nature machine intelligence

Article

https://doi.org/10.1038/s42256-026-01229-5

A collaborative constrained graph diffusion model for the generation of realistic synthetic molecules

Received: 21 May 2025

Accepted: 26 March 2026

Published online: 04 May 2026

Check for updates

Manuel Ruiz-Botella [1], Marta Sales-Pardo [1,2] ✉ & Roger Guimerà [1,2,3] ✉

Developing new molecular compounds is crucial to address pressing challenges, from health to environmental sustainability. However, exploring the molecular space to discover new molecules is difficult owing to the vastness of the space. Here we introduce CoCoGraph, a collaborative and constrained graph diffusion model capable of generating molecules that are guaranteed to be chemically valid. Thanks to the constraints built into the model and to the collaborative mechanism, CoCoGraph outperforms state-of-the-art approaches on standard benchmarks while being more efficient. Analysis of 36 chemical properties also demonstrates that CoCoGraph generates molecules with distributions more closely matching real molecules than current models. To illustrate the potential of the model, we created a database of 8.2 million synthetically generated molecules, show how this database and CoCoGraph could be used for molecular discovery and conduct a Turing-like test with organic chemistry experts to further assess the plausibility of the generated molecules, and the potential biases and limitations of CoCoGraph.

Discovering and developing new molecular compounds is key to addressing several pressing challenges, including facilitating the development of new drugs[1], the creation of advanced materials[2], the design of more environmentally friendly refrigerants[3], the identification of unknown metabolites[4] or the discovery of molecules that bind to disease-associated target proteins[5]. However, the vastness of the space of drug-like molecules (~10^{60} molecules[6]) renders the discovery of new compounds a high-dimensional and exceedingly complex problem. Consequently, these areas stand to benefit greatly from artificial intelligence tools that can generate new molecules with desired properties or reconstruct molecules from available molecular information.

Traditionally, algorithms for molecule generation relied on rule-based models and optimization[7–9]. However, these classical approaches were limited to modifying existing molecules rather than generating new ones[10]. With the progression of deep learning, new generative models for molecules have been developed by using variational autoencoders[11], generative adversarial networks[12] and graph neural networks (GNNs)[13]. Despite their improved performance, these models still faced challenges related to scalability, computational efficiency, molecular validity and/or adherence to chemical constraints[14–16]. Moreover, these models exhibited limited generalization capabilities, often struggling to generate molecules that deviate from those seen during training[17].

Advances in diffusion models[18–21], which originated from image generation[22,23], have led to innovative generative techniques that alleviate some of these shortcomings. In the context of molecule generation[24–26], diffusion models consist of a process that progressively adds noise (atoms and/or bonds) to a molecular graph, followed by a denoising process that learns to reconstruct molecules by removing the noise. The denoising process is then used to generate new molecules. To better handle molecular constraints and improve sampling efficiency, graph-based diffusion models[27,28] such as DiGress[29] and CDGS[30] employ discrete noising processes. Theoretically, this approach facilitates the generation of valid molecules and is able to produce

[1]Department of Chemical Engineering, Universitat Rovira i Virgili, Tarragona, Catalonia. [2]Center for Computational Science and Applied Mathematics (ComSCIAM), Universitat Rovira i Virgili, Tarragona, Catalonia. [3]ICREA, Barcelona, Catalonia. ✉e-mail: marta.sales@urv.cat; roger.guimera@urv.cat

new molecules not seen during training. Nevertheless, creating models that accurately reflect the original molecular distributions while also generalizing to new molecules remains a challenge[16,31,32]. More recent approaches have explored alternative formulations to address these limitations, including score-based generative models with continuous states and continuous time, such as GDSS[26] and GruM[33], and flow-matching techniques, such as DeFoG[34]. Although these methods represent important advances in molecular generation, they still face challenges in accurately capturing the full complexity of molecular property distributions. Therefore, further improvements in generative techniques for molecules are necessary to ensure both efficient generation of chemically valid molecules and the comprehensive exploration of the chemical space.

