mathematics

MDPI

Article

Secure and Differentially Private Federated Graph Learning for Molecular Property Prediction

Yumeng You and Jiaxin Chen *

College of Chemistry and Chemical Engineering, Donghua University, Shanghai 201620, China; 231010610@mail.dhu.edu.cn
* Correspondence: chjiaxin@dhu.edu.cn

Abstract

Chemical artificial intelligence increasingly relies on molecular property prediction models trained from proprietary compound libraries, bioassay records, and reaction-screening data. However, these data often contain commercially sensitive structures, confidential activity labels, and privacy-relevant experimental metadata, making direct centralization impractical. This paper proposes PrivMol, a privacy-preserving computational chemistry framework for federated molecular representation learning. PrivMol introduces two novel algorithms: Secure Substructure-Aware Federated Optimization and Differentially Private Molecular Gradient Calibration. The first algorithm decomposes molecular graphs into privacy-sensitive and task-relevant substructure regions, enabling local clients to train graph neural networks while transmitting only securely aggregated model updates. The second algorithm adaptively calibrates clipping and perturbation according to atom- and substructure-level contribution scores, reducing unnecessary utility loss on chemically informative fragments while retaining formal differential privacy guarantees. To improve robustness under heterogeneous chemical spaces, PrivMol incorporates local contrastive molecular alignment without exposing raw molecules, labels, scaffolds, substructure masks, or embeddings. Experimental evaluation on widely used public molecular benchmarks, including ESOL, FreeSolv, Lipophilicity, BBBP, BACE, HIV, and Tox21, demonstrates that PrivMol provides a favorable trade-off among prediction accuracy, communication efficiency, empirical leakage resistance, and privacy protection. The study offers a practical route toward secure collaborative chemical intelligence for computer-aided drug discovery, toxicology prediction, and materials informatics.

Keywords: privacy-preserving machine learning; computational chemistry; molecular property prediction; federated learning; graph neural networks; differential privacy; secure aggregation; drug discovery

MSC: 62R07

Check for updates

Academic Editors: Chunchun Wang and Davide Valenti

Received: 29 May 2026
Revised: 18 June 2026
Accepted: 30 June 2026
Published: 8 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Molecular property prediction is a core problem in computational chemistry, drug discovery, toxicology assessment, and materials informatics. Given a molecular graph $G = (V, E, \mathbf{X}, \mathbf{B})$, the goal is to learn a predictor $f_\theta(G)$ that estimates a physicochemical, pharmacokinetic, bioactivity, or toxicity property from atom and bond information. Public benchmarks such as MoleculeNet standardized tasks including ESOL, FreeSolv, Lipophilicity, BBBP, BACE, HIV, and Tox21 [1]. Graph neural networks are now widely

adopted because they naturally process atoms as nodes and bonds as edges [2–4]. Recent advances further improve representation quality through pre-training, graph transformers, contrastive learning, chemical language modeling, and fragment-level interpretation [5–11].

In practice, high-value molecular datasets are distributed across pharmaceutical companies, hospitals, academic laboratories, contract research organizations, and autonomous synthesis platforms. These datasets may contain unpublished lead compounds, proprietary scaffolds, confidential assay outcomes, toxicological measurements, and failed experiments that are scientifically valuable but commercially sensitive. Direct data pooling is therefore often infeasible. Federated learning offers a promising alternative by allowing clients to train a shared model while retaining private datasets $\mathcal{D}_k$ locally [12,13]. However, standard federated optimization is not sufficient for privacy-sensitive chemistry. Even when raw molecules are not uploaded, gradients may leak information about rare scaffolds, active compounds, assay distributions, and client-specific chemical series.

The privacy challenge is also structural. A full molecule can reveal an unpublished lead compound, but a substructure can reveal a protected pharmacophore, toxicophore, ligand, catalyst motif, or proprietary building block. Existing privacy techniques such as secure aggregation and differential privacy provide important foundations [14–16], yet generic differentially private clipping treats all molecules and fragments uniformly. Uniform noise may destroy common useful signals while still failing to explicitly regulate rare high-value substructures. Meanwhile, federated chemical data are highly heterogeneous: different clients may explore different scaffold families and assay regimes. Federated graph learning research has recognized similar non-IID and structural-shift issues [17–21], and federated molecular learning has shown that scaffold heterogeneity is a major obstacle [22].

This paper proposes PrivMol, a privacy-preserving federated molecular graph learning framework. Each client k owns $\mathcal{D}_k = \{(G_{k,i}, y_{k,i})\}_{i=1}^{N_k}$ and locally extracts chemically meaningful substructures $\mathcal{S}(G) = \{S_1, \ldots, S_m\}$. These substructures are never transmitted; they only guide local learning and privacy calibration. PrivMol introduces two algorithms. Secure Substructure-Aware Federated Optimization (SSFO) combines supervised prediction, local substructure consistency, contrastive molecular alignment, and proximal drift control. Differentially Private Molecular Gradient Calibration (DPMGC) converts atom scores c_v and substructure scores $c(S_r)$ into bounded sensitivity indicators, adaptively clips molecular gradients under a public bound C_0, injects Gaussian noise for (ε, δ)-differential privacy, and then uploads $\widetilde{\Delta}_k^t$ through secure aggregation.

The central idea is that chemical structure should guide both learning and privacy. Substructure-aware models have been used to explain molecular predictions [23,24]; PrivMol uses substructures differently, as local optimization and privacy-calibration units. A rare high-contribution scaffold is clipped more conservatively, whereas common low-risk molecular patterns can still provide useful gradients. PrivMol is therefore not intended as a direct stacking of FedAvg or FedProx, contrastive learning, DP-SGD, and secure aggregation. The framework introduces a molecule–substructure interface in which the same locally extracted chemical units define auxiliary representation learning, bounded contribution scores, and adaptive privacy calibration, while the server sees only securely aggregated noisy updates. This design separates PrivMol from scaffold-aware federated molecular learning, which primarily addresses non-IID scaffold partitions, because PrivMol explicitly treats molecules, scaffolds, and local substructure contributions as protected units. The contributions are threefold. First, we formulate federated molecular property prediction with molecule-level and substructure-level privacy concerns. Second, we propose SSFO and DPMGC as chemically informed algorithms for heterogeneous, private, and secure federated graph learning. Third, we evaluate PrivMol on seven public benchmarks

and show improved privacy–utility trade-offs over FedAvg, FedProx, DP-FedAvg, and secure-aggregation-only variants.

The remainder of this paper is organized as follows. Section 2 reviews molecular representation learning, federated graph learning, and privacy-preserving graph learning. Section 3 defines the notation and threat model. Section 4 presents SSFO, DPMGC, and analysis. Section 5 reports the experimental evaluation. Section 6 concludes the paper.

2. Related Work

2.1. Molecular Representation Learning for Property Prediction

Molecular property prediction has become a central task in computational chemistry, drug discovery, toxicology, and materials science. Early molecular learning methods often relied on hand-crafted descriptors, circular fingerprints, physicochemical features, or substructure counts. Although these descriptors are efficient and chemically interpretable, they are limited by predefined feature templates and may fail to capture task-specific structural patterns. MoleculeNet significantly advanced this area by introducing standardized molecular benchmarks, data splits, and evaluation metrics, making it easier to compare molecular machine learning models across quantum chemistry, biophysics, physiology, and physical chemistry tasks [1]. This benchmark-driven setting motivated the development of graph-based molecular encoders that learn representations directly from atom–bond connectivity.

Graph neural networks have become a dominant approach for molecular property prediction because molecules are naturally represented as graphs. Message Passing Neural Networks (MPNNs) established a general framework in which atom representations are updated by aggregating messages from neighboring atoms and bonds [2]. This formulation provides a flexible abstraction for many molecular GNN architectures and allows chemical information to be propagated across local neighborhoods. Directed message passing and Chemprop-style architectures further showed that propagating messages along directed bonds can improve molecular prediction by reducing redundant atom-level loops and emphasizing bond-centered chemical environments [4]. Theoretical studies on graph neural network expressiveness clarified the relationship between message-passing GNNs and graph isomorphism tests, showing when GNNs can or cannot distinguish different graph structures [3]. These studies motivate our use of a message-passing molecular encoder f_θ as the backbone of PrivMol.

Beyond supervised molecular GNNs, self-supervised and foundation-style molecular learning has rapidly advanced. These methods aim to learn transferable chemical representations from large unlabeled or weakly labeled molecular corpora. GROVER pre-trained a graph transformer using contextual property prediction and motif-level prediction objectives, demonstrating that chemical motif information can improve downstream molecular tasks [5]. Molecular contrastive learning improved transfer by constructing augmented molecular graph views and encouraging representation consistency across related views [6]. Graphium introduced large-scale multi-task molecular datasets, model implementations, and tooling for graph foundation models, enabling systematic studies of pre-training and transfer in molecular graph learning [8]. MolE combined graph-transformer pretraining with biological multitask learning to improve molecular representation learning in biologically relevant settings [7]. MolTRES incorporated generator–discriminator training and external scientific knowledge for chemical language representation learning [9]. CheMeleon studied descriptor-based pre-training for molecular property prediction, showing that descriptor information can still provide strong inductive bias when integrated with modern learning pipelines [10]. RoFt-Mol benchmarked robust fine-tuning strategies for molecular

graph foundation models and highlighted the importance of reliable adaptation under distribution shifts [11].

Although these pre-training and foundation-style approaches have improved molecular representation learning, most of them assume that training data can be centrally collected or that pre-training data can be pooled at scale. This assumption is often unrealistic in pharmaceutical and biomedical settings, where molecular structures, biological assay labels, scaffolds, and lead series may be proprietary or regulated. Different institutions may own complementary chemical libraries but be unable to share raw data due to intellectual property, privacy, contractual, or regulatory constraints. PrivMol differs from centralized molecular foundation models by assuming that downstream molecular data remain distributed across isolated clients. Instead of pooling molecular graphs, PrivMol enables collaborative training through privacy-preserving federated optimization.

Recent architecture and knowledge-integration studies also inform the design of PrivMol. Chain-aware GNNs encode molecular chains and long-range path information, which can be important for capturing dependencies that are not fully represented by shallow local neighborhoods [25]. MPCD integrates common and domain knowledge in a multitask graph transformer, showing that external chemical knowledge and task-specific information can improve molecular prediction [26]. Kolmogorov–Arnold graph neural networks improve expressivity and interpretability for molecular property prediction by introducing more flexible functional transformations [27]. Multilevel fusion methods combine local graph features, global graph representations, and molecular fingerprints to exploit complementary structural signals [28]. Knowledge-guided graph self-supervised learning injects orbital-level chemical knowledge into molecular representation learning, demonstrating the value of chemically informed pre-training objectives [29]. Improved graph transformer networks with multitask learning have also been proposed for molecular property prediction, further emphasizing the usefulness of attention-based architectures and shared representations across related endpoints [30]. In addition, studies using high-quality experimental datasets have examined the reliability and practical utility of GNNs for molecular prediction [31].

