Enhancing Chemical Explainability Through Counterfactual Masking

Łukasz Janisiów[1,2,3], **Marek Kochańczyk**[1*], **Bartosz Michał Zieliński**[1], **Tomasz Danel**[1,4]

[1] Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków 30-348, Poland
[2] Doctoral School of Exact and Natural Sciences, Jagiellonian University, Kraków 30-348, Poland
[3] Faculty of Economic Sciences, University of Warsaw, Warsaw 00-241, Poland
[4] Faculty of Chemistry, Jagiellonian University, Kraków 30-387, Poland
tomasz.danel@uj.edu.pl

Abstract

Molecular property prediction is a crucial task that guides the design of new compounds, including drugs and materials. While explainable artificial intelligence methods aim to scrutinize model predictions by identifying influential molecular substructures, many existing approaches rely on masking strategies that remove either atoms or atom-level features to assess importance via fidelity metrics. These methods, however, often fail to adhere to the underlying molecular distribution and thus yield unintuitive explanations. In this work, we propose counterfactual masking, a novel framework that replaces masked substructures with chemically reasonable fragments sampled from generative models trained to complete molecular graphs. Rather than evaluating masked predictions against implausible zeroed-out baselines, we assess them relative to counterfactual molecules drawn from the data distribution. Our method offers two key benefits: (1) molecular realism that underpins robust and distribution-consistent explanations, and (2) meaningful counterfactuals that directly indicate how structural modifications may affect predicted properties. We demonstrate that counterfactual masking is well-suited for benchmarking model explainers and yields more actionable insights across multiple datasets and property prediction tasks. Our approach bridges the gap between explainability and molecular design, offering a principled and generative path toward explainable machine learning in chemistry.

Code — https://github.com/gmum/counterfactual-masking
Extended version — https://arxiv.org/abs/2508.18561

Introduction

Molecular property prediction has emerged as a cornerstone of modern drug discovery and materials science, promising to dramatically accelerate the identification of compounds with desired characteristics. By leveraging machine learning (ML) algorithms to predict properties such as binding affinity, toxicity, solubility, and biological activity, researchers can efficiently navigate the vast chemical space without exhaustive experimental testing (Wieder et al. 2020; Dara et al. 2022). However, the increasing complexity of ML models presents a significant challenge: while deep neural networks and ensemble methods deliver impressive predictive performance, they often function as inscrutable "black boxes," offering little insight into the structural features that drive their predictions (Biecek and Samek 2024; Longo et al. 2024). This opacity is particularly problematic in chemistry and pharmaceutical development, where understanding structure–property relationships is essential for rational molecular design, regulatory approval, and scientific knowledge advancement. Interpretable models that can explain their predictions in chemically meaningful terms are, therefore, not merely desirable but necessary to bridge the gap between statistical performance and actionable scientific insights (Jimenez-Luna, Grisoni, and Schneider 2020; Wong et al. 2024).

Graph masking has emerged as a fundamental technique for explaining molecular property predictions, serving both as the foundation for certain explainable AI (XAI) methods and as the basis for evaluation metrics like fidelity (Bugueño, Biswas, and de Melo 2024). By systematically removing or obscuring specific atoms or substructures, masking approaches aim to quantify the contribution of each component to the final prediction. However, conventional masking strategies suffer from critical limitations in the molecular domain. When atoms or bonds are naively masked, the resulting structures often become chemically implausible or physically impossible, generating examples that fall outside the training distribution. This out-of-distribution problem undermines the reliability of both the explanations and their evaluation metrics. Furthermore, current masking approaches inadvertently leak information about the original graph topology to the model—even when certain atoms are "masked," their structural relationships and connectivity patterns remain implicitly encoded in the modified graph. This information leakage creates a false sense of explanation quality, as models may still leverage these implicit structural cues rather than truly operating without knowledge of the masked components. Thus, developing more sophisticated masking techniques that preserve chemical validity while effectively controlling information flow is essential for advancing trustworthy explanations in molecular ML.

Current approaches to molecular property prediction primarily rely on *post-hoc* explanation methods such as

[*] On leave from the Institute of Fundamental Technological Research of the Polish Academy of Sciences, Warsaw 02-106, Poland.