Here we introduce a collaborative constrained discrete diffusion model (CoCoGraph) that has two key mechanisms (Fig. 1). First and foremost, we employ a discrete process that involves double edge swapping (DES) and constrains each atom to always have the correct valence[35–37], maintaining chemical properties such as molecular weight, number of atoms, number of bonds and molecular formula. Second, we introduce a collaborative mechanism in which two models are trained at each step of the denoising process. The first model (diffusion model) is trained to predict the DES operation to be reverted at each denoising step, taking as input molecular graph features and the diffusion time step. The second model (time model) learns to predict the time step of the diffusion process and collaborates with the diffusion model by informing it of how close the molecular graph is to a valid molecule, so that the diffusion model can adapt its DES predictions to the actual (as opposed to the expected) progress of the denoising process.

CoCoGraph generates new molecules with 100% chemical validity, as defined in the most comprehensive existing benchmark[38], and beats the state-of-the-art on this benchmark. In addition, the distributions of chemical properties of these generated molecules are statistically closer to those of the known chemical universe than those produced by current generative models. Importantly, our constrained collaborative approach achieves these results with up to an order of magnitude fewer parameters than existing models and better sampling efficiency. The lightness of the model allows us to create a dataset with 8.2 million synthetically generated molecular structures, over which we conduct a Turing-like test to determine whether human experts can distinguish between generated and real molecules. We find that chemists with at least undergraduate training in organic chemistry can only identify the real molecules with 62% accuracy (59% for those without postgraduate education). For specific types of molecules, performance is statistically compatible with 50% accuracy, which thus indicates that real and generated molecules are similarly perceived (although the experiment does not allow confirming that real and generated molecules are fully indistinguishable). Furthermore, we show that CoCoGraph can be used for inpainting-based applications, where molecular fragments can be added to existing molecules while preserving their core structure, enabling targeted molecular design. Our results underscore the effectiveness of our model in navigating the vast chemical universe, and highlight the potential of our approach for real-world applications.

Results

A collaborative constrained diffusion model for the generation of graphs with fixed degree sequence

Discrete graph-based diffusion models for molecule generation such as DiGress[29] and CDGS[30] have been shown to be superior to continuous ones. More recent models such as Construct[39] have introduced constraint-aware diffusion processes specifically designed for graphs. By using a noising diffusion process that is aware of some chemical constraints, and automatically satisfies them, these models are able to enforce specific chemical rules and properties during the generation process[40,41].

Here we constrain even further the molecular graphs considered during a given diffusion process, so that each noising/denoising step preserves the nodes (atoms and thus molecular formula) and degree sequence (exact number of bonds per atom; that is, valence)[35–37]. To achieve this, at each noising diffusion step we swap two edges, so that two bonds AB and CD within the molecule are randomly selected and removed, and two new bonds AC and BD are formed (Fig. 1). By doing this, molecular graphs diffuse into a Molloy–Reed distribution[35,36,42], which is the maximum-entropy distribution over the space of graphs with fixed degree sequence. The satisfaction of chemical constraints by construction implies that: (1) invalid molecules not satisfying the constraints are never generated; (2) the molecular structure search space is vastly reduced; (3) the chemical constraints do not need to be learned; and therefore (4) models can be much smaller and focus on learning more subtle molecular features.

The denoising diffusion model learns to undo these edge swaps (Fig. 1; Methods). It takes as input the time step t of the diffusion process and the molecular graph, which runs through three graph neural layers. The resulting node embeddings, together with edge features, molecular graph features and time, are used to estimate the plausibility of each edge swap by means of a feedforward network. Note that although CoCoGraph typically starts by noising a real molecule, it can also perform generation of molecules starting directly from a random molecular graph built from an arbitrary molecular formula (Supplementary Section 1).