A closely related direction is interpretable molecular learning. In molecular science, accurate predictions are not sufficient; chemists often need to know which atoms, bonds, fragments, or scaffolds contribute to a prediction. Interpretability methods such as Substructure Mask Explanation and FragNet show that chemically meaningful substructures can serve as explanation units and can reveal functional groups or fragments that drive model decisions [23,24]. These works support the view that substructures are not merely auxiliary features but important semantic units in molecular prediction. PrivMol adopts this chemical insight in a different way: instead of using substructures only for post hoc explanation, it uses atom-level and substructure-level signals to guide private optimization and privacy calibration inside each client. In this sense, PrivMol connects interpretable molecular learning with privacy-preserving federated training.

2.2. Federated and Privacy-Preserving Graph Learning

Federated learning enables multiple clients to collaboratively train a model without directly sharing their local data. FedAvg is the standard federated optimization algorithm in which clients perform local updates and the server averages their model updates [12]. FedProx extends this setting by adding a proximal term to stabilize local training under statistical heterogeneity [13]. These methods are especially relevant for molecular learning because participating organizations may have private chemical libraries, assay results, or biological activity labels that cannot be centralized. However, molecular federated learning is more difficult than standard federated learning because molecular data are highly heterogeneous

across clients. Different clients may focus on different target families, scaffold series, compound sizes, experimental protocols, or label distributions. Several stronger non-IID and personalized federated optimizers have been proposed, including SCAFFOLD with control variates, FedDyn with dynamic regularization, FedBN with local normalization, and FedAMP with attentive personalized collaboration [32–35]. These algorithms mainly target optimization drift, feature shift, or personalization under heterogeneous clients; they do not by themselves define chemically meaningful protected substructure units or molecularly calibrated differentially private clipping. Similarly, adaptive clipping and client-level differentially private FL provide important privacy baselines [36,37], but they adapt clipping at the update or client level rather than by atom and substructure contribution. PrivMol is complementary to these methods: DPMGC could be coupled with stronger optimizers in future work, while the present experiments focus on isolating molecular substructure-aware privacy calibration against standard federated and DP-FedAvg baselines.

Federated graph learning generalizes federated learning to graph-structured data, where privacy and heterogeneity arise not only from features and labels but also from graph topology. Surveys of federated graph neural networks show that graph data introduce unique challenges, including structural non-IID distributions, topology leakage, missing cross-client edges, client-specific graph domains, and graph-level privacy risks [17]. OpenFGL provides a large benchmark and unified evaluation environment for federated graph learning, supporting systematic comparison of algorithms under different graph tasks and heterogeneity settings [19]. Federated semantic and structural learning studies how node-level and graph-level distortions affect federated graph models and how semantic and structural information can be aligned across clients [18]. Personalized and privacy-preserving federated graph neural networks address client-specific graph distributions by allowing local adaptation while still benefiting from collaborative training [20]. Personalized federated learning on graph-structured data further explores collaboration graphs among clients so that related clients can share more useful information while reducing negative transfer from dissimilar clients [21].

Molecular federated learning has additional chemical challenges beyond general graph federated learning. In molecular datasets, the graph instances are usually independent molecules rather than nodes in one large social or citation graph. The main distribution shifts often come from scaffold heterogeneity, assay heterogeneity, and chemical library bias. FedChem showed that scaffold heterogeneity can substantially degrade standard federated molecular prediction and proposed reweighting strategies to improve learning under molecular non-IID distributions [22]. This observation is important because scaffold splits are widely used to evaluate molecular generalization, and clients in real-world collaborative drug discovery may naturally correspond to different scaffold families or chemical programs. PrivMol builds on this line of work but focuses on a stronger privacy setting. It assumes that client updates may reveal chemical information and therefore introduces explicit privacy calibration based on molecule-level and substructure-level contributions.

Privacy-preserving graph learning remains challenging because graph data are inherently relational. In ordinary tabular data, a record may be treated as an independent protected unit. In graph data, however, one node, edge, molecule, or substructure can influence multiple messages, embeddings, and gradients. As a result, structural patterns can leak through learned representations or model updates. Secure aggregation is a practical cryptographic technique that prevents the server from observing individual client updates by allowing it to recover only the aggregate update [14]. This is useful in federated molecular learning because an individual update may contain signals about a client's private chemical space. However, secure aggregation alone does not provide a formal bound on what can be inferred from the aggregate or from the sequence of released global models.

Differential privacy provides a complementary formal privacy guarantee by bounding the influence of a protected unit on the output of a randomized mechanism [15]. In deep learning, differentially private stochastic gradient descent clips per-example gradients and adds calibrated noise to limit the contribution of any individual training example [16]. This mechanism has inspired many privacy-preserving learning systems, but applying it to graph and molecular learning is nontrivial. The protected unit may be a node, an edge, an entire graph, a molecule, a scaffold, or a substructure-level contribution. Moreover, graph neural networks propagate information through neighborhoods, so the sensitivity of a gradient may depend on local structure and message-passing depth. In molecular settings, the challenge is further complicated by the fact that chemically meaningful fragments, such as rings or functional groups, may be more sensitive than individual atoms because they can reveal proprietary scaffold information.

Recent studies have examined privacy threats and defenses in federated and graph neural network settings. Surveys of privacy in graph neural networks emphasize that attacks on graph models go beyond ordinary membership inference and may include structure inference, property inference, edge inference, and reconstruction attacks [38]. Local differential privacy has been used to protect federated graph neural networks by perturbing client-side information before it is shared [39]. PPFedGNN studies secure privacy-preserving federated graph learning for social networks, where relational structure and user attributes must be protected [40]. Vertical federated graph convolutional networks consider settings in which graph features are vertically partitioned across parties, requiring privacy mechanisms for collaborative representation learning without exposing raw feature views [41]. Property inference attacks against federated graph neural networks show that model updates may reveal sensitive graph-level or client-level properties even when raw data are never shared [42]. These findings motivate a layered privacy design in which update visibility is restricted and the influence of private data on released models is formally bounded.

PrivMol follows this layered design. Secure aggregation prevents the server from directly inspecting individual protected updates, while Differentially Private Molecular Gradient Calibration limits the influence of molecules and substructure-level contributions on the released aggregate. This combination is important because the two mechanisms address different leakage channels. Secure aggregation hides individual client updates but does not by itself limit what can be inferred from the aggregate. Differential privacy limits the contribution of protected units but may still expose noisy individual updates if used without secure aggregation. By combining both, PrivMol reduces the risk of direct update inspection, membership inference, property inference, and substructure leakage in federated molecular prediction.

2.3. Position of PrivMol

PrivMol sits at the intersection of molecular graph learning, federated graph learning, and privacy-preserving optimization. Compared with molecular encoders such as D-MPNN, GROVER, Chain-aware GNN, MPCD, and KA-GNN [4,5,25–27], PrivMol is not only a molecular representation model but a complete federated training framework for isolated clients. These molecular encoders improve how molecules are represented, how messages are propagated, or how chemical knowledge is incorporated, but they generally do not address the setting in which multiple institutions collaboratively train without revealing raw graphs, labels, scaffolds, or substructures. PrivMol can use such molecular encoders as backbones, while adding privacy-preserving communication and substructure-aware calibration around them.

Compared with molecular foundation models and pre-training frameworks such as Graphium, MolE, MolTRES, CheMeleon, and RoFt-Mol [7–11], PrivMol does not assume centralized access to large molecular datasets or unrestricted pooling of downstream task data. Foundation models are powerful because they can exploit large-scale pre-training and transfer learning, but their deployment in sensitive pharmaceutical or biomedical collaborations may still require privacy-preserving adaptation. PrivMol addresses this complementary problem by enabling clients to train or fine-tune molecular predictors while keeping proprietary molecular graphs and labels local. Thus, PrivMol can be viewed as a privacy-preserving framework that can potentially support both training from scratch and federated adaptation of pretrained molecular encoders.

Compared with general federated graph learning methods [17–21], PrivMol is tailored to molecular graph-level prediction. General federated graph methods often focus on social, citation, recommendation, or node-classification graphs, where cross-client edges, node identities, and relational topology are central issues. In contrast, PrivMol focuses on molecular graphs, where each graph corresponds to a molecule and sensitive information is often encoded in atom types, bond patterns, scaffolds, functional groups, and assay labels. This difference changes both the learning objective and the privacy objective. In PrivMol, the local chemical substructure set $\mathcal{S}(G)$ provides a chemically meaningful layer that is absent from most generic graph federated learning methods.

Compared with FedChem [22], PrivMol addresses a broader privacy objective. FedChem highlights the importance of scaffold heterogeneity and shows that molecular non-IID distributions can harm federated optimization. PrivMol shares this motivation but additionally treats molecules, scaffolds, and substructures as sensitive information that may be inferred from updates. It therefore introduces local atom and substructure scores, denoted by c_v and $c(S_r)$, to guide private optimization and calibration. These scores remain inside each client and are used to construct protected updates rather than being transmitted as explicit chemical evidence. This distinction is conceptual as well as algorithmic. Scaffold-aware FL primarily changes how clients are partitioned or regularized under scaffold shift, whereas PrivMol changes the protected unit and the clipping rule: a molecule-level gradient is first decomposed through local chemical fragments, converted into a bounded sensitivity indicator $\omega_{k,i}$, clipped by a public worst-case bound, noised for differential privacy, and then hidden by secure aggregation. Thus, the framework protects against update-level exposure of rare local fragments even when the global task is only graph-level property prediction [43–46].

Compared with privacy-preserving graph methods such as local-DP federated GNNs, PPFedGNN, vertical federated GCNs, and property-inference-aware defenses [39–42], PrivMol introduces a chemical layer into privacy-preserving graph learning. Existing methods mainly protect generic graph attributes, node features, edges, or client properties. PrivMol instead recognizes that in molecular learning, sensitive information may correspond to chemically meaningful motifs, such as functional groups, rings, scaffolds, or pharmacophoric fragments. By defining local substructures $\mathcal{S}(G)$ and using atom-level and substructure-level scores c_v, $c(S_r)$, and $\omega_{k,i}$, PrivMol aligns privacy calibration with chemical semantics.