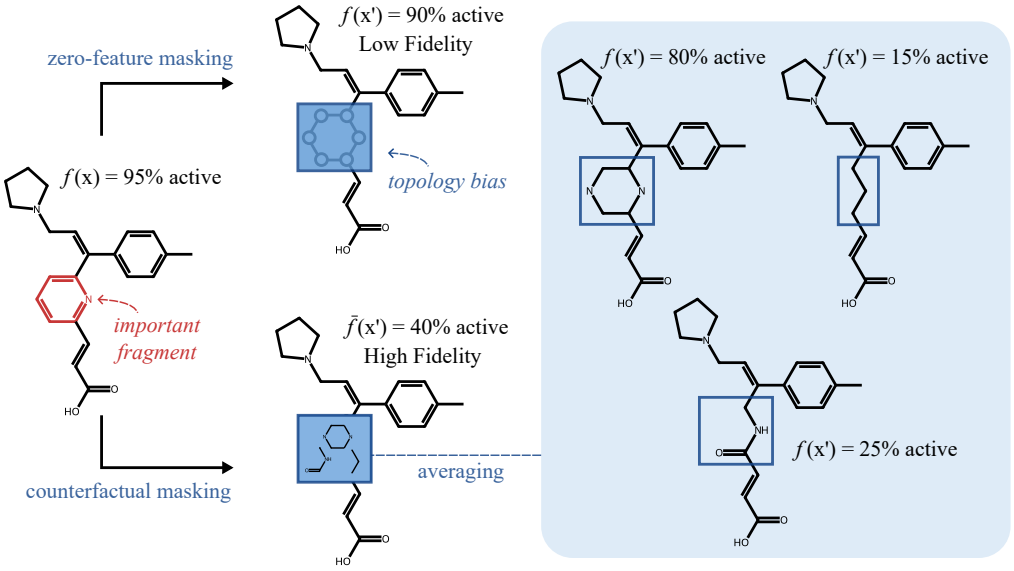

Figure 1: Comparison of zero-feature masking and our CM method. The former strategy preserves the topology of the masked fragment, which may lead to predictive bias. CM generates multiple fragment replacements, enabling more robust and trustworthy evaluation of model explanations.

gradient-based attribution, attention mechanisms, and feature importance scores (Pope et al. 2019). These techniques attempt to highlight atoms or substructures that significantly influence predictions, providing chemists with visual maps of "important" molecular regions. However, conventional explanation methods struggle to address the holistic nature of molecular design, where virtually all components contribute to functionality through precise atom arrangements. Identifying which atoms are "important" offers limited insight when the entire structure has been carefully optimized. What chemists truly need is to understand why these structural elements matter and how they might be modified to achieve the desired properties. This gap between highlighting important features and providing actionable insights motivates the development of counterfactual explanations, which instead focus on answering the critical question of what other molecular design choices would lead to a change in chemical properties, e.g., higher potency or lower toxicity (Gleeson et al. 2011; Guengerich 2011).

In this paper, we present counterfactual masking (CM) for molecular graphs, a novel method designed to enhance the explanations of ML model predictions (Figure 1). For every key molecular fragment identified with the explanation method, we generate a set of counterfactual explanations that replace the original fragment. This visually illustrates the importance of these fragments in relation to specific properties. Our approach aims to provide clearer insights into why certain molecular features contribute to predictive outcomes. The contribution of this paper can be summarized as follows:

1. We introduce a new masking method for molecular graphs that hides molecular fragments by replacing them with a set of alternative generated fragments, ensuring that masked molecules are valid in-distribution molecules.
2. We use our masking technique to provide additional insights for the explanation model by producing counterfactual explanations that show why these structures might be important for the model.

Related Work

Factual explanations. Factual explanation methods for GNNs can be broadly categorized into gradient-based, perturbation-based, surrogate-based, and self-interpretable approaches. Gradient-based techniques, such as CAM (Zhou et al. 2016), Grad-CAM (Selvaraju et al. 2017), and Integrated Gradients (Sundararajan, Taly, and Yan 2017), estimate feature relevance through backpropagation. Perturbation-based methods, including GNNExplainer (Ying et al. 2019) and SubgraphX (Yuan et al. 2021), generate explanations by learning masks or searching over subgraphs that are most influential to model predictions. In contrast, self-interpretable GNNs attempt to provide inherently explainable reasoning by embedding interpretability directly into the architecture (Zhang et al. 2022). While factual methods assign importance scores to nodes, edges, or subgraphs, they often lack mechanisms to verify the factual correctness of the highlighted structures on real-world datasets. Moreover, most approaches offer limited insight into why a particular structure is deemed important.

Counterfactual explanations. Unlike factual methods, counterfactual explanations provide additional insight into model predictions by showing similar instances that are predicted differently from the original (Kirilenko et al. 2024). In chemistry, research on counterfactual methods is still limited compared to other fields, but several methods have been pro-

posed. Wellawatte, Seshadri, and White (2022) introduced MMACE, a model-agnostic approach that generates counterfactual examples by insertions, replacements, or deletions in the SELFIES (Krenn et al. 2020) representation of the compound. Numeroso and Bacciu (2021) proposed MEG, a method employing reinforcement learning to find minimal atom-level modifications that change the model prediction. Other methods tweak the latent representation to sample close analogs of a compound, e.g. by using a variational autoencoder (Wang et al. 2024). However, many counterfactual methods are difficult to apply due to the discrete nature of molecular graphs, and some of the methods introduce modifications that harm synthetic accessibility and lack localization of the changes.

Generative models for structure optimization. Our method generates replacements only in the parts of the molecule that are predicted as important. Some methods have been proposed for such context-constrained generative modeling. For example, models such as DiffLinker (Igashov et al. 2024) and DeLinker (Imrie et al. 2020) were proposed to generate fragments between two or more molecular fragments. Polishchuk (2020) introduced CReM, a method that finds fragments in compound databases that were already seen in the same molecular context, ensuring the synthesizability of the compounds generated. Recently, Lee et al. (2025) proposed GenMol, a textual masked discrete diffusion model capable of generating linkers and scaffold decorations.

Methods

Let us define a molecular graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, F)$, where $\mathcal{V}$ is a set of nodes, $\mathcal{E}$ is a set of edges, and $F : \mathcal{V} \to \mathbb{R}^d$ is a function that assigns features to nodes. Factual explanation techniques for molecular graphs identify the atoms that significantly affect model predictions. This can be formally described by a function $E : \mathcal{V} \to \{0, 1\}$ that assigns a value of one to nodes considered crucial for the prediction. We now present our CM method, which evaluates the influence of these important fragments and offers alternative subgraphs for complementary counterfactual explanations.