When using only the diffusion model, we observe that the progress of the diffusion process is not uniform across the training set—even after scaling the number of steps by molecular graph size (number of bonds), some molecules are quickly randomized whereas for others it takes much longer. Therefore, the time feature in the diffusion model turns out to be not very informative. Based on this observation, we introduce a time model (Fig. 1; Methods) that estimates how far the molecular graph is from a real molecule. This model takes as input the molecular graph and returns a normalized time, which is fed during sampling into the diffusion model instead of the actual time step, and thus collaborates with the diffusion model by providing more relevant information about the actual position within the diffusion trajectory. Furthermore, at the conclusion of the sampling process, the model chooses the molecule with the smallest predicted time throughout the whole trajectory (that is, in principle, the one closest to a real molecule), rather than the last generated molecule. The architecture of the time model is very similar to that of the diffusion model—the input graph goes through three graph neural layers, and the produced embeddings are used to predict time (Fig. 1). An ablation study demonstrates that, although the collaborative mechanism is less crucial than the constraining mechanism, it improves the realism of the generated molecules (Table 1; Supplementary Section 3).

A key consequence of our collaborative constrained diffusion approach CoCoGraph is a significantly more efficient model architecture that requires up to an order of magnitude fewer parameters than state-of-the-art approaches and samples new molecules faster (Table 1). By incorporating chemical constraints directly into the diffusion process, our model inherently preserves chemical validity rather than needing to learn these rules. This reduction in model complexity translates into lower computational requirements, making molecule generation more accessible. Importantly, our design based on constraints enables the model to allocate its learning capacity into capturing structural patterns of real molecules, resulting in better performance despite its smaller size, as we show next.

CoCoGraph outperforms existing generative models for molecules on a standard benchmark

To comprehensively evaluate the performance of CoCoGraph, we compare it with state-of-the-art molecular generative models using the GuacaMol[38] benchmark suite. This evaluation framework provides