3. Preliminaries

3.1. Molecular Graph Representation and Prediction Task

A molecule is represented as an attributed graph $G = (V, E, \mathbf{X}, \mathbf{B})$, where $V = \{v_1, \ldots, v_n\}$ is the atom set, $E \subseteq V \times V$ is the bond set, $\mathbf{X} \in \mathbb{R}^{n \times d_a}$ contains atom features, and $\mathbf{B} \in \mathbb{R}^{|E| \times d_b}$ contains bond features. Atom features include atom type, formal charge, aromaticity, chirality, hybridization, and hydrogen count; bond features include bond type,

conjugation, ring membership, and stereochemistry. The label y may be continuous for regression or binary/multi-label for classification.

A graph neural network computes atom states from $\mathbf{h}_v^{(0)} = \mathbf{x}_v$. At layer l, $\mathbf{h}_v^{(l)} = \mathrm{UPD}^{(l)}(\mathbf{h}_v^{(l-1)}, \mathrm{AGG}^{(l)}(\{\mathbf{h}_u^{(l-1)}, \mathbf{b}_{uv} : u \in \mathcal{N}(v)\}))$, where $\mathcal{N}(v)$ is the neighbor set. After L layers, a readout produces $\mathbf{z}_G = \mathrm{READOUT}(\{\mathbf{h}_v^{(L)} : v \in V\})$, and the prediction is $f_\theta(G) = q_\theta(\mathbf{z}_G)$. For $\mathcal{D} = \{(G_i, y_i)\}_{i=1}^N$, the empirical objective is $\mathcal{L}(\theta; \mathcal{D}) = N^{-1} \sum_{i=1}^N \ell(f_\theta(G_i), y_i)$.

3.2. Federated Molecular Learning Setting

PrivMol considers K clients. Client k owns $\mathcal{D}_k = \{(G_{k,i}, y_{k,i})\}_{i=1}^{N_k}$, and the global dataset is $\mathcal{D} = \bigcup_{k=1}^K \mathcal{D}_k$ with $N = \sum_k N_k$. Raw molecular graphs, labels, scaffolds, substructure identities, and metadata are never uploaded. At communication round t, the server broadcasts θ^t to selected clients $\mathcal{C}^t$. Client k locally optimizes from θ^t and forms $\Delta_k^t = \theta_k^t - \theta^t$. Standard aggregation would use $\theta^{t+1} = \theta^t + \sum_{k \in \mathcal{C}^t} \alpha_k^t \Delta_k^t$, but direct updates may leak chemical information.

Federated molecular data are typically non-IID. Local distributions $\pi_k(G, y)$ may differ in scaffold families, functional groups, molecular sizes, assay protocols, and label balance. We define $\mathcal{S}(G) = \{S_1, \ldots, S_m\}$ as the chemically meaningful substructures extracted from G, where each S_r is a connected atom-induced subgraph. In PrivMol, $\mathcal{S}(G)$ is computed and used only locally. This allows substructures to guide learning and privacy calibration without exposing scaffold-level evidence. In our implementation, $\mathcal{S}(G)$ is the deterministic union of four fragment families: the Bemis–Murcko scaffold when present, ring systems from the molecular cycle basis, functional groups matched by a fixed SMARTS dictionary, and BRICS-type cleavable-bond fragments or radius-one atom neighborhoods for acyclic regions [47,48]. Fragments with fewer than two heavy atoms are discarded, duplicate atom sets are merged, overlapping fragments are retained only when their Jaccard overlap is below 0.8, and at most $m_{\max} = 16$ fragments are kept per molecule by sorting first by heavy-atom size and then by local rarity. If no valid fragment is detected, the whole molecule is used as S_1. These rules are fixed before training and use only each client's local molecular graph.

3.3. Privacy Objective and Threat Model

The server is honest but curious: it follows the protocol but may attempt to infer client molecules, labels, scaffolds, or substructures from updates and model snapshots. Other clients may also attempt to infer whether another client owns a molecule or chemical motif. PrivMol combines secure aggregation and differential privacy. Secure aggregation lets the server observe only $\sum_{k \in \mathcal{C}^t} \widetilde{\Delta}_k^t$, not each individual $\widetilde{\Delta}_k^t$. Differential privacy bounds the influence of a protected unit on released models.

A randomized mechanism $\mathcal{M}$ is (ε, δ)-differentially private if, for neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ differing in one protected unit and any output set $\mathcal{O}$, $\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{O}] \leq e^\varepsilon \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{O}] + \delta$. The protected unit may be a molecule (G, y) or a masked substructure-level contribution. DPMGC clips and perturbs updates as $\widetilde{\Delta}_k^t = \bar{\Delta}_k^t + \boldsymbol{\xi}_k^t$, using atom scores c_v and substructure scores $c(S_r)$ to calibrate clipping while retaining a public sensitivity bound.

4. Methodology

4.1. Overview of PrivMol

PrivMol is designed for federated molecular property prediction when each client owns confidential molecules and labels. At round t, the server broadcasts θ^t to selected clients $\mathcal{C}^t$. Each client optimizes locally on $\mathcal{D}_k$, computes a calibrated update

$\widetilde{\Delta}_k^t$, and submits it through secure aggregation. The server updates the model as $\theta^{t+1} = \theta^t + \eta_t \sum_{k \in \mathcal{C}^t} \alpha_k^t \widetilde{\Delta}_k^t$, where η_t is the server learning rate and α_k^t is the aggregation weight. Figure 1 gives the overall PrivMol pipeline and shows that raw molecules, labels, scaffolds, substructure masks, contribution scores, and embeddings never leave the local clients.

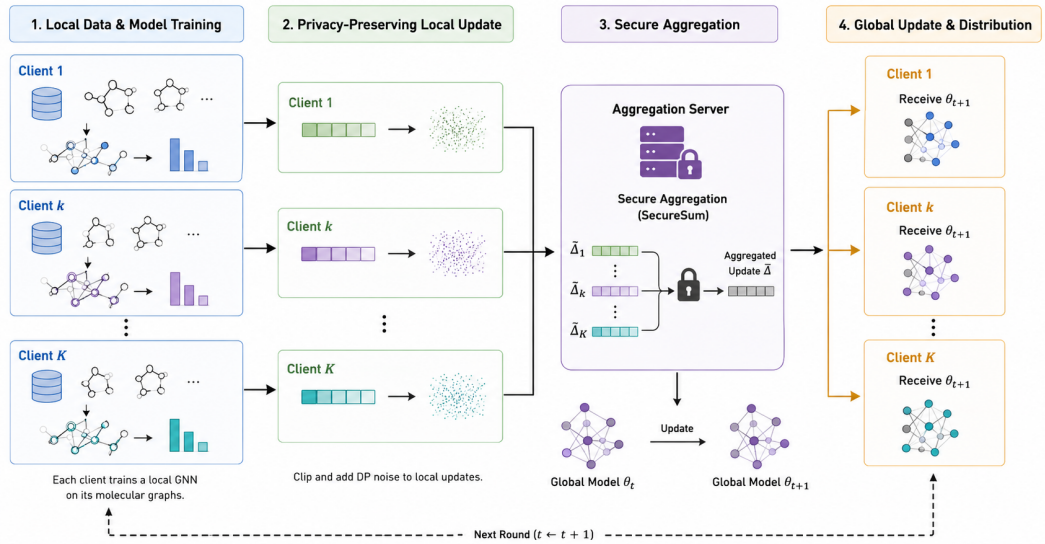

Figure 1. General pipeline of PrivMol. Each client keeps molecular graphs, labels, scaffolds, and substructure information locally. Atom-level and substructure-level information guide local training and DPMGC-based privacy calibration. The protected updates are transmitted through secure aggregation, and the server updates the global model using only the aggregated noisy update.

The framework is built around local chemical substructures. For molecule G, client k extracts $\mathcal{S}(G) = \{S_1, \ldots, S_m\}$ using ring systems, functional groups, scaffold fragments, or radius-bounded atom neighborhoods. The molecular embedding is $\mathbf{z}_G = \mathrm{READOUT}(\{\mathbf{h}_v^{(L)} : v \in V\})$, and the embedding of substructure S_r is $\mathbf{z}_{S_r} = \mathrm{READOUT}_s(\{\mathbf{h}_v^{(L)} : v \in V(S_r)\})$. These embeddings support prediction and local calibration, but they are not uploaded.

The local objective combines supervised prediction, substructure consistency, contrastive alignment, and proximal drift control: $\mathcal{J}_k^t(\theta) = \mathcal{L}_k^{\mathrm{sup}}(\theta) + \lambda_s \mathcal{L}_k^{\mathrm{sub}}(\theta) + \lambda_c \mathcal{L}_k^{\mathrm{con}}(\theta) + \frac{\mu}{2}\|\theta - \theta^t\|_2^2$. Here $\mathcal{L}_k^{\mathrm{sup}} = N_k^{-1} \sum_i \ell(f_\theta(G_{k,i}), y_{k,i})$. The substructure term helps preserve local chemical evidence, the contrastive term stabilizes representations under client heterogeneity, and the proximal term limits drift from the global model. The contribution scores are not extra information sent to the server. They are recomputed locally from the current mini-batch and model state during training, used immediately to choose a bounded clipping radius, and then discarded or retained only in local moving averages for stable normalization.

Algorithm 1 shows that all chemically interpretable quantities remain local. The server receives neither scaffold identities nor atom-level scores. This separation is essential because a rare substructure may reveal a lead series even if the full molecule is not transmitted.

4.2. Secure Substructure-Aware Federated Optimization

SSFO trains the molecular encoder while accounting for client heterogeneity and chemical locality. For each substructure $S_{k,i,r} \in \mathcal{S}(G_{k,i})$, client k computes a contribution score $c(S_{k,i,r})$ and converts scores into weights $w_{k,i,r} = \exp(c(S_{k,i,r})/\tau_s) / \sum_q \exp(c(S_{k,i,q})/\tau_s)$, where τ_s is a temperature. These weights guide local learning but are never released.