Counterfactual Masking

CM generates a set of alternative molecules by replacing fragments identified by a factual explanation technique.

Step 1: Identification of important subgraphs. All important connected subgraphs are extracted. First, all the nodes crucial for the prediction are identified using a factual explanation method, resulting in a set of key nodes $\mathcal{V}_{\text{imp}} = \{v : v \in \mathcal{V} \wedge E(v) = 1\}$. Next, all these key nodes and their connected edges are removed, and the remaining graph serves as a context $\mathcal{C}$ for the generative methods to fill in the missing fragments. The context should also include attachment points $\mathcal{A}$, which are nodes from $\mathcal{C}$ that were connected to any of the removed key nodes.

Step 2: Regeneration of the removed fragments. A generative model g is employed to replace each removed subgraph based on the context and the attachment points, modeling the distribution of feasible molecules $p(x \mid \mathcal{C}, \mathcal{A})$. Then, CM is a set of molecules produced by drawing multiple samples from the generator.

Properties of our masking method. Since CM is sampled from a distribution of feasible molecules, we can evaluate the quality of factual explanations by comparing the original prediction with the average case from that distribution rather than using artificially zeroed-out features. In the case of classification, we can present molecules that are predicted to be in a different class as counterfactual examples.

Generative model. As a generative model for fragment replacement, we use two different approaches. One is based on the CReM algorithm, which uses fragments extracted from the ChEMBL database. In this approach, only fragments that were already seen in the same atom context can be used, thus increasing the synthetic accessibility of the generated replacements. In this case, the entire molecular ring is replaced if any atom within it is found to be influential because CReM does not operate on partial rings.

Another generative method used is DiffLinker, a diffusion model that produces a linker between fragments placed in 3-D space, trained on the ZINC dataset (Irwin and Shoichet 2005). In this method, the input molecules are embedded in 3-D space by applying a force field method to generate a stable conformation of a molecule. This ensures that the generated fragment replacement occupies a similar amount of space between fragments being connected. A considerable disadvantage of this method is that the diffusion model works at the atomic level, sometimes resulting in molecules that are unsynthesizable or even invalid.

Common Substructure Pair Dataset

To evaluate the performance of masking methods, we introduced a new dataset. Starting with the Solubility dataset (Sorkun, Khetan, and Er 2019) from the Therapeutics Data Commons (TDC) collection (Huang et al. 2021), we applied BRICS decomposition (Degen et al. 2008) to fragment each molecule into chemically meaningful substructures. These substructures were then used to query the PubChem database (Kim et al. 2024) to identify larger molecules (superstructures) in which each substructure appears as a subgraph. For each source fragment, multiple matching superstructures were identified, resulting in a dataset of 881 pairs of the form (source fragment, superstructures). On average, each source fragment is associated with 52 chemically distinct superstructures.

Results

In this section, we first demonstrate that the widely used masking technique of zeroing features is flawed and our masking strategy resolves this issue. Next, we show the utility of our method in producing counterfactual explanations. CM is then used to benchmark XAI techniques. We conclude by discussing the limitations of our method.

Enhanced Atom Masking Strategy

Masking is commonly used for creating or evaluating graph-based explanation methods. However, current masking tech-

Masking	Single Anchor			Multiple Anchors			Combined		
	$\lvert\Delta\hat{y}\rvert \downarrow$	Validity ↑	Size ↑	$\lvert\Delta\hat{y}\rvert \downarrow$	Validity ↑	Size ↑	$\lvert\Delta\hat{y}\rvert \downarrow$	Validity ↑	Size ↑
No masking	2.81	100%	517	2.52	100%	638	2.91	100%	783
Feature zeroing	2.08	100%	517	1.60	100%	638	1.80	100%	783
CM (CReM)	1.68	88%	454	1.32	53%	341	1.72	65%	510
CM (DiffLinker)	1.81	67%	347	1.59	61%	386	1.55	65%	508

Table 1: The effectiveness of masking methods in preventing information leakage from the masked subgraph. The difference in predictions ($|\Delta\hat{y}|$) should be close to zero when only the common subgraph is retained, while the rest of the molecule is masked. Results are shown for three scenarios: single anchor (a non-shared fragment between the two molecules has one attachment point to the common substructure), multiple anchors (the fragment has more than one attachment point), and combined (the fragment can have one or more attachment points). Examples of molecules for each scenario are provided in Appendix B.

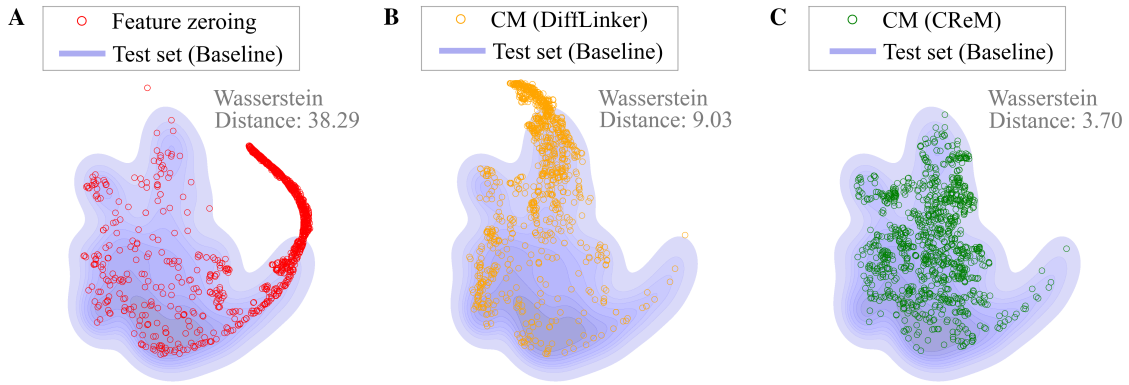

Figure 2: t-SNE visualization of molecular embeddings comparing the test set distribution with molecules containing parts masked by various methods. Wasserstein distances quantify the distributional divergence from the test set reference distribution.

