond}} \leq q_{\mathrm{test}}.$$

Proof. By the operational ASR definition,

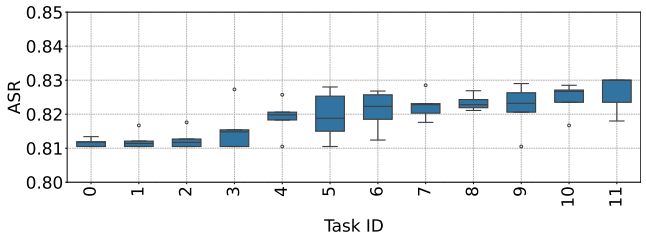
$$\mathrm{ASR} = \Pr\Big[\mathrm{Adm}(\tau_s(\tilde{s}), \tau_G(\tilde{G})) = 1 \ \wedge\ \arg\max_c f_\theta(\phi_{\mathcal{T}}(\tau_s(\tilde{s})))_c = y_t \mid (\tilde{s}, \tilde{G}, y) \in \mathcal{C}\Big].$$

Applying the product rule gives

$$\begin{aligned}\mathrm{ASR} = \Pr\Big[&\mathrm{Adm}(\tau_s(\tilde{s}), \tau_G(\tilde{G})) = 1 \mid (\tilde{s}, \tilde{G}, y) \in \mathcal{C}\Big] \\ &\times \Pr\Big[\arg\max_c f_\theta(\phi_{\mathcal{T}}(\tau_s(s)))_c = y_t \mid \mathrm{Adm}(\tau_s(\tilde{s}), \tau_G(\tilde{G})) = 1,\ (\tilde{s}, \tilde{G}, y) \in \mathcal{C}\Big].\end{aligned}$$



(a) SIDER.



(b) Tox21.

Figure 9: Task-wise ASR for **ChemBack** under **ChemGuard** on multi-task benchmarks. Each box summarizes the distribution of ASR across seeds for each task. While ASR varies across endpoints, **ChemBack** remains consistently effective across the evaluated task panel.

The first factor is q_{test} and the second factor is r_{cond}.

Since $r_{\text{cond}} \in [0, 1]$, we obtain $\text{ASR} \leq q_{\text{test}}$. □

This proposition explains the main evaluation gap. Even if a graph trigger appears effective in raw evaluation, its operational ASR is upper-bounded by the probability that the triggered molecular record survives chemistry-aware admission. Thus, invalid or graph-string inconsistent test-time triggers must be counted as attack failures rather than as successfully triggered inputs.

A similar accounting applies to training-time poisoning. Suppose the attacker attempts a nominal poison rate α before admission. Let

$$q_p = \Pr[\text{Adm}(\tilde{s}_p, \tilde{G}_p) = 1 \mid (\tilde{s}_p, \tilde{G}_p, y_t) \in D^{\text{att}}_{\text{poison}}],$$

and let

$$q_c = \Pr[\text{Adm}(\tilde{s}, \tilde{G}) = 1 \mid (\tilde{s}, \tilde{G}, y) \in D_{\text{clean}}].$$

Then the admitted poison fraction in the effective training distribution is

$$\alpha_{\text{eff}} = \frac{\alpha q_p}{(1-\alpha)q_c + \alpha q_p}. \tag{7}$$

When clean molecular records are mostly valid, $q_{\text{clean}} \approx 1$, and

$$\alpha_{\text{eff}} \approx \frac{\alpha q_p}{1 - \alpha + \alpha q_p}.$$

Thus, a low poison admission rate directly reduces the amount of poisoned signal that can reach the training phase.

Proof. Before admission, the training distribution is a mixture with clean mass $(1-\alpha)$ and attempted poison mass α. After admission, clean records contribute mass $(1-\alpha)q_c$, while attempted poisons contribute mass αq_p. Normalizing the admitted poison mass by the total admitted mass gives Equation (7). □

O.2 ChemBack Admissibility by Construction

The second result formalizes why **ChemBack** reports EPR $= 100\%$ for submitted poisons. This does not mean that every attempt to anchor an internal motif succeeds. Rather, invalid internal attempts are rejected during search; only admitted poisoned records are submitted to the poisoned training set.