Algorithm 1 PrivMol Training Overview

Require: Clients $\{1, \ldots, K\}$, private datasets $\{\mathcal{D}_k\}_{k=1}^{K}$, initial model θ^0, rounds T, local steps E, target (ε, δ)
Ensure: Global predictor f_{θ^T}

```
for t = 0 to T − 1 do
    The server samples C^t and broadcasts θ^t.
    for each client k ∈ C^t in parallel do
        Extract local substructures S(G_{k,i}) for molecules in D_k.
        Optimize J_k^t(θ) for E local steps.
        Estimate atom scores c_v and substructure scores c(S_r).
        Apply DPMGC to obtain Δ̃_k^t.
        Submit Δ̃_k^t through secure aggregation.
    end for
    The server obtains only Σ_{k∈C^t} Δ̃_k^t and updates θ^{t+1}.
end for
return f_{θ^T}
```

The substructure consistency loss is $\mathcal{L}_k^{\text{sub}}(\theta) = N_k^{-1} \sum_i \sum_{r=1}^{m_{k,i}} w_{k,i,r} \ell(g_\theta(S_{k,i,r}), y_{k,i})$, where g_θ is a lightweight substructure head. This term does not require isolated fragments to determine the property; rather, it encourages the encoder to retain information from fragments repeatedly associated with the molecular label. For example, charged groups, heterocycles, aromatic rings, and hydrogen-bond donors may receive high local relevance in solubility, permeability, toxicity, or activity tasks.

The contrastive alignment term uses two chemically valid augmented views $G_{k,i}^a$ and $G_{k,i}^b$, generated by atom-feature masking, bond-feature masking, or substructure-preserving perturbation. With projected vectors $\mathbf{u}_{k,i}^a$ and $\mathbf{u}_{k,i}^b$, $\mathcal{L}_k^{\text{con}} = -|\mathcal{B}_k|^{-1} \sum_{i \in \mathcal{B}_k} \log \frac{\exp(\text{sim}(\mathbf{u}_{k,i}^a, \mathbf{u}_{k,i}^b)/\tau_c)}{\sum_{j \in \mathcal{B}_k} \exp(\text{sim}(\mathbf{u}_{k,i}^a, \mathbf{u}_{k,j}^b)/\tau_c)}$. Since all views and embeddings stay within the client, this alignment does not create a new communication channel.

After E local steps, client k obtains $\Delta_k^t = \theta_k^{t,E} - \theta^t$. SSFO never uploads Δ_k^t directly. Instead, DPMGC converts it into $\widetilde{\Delta}_k^t$, and secure aggregation masks it. Pairwise masks cancel after summation, so the server obtains $\sum_{k \in \mathcal{C}^t} \widetilde{\Delta}_k^t$ but cannot isolate any individual update.

Algorithm 2 is chemically informed but privacy-preserving. Substructure information improves the local update before perturbation, while secure aggregation ensures that no individual calibrated update is visible to the server.

Algorithm 2 Secure Substructure-Aware Federated Optimization

Require: Global model θ^t, selected clients $\mathcal{C}^t$, local steps E, local learning rate η_ℓ
Ensure: Protected aggregate update for round t

```
for each client k ∈ C^t in parallel do
    Set θ_k^{t,0} = θ^t.
    Locally extract S(G_{k,i}) and masks for each molecule in D_k.
    Compute z_{G_{k,i}}, z_{S_{k,i,r}}, c(S_{k,i,r}), and w_{k,i,r}.
    for e = 0 to E − 1 do
        Construct augmented molecular views for contrastive alignment.
        Minimize J_k^t(θ).
        Update θ_k^{t,e+1} = θ_k^{t,e} − η_ℓ ∇_θ J_k^t(θ_k^{t,e}).
    end for
    Compute Δ_k^t = θ_k^{t,E} − θ^t.
    Apply DPMGC and secure aggregation masking to upload a protected update.
end for
```

4.3. Differentially Private Molecular Gradient Calibration

DPMGC transforms unprotected local gradients into private calibrated gradients. For molecule $G_{k,i}$ in mini-batch $\mathcal{B}_k^t$, let $\mathbf{g}_{k,i}^t = \nabla_\theta \mathcal{J}_{k,i}^t(\theta)$ be its per-molecule gradient. Atom scores combine gradient saliency and readout contribution. Specifically, the saliency part is $s_v = \|\partial \ell_{k,i}/\partial \mathbf{h}_v^{(L)}\|_2 / (\sum_{u \in V(G_{k,i})} \|\partial \ell_{k,i}/\partial \mathbf{h}_u^{(L)}\|_2 + \epsilon_{\text{num}})$, and the readout part is $a_v = \|\partial \mathbf{z}_{G_{k,i}}/\partial \mathbf{h}_v^{(L)}\|_1 / (\sum_{u \in V(G_{k,i})} \|\partial \mathbf{z}_{G_{k,i}}/\partial \mathbf{h}_u^{(L)}\|_1 + \epsilon_{\text{num}})$. The atom contribution score is $c_v = \rho s_v + (1-\rho) a_v$, where $\rho \in [0,1]$ is fixed before training and ϵ_{num} is a small numerical constant.

For substructure $S_{k,i,r}$, DPMGC computes $c(S_{k,i,r}) = |V(S_{k,i,r})|^{-1} \sum_{v \in V(S_{k,i,r})} c_v + \gamma r(S_{k,i,r})$, where $r(S_{k,i,r}) = \log((N_k + \beta)/(n_k(S_{k,i,r}) + \beta)) / \log((N_k + \beta)/\beta)$ is a locally computed rarity score, $n_k(S_{k,i,r})$ counts matching fragments in client k, and $\beta > 0$ is a smoothing constant. Scores are recomputed during local training rather than fixed once at preprocessing, so they reflect the current encoder while remaining entirely local.

Raw scores may depend on private data, so DPMGC bounds them before they affect any released update. The normalized score is $\bar{c}(S_{k,i,r}) = \text{clip}((c(S_{k,i,r}) - m_k)/s_k, 0, c_{\max})$, where m_k and s_k are local robust statistics and $c_{\max}$ is public. The molecule-level sensitivity indicator is $\omega_{k,i} = 1 + \lambda_\omega \sum_r w_{k,i,r} \bar{c}(S_{k,i,r})$, which satisfies $1 \le \omega_{k,i} \le 1 + \lambda_\omega c_{\max}$. The adaptive clipping bound is $C_{k,i} = C_0/\sqrt{\omega_{k,i}} \le C_0$, so every clipped gradient remains bounded by the public constant C_0 regardless of the private score values.

The clipped gradient is $\bar{\mathbf{g}}_{k,i}^t = \mathbf{g}_{k,i}^t \min(1, C_{k,i}/\|\mathbf{g}_{k,i}^t\|_2)$. After averaging clipped gradients, client k samples $\boldsymbol{\xi}_k^t \sim \mathcal{N}(\mathbf{0}, \sigma_t^2 C_0^2 \mathbf{I}/|\mathcal{B}_k^t|^2)$ and forms $\widetilde{\mathbf{g}}_k^t = |\mathcal{B}_k^t|^{-1} \sum_i \bar{\mathbf{g}}_{k,i}^t + \boldsymbol{\xi}_k^t$. The calibrated update is accumulated across local steps as $\widetilde{\Delta}_k^t = -\eta_\ell \sum_e \widetilde{\mathbf{g}}_k^{t,e}$.

Proposition 1. *Assume each protected molecular contribution is clipped by Algorithm 3 with norm at most the public constant C_0 and Gaussian noise with multiplier σ_t is added to the averaged clipped gradient. For add-or-remove molecule adjacency, the ℓ_2 sensitivity of a mini-batch average is at most $C_0/|\mathcal{B}_k^t|$; for replace-one adjacency, it is at most $2C_0/|\mathcal{B}_k^t|$. Therefore, for a fixed sampling rate and fixed number of local optimization steps, the released sequence of calibrated updates satisfies (ε, δ)-differential privacy under the Gaussian accountant used by PrivMol. The same statement applies to masked substructure-level adjacency because each masked contribution is clipped inside the same public C_0 ball.*

Algorithm 3 Differentially Private Molecular Gradient Calibration

Require: Mini-batch $\mathcal{B}_k^t$, base bound C_0, noise multiplier σ_t, score bound $c_{\max}$, calibration strength λ_ω
Ensure: Private gradient $\widetilde{\mathbf{g}}_k^t$

```
for each molecule G_{k,i} ∈ B_k^t do
    Compute atom scores c_v and substructure scores c(S_{k,i,r}).
    Normalize scores as c̄(S_{k,i,r}) = clip((c(S_{k,i,r}) − m_k)/s_k, 0, c_max).
    Compute ω_{k,i} = 1 + λ_ω Σ_r w_{k,i,r} c̄(S_{k,i,r}) and C_{k,i} = C_0/√ω_{k,i}.
    Clip g_{k,i}^t to obtain ḡ_{k,i}^t.
end for
Average clipped gradients and add Gaussian noise with multiplier σ_t.
return g̃_k^t
```

The data-dependent scores do not create an additional released channel. They are computed locally, are not transmitted, and only determine a deterministic clipped vector whose norm is still bounded by C_0. Thus, for any two neighboring local datasets, the released vector is the Gaussian mechanism applied to a function with the same public worst-case sensitivity bound. Secure aggregation is applied after perturbation and is a post-processing step, so it cannot weaken the differential privacy guarantee.

4.4. Privacy, Utility, and Complexity Analysis

The protection of PrivMol is layered. DPMGC bounds the influence of each molecule or protected substructure contribution before noise injection, while secure aggregation hides individual calibrated updates. Under add-or-remove adjacency, the sensitivity of the averaged clipped gradient is at most $C_0/|\mathcal{B}_k^t|$; under replace-one adjacency, it is at most $2C_0/|\mathcal{B}_k^t|$. Secure aggregation is post-processing of already perturbed updates, so it cannot weaken the differential privacy guarantee.

Utility improves because privacy control is aligned with chemical structure. SSFO extracts useful substructure evidence before update release, while DPMGC prevents rare high-gradient scaffolds from dominating the released update. Let $\mathcal{J}(\theta) = \sum_k (N_k/N)\mathcal{J}_k(\theta)$ be the global objective without privacy noise. The update can be written as $\theta^{t+1} = \theta^t + \eta_t \sum_{k\in\mathcal{C}^t} \alpha_k^t(\bar{\Delta}_k^t + \boldsymbol{\xi}_k^t)$, where $\bar{\Delta}_k^t$ is clipped and $\boldsymbol{\xi}_k^t$ is privacy noise. To make the descent terms explicit, define the client heterogeneity at round t as $H_t = \sum_{k=1}^K p_k \|\nabla\mathcal{J}_k(\theta^t) - \nabla\mathcal{J}(\theta^t)\|_2^2$, where $p_k = N_k/N$, and define the clipping bias as $b_t = \mathbb{E}[\bar{\mathbf{g}}^t \mid \theta^t] - \nabla\mathcal{J}(\theta^t)$. Let $q = |\mathcal{C}^t|/K$ be the client participation rate, $B = |\mathcal{B}_k^t|$ the mini-batch size, E the number of local epochs, and $P = |\theta|$ the number of trainable parameters.