https://doi.org/10.1038/s42256-026-01229-5

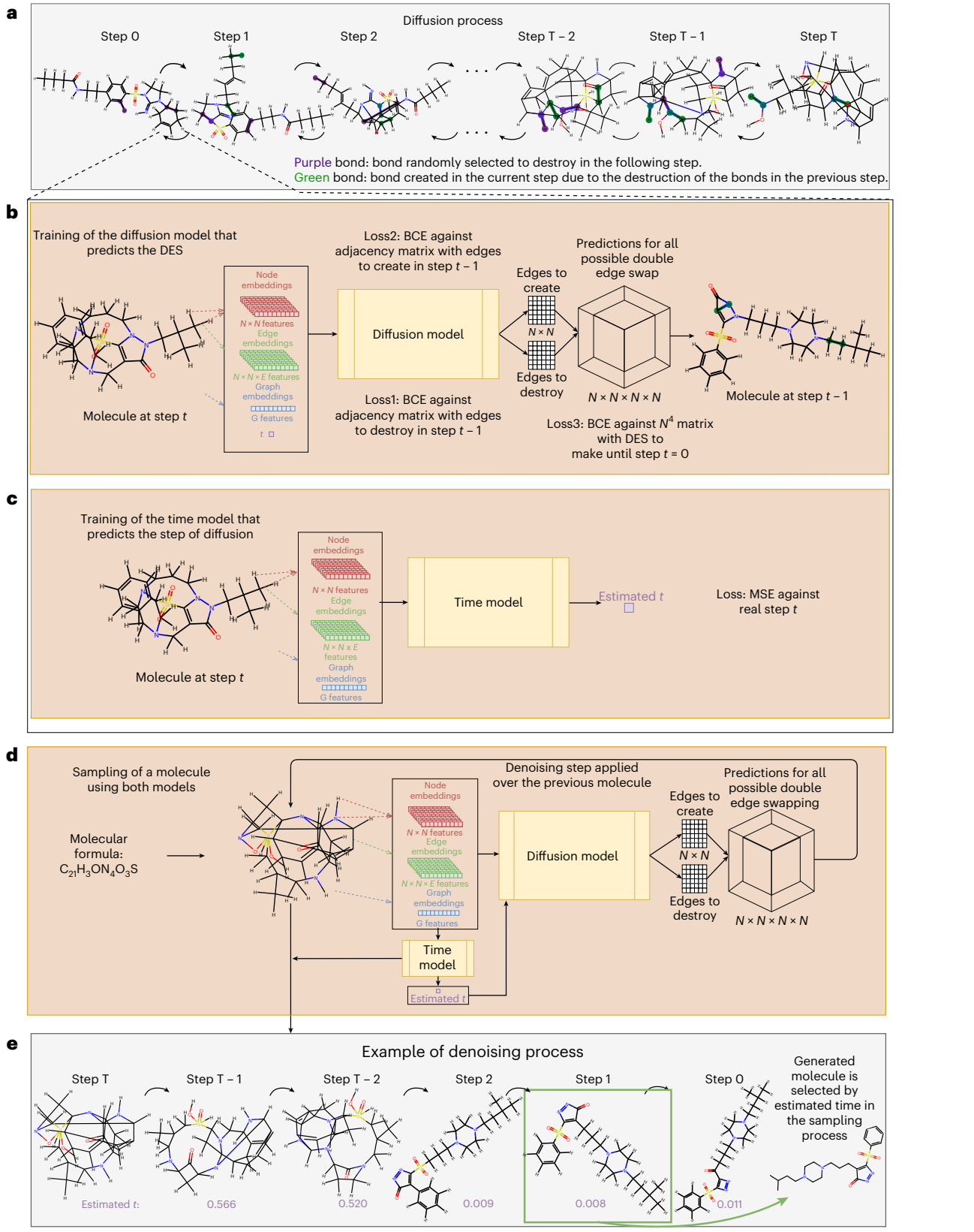


Fig. 1 | Constrained collaborative graph diffusion model, CoCoGraph. **a**, Constrained diffusion process. We introduce noise in the molecular graph by swapping two chemical bonds at each step. We then train diffusion and time models to revert this process. **b**, Diffusion model. At each step, it receives molecular features and the time step as an input and assigns a score to all possibilities of edge swaps. **c**, Time model. It receives molecular features and estimates the time step of the current molecular graph. **d**,**e**, Sampling model (**d**) and example (**e**). We use trained diffusion and time models in collaboration to generate a trajectory of denoising starting from a random molecular graph with a defined molecular formula. We then select the molecule with the smallest predicted time as the generated molecule (**e**).

https://doi.org/10.1038/s42256-026-01229-5

Table 1 | Model comparison on the GuacaMol benchmark evaluated against the PubChem database

Model	Number of parameters	Valid (%)	Valid and unique (%)	Valid, unique and new (%)	KL divergence (%)	Samples per second
JTVAE[a]	5.3 million	**100**	99.9	–	55.9 ± 0.8	–
DiGress[b]	4.6 million	83.4	83.4	81.6	91.4 ± 0.1	0.75
GDSS[a]	**0.074 million**	95.7	94.3	91.0	69.8 ± 0.4	**7.6**
GruM[a]	8.2 million	98.5	98.4	92.9	86.6 ± 0.3	1.14
DeFoG strict[b]	4.6 million	90.4	90.4	85.3	93.6 ± 0.1	0.40[d]
DeFoG relaxed[b]	4.6 million	98.5	98.5	93.0	93.4 ± 0.2	0.40[d]
CoCoGraph (FPS) no time[c]	3.1 million	**100**	**100**	**97.5**	94.4 ± 0.6	1.16
CoCoGraph (BASE)[c]	0.471 million + 0.063 million	**100**	99.8	95.7	95.7 ± 0.4	1.19
CoCoGraph (FPS)[c]	3.1 million+1.3 million	**100**	99.9	95.7	**96.3 ± 0.4**	0.98