niques, such as feature zeroing, often produce samples outside the training distribution and may reveal information about the original graph topology. In this section, we examine our CM approach in which masked fragments are replaced with chemically plausible alternatives. We compare this method to the traditional feature zeroing baseline, where all node features within the masked fragments are set to zero.

Setup. To assess the effectiveness of various masking techniques, we used the common substructure pair dataset described in the Methods section. For each substructure, we selected a pair of superstructures that show the greatest difference in solubility predictions from a GNN regression model described below. In each pair, we masked the molecular fragments that were not shared between the two molecules and measured the resulting difference in the model's predictions. An ideal masking technique should completely obscure the masked fragment, resulting in identical predictions for both molecules in the pair.

Model. As the regression model, we used a Graph Isomorphism Network (GIN) (Xu et al. 2019) with three layers, each with a hidden size of 512. Global mean pooling was used to aggregate node features. The model was trained for 300 epochs with a batch size of 64 using the Adam optimizer and mean squared error (MSE) loss. During training, a dropout rate of 30% was applied.

Metrics. We used three metrics to evaluate the performance of each masking method. $|\boldsymbol{\Delta\hat{y}}|$ measures the absolute difference in predicted values between two molecules sharing the same substructure. **Validity** is the percentage of molecular pairs for which the method generates valid masked molecules. **Size** is the total number of pairs to which the method can be successfully applied.

Results. Table 1 compares the CM approach using two generators with the baseline that sets the features of masked nodes to zero. All masking methods reduce prediction differences compared to unmasked pairs, with CM consistently outperforming the baseline in all three scenarios. CReM sampling produces the lowest prediction difference in two scenarios, while DiffLinker performs best in the combined scenario. However, CM depends on generative models, which may not always produce valid fragments, whereas feature zeroing provides complete coverage of the dataset.

Figure 2 displays the distribution of molecular embeddings from the single anchor part of the dataset (for other scenarios, see Appendix B) after masking, compared to the test set of the solubility dataset. Since the test set is randomly sampled, it reflects the training data distribution. Embeddings of molecules masked with CM align more closely with this training distribution than those masked with feature zeroing, as evidenced by lower Wasserstein distances, especially for the CReM model. Feature zeroing often yields

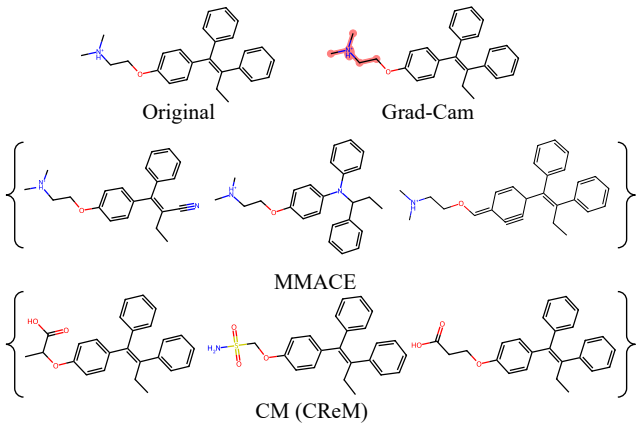

Figure 3: Comparison of counterfactual examples generated via MMACE and CM (CReM). The initial model prediction indicated that the original molecule is a hERG blocker.

molecules outside the training distribution, reducing prediction and explanation reliability. This occurs because zero-masked features provide the model with misleading information about atomic properties, frequently representing chemically invalid molecules. In contrast, CM provides a more robust and chemically grounded comparison by contrasting the original molecule's prediction with the average prediction of chemically plausible alternatives, offering a meaningful measure of the influence of masked molecular fragments.

Counterfactual Explanations

Although originally designed for masking tasks, CM can effectively generate realistic, chemically valid counterfactual examples by ensuring molecules stay within the data distribution. It also enables targeted local modifications to specific fragments, improving interpretability by offering insights into how substructure changes affect predicted classes.

Dataset. To evaluate counterfactual explanations, we used three binary classification datasets from TDC: prediction of inhibition of cytochrome P450 enzymes CYP3A4 and CYP2D6, and hERG channel blockage (hERG), a key cardiotoxicity indicator (Karim et al. 2021). We trained a GIN classification model using 80% of each dataset, while the remaining 20% was reserved for counterfactual generation.

Explained model. To predict molecular properties, we used a GIN model with three layers, each with a hidden size of 512. Global mean pooling was applied to aggregate node features. The model was trained for 300 epochs with a batch size of 16, using the Adam optimizer and binary cross-entropy loss. During training a dropout rate of 30% was applied and early stopping was used if the validation performance had not improved in the last 20 epochs.

Counterfactual models and baselines. To generate counterfactuals using the CM approach, we first applied Grad-CAM to identify the top 20% most important atoms in the molecule. These atoms were then replaced with new fragments using either the CReM or DiffLinker framework to produce counterfactual examples. After generation, post-hoc filtering was applied to select alternative molecules that successfully changed the prediction class. From these, a subset was selected to maximize similarity to the original molecule while maintaining diversity among the already selected counterfactuals. The first example was chosen based on its highest similarity to the original sample, and subsequent counterfactuals were selected by balancing similarity to the original with diversity relative to those already chosen.

As baseline methods for counterfactual generation, we used GNNExplainer and Nearest Neighbor (NN). For GNNExplainer, counterfactuals were generated by removing the top 10% of nodes identified as the most influential. For the NN method, counterfactuals were defined as the most structurally similar molecules from the training set that belong to the opposite prediction class. Additionally, we benchmarked our approach against MMACE, which produces counterfactuals by performing up to two targeted structural modifications (deletion, replacement, or insertion) to the original molecule.

Metrics. We evaluated counterfactual generation using five metrics. **Before success rate (BSR)** measures the proportion of generated molecules that change the prediction class before any filtering is applied, while **success rate** measures this proportion after filtering. **Similarity** indicates the average similarity between the original molecule and the filtered counterfactuals, calculated using the Tanimoto distance between molecular fingerprints. **Diversity** captures the variation among the generated counterfactuals, based on the Tanimoto distance between their fingerprints. Finally, **synthetic accessibility (SA)** evaluates the average ease of synthesizing the filtered counterfactuals and reflects their practical usability in real-world laboratory settings. A lower SA score indicates more realistic and accessible counterfactuals.

Results. The performance of counterfactual generation methods is summarized in Table 2. Before filtering, the NN baseline shows the highest success rate, as it directly searches for molecules that change the prediction class. Among generative methods, CM (DiffLinker) achieves the highest rate of class changes in two out of three datasets. After filtering, all methods except GNNExplainer, which provides only one counterfactual per instance, reach a success rate of 100%. MMACE generates the most similar and diverse molecules, as it is explicitly optimized for these metrics. However, CM (CReM) remains close in both statistics in all tested datasets while maintaining a lower synthetic accessibility score. MMACE frequently produces chemically implausible molecules, leading to higher synthetic accessibility scores, as illustrated in Figure 3. The only exception is the NN baseline, which exhibits greater diversity than both because of its larger distance from the original molecules in chemical space. Overall, combining Grad-CAM with CM (CReM) offers effective counterfactuals, on par with MMACE, but with improved synthesizability.

Figure 3 displays counterfactuals generated by MMACE and CM (CReM) for a hERG inhibitor. The top 20% most

Dataset	Method	BSR ↑	Success Rate ↑	Similarity ↑	Diversity ↑	SA ↓
CYP 3A4	GNNExplainer	0.16 ± 0.04	0.16 ± 0.04	0.33 ± 0.00	0.00 ± 0.00	3.67 ± 0.04
	NN	1.00 ± 0.00	1.00 ± 0.00	0.37 ± 0.00	0.66 ± 0.00	2.53 ± 0.01
	MMACE	0.27 ± 0.00	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.60 \pm 0.01}$	$\mathbf{0.52 \pm 0.01}$	3.34 ± 0.02
	CM (CReM)	0.21 ± 0.01	$\mathbf{1.00 \pm 0.00}$	0.52 ± 0.04	0.46 ± 0.04	$\mathbf{2.71 \pm 0.08}$
	CM (DiffLinker)	$\mathbf{0.41 \pm 0.02}$	$\mathbf{1.00 \pm 0.00}$	0.41 ± 0.05	0.41 ± 0.01	3.91 ± 0.18
CYP 2D6	GNNExplainer	0.10 ± 0.10	0.10 ± 0.10	0.35 ± 0.05	0.00 ± 0.00	3.62 ± 0.19
	NN	1.00 ± 0.00	1.00 ± 0.00	0.32 ± 0.00	0.68 ± 0.00	2.53 ± 0.02
	MMACE	$\mathbf{0.18 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.52 \pm 0.00}$	$\mathbf{0.57 \pm 0.00}$	3.45 ± 0.03
	CM (CReM)	0.08 ± 0.00	$\mathbf{1.00 \pm 0.00}$	0.44 ± 0.01	0.49 ± 0.01	$\mathbf{2.79 \pm 0.04}$
	CM (DiffLinker)	0.15 ± 0.02	$\mathbf{1.00 \pm 0.00}$	0.46 ± 0.01	0.38 ± 0.02	3.91 ± 0.16
hERG	GNNExplainer	0.21 ± 0.06	0.21 ± 0.06	0.31 ± 0.08	0.00 ± 0.00	4.20 ± 0.05
	NN	1.00 ± 0.00	1.00 ± 0.00	0.30 ± 0.02	0.67 ± 0.02	3.14 ± 0.09
	MMACE	0.39 ± 0.02	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.65 \pm 0.01}$	$\mathbf{0.48 \pm 0.01}$	3.69 ± 0.02
	CM (CReM)	0.30 ± 0.07	$\mathbf{1.00 \pm 0.00}$	0.51 ± 0.03	0.47 ± 0.04	$\mathbf{3.10 \pm 0.09}$
	CM (DiffLinker)	$\mathbf{0.61 \pm 0.07}$	$\mathbf{1.00 \pm 0.00}$	0.41 ± 0.01	$\mathbf{0.48 \pm 0.01}$	3.63 ± 0.14

Table 2: Counterfactual evaluation metrics. The numbers in the table represent mean ± s.d. BSR refers to the success rate before filtering generated molecules, while SA represents synthetic accessibility, with lower scores indicating easier synthesis.

influential atoms identified by Grad-CAM point to a protonated tertiary amine, a common structural feature of hERG blockers (Garrido et al. 2020). MMACE often generates molecules that cannot exist in nature, such as those containing a triple bond within a ring. In contrast, CM (CReM) creates chemically plausible, targeted modifications. For example, CReM replaces a positively charged tertiary amine with neutral groups, potentially reducing hERG inhibition risk by limiting interactions with aromatic amino acids. Additional examples can be found in the Appendix C.

Improved Benchmark of XAI Methods

In the widely used fidelity metric, the predictions are made for the original graph and the graph with masked important atoms. Then, the difference in accuracy can be measured to assess the robustness of the explanation method. However, masking atom features with zeros creates out-of-distribution molecular graphs, which do not accurately reflect the importance of these atoms. The messages in such graphs can still be passed through masked nodes in graph neural networks. It is thus desirable to be able to evaluate explainers with alternative masking methods, and our generative counterfactual approach addresses this necessity. CM can be used to measure the quality of the factual explanation.

Datasets and models. We used five publicly available datasets from the TDC. Three datasets were for binary classification tasks: CYP2D6, CYP3A4, and hERG; two were for regression tasks: predicting Lipophilicity and Solubility in water. As a preprocessing step, molecules were stripped of ions, and molecules with fewer than five heavy atoms were removed. We evaluated six distinct graph neural network architectures and regularizations: three variants of the Graph Isomorphism Network (GIN): large GIN (2,648,065 parameters trained at 30% dropout); medium GIN (170,497 parameters at 15% dropout), and small GIN (11,905 parameters, no dropout), one GIN model with additional edge attributes (106,189 parameters at 10% dropout), one GIN model with residual connections (56,577 parameters at 10% dropout), and one Graph Attention Network model (Veličković et al. 2018) (3 attention heads, 176,385 parameters at 10% dropout). A detailed definition of the model architectures is presented in the Appendix A. For each dataset and model architecture, we performed three independent trainings. In each training, the data was randomly split into training (75%), validation (10%), and test (15%) sets. Models were trained using the Adam optimizer with an initial learning rate of 3×10^{-4} and a batch size of 64 molecules. The learning rate was reduced by a factor of 0.85 every 10 epochs. Binary cross-entropy and smooth L1 were used as loss functions for classification and regression tasks, respectively. Training was carried out for a maximum of 300 epochs, with an early stopping condition that terminated training if the validation loss did not improve for 50 consecutive epochs. In this way, for each of the five chemical datasets, we obtained 18 trained models and corresponding test sets.

Experimental setup. For each combination of masking techniques and datasets, we used five explainability methods to identify the atoms most responsible for a specific prediction: Grad-CAM, Integrated Gradients, GNNExplainer, saliency map, and, as a baseline, random assignment. For each molecule, we determined two distinct sets of atoms: the top 10% that most strongly increased the predicted value and, separately, the top 10% that most strongly decreased it. GNNExplainer was configured to run 100 iterations and mask node attributes, with the influence direction determined by an auxiliary Grad-CAM assessment. The influential atoms identified were then masked using one of three techniques: (1) CM (CReM), (2) CM (DiffLinker), or (3) simple atom feature ablation (setting features to zero).

To ensure the validity of our analysis, we first filtered the generated molecules. When the new, replacement atoms were indicated by an explainer as contributing to the prediction in the same direction but to an even greater extent than

Masking method	Explanation method	Dataset				
		CYP 2D6	CYP 3A4	Lipophilicity	Solubility	hERG
Feature zeroing	Grad-CAM	$84 \pm 13\%$	$79 \pm 23\%$	$84 \pm 17\%$	$76 \pm 23\%$	$79 \pm 21\%$
	Integrated Gradients	$76 \pm 13\%$	$84 \pm 14\%$	$81 \pm 17\%$	$77 \pm 12\%$	$70 \pm 17\%$
	GNNExplainer	$62 \pm 22\%$	$58 \pm 35\%$	$55 \pm 41\%$	$56 \pm 32\%$	$56 \pm 38\%$
	Saliency	$56 \pm 19\%$	$63 \pm 31\%$	$60 \pm 30\%$	$52 \pm 25\%$	$53 \pm 31\%$
	Random	$50 \pm 16\%$	$50 \pm 26\%$	$50 \pm 27\%$	$50 \pm 19\%$	$50 \pm 30\%$
CM (DiffLinker)	Grad-CAM	$56 \pm 28\%$	$59 \pm 21\%$	$56 \pm 38\%$	$55 \pm 36\%$	$54 \pm 40\%$
	Integrated Gradients	$53 \pm 30\%$	$54 \pm 24\%$	$54 \pm 39\%$	$54 \pm 40\%$	$53 \pm 37\%$
	GNNExplainer	$52 \pm 28\%$	$52 \pm 24\%$	$51 \pm 36\%$	$52 \pm 38\%$	$51 \pm 41\%$
	Saliency	$50 \pm 30\%$	$49 \pm 23\%$	$52 \pm 38\%$	$51 \pm 39\%$	$50 \pm 39\%$
	Random	$50 \pm 28\%$	$50 \pm 24\%$	$50 \pm 37\%$	$50 \pm 39\%$	$50 \pm 39\%$
CM (CReM)	Grad-CAM	$71 \pm 10\%$	$72 \pm 13\%$	$76 \pm 14\%$	$63 \pm 21\%$	$72 \pm 16\%$
	Integrated Gradients	$52 \pm 13\%$	$56 \pm 17\%$	$61 \pm 12\%$	$68 \pm 14\%$	$54 \pm 21\%$
	GNNExplainer	$60 \pm 14\%$	$59 \pm 21\%$	$57 \pm 28\%$	$52 \pm 30\%$	$58 \pm 21\%$
	Saliency	$51 \pm 12\%$	$53 \pm 19\%$	$53 \pm 17\%$	$49 \pm 24\%$	$51 \pm 18\%$
	Random	$50 \pm 13\%$	$50 \pm 18\%$	$50 \pm 22\%$	$49 \pm 28\%$	$50 \pm 22\%$

Table 3: Consistency of masking methods and explanation methods. The numbers in the table represent mean $\pm$ s.d. of consistency scores, which are the percentage of times that CM of influential atoms (as identified by an explainer) caused the model's prediction to change in the expected direction. A higher score within each masking method means the explainer is more effective and reliable under this masking strategy. A score of 50% is the random baseline, indicating no better than chance.

the original atoms, the generated molecule was discarded. In this way, the remaining molecules were expected to have their most influential atoms masked counterfactually, meaning that the masked region's original influence has been attenuated or reversed. This selection procedure was applied to molecules generated by all three masking techniques. We then assessed whether masking influential atoms led to the expected change in the model's overall prediction. We calculated a consistency score, defined as the proportion of counterfactually masked molecules where the property prediction changed in the direction anticipated by the explainer (e.g., the prediction decreased after masking atoms identified as increasing the value). This consistency score was calculated for 3 independent data splits $\times$ 6 model architectures $\times$ 2 directions of the change in a predicted molecular property.

Benchmark results. Table 3 reports the aggregate statistics for consistency scores, summarizing the results from every combination of the 5 (test) datasets, 5 explanation methods, and 3 masking techniques. Statistics are computed for 300 randomly selected molecules from each test dataset. Detailed results for separate models are provided in the supplementary tables in the Appendix D.

Across all five datasets, the combination of the Grad-CAM explainer with the standard masking by feature zeroing consistently yields the highest average consistency scores (ranging from 76% to 84%). However, it should be remembered that this method masks atom features but leaves the graph structure intact, potentially leaking information and inflating consistency scores. CReM, acting through molecular replacement under more stringent chemical requirements, still achieves strong consistency scores, particularly with Grad-CAM (ranging from 63% to 76%). These results are well above the 50% random baseline (and above the results for DiffLinker), demonstrating that CReM effectively creates meaningful counterfactuals that validate the explainer's output. A qualitative example of compound solubility analysis and explanation method evaluation using CM is provided in Appendix E.

Limitations

Although CM brings chemical realism to XAI model evaluation, it has important limitations that should be addressed in future research. One significant limitation is the validity of molecules generated by current models conditioned on molecular fragment context. CReM struggles to fill in fragments with more than two attachment points. Conversely, DiffLinker can even complete partial rings but often fails to produce chemically feasible molecules. Because sampling multiple compounds from generative models is necessary, CM also demands more computational resources for model evaluation.

Conclusions

In this paper, we introduced CM, a method that combines factual explanation techniques for molecular graphs with structure-conditioned generative models to generate a set of counterfactual examples. This approach improves the evaluation of XAI methods for molecular property prediction by creating molecules with fragment replacements that serve as better references than masked molecules with artificially removed nodes or features. Additionally, we demonstrate how CM can be used as a counterfactual explanation technique that performs precise, local replacements of fragments essential for the model's prediction. We hope that our method will advance the development of more effective XAI methods and accelerate molecular design workflows.

Acknowledgments

This study was funded by the "Interpretable and Interactive Multimodal Retrieval in Drug Discovery" project. The "Interpretable and Interactive Multimodal Retrieval in Drug Discovery" project (FENG.02.02-IP.05-0040/23) is carried out within the First Team programme of the Foundation for Polish Science co-financed by the European Union under the European Funds for Smart Economy 2021-2027 (FENG). We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2025/018272.

References

Biecek, P.; and Samek, W. 2024. Position: explain to question not to justify. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Bugueño, M.; Biswas, R.; and de Melo, G. 2024. Graph-Based Explainable AI: A Comprehensive Survey.

Dara, S.; Dhamercherla, S.; Jadav, S. S.; Babu, C. M.; and Ahsan, M. J. 2022. Machine Learning in Drug Discovery: A Review. *Artificial Intelligence Review*, 55(3): 1947–1999.

Degen, J.; Wegscheid-Gerlach, C.; Zaliani, A.; and Rarey, M. 2008. On the Art of Compiling and Using 'Drug-Like' Chemical Fragment Spaces. *ChemMedChem*, 3(10): 1503–1507.

Garrido, A.; Lepailleur, A.; Mignani, S. M.; Dallemagne, P.; and Rochais, C. 2020. hERG toxicity assessment: Useful guidelines for drug design. *European Journal of Medicinal Chemistry*, 195: 112290.

Gleeson, M. P.; Hersey, A.; Montanari, D.; and Overington, J. 2011. Probing the links between in vitro potency, ADMET and physicochemical parameters. *Nature Reviews Drug Discovery*, 10(3): 197–208.

Guengerich, F. P. 2011. Mechanisms of Drug Toxicity and Relevance to Pharmaceutical Development. *Drug Metabolism and Pharmacokinetics*, 26(1): 3–14.

Huang, K.; Fu, T.; Gao, W.; Zhao, Y.; Roohani, Y.; Leskovec, J.; Coley, C. W.; Xiao, C.; Sun, J.; and Zitnik, M. 2021. Therapeutics Data Commons: Machine Learning Datasets and Tasks for Drug Discovery and Development. *Proceedings of Neural Information Processing Systems, NeurIPS Datasets and Benchmarks*.

Igashov, I.; Stärk, H.; Vignac, C.; Schneuing, A.; Satorras, V. G.; Frossard, P.; Welling, M.; Bronstein, M.; and Correia, B. 2024. Equivariant 3D-conditional diffusion model for molecular linker design. *Nature Machine Intelligence*, 6(4): 417–427.

Imrie, F.; Bradley, A. R.; van der Schaar, M.; and Deane, C. M. 2020. Deep generative models for 3D linker design. *Journal of chemical information and modeling*, 60(4): 1983–1995.

Irwin, J. J.; and Shoichet, B. K. 2005. ZINC- a free database of commercially available compounds for virtual screening. *Journal of chemical information and modeling*, 45(1): 177–182.

Jimenez-Luna, J.; Grisoni, F.; and Schneider, G. 2020. Drug discovery with explainable artificial intelligence. *Nature Machine Intelligence*, 2(10): 573–584.

Karim, A.; Lee, M.; Balle, T.; and Sattar, A. 2021. CardioTox net: a robust predictor for hERG channel blockade based on deep learning meta-feature ensembles. *J Cheminform*, 13: 60.

Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; and Bolton, E. E. 2024. PubChem 2025 update. *Nucleic Acids Research*, 53(D1): D1516–D1525.

Kirilenko, D.; Barbiero, P.; Gjoreski, M.; Luštrek, M.; and Langheinrich, M. 2024. Generative Models for Counterfactual Explanations. *View Article*.

Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; and Aspuru-Guzik, A. 2020. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4): 045024.

Lee, S.; Kreis, K.; Veccham, S. P.; Liu, M.; Reidenbach, D.; Peng, Y.; Paliwal, S.; Nie, W.; and Vahdat, A. 2025. Genmol: A drug discovery generalist with discrete diffusion.

Longo, L.; Brcic, M.; Cabitza, F.; Choi, J.; Confalonieri, R.; Ser, J. D.; Guidotti, R.; Hayashi, Y.; Herrera, F.; Holzinger, A.; Jiang, R.; Khosravi, H.; Lecue, F.; Malgieri, G.; Páez, A.; Samek, W.; Schneider, J.; Speith, T.; and Stumpf, S. 2024. Explainable Artificial Intelligence (XAI) 2.0: A manifesto of open challenges and interdisciplinary research directions. *Information Fusion*, 106: 102301.

Numeroso, D.; and Bacciu, D. 2021. Meg: Generating molecular counterfactual explanations for deep graph networks. In *2021 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Polishchuk, P. 2020. CReM: chemically reasonable mutations framework for structure generation. *Journal of Cheminformatics*, 12(1): 28.

Pope, P. E.; Kolouri, S.; Rostami, M.; Martin, C. E.; and Hoffmann, H. 2019. Explainability Methods for Graph Convolutional Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Sorkun, M. C.; Khetan, A.; and Er, S. 2019. AqSolDB, a curated reference set of aqueous solubility and 2D descriptors for a diverse set of compounds. *Scientific Data*, 6(1): 143.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, 3319–3328. PMLR.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Wang, D.; Antoniades, A.; Luong, K.-D.; Zhang, E.; Kosan, M.; Li, J.; Singh, A.; Wang, W. Y.; and Li, L. 2024. Global

Human-guided Counterfactual Explanations for Molecular Properties via Reinforcement Learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2991–3000.

Wellawatte, G. P.; Seshadri, A.; and White, A. D. 2022. Model agnostic generation of counterfactual explanations for molecules. *Chemical science*, 13(13): 3697–3705.

Wieder, O.; Kohlbacher, S.; Kuenemann, M.; Garon, A.; Ducrot, P.; Seidel, T.; and Langer, T. 2020. A compact review of molecular property prediction with graph neural networks. *Drug Discovery Today: Technologies*, 37: 1–12.

Wong, F.; Zheng, E. J.; Valeri, J. A.; Donghia, N. M.; Anahtar, M. N.; Omori, S.; Li, A.; Cubillos-Ruiz, A.; Krishnan, A.; Jin, W.; Manson, A. L.; and Fr, J. 2024. Discovery of a structural class of antibiotics with explainable deep learning. *Nature*, 626(7997): 177–185.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

Ying, Z.; Bourgeois, D.; You, J.; Zitnik, M.; and Leskovec, J. 2019. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32.

Yuan, H.; Yu, H.; Wang, J.; Li, K.; and Ji, S. 2021. On explainability of graph neural networks via subgraph explorations. In *International conference on machine learning*, 12241–12252. PMLR.

Zhang, Z.; Liu, Q.; Wang, H.; Lu, C.; and Lee, C. 2022. Protgnn: Towards self-explaining graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 9127–9135.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929.