Theorem 1. *Assume $\mathcal{J}$ is L-smooth and lower bounded by $\mathcal{J}^\star$, unclipped stochastic gradients are unbiased with variance at most ν^2, the selected clients are sampled uniformly with participation rate q, and Algorithm 3 clips each protected contribution by the public bound C_0. For sufficiently small local and server learning rates, the iterates generated by Algorithms 2 and 3 satisfy the non-convex stationarity bound:*

$$T^{-1}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla\mathcal{J}(\theta^t)\|_2^2 \le O\left(\frac{\mathcal{J}(\theta^0)-\mathcal{J}^\star}{\eta T}\right) + O\left(\frac{\eta L\nu^2}{qKB}\right) + O\left(\frac{\eta LP\sigma_t^2 C_0^2}{qKB^2}\right) + O\left(E^2\eta_\ell^2\bar{H}\right) + O\left(T^{-1}\sum_t \mathbb{E}\|b_t\|_2^2\right), \tag{1}$$

where $\bar{H} = T^{-1}\sum_t H_t$. The four additive terms correspond to stochastic optimization error, Gaussian privacy noise, local-update/client heterogeneity amplified by E, and clipping bias.

Proof. By L-smoothness, one-step descent is bounded by the inner product between $\nabla\mathcal{J}(\theta^t)$ and the expected aggregated update plus a quadratic variance term. Decomposing the update into the true global gradient, the stochastic-gradient error, client-sampling error, local drift over E steps, clipping bias b_t, and Gaussian noise gives the displayed terms. The privacy-noise variance is $P\sigma_t^2 C_0^2/B^2$ per selected client update and is reduced by aggregation over approximately qK clients. The local drift term follows from smoothness and bounded client heterogeneity; the proximal penalty μ and contrastive alignment weight λ_c reduce this term empirically by limiting client drift and representation disagreement. This is a standard non-convex descent argument for clipped noisy federated SGD, adapted here to the public C_0 bound of DPMGC [32,49]. □

The computational cost is close to ordinary federated graph neural network training. For a molecule with $|V|$ atoms and $|E|$ bonds, an L-layer encoder with hidden dimension d costs $O(L(|V|+|E|)d^2)$. If at most m substructures are pooled per molecule, masked pooling adds $O(m|V|d)$ in the dense worst case and less with sparse masks. Communication remains $O(P)$ per selected client per round, where $P = |\theta|$, because clients upload one protected update vector and do not upload substructures, fingerprints, embeddings, or labels.

5. Experiments

5.1. Experimental Setting

The experiments evaluate whether PrivMol improves molecular property prediction when chemical datasets are distributed across privacy-sensitive clients. We use seven public datasets from MoleculeNet and DeepChem. ESOL, FreeSolv, and Lipophilicity are regression tasks evaluated by RMSE; BBBP, BACE, HIV, and Tox21 are classification or multi-label classification tasks evaluated by ROC-AUC. Table 1 lists the dataset setting and public sources. Molecules are converted from SMILES strings into graphs using the atom and bond descriptors defined in Section 3.

Table 1. Dataset configuration used in the experiments. Regression tasks are evaluated by RMSE, and classification tasks are evaluated by ROC-AUC.

Dataset	Task Type	Chemical Objective	Molecules	Metric	Public Source URL
ESOL	Regression	Aqueous solubility	1128	RMSE ↓	https://deepchemdata.s3-us-west-1.amazonaws.com/datasets/delaney-processed.csv (accessed on 2 May 2026)
FreeSolv	Regression	Hydration free energy	642	RMSE ↓	https://deepchemdata.s3-us-west-1.amazonaws.com/datasets/SAMPL.csv (accessed on 2 May 2026)
Lipophilicity	Regression	Octanol/water distribution	4200	RMSE ↓	https://deepchemdata.s3-us-west-1.amazonaws.com/datasets/Lipophilicity.csv (accessed on 2 May 2026)
BBBP	Classification	Blood–brain barrier penetration	2039	ROC-AUC ↑	https://deepchemdata.s3-us-west-1.amazonaws.com/datasets/BBBP.csv (accessed on 2 May 2026)
BACE	Classification	BACE inhibitor activity	1513	ROC-AUC ↑	https://deepchemdata.s3-us-west-1.amazonaws.com/datasets/bace.csv (accessed on 2 May 2026)
HIV	Classification	HIV replication inhibition	41,127	ROC-AUC ↑	https://deepchemdata.s3-us-west-1.amazonaws.com/datasets/HIV.csv (accessed on 2 May 2026)
Tox21	Multi-label classification	Toxicity over 12 targets	7831	ROC-AUC ↑	https://deepchemdata.s3-us-west-1.amazonaws.com/datasets/tox21.csv.gz (accessed on 2 May 2026)

Each dataset uses an 80/10/10 scaffold split. The training partition is then distributed to $K = 10$ clients by scaffold-aware Dirichlet allocation with concentration $\alpha_{\text{dir}} = 0.5$ unless otherwise stated. Each communication round selects $|\mathcal{C}^t| = 5$ clients. The graph encoder has five message-passing layers, hidden dimension $d = 300$, attention pooling, and a two-layer prediction head. Table 2 reports the default federated, model, optimization, and privacy hyperparameters.

Figure 2 visualizes the dataset-size profile used in Table 1.

Figure 2. Dataset size profile for the experimental setting. Lighter bars correspond to regression datasets, and darker bars correspond to classification datasets. The vertical axis reports $\log_{10}$ molecule count, while the original counts are printed above the bars to avoid ambiguity from nonlinear scaling.

The baselines are Centralized GNN, Local-only GNN, FedAvg, FedProx, SSFO with secure aggregation but without differential privacy, and DP-FedAvg. The centralized model

is non-private and is included only as an upper reference. All results are averaged over five random seeds.

Table 2. Default federated learning, graph learning, and privacy hyperparameters. These settings are used unless an ablation explicitly changes them.

Category	Hyperparameter	Default Value
Federated training	Number of clients K	10
Federated training	Selected clients per round $\lvert\mathcal{C}^t\rvert$	5
Federated training	Communication rounds T	100; 150 for HIV
Federated training	Local epochs E	3
Federated training	Partition concentration α_{dir}	0.5
Graph model	Layers L and hidden dimension d	5 and 300
Optimization	Local learning rate η_ℓ	1.0×10^{-3}
PrivMol objective	λ_s, λ_c, μ	0.20, 0.10, 0.01
DPMGC	C_0, $c_{\max}$, λ_ω	1.0, 3.0, 0.50
DPMGC	Privacy target (ε, δ)	$(4.0, 10^{-5})$

5.2. Prediction Performance and Privacy–Utility Trade-Off

Table 3 reports the main predictive performance. PrivMol gives the strongest privacy-preserving results on all seven datasets. Compared with DP-FedAvg under the same default privacy target, PrivMol reduces RMSE on ESOL, FreeSolv, and Lipophilicity and improves ROC-AUC on BBBP, BACE, HIV, and Tox21. The secure-aggregation-only SSFO variant remains close to the centralized upper reference but does not provide differential privacy, whereas PrivMol retains much of its utility after privacy calibration.

Table 3. Main predictive performance. Regression datasets are evaluated by RMSE, where lower is better; classification datasets are evaluated by ROC-AUC, where higher is better. Results are mean ± standard deviation over five seeds.

Method	ESOL ↓	FreeSolv ↓	Lipophilicity ↓	BBBP ↑	BACE ↑	HIV ↑	Tox21 ↑
Centralized GNN	0.588 ± 0.018	1.195 ± 0.047	0.633 ± 0.012	0.923 ± 0.010	0.852 ± 0.013	0.785 ± 0.006	0.829 ± 0.005
Local-only GNN	0.742 ± 0.031	1.570 ± 0.085	0.781 ± 0.022	0.865 ± 0.021	0.779 ± 0.020	0.735 ± 0.009	0.781 ± 0.010
FedAvg	0.675 ± 0.026	1.401 ± 0.071	0.718 ± 0.019	0.893 ± 0.017	0.812 ± 0.018	0.762 ± 0.008	0.804 ± 0.007
FedProx	0.654 ± 0.024	1.365 ± 0.066	0.700 ± 0.018	0.902 ± 0.015	0.823 ± 0.016	0.768 ± 0.007	0.811 ± 0.007
SSFO+SA, no DP	0.611 ± 0.020	1.257 ± 0.055	0.660 ± 0.015	0.919 ± 0.011	0.844 ± 0.014	0.781 ± 0.006	0.826 ± 0.006
DP-FedAvg	0.735 ± 0.034	1.543 ± 0.083	0.774 ± 0.023	0.867 ± 0.020	0.784 ± 0.019	0.738 ± 0.009	0.782 ± 0.010
PrivMol	0.623 ± 0.021	1.288 ± 0.058	0.669 ± 0.016	0.915 ± 0.012	0.839 ± 0.014	0.778 ± 0.006	0.822 ± 0.006

Figure 3 summarizes the average rank implied by Table 3. It is retained as a compact visualization of the tabulated results rather than as an additional experiment.

Table 4 evaluates privacy–utility behavior under different privacy budgets. Membership AUC measures whether an attacker can distinguish training molecules from held-out molecules, and Scaffold Hit@10 measures whether an attacker can recover client-enriched scaffolds. These empirical attacks complement, but do not replace, the formal differential privacy guarantee.

Table 4. Privacy–utility comparison under different privacy budgets. Lower RMSE, lower attack scores, and higher ROC-AUC indicate better behavior.