For the BASE and FPS versions of CoCoGraph and baseline models, we show the number of parameters, the percentage of valid molecules, the percentage of valid and unique generated molecules, the percentage of distinct generated molecules not in the known chemical universe as represented by PubChem (filtered to ≤70 atoms without any overlap with the training datasets, yielding 94.7 million reference molecules), the KL divergence score and the number of samples per second. All models were evaluated on generated molecules with up to 70 atoms. KL divergence score represents the mean and standard deviation over five runs against 1 million molecules randomly sampled from filtered PubChem. Bold indicates the best performance, underlining indicates second-best performance. The training dataset is indicated by a superindex next to each model's name: [a]ZINC250K; [b]GuacaMol; [c]our dataset. [d]Inference times are reported from the original publication; all other measurements were performed on identical hardware.

standardized metrics to assess the quality and diversity of generated molecules. We compare CoCoGraph against six state-of-the-art models: the junction tree variational autoencoder (JTVAE)[11], DiGress[29], GDSS[26], GruM[33] and DeFog in both strict and relaxed modes[34]. To ensure fair comparison across models trained on different datasets, we evaluate all models against a processed PubChem reference database filtered to ≤70 atoms and with no overlap with any training dataset (94.7 million molecules), which provides a comprehensive representation of the known chemical universe and allows reliable evaluation of generalization ability. As an additional naive baseline, we also consider a model that makes random modifications to real molecules (Supplementary Section 4).

We consider two different CoCoGraph models (Table 1; Methods). The BASE version of our model requires 534,000 parameters in total (471,000 for the diffusion model and 63,000 for the time model), an order of magnitude fewer than all comparators except GDSS (74,000). The fingerprint-enhanced (FPS) CoCoGraph model, which incorporates molecular fingerprints as additional inputs to improve edge swapping prediction in the diffusion model and time prediction in the time model, uses 4.4 million parameters, which is still fewer than all baselines except GDSS.

Both CoCoGraph models achieve 100% chemical validity, which is a direct consequence of the constrained diffusion approach inherently obeying chemical rules throughout the diffusion process. Perfect validity is achieved without sacrificing uniqueness (99.8% for BASE and 99.9% for FPS) or novelty (95.7% for both versions), demonstrating that imposing valence constraints does not overly restrict chemical space exploration. Generated molecules are also distinct from the original molecules and random molecular graphs that serve as seed for the generation process (Supplementary Section 2). Some non-diffusion approaches such as JTVAE also yield perfect validity; however, as we show next, these models generally have much poorer property-distribution matching (Table 1). Other diffusion models show lower validity rates. Notably, all CoCoGraph variants achieve the higher novelty rate among all evaluated models indicating superior exploration of the chemical space.

Beyond validity, uniqueness and novelty, the distribution of physicochemical properties for generated molecules is the critical metric for evaluating generative models. We aim to generate molecules that are new but plausible, that is, that have physicochemical properties statistically similar to those of real molecules. The GuacaMol benchmark quantifies this by measuring the Kullback–Leibler (KL) divergence between distributions of ten physicochemical properties of generated molecules and those of real molecules. As demonstrated by the KL divergence scores (Table 1), CoCoGraph generates molecules whose property distributions most closely match those of real molecules, achieving scores of 95.7% and 96.3% for the BASE and FPS versions. respectively. (The CoCoGraph time-ablated FPS model achieves significantly lower KL divergence of 94.4%, indicating that the time model improves distribution learning; Supplementary Section 3). These scores significantly outperform all baseline models. The CoCoGraph FPS model achieves the best overall performance, demonstrating that our constrained collaborative approach more accurately captures the underlying distribution of molecular properties than existing approaches, producing molecules that better reflect the characteristics of molecules in the known chemical universe.

To provide more nuance, we analyse in detail each of the ten molecular properties used in the GuacaMol benchmark. For this analysis, we focus on the FPS CoCoGraph model (Fig. 2). The distributions of molecular weight, internal similarity, Bertz complexity index and number of aromatic rings are best approximated by CoCoGraph (Fig. 2a,c,d,e). For molecular log *P* (Fig. 2b), CoCoGraph outperforms JTVAE and DiGress but shows slightly lower performance than DeFoG models. For hydrogen-bond donors, all baseline models achieve better distribution matching. Overall, CoCoGraph FPS outperforms JTVAE and DiGress on eight out of ten properties, and both DeFoG variants on six out of ten (Fig. 2g–j). The BASE CoCoGraph model performs slightly worse than the FPS model, but still better than the comparators (Extended Data Fig. 1).