Proposition O.2 (**ChemBack** submitted poisons pass **ChemGuard**). *Assume the attachment operator* $\mathrm{Att}_{\mathcal{T}}(G, m_k, i, j)$ *returns a molecular record* $(\tilde{s}_p, \tilde{G}_p)$ *only if*

$$\phi_{\mathcal{T}}(\tilde{s}_p) \neq \varnothing \quad \textit{and} \quad \mathrm{Topo}(\phi_{\mathcal{T}}(\tilde{s}_p)) = \mathrm{Topo}(\tilde{G}_p).$$

Otherwise, the operator returns $\perp$. *If the final submitted poisoned set* $D^{\mathrm{sub}}_{\mathrm{poison}}$ *contains only records returned by* $\mathrm{Att}_{\mathcal{T}}$, *then*

$$\mathrm{ChemGuard}_{\mathcal{T}}(\tilde{s}_p, \tilde{G}_p) = 1 \quad \textit{for all } (\tilde{s}_p, \tilde{G}_p, y_t) \in D^{\mathrm{sub}}_{\mathrm{poison}},$$

and therefore

$$\mathrm{EPR} = 1.$$

Proof. By the definition of **ChemGuard**,

$$\mathrm{ChemGuard}_{\mathcal{T}}(\tilde{s}_p, \tilde{G}_p) = \mathbf{1}[\phi_{\mathcal{T}}(\tilde{s}_p) \neq \varnothing] \cdot \mathbf{1}[\mathrm{Topo}(\phi_{\mathcal{T}}(\tilde{s}_p)) = \mathrm{Topo}(\tilde{G}_p)].$$

For every submitted poison generated by $\mathrm{Att}_{\mathcal{T}}$, both indicators are one by assumption. Thus, $\mathrm{ChemGuard}_{\mathcal{T}}(\tilde{s}_p, \tilde{G}_p) = 1$ for every submitted poisoned record. The EPR is the average of this indicator over submitted attempted poisons, so $\mathrm{EPR} = 1$. □

This proposition is intentionally about submitted poisoned records, not about all internal candidate proposals. A search procedure may explore many invalid motif-anchor actions, but **ChemBack** filters them before submission. This distinction is important because EPR measures the poison signal that can enter training, not the number of invalid candidates explored during black-box trigger search.

O.3 Clean-risk Preservation under Rare Triggers

Backdoor attacks aim to preserve clean utility while changing behavior only when the trigger is present. The following simple bound explains why rare molecular triggers can preserve clean accuracy, provided that the trigger predicate rarely appears naturally in clean non-target data.

Let $z(G) \in \{0, 1\}$ denote a trigger predicate, where $z(G) = 1$ if molecule G contains the trigger substructure. Let f_0 be a clean classifier. Define the idealized backdoored classifier

$$f_{\mathrm{bd}}(G) = \begin{cases} y_t, & z(G) = 1, \\ f_0(G), & z(G) = 0. \end{cases}$$

Proposition O.3 (Rare triggers preserve clean risk). *Let* $R_{\mathrm{clean}}(f) = \Pr_{(G,y)\sim P_{\mathrm{clean}}}[f(G) \neq y]$. *Then*

$$R_{\mathrm{clean}}(f_{\mathrm{bd}}) \leq R_{\mathrm{clean}}(f_0) + \Pr_{(G,y)\sim P_{\mathrm{clean}}}[z(G) = 1,\ y \neq y_t].$$

Proof. For any clean sample (G, y), if $z(G) = 0$, then $f_{\mathrm{bd}}(G) = f_0(G)$ and the two classifiers have identical error. If $z(G) = 1$ and $y = y_t$, then $f_{\mathrm{bd}}(G) = y_t = y$, so the backdoored classifier is correct on that sample. The only additional clean error caused by the trigger rule can occur when $z(G) = 1$ and $y \neq y_t$. Therefore, pointwise,

$$\mathbf{1}[f_{\mathrm{bd}}(G) \neq y] \leq \mathbf{1}[f_0(G) \neq y] + \mathbf{1}[z(G) = 1,\ y \neq y_t].$$

Taking expectation over P_{clean} proves the claim. □

This result justifies using rare motifs to reduce the occurrence of accidental clean triggers. However, rarity alone is not a stealth guarantee. A rare motif may still be structurally or representationally outlying. This is why **ChemBack** additionally selects among admitted candidates using Tanimoto similarity to clean target-class molecules.

O.4 Tanimoto Similarity and Representation Proximity

ChemBack does not use a clean victim model during trigger selection. It ranks admitted candidates by fingerprint-based Tanimoto similarity to clean target-class molecules. This subsection explains why such a chemistry-native structural signal can align with the post-hoc MD^2 diagnostic without making MD^2 part of the attack.

Let $\mathrm{FP}(G) \in \{0,1\}^d$ be a molecular fingerprint and define the Tanimoto distance

$$d_{\mathrm{Tan}}(G, G') = 1 - \mathrm{Tan}(\mathrm{FP}(G), \mathrm{FP}(G')).$$

Let $h_\theta(G) \in \mathbb{R}^p$ denote the penultimate representation of a clean reference model used only for diagnostics. For the clean target class y_t, let μ_t and Σ_t be the empirical mean and covariance of clean target embeddings. We use a ridge-regularized covariance

$$\widetilde{\Sigma}_t = \Sigma_t + \rho I, \quad \rho > 0,$$

and define

$$\mathrm{MD}_t(G) = \left\| \widetilde{\Sigma}_t^{-1/2} \left(h_\theta(G) - \mu_t \right) \right\|_2 .$$

Thus $\mathrm{MD}_t^2(G)$ is the squared Mahalanobis distance to the clean target class.

Assumption O.4 (Local fingerprint-representation smoothness)**.** For the molecular neighborhood considered by **ChemBack**, there exists $L > 0$ such that for any admitted molecules G and G',

$$\|h_\theta(G) - h_\theta(G')\|_2 \le L\, d_{\mathrm{Tan}}(G, G').$$

This assumption does not grant the attacker access to h_θ. It is only a post-hoc regularity condition used to explain why structural similarity can align with learned representation proximity.

Proposition O.5 (Tanimoto target similarity implies bounded MD under local smoothness)**.** *Suppose Assumption O.4 holds. Let G_p be a **ChemGuard**-admissible poison. Assume there exists a clean target-class molecule G_z such that*

$$d_{\mathrm{Tan}}(G_p, G_z) \le \epsilon \quad \textit{and} \quad \mathrm{MD}_t^2(G_z) \le r.$$

Let $\lambda_{\min}(\widetilde{\Sigma}_t)$ be the smallest eigenvalue of $\widetilde{\Sigma}_t$. Then

$$\mathrm{MD}_t(G_p) \le \sqrt{r} + \frac{L}{\sqrt{\lambda_{\min}(\widetilde{\Sigma}_t)}}\epsilon,$$

and consequently,

$$\mathrm{MD}_t^2(G_p) \le \left(\sqrt{r} + \frac{L}{\sqrt{\lambda_{\min}(\widetilde{\Sigma}_t)}}\epsilon \right)^2 .$$

Proof. By the triangle inequality in the Mahalanobis norm,

$$\begin{aligned} \mathrm{MD}_t(G_p) &= \left\| \widetilde{\Sigma}_t^{-1/2} (h_\theta(G_p) - \mu_t) \right\|_2 \\ &\le \left\| \widetilde{\Sigma}_t^{-1/2} (h_\theta(G_z) - \mu_t) \right\|_2 + \left\| \widetilde{\Sigma}_t^{-1/2} (h_\theta(G_p) - h_\theta(G_z)) \right\|_2 . \end{aligned}$$

The first term is $\mathrm{MD}_t(G_z) \le \sqrt{r}$. For the second term,

$$\left\| \widetilde{\Sigma}_t^{-1/2} (h_\theta(G_p) - h_\theta(G_z)) \right\|_2 \le \|\widetilde{\Sigma}_t^{-1/2}\|_2 \cdot \|h_\theta(G_p) - h_\theta(G_z)\|_2.$$

Since

$$\|\widetilde{\Sigma}_t^{-1/2}\|_2 = \frac{1}{\sqrt{\lambda_{\min}(\widetilde{\Sigma}_t)}},$$

and by Assumption O.4,

$$\|h_\theta(G_p) - h_\theta(G_z)\|_2 \le L\, d_{\mathrm{Tan}}(G_p, G_z) \le L\epsilon,$$

we obtain

$$\mathrm{MD}_t(G_p) \le \sqrt{r} + \frac{L}{\sqrt{\lambda_{\min}(\widetilde{\Sigma}_t)}}\epsilon.$$

Squaring both sides gives the bound on $\mathrm{MD}_t^2(G_p)$. □

This proposition provides a formal explanation for the post-hoc diagnostic in the main text. If Tanimoto similarity is high, then ϵ is small. Under local smoothness, a structurally target-similar poison is also close to the target region in the clean model's representation space. Importantly, the attack need not know h_θ, μ_t, or Σ_t. These quantities are used only by the evaluator after trigger generation.

P Technical Limitations

This section briefly expands on the limitations discussed in the main conclusion.

ChemGuard is a first-layer admission rule, not a complete defense. ChemGuard formalizes molecular sanitization and graph-string consistency checks that are already implicit in many molecular learning pipelines. It does not certify robustness against all chemistry-valid attacks. It also does not cover all domain-specific screening procedures used in medicinal chemistry or materials discovery.

Toolkit sanitization is not full chemical realism. Passing RDKit, Indigo, or Open Babel sanitization does not imply synthetic accessibility, biological relevance, assay compatibility, or expert approval. These toolkits enforce important but limited notions of molecular validity. Future work should incorporate deeper chemistry constraints, including synthetic accessibility, retrosynthetic feasibility, reaction validity, PAINS filters, medicinal-chemistry rules, and expert review.

ChemBack uses a controlled single-step attachment space. ChemBack currently uses motif attachment through a single chemically valid bond. This controlled design isolates the effect of chemistry-aware admission and makes the attack interpretable. However, richer multi-step or reaction-like transformations may produce broader classes of valid molecular backdoors. Studying such transformations is an important future direction.

Tanimoto similarity is a model-free structural signal, not a universal stealth guarantee. ChemBack uses Tanimoto similarity to select poisons structurally close to the clean target class. Post-hoc diagnostics show alignment with clean-model MD^2 and chemistry-facing descriptors, but these diagnostics do not guarantee human-level stealth or robustness against all possible structural detectors. The stealth claims in this work are therefore limited to the evaluated diagnostics.

Benchmarks are proxies for deployed molecular pipelines. MoleculeNet benchmarks provide standardized evaluation settings, but deployed molecular ML systems may include proprietary validation rules, temporal data shifts, assay-specific curation, human review, or additional domain filters. Our results show that chemistry-aware admission materially changes molecular backdoor evaluation, but further work is needed to study more complex deployment pipelines.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the main claims: the admission gap in molecular backdoor evaluation, the **ChemGuard** admission protocol, and the **ChemBack** attack under a model-free data-poisoning threat model. The claims are supported by the main experiments in Section 4, the discussion in Section 5, and additional analyses in the appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations and scope in Section 5 and provides a more detailed technical discussion in Appendix P. These include the limited scope of **ChemGuard**, the fact that **ChemBack** is model-free but not target-free, limitations of Tanimoto similarity, and the restriction to controlled motif-attachment transformations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The theoretical results are provided in Appendix O. Each proposition states its assumptions explicitly, including the admission rule, submitted-poison construction, rare-trigger condition, and local fingerprint-representation smoothness assumption, and each result is accompanied by a proof.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper specifies datasets, split protocols, evaluation modes, metrics, victim architecture, optimizer, poison rate, baselines, and chemistry-aware admission rules in Section 4 and Appendices C–D. Additional implementation details, search strategies, toolkit validation, sensitivity analyses, and robustness results are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The datasets are public molecular benchmarks, and the ChemBack implementation, installation instructions, experiment configurations, and training and evaluation commands are publicly available at `https://github.com/sail-research/chemback-backdoor-molecule`. However, the release does not include complete implementations of all reported baselines, defenses, and the GROVER setting, as documented in its experiment map. We therefore answer No for complete reproduction of the main experimental comparisons.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 specifies the datasets, split protocols, evaluation modes, victim model, optimizer, poison rate, and baselines. Appendix D provides additional training details, while Appendices F, G, and H describe trigger construction, attachment feasibility, and search strategies.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The main experimental tables report mean±standard deviation over five fixed random seeds. The paper states in Section 4.1 that results are averaged over $S = 5$ seeds, and the same convention is used in the appendix tables and sensitivity analyses.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Appendix H.2 reports representative runtime and resource overhead, including trigger-search time, full runtime, peak GPU memory, and average CPU utilization across the main datasets and search variants. The appendix also distinguishes the overhead of graph-only baselines, RL search, exhaustive search, and greedy search.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The work studies security risks and defenses for molecular machine learning using public benchmark datasets and does not involve human subjects, private data, or deployment on real users. The paper discusses the dual-use nature and limitations of the attack in Section 5 and Appendix P.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The positive impact is to improve the realism of molecular backdoor evaluation and motivate chemistry-aware admission checks. The negative impact is the dual-use nature of backdoor attack research in molecular ML. This is discussed through the motivating scenario in Appendix A, the defense discussion in Section 5, and the limitations in Appendix P.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release a high-risk generative model, scraped dataset, or deployed system. The work uses public molecular benchmarks and studies a dual-use security problem; the risks and mitigation framing are discussed in the paper, but no high-risk model or dataset release requiring access safeguards is introduced.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper cites the original sources for the datasets, models, molecular toolkits, and baseline methods. The released ChemBack code is licensed under MIT, while third-party assets retain their respective licenses. An asset-by-asset inventory of all existing assets' licenses and terms of use is not provided in the manuscript, so we answer No.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released ChemBack software is documented at `https://github.com/sail-research/chemback-backdoor-molecule`, including dependencies, installation, configurations, execution commands, metric definitions, and the scope of supported experiments. The repository includes a license and citation metadata. No new dataset or model checkpoint is released.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: The paper does not involve crowdsourcing, human-subject experiments, user studies, or human annotation.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human subjects, crowdsourcing, or collection of human data, so IRB approval or equivalent human-subjects review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core methodology, experiments, and analysis do not use LLMs as an important, original, or non-standard component. Any use of LLMs for writing, editing, or formatting assistance does not affect the method, experiments, or scientific claims.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.