Method and Budget	ESOL RMSE ↓	BBBP ROC-AUC ↑	BACE ROC-AUC ↑	HIV ROC-AUC ↑	Membership AUC ↓	Scaffold Hit@10 ↓
DP-FedAvg, $\varepsilon = 2.0$	0.812 ± 0.039	0.842 ± 0.022	0.755 ± 0.021	0.714 ± 0.010	0.548 ± 0.011	0.184 ± 0.018
PrivMol, $\varepsilon = 2.0$	0.675 ± 0.026	0.894 ± 0.015	0.812 ± 0.017	0.754 ± 0.008	0.521 ± 0.008	0.112 ± 0.013
DP-FedAvg, $\varepsilon = 4.0$	0.735 ± 0.034	0.867 ± 0.020	0.784 ± 0.019	0.738 ± 0.009	0.573 ± 0.012	0.221 ± 0.020
PrivMol, $\varepsilon = 4.0$	0.623 ± 0.021	0.915 ± 0.012	0.839 ± 0.014	0.778 ± 0.006	0.533 ± 0.009	0.135 ± 0.014
DP-FedAvg, $\varepsilon = 8.0$	0.704 ± 0.030	0.879 ± 0.018	0.800 ± 0.018	0.751 ± 0.008	0.611 ± 0.015	0.265 ± 0.023
PrivMol, $\varepsilon = 8.0$	0.603 ± 0.020	0.919 ± 0.011	0.846 ± 0.013	0.782 ± 0.006	0.558 ± 0.010	0.166 ± 0.017

To quantify the privacy cost against the centralized non-private upper reference, Table 5 reports the relative performance degradation of PrivMol for every dataset and privacy budget. For RMSE datasets, the degradation is computed as $100 \times (\mathrm{RMSE}_{\mathrm{PrivMol}} - \mathrm{RMSE}_{\mathrm{Centralized}})/\mathrm{RMSE}_{\mathrm{Centralized}}$; for ROC-AUC datasets, it is computed as $100 \times (\mathrm{AUC}_{\mathrm{Centralized}} - \mathrm{AUC}_{\mathrm{PrivMol}})/\mathrm{AUC}_{\mathrm{Centralized}}$.

Figure 4 visualizes the same privacy–utility trend for the datasets reported in Table 4; it is retained to show the monotonic change as the privacy budget is relaxed.

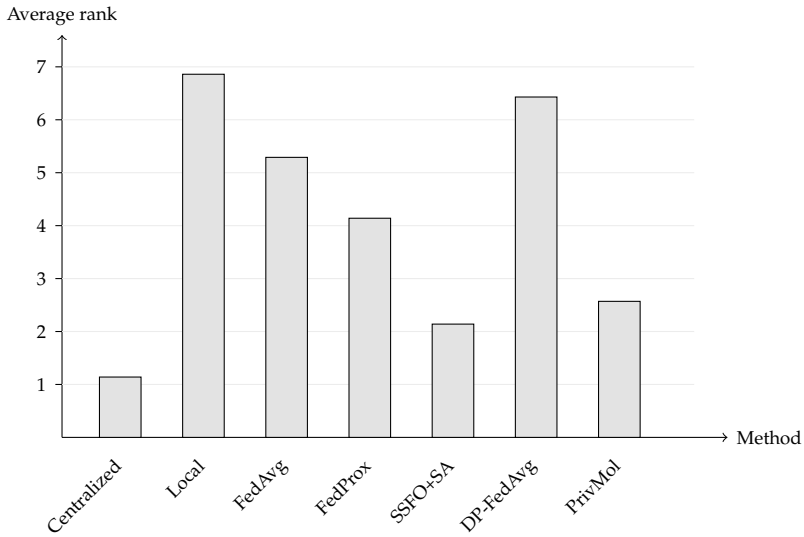

Figure 3. Average rank across the seven datasets in Table 3. Lower values indicate better overall predictive performance. The bars correspond, from left to right, to Centralized GNN, Local-only GNN, FedAvg, FedProx, SSFO with secure aggregation but without differential privacy, DP-FedAvg, and PrivMol.

Table 5. Relative utility degradation of PrivMol compared with the centralized non-private GNN under different privacy budgets. Lower values indicate a smaller privacy-induced utility cost.

Privacy Budget	ESOL	FreeSolv	Lipophilicity	BBBP	BACE	HIV	Tox21
$\varepsilon = 2.0$	14.8%	17.7%	14.4%	3.1%	4.7%	3.9%	3.6%
$\varepsilon = 4.0$	6.0%	7.8%	5.7%	0.9%	1.5%	0.9%	0.8%
$\varepsilon = 8.0$	2.6%	4.5%	3.0%	0.4%	0.7%	0.4%	0.4%

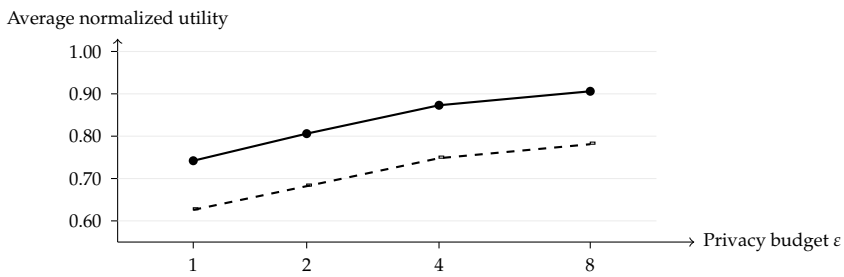

Figure 4. Privacy–utility trade-off across privacy budgets. The solid line with circular markers denotes PrivMol, and the dashed line with square markers denotes DP-FedAvg. The plotted score is average normalized predictive utility computed from ESOL, BBBP, BACE, and HIV in Table 4.

5.3. Ablation, Robustness, and Practical Cost

This subsection studies which components of PrivMol are responsible for the gains and whether the method remains robust under heterogeneous chemical spaces. Table 6 evaluates five variants: removing substructure consistency, removing contrastive alignment, removing proximal drift control, replacing DPMGC with uniform differentially private

clipping, and removing secure aggregation. Every component contributes, and the largest degradation is caused by replacing DPMGC with uniform clipping.

Table 6. Ablation study under the default privacy budget $(\varepsilon, \delta) = (4.0, 10^{-5})$. Results are reported for all seven datasets; regression datasets use RMSE, and classification datasets use ROC-AUC.

Variant	ESOL ↓	FreeSolv ↓	Lipophilicity ↓	BBBP ↑	BACE ↑	HIV ↑	Tox21 ↑
PrivMol without substructure consistency, $\lambda_s = 0$	0.657 ± 0.025	1.352 ± 0.064	0.701 ± 0.019	0.904 ± 0.014	0.826 ± 0.016	0.768 ± 0.007	0.811 ± 0.007
PrivMol without contrastive alignment, $\lambda_c = 0$	0.641 ± 0.023	1.331 ± 0.061	0.684 ± 0.017	0.900 ± 0.015	0.818 ± 0.018	0.761 ± 0.008	0.803 ± 0.008
PrivMol without proximal drift control, $\mu = 0$	0.636 ± 0.022	1.322 ± 0.060	0.681 ± 0.017	0.909 ± 0.013	0.827 ± 0.015	0.772 ± 0.007	0.812 ± 0.007
SSFO with uniform DP clipping	0.671 ± 0.027	1.421 ± 0.072	0.713 ± 0.020	0.889 ± 0.018	0.811 ± 0.018	0.756 ± 0.008	0.801 ± 0.009
DPMGC without secure aggregation	0.626 ± 0.022	1.294 ± 0.059	0.671 ± 0.016	0.913 ± 0.012	0.837 ± 0.015	0.777 ± 0.006	0.820 ± 0.006
Full PrivMol	0.623 ± 0.021	1.288 ± 0.058	0.669 ± 0.016	0.915 ± 0.012	0.839 ± 0.014	0.778 ± 0.006	0.822 ± 0.006

Figure 5 summarizes the average normalized performance loss of the ablated variants relative to full PrivMol, using the results in Table 6.

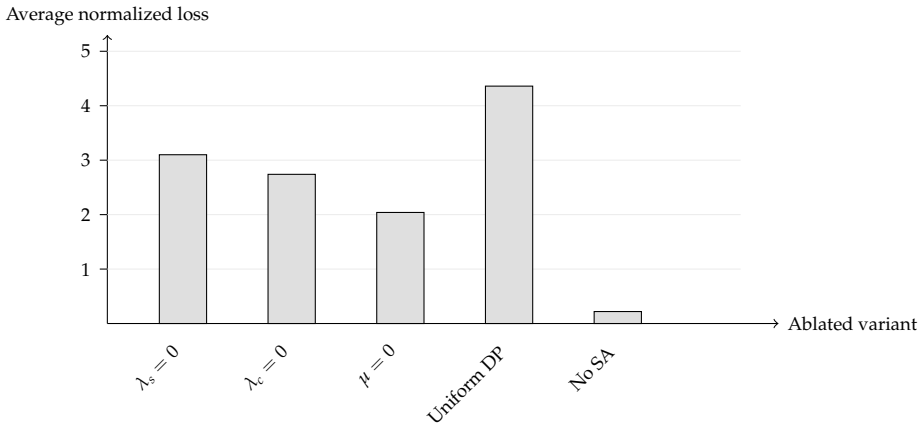

Figure 5. Average normalized performance loss of ablated variants relative to full PrivMol, computed from all seven datasets in Table 6. The bars from left to right correspond to removing substructure consistency, removing contrastive alignment, removing proximal drift control, replacing DPMGC with uniform differentially private clipping, and removing secure aggregation.

Robustness to client heterogeneity is evaluated by varying the scaffold-aware Dirichlet concentration α_{dir}. Table 7 reports BBBP and BACE results. Smaller α_{dir} produces more heterogeneous client partitions. PrivMol remains strongest in all tested settings, and the advantage is largest when clients have highly biased scaffold distributions.

Table 7. Robustness to scaffold-aware client heterogeneity. Smaller α_{dir} indicates more heterogeneous client partitions. Metrics are ROC-AUC.

	BBBP ROC-AUC ↑				BACE ROC-AUC ↑			
Method	**0.1**	**0.5**	**1.0**	**5.0**	**0.1**	**0.5**	**1.0**	**5.0**
FedAvg	0.861 ± 0.022	0.893 ± 0.017	0.902 ± 0.015	0.912 ± 0.013	0.782 ± 0.021	0.812 ± 0.018	0.821 ± 0.017	0.832 ± 0.015
FedProx	0.875 ± 0.020	0.902 ± 0.015	0.910 ± 0.014	0.916 ± 0.012	0.797 ± 0.020	0.823 ± 0.016	0.831 ± 0.016	0.839 ± 0.014
DP-FedAvg	0.832 ± 0.024	0.867 ± 0.020	0.878 ± 0.018	0.889 ± 0.017	0.746 ± 0.023	0.784 ± 0.019	0.795 ± 0.018	0.806 ± 0.017
PrivMol	0.895 ± 0.016	0.915 ± 0.012	0.920 ± 0.011	0.924 ± 0.010	0.818 ± 0.017	0.839 ± 0.014	0.845 ± 0.013	0.850 ± 0.012

Figure 6 visualizes the BBBP robustness trend from Table 7 as the scaffold-aware Dirichlet concentration changes.

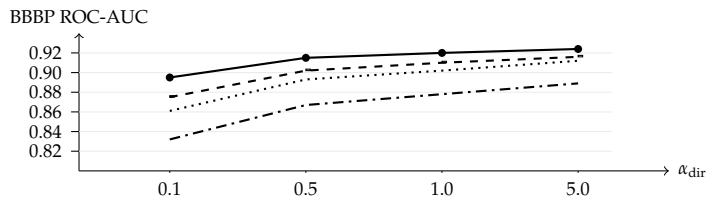

Figure 6. BBBP robustness under scaffold-aware client heterogeneity. The solid line with circular markers denotes PrivMol; the dashed line with square markers denotes FedProx; the dotted line denotes FedAvg; and the dash–dot line denotes DP-FedAvg.

Table 8 reports runtime and communication cost. PrivMol adds local computation for substructure pooling and DPMGC, but its upload size remains close to standard federated learning because only protected parameter updates are sent. The runtime numbers in Table 8 use the communication-round settings in Table 2. The secure aggregation component measures pairwise mask generation and update masking; in our simulation, the overhead is dominated by local cryptographic masking rather than wide-area network latency. Since clients upload one protected parameter vector of dimension P, communication remains $O(P)$ per selected client per round, while masking cost scales approximately linearly with $|\mathcal{C}^t|$ and DPMGC scoring scales with the molecules and fragments processed locally.

Table 8. Efficiency comparison. Runtime is average seconds per communication round, and upload cost is megabytes per selected client per round.

	BBBP		**BACE**		**HIV**		**Tox21**	
Method	**Time ↓**	**Upload ↓**	**Time ↓**	**Upload ↓**	**Time ↓**	**Upload ↓**	**Time ↓**	**Upload ↓**
FedAvg	18.4 ± 0.7	7.8	17.9 ± 0.6	7.8	42.6 ± 1.3	7.8	31.8 ± 1.1	7.8
FedProx	19.2 ± 0.7	7.8	18.6 ± 0.7	7.8	44.1 ± 1.4	7.8	32.7 ± 1.2	7.8
DP-FedAvg	21.1 ± 0.8	7.8	20.9 ± 0.8	7.8	47.2 ± 1.5	7.8	35.8 ± 1.2	7.8
SSFO+SA, no DP	22.8 ± 0.8	7.9	22.4 ± 0.8	7.9	49.3 ± 1.6	7.9	37.6 ± 1.3	7.9
PrivMol	23.7 ± 0.9	7.9	23.4 ± 0.9	7.9	51.7 ± 1.7	7.9	39.7 ± 1.4	7.9

Figure 7 visualizes the PrivMol runtime composition under the communication-round settings in Table 2.

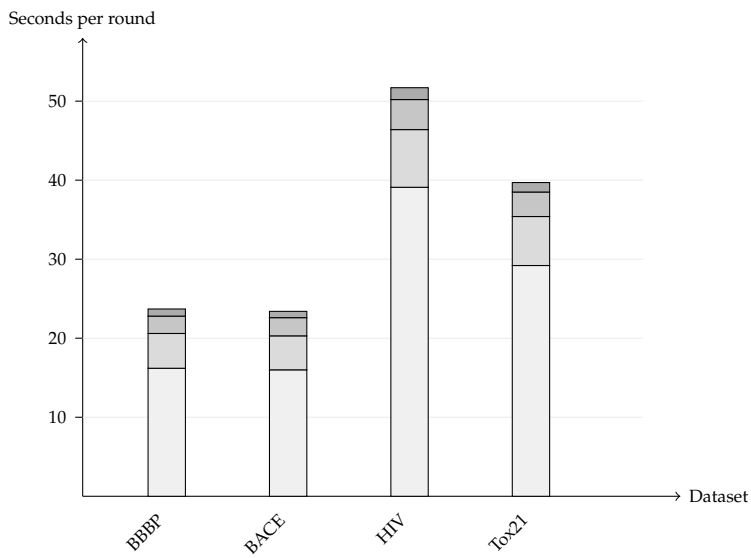

Figure 7. Runtime breakdown of PrivMol. Within each stacked bar, the lightest segment denotes ordinary graph neural network computation, the second-lightest segment denotes substructure extraction and masked pooling, the darker segment denotes DPMGC scoring and perturbation, and the darkest segment denotes secure aggregation masking.

Table 9 reports empirical leakage resistance. PrivMol achieves lower membership AUC and Scaffold Hit@10 and higher reconstruction error than the baselines.

Table 9. Empirical leakage resistance on BACE and Tox21. Membership AUC and Scaffold Hit@10 are lower-is-better attack success indicators, while reconstruction error is higher-is-better.

	BACE			Tox21		
Method	**Membership AUC ↓**	**Scaffold Hit@10 ↓**	**Recon. Error ↑**	**Membership AUC ↓**	**Scaffold Hit@10 ↓**	**Recon. Error ↑**
FedAvg	0.681 ± 0.019	0.356 ± 0.030	0.418 ± 0.025	0.646 ± 0.017	0.314 ± 0.027	0.462 ± 0.024
FedProx	0.664 ± 0.018	0.331 ± 0.028	0.441 ± 0.026	0.632 ± 0.016	0.295 ± 0.025	0.481 ± 0.023
SSFO+SA, no DP	0.612 ± 0.016	0.247 ± 0.023	0.586 ± 0.027	0.594 ± 0.015	0.226 ± 0.021	0.611 ± 0.026
DP-FedAvg	0.573 ± 0.012	0.221 ± 0.020	0.671 ± 0.024	0.561 ± 0.012	0.204 ± 0.019	0.689 ± 0.023
PrivMol	0.533 ± 0.009	0.135 ± 0.014	0.782 ± 0.020	0.526 ± 0.009	0.128 ± 0.013	0.801 ± 0.019

Figure 8 summarizes the BACE attack-risk profile by converting reconstruction error into reconstruction risk so that lower values are better on all axes.

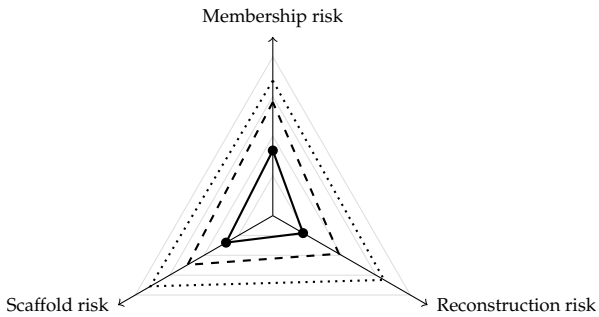

Figure 8. Attack-risk profile on BACE. The dotted triangle denotes FedAvg, the dashed triangle denotes DP-FedAvg, and the solid triangle denotes PrivMol. Membership risk is derived from membership AUC, scaffold risk is derived from Scaffold Hit@10, and reconstruction risk is derived from normalized reconstruction error. Smaller triangles indicate stronger empirical leakage resistance.

Figures 3–6 and 8 are summary visualizations of the corresponding tables. They are retained to expose aggregate rank, privacy–utility slope, ablation sensitivity, heterogeneity robustness, and attack-risk geometry, without making claims beyond the tabulated values. The evidence from Tables 6–9 and Figures 5–8 shows that PrivMol is not a single-mechanism improvement. SSFO improves representations under scaffold heterogeneity, DPMGC improves the privacy–utility trade-off by bounded molecular calibration, and secure aggregation prevents direct server inspection of individual updates.

6. Conclusions

This paper proposed PrivMol, a privacy-preserving federated molecular graph learning framework for collaborative molecular property prediction under chemical data confidentiality constraints. PrivMol treats privacy not only as a molecule-level issue but also as a substructure-level issue, reflecting the realistic risk that proprietary scaffolds, pharmacophores, toxicophores, or confidential functional fragments may be exposed through federated updates. The proposed Secure Substructure-Aware Federated Optimization algorithm improves learning under scaffold-heterogeneous clients by combining supervised molecular prediction, local substructure consistency, contrastive molecular alignment, and proximal drift control, while the proposed Differentially Private Molecular Gradient Calibration algorithm converts atom scores c_v and substructure scores $c(S_r)$ into bounded molecule-level sensitivity indicators for adaptive clipping and Gaussian perturbation under a target privacy budget (ε, δ). Together with secure aggregation, PrivMol ensures that the server observes only the aggregate calibrated update $\sum_{k \in \mathcal{C}^t} \widetilde{\Delta}_k^t$ rather than raw molecules,

labels, substructure masks, embeddings, or individual client updates. Experiments on ESOL, FreeSolv, Lipophilicity, BBBP, BACE, HIV, and Tox21 under simulated scaffold-aware federated partitions show that PrivMol achieves a stronger privacy–utility trade-off than FedAvg, FedProx, uniform DP-FedAvg, and secure-aggregation-only variants, while maintaining practical runtime and communication costs in this benchmark setting. These results suggest that PrivMol is a promising direction for privacy-sensitive collaborative molecular learning; however, real-world deployment still requires validation under cross-institutional protocols, regulatory constraints, collusion risks, industrial-scale models, and proprietary scaffold-leakage scenarios.

Author Contributions: Conceptualization, Y.Y. and J.C.; Methodology, Y.Y. and J.C.; Software, Y.Y. and J.C.; Validation, Y.Y. and J.C.; Formal analysis, Y.Y. and J.C.; Investigation, Y.Y. and J.C.; Resources, Y.Y. and J.C.; Data curation, Y.Y. and J.C.; Writing—original draft, Y.Y. and J.C.; Writing—review and editing, Y.Y. and J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors decare no conflicts of interest.

References

1. Wu, Z.; Ramsundar, B.; Feinberg, E.N.; Gomes, J.; Geniesse, C.; Pappu, A.S.; Leswing, K.; Pande, V.S. MoleculeNet: A benchmark for molecular machine learning. *Chem. Sci.* **2018**, *9*, 513–530. [CrossRef] [PubMed]
2. Gilmer, J.; Schoenholz, S.S.; Riley, P.F.; Vinyals, O.; Dahl, G.E. Neural message passing for quantum chemistry. In Proceedings of the 34th International Conference on Machine Learning, PMLR (Proceedings of Machine Learning Research), Sydney, Australia, 6–11 August 2017; Volume 70, pp. 1263–1272.
3. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How powerful are graph neural networks? In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
4. Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; et al. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.* **2019**, *59*, 3370–3388. [CrossRef] [PubMed]
5. Rong, Y.; Bian, Y.; Xu, T.; Xie, W.; Wei, Y.; Huang, W.; Huang, J. Self-supervised graph transformer on large-scale molecular data. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–12 December 2020; Volume 33, pp. 12559–12571.
6. Wang, Y.; Wang, J.; Cao, Z.; Barati Farimani, A. Molecular contrastive learning of representations via graph neural networks. *Nat. Mach. Intell.* **2022**, *4*, 279–287. [CrossRef]
7. Méndez-Lucio, O.; Nicolaou, C.A.; Earnshaw, B. MolE: A foundation model for molecular graphs using disentangled attention. *Nat. Commun.* **2024**, *15*, 9431. [CrossRef] [PubMed]
8. Beaini, D.; Huang, S.; Cunha, J.A.; Li, Z.; Moisescu-Pareja, G.; Dymov, O.; Maddrell-Mander, S.; McLean, C.; Wenkel, F.; Müller, L.; et al. Towards foundational models for molecular learning on large-scale multi-task datasets. In Proceedings of the International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
9. Park, J.H.; Kim, Y.; Lee, M.; Park, H.; Lee, S. MolTRES: Improving chemical language representation learning for molecular property prediction. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, FL, USA, 12–16 November 2024; pp. 14241–14254. [CrossRef]
10. Burns, J.W.; Zalte, A.S.; Green, W.H. Descriptor-based foundation models for molecular property prediction. *arXiv* **2025**, arXiv:cs.LG/2506.15792. [CrossRef]
11. Liu, S.; Zou, D.; Shoghi, N.; Fung, V.; Liu, K.; Li, P. RoFt-Mol: Benchmarking robust fine-tuning with molecular graph foundation models. *arXiv* **2025**, arXiv:cs.LG/2509.00614. [CrossRef]
12. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, PMLR (Proceedings of Machine Learning Research), Sydney, Australia, 6–11 August 2017; Volume 54, pp. 1273–1282.
13. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. In Proceedings of the Machine Learning and Systems, Austin, TX, USA, 2–4 March 2020; Volume 2, pp. 429–450.

14. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*; Association for Computing Machinery: New York, NY, USA, 2017; pp. 1175–1191. [CrossRef]
15. Dwork, C.; McSherry, F.; Nissim, K.; Smith, A. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Theory of Cryptography*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3876, pp. 265–284. [CrossRef]
16. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H.B.; Mironov, I.; Talwar, K.; Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 308–318. [CrossRef]
17. Liu, R.; Xing, P.; Deng, Z.; Li, A.; Guan, C.; Yu, H. Federated graph neural networks: Overview, techniques and challenges. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 4279–4295. [CrossRef] [PubMed]
18. Huang, W.; Wan, G.; Ye, M.; Du, B. Federated graph semantic and structural learning. In Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, Macao SAR, China, 19–25 August 2023; pp. 3830–3838. [CrossRef] [PubMed]
19. Li, X.; Zhu, Y.; Pang, B.; Yan, G.; Yan, Y.; Li, Z.; Wu, Z.; Zhang, W.; Li, R.H.; Wang, G. OpenFGL: A comprehensive benchmark for federated graph learning. In Proceedings of the VLDB Endowment, London, UK, 1–5 September 2025; Volume 18, pp. 1305–1320. [CrossRef]
20. Liu, Y.; Li, H.; Hao, M. Personalized and privacy-preserving federated graph neural network. *Front. Phys.* **2024**, *12*, 1383276. [CrossRef]
21. Liang, W.; Zhao, Y.; She, R.; Li, Y.; Tay, W.P. FedSheafHN: Personalized federated learning on graph-structured data. *arXiv* **2024**, arXiv:cs.LG/2405.16056. [CrossRef]
22. Zhu, W.; Luo, J.; White, A.D. Federated learning of molecular properties with graph neural networks in a heterogeneous setting. *Patterns* **2022**, *3*, 100521. [CrossRef] [PubMed]
23. Wu, Z.; Wang, J.; Du, H.; Jiang, D.; Kang, Y.; Li, D.; Pan, P.; Deng, Y.; Cao, D.; Hsieh, C.Y.; et al. Chemistry-intuitive explanation of graph neural networks for molecular property prediction with substructure masking. *Nat. Commun.* **2023**, *14*, 2585. [CrossRef] [PubMed]
24. Panapitiya, G.; Gao, P.; Maupin, C.M.; Saldanha, E.G. FragNet: A graph neural network for molecular property prediction with four levels of interpretability. *J. Am. Chem. Soc.* **2026**, *148*, 9930–9950. [CrossRef] [PubMed]
25. Wang, H.; Zhang, A.; Zhong, Y.; Tang, J.; Zhang, K.; Li, P. Chain-aware graph neural networks for molecular property prediction. *Bioinformatics* **2024**, *40*, btae574. [CrossRef] [PubMed]
26. Yang, X.; Duan, Y.; Cheng, Z.; Li, K.; Liu, Y.; Zeng, X.; Cao, D. MPCD: A multitask graph transformer for molecular property prediction by integrating common and domain knowledge. *J. Med. Chem.* **2024**, *67*, 21303–21316. [CrossRef] [PubMed]
27. Li, L.; Zhang, Y.; Wang, G.; Xia, K. Kolmogorov–Arnold graph neural networks for molecular property prediction. *Nat. Mach. Intell.* **2025**, *7*, 1346–1354. [CrossRef]
28. Liu, X.; Fan, C.; Liu, Y.; Li, H.B. Multilevel fusion graph neural network for molecule property prediction. *J. Chem. Inf. Model.* **2025**, *65*, 9034–9048. [CrossRef] [PubMed]
29. To, V.T.; Van Nguyen, P.C.; Truong, G.B.; Phan, T.M.; Phan, T.L.; Fagerberg, R.; Stadler, P.F.; Truong, T.N. KGG: Knowledge-guided graph self-supervised learning to enhance molecular property predictions. *J. Chem. Inf. Model.* **2025**, *65*, 9443–9458. [CrossRef] [PubMed]
30. Zhao, X.; Zhang, S.; Zhang, T.; Li, H.; Cao, Y. Molecular property prediction based on improved graph transformer network and multitask joint learning strategy. *J. Chem. Inf. Model.* **2025**, *65*, 10561–10572. [CrossRef] [PubMed]
31. Qu, C.; Schneider, B.I.; Kearsley, A.J.; Keyrouz, W.; Allison, T.C. Applying graph neural network models to molecular property prediction using high-quality experimental data. *Artif. Intell. Chem.* **2024**, *2*, 100050. [CrossRef]
32. Karimireddy, S.P.; Kale, S.; Mohri, M.; Reddi, S.J.; Stich, S.U.; Suresh, A.T. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In Proceedings of the 37th International Conference on Machine Learning, PMLR (Proceedings of Machine Learning Research), Virtual, 13–18 July 2020; Volume 119, pp. 5132–5143.
33. Acar, D.A.E.; Zhao, Y.; Matas, R.; Mattina, M.; Whatmough, P.N.; Saligrama, V. Federated Learning Based on Dynamic Regularization. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021.
34. Li, X.; Jiang, M.; Zhang, X.; Kamp, M.; Dou, Q. FedBN: Federated Learning on Non-IID Features via Local Batch Normalization. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021.
35. Huang, Y.; Chu, L.; Zhou, Z.; Wang, L.; Liu, J.; Pei, J.; Zhang, Y. Personalized Cross-Silo Federated Learning on Non-IID Data. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 7865–7873.
36. Andrew, G.; Thakkar, O.; McMahan, H.B.; Ramaswamy, S. Differentially Private Learning with Adaptive Clipping. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–14 December 2021; Volume 34, pp. 17455–17466.
37. Geyer, R.C.; Klein, T.; Nabi, M. Differentially Private Federated Learning: A Client Level Perspective. *arXiv* **2017**, arXiv:1712.07557.

38. Zhang, Y.; Zhao, Y.; Li, Z.; Cheng, X.; Wang, Y.; Kotevska, O.; Yu, P.S.; Derr, T. A survey on privacy in graph neural networks: Attacks, preservation, and applications. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 7497–7515. [CrossRef]
39. Liu, Y.; Zhang, Y.; He, P.; Fang, S. FedGNNLDP: Federated graph neural network with locally differential privacy. *Comput. Secur.* **2026**, *161*, 104757. [CrossRef]
40. Feng, Y.; Qian, Q. PPFedGNN: An efficient privacy-preserving federated graph neural network method for social network analysis. *IEEE Trans. Comput. Soc. Syst.* **2025**, *12*, 4481–4500. [CrossRef]
41. Wang, G.; Li, Q.; Liu, X.; Yan, X.; Dong, Q.; Wu, H.; Kong, X.; Zhou, L. VFGCN: A vertical federated learning framework with privacy preserving for graph convolutional network. *IEEE Trans. Dependable Secur. Comput.* **2025**, *22*, 5617–5631. [CrossRef]
42. Liu, J.; Chen, B.; Xue, B.; Guo, M.; Xu, Y. PIAFGNN: Property inference attacks against federated graph neural networks. *Comput. Mater. Contin.* **2025**, *82*, 1857–1877. [CrossRef]
43. Lai, C.H.; Wu, T.E.; Wang, C.C. Enhancing Information Security in Smart Manufacturing Through Least Significant Bit Steganography in Engineering Drawings. *J. Comput. Inf. Sci. Eng.* **2025**, *25*, 091006. [CrossRef]
44. Weng, Y.; Yang, K.; Liu, Z.; He, W.; Tang, X. Lgvlm-miot: A lightweight generative visual-language model for multilingual iot applications. *IEEE Internet Things J.* **2025**, *12*, 13311–13326.
45. Yang, T.; Qian, Z.; Hang, N.; Liu, M. S-PINN: Stabilized physics-informed neural networks for alleviating barriers between multi-level co-optimization. *Comput. Methods Appl. Mech. Eng.* **2025**, *447*, 118348.
46. Roy, T.; Zhou, Y. From agricultural waste to building insulation: A chemical-free composite from banana stem and jackfruit seed starch. *Bioresour. Technol.* **2026**, *452*, 134591. [CrossRef] [PubMed]
47. Bemis, G.W.; Murcko, M.A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39*, 2887–2893. [CrossRef] [PubMed]
48. Degen, J.; Wegscheid-Gerlach, C.; Zaliani, A.; Rarey, M. On the Art of Compiling and Using "Drug-Like" Chemical Fragment Spaces. *ChemMedChem* **2008**, *3*, 1503–1507. [CrossRef] [PubMed]
49. Ghadimi, S.; Lan, G. Stochastic First- and Zeroth-Order Methods for Nonconvex Stochastic Programming. *SIAM J. Optim.* **2013**, *23*, 2341–2368. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.