This improvement across multiple models and molecular characteristics underscores the effectiveness of our constrained collaborative approach in generating chemically valid and structurally new molecules that are both realistic and diverse in terms of their physicochemical properties. For example, although JTVAE-generated molecules are always valid and plausible, the analysis of the distributions shows that their physicochemical properties are restricted to narrow ranges, indicating limited diversity. The same, although to a much lesser extent, is generally true for molecules generated by other baselines. In terms of computational efficiency, CoCoGraph achieves the second fastest inference speed, generating 1.19 molecules per second (BASE) and 0.98 molecules per second (FPS). Although GDSS achieves faster generation than CoCoGraph, it exhibits considerably lower performance on the matching of physicochemical property distributions. Thus, CoCoGraph achieves the best generation quality while being computationally efficient.

https://doi.org/10.1038/s42256-026-01229-5

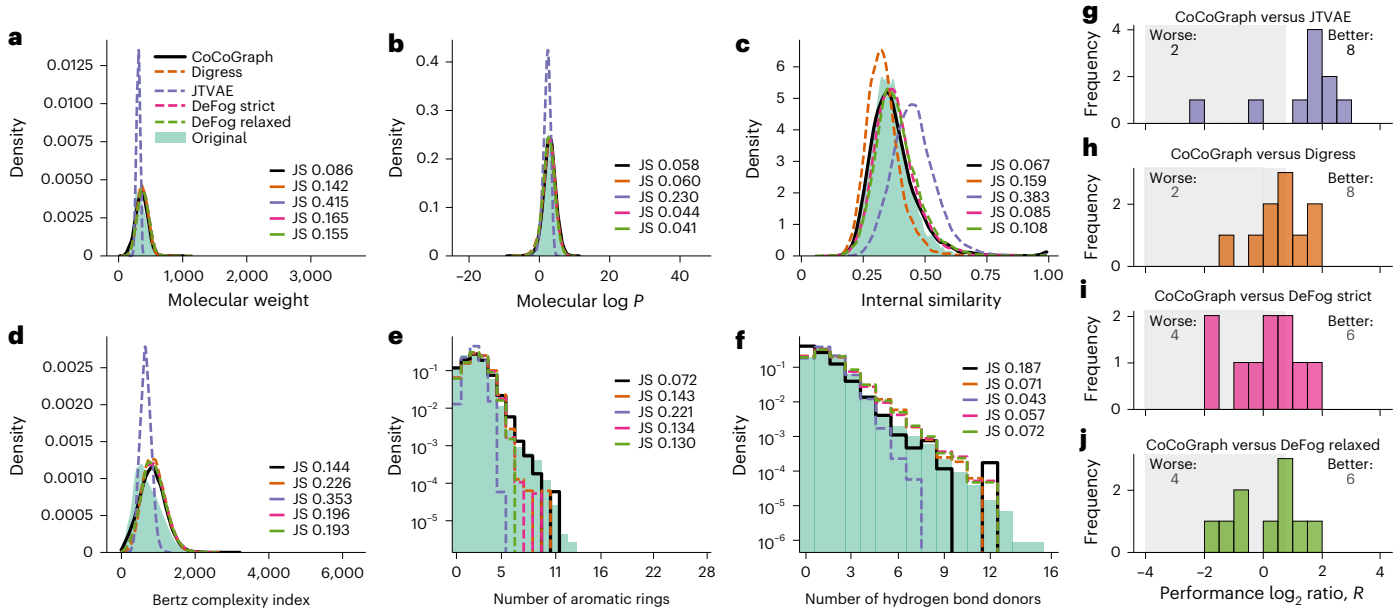


Fig. 2 | Performance comparison of CoCoGraph FPS on GuacaMol benchmark properties. a–f, Distributions of six molecular properties: molecular weight (**a**); molecular log *P* (**b**); internal similarity (**c**); Bertz complexity index (**d**); number of aromatic rings (**e**); and number of hydrogen-bond donors (**f**). For each property, the distribution of values calculated for molecules generated by CoCoGraph (black line) is compared with that of the original molecules from the PubChem reference database (green area) and with those of molecules generated by JTVAE (purple dashed line), DiGress (orange dashed line), DeFoG strict (pink dashed line) and DeFoG relaxed (light green dashed line). We show JS distance values between each model and the original distribution. **g–j**, Performance $\log_2$ ratio of JS distances between CoCoGraph FPS and comparator models for the ten properties in the GuacaMol benchmark[38]: JTVAE (**g**); Digress (**h**); DeFog strict (**i**); DeFog relaxed (**j**). Positive values indicate CoCoGraph FPS outperforming the comparator model, whereas negative values indicate poorer performance.

Molecules generated by CoCoGraph are plausible on a wide range of chemical properties

Although standard benchmarks such as GuacaMol provide a useful starting point for evaluating molecular generative models, they only assess performance on a limited set of physicochemical properties. In addition, as soon as a benchmark becomes standard, it starts being used during algorithm design and evaluation, potentially leading to overfitting of the corresponding properties. To provide a more comprehensive evaluation of our model's ability to generate chemically plausible molecules, we extended our analysis to a diverse set of 36 chemical properties. To reduce selection bias, we employed OpenAI's O1-mini model, an external agent not trained on our model or aware of our training data and results, to identify a representative and diverse set of molecular descriptors that can be calculated with RDKit[43]. This approach avoids a potential, unconscious tendency to favour properties where our model performs well, and reduces the likelihood of selecting properties that are biased toward our model's training or performance characteristics. The properties span a wide range of molecular characteristics including size and composition metrics, topological features, electronic properties and drug-likeness indicators (Extended Data Table 1).

Following the same methodology as in the previous section, we calculated Jensen–Shannon (JS) distances between the distributions of each property for molecules in the PubChem reference database and molecules generated by the different models that we consider (Extended Data Table 2). In Fig. 3a–j, we show the distributions of ten properties.

The comprehensive comparison across all 36 properties is summarized in Fig. 3k–n, which shows the $\log_2$ ratio of JS distances between CoCoGraph and each baseline. CoCoGraph outperforms other baselines in at least 66.6% of properties. The ten properties shown in Fig. 3a–j were selected to reflect this performance distribution: eight properties (Fig. 3a–g,i) show CoCoGraph outperforming all competing models; Fig. 3h shows quantitative estimate of drug likeness, where all models except JTVAE outperform CoCoGraph; and Fig. 3j shows NHOH count, where all baselines outperform CoCoGraph. CoCoGraph shows particular strength in topological features (Balaban's *J* index), electronic properties (number of valence electrons and maximum absolute partial charge) and structural descriptors (heavy atom count, ring count, topological polar surface area (TPSA))—properties that are critical for applications in medicinal chemistry and drug discovery. In Extended Data Fig 2, we show that the BASE CoCoGraph model also outperforms, overall, the benchmark algorithms, despite its much smaller number of parameters. These results further validate the effectiveness of our constrained collaborative approach.

A large database of realistic synthetic molecules

The computational efficiency of CoCoGraph, with its reduced parameter count and high sampling efficiency, enables molecule generation at scale with modest computational resources. Our model produces thousands of chemically valid molecules per hour on a single mid-range graphics processing unit (GPU), allowing us to create a comprehensive database containing 8.2 million molecules, with only 7.1% redundancy. This high efficiency, combined with the 95.7% novelty rate demonstrated in Table 1, means that our database contains approximately 7.3 million new and unique, chemically valid molecules that are not present in the PubChem database (see Extended Data Fig. 3 for a random sample of 50 generated molecules). Such a large-scale database of chemically valid and new molecules may be a valuable resource for exploring new regions of drug-like chemical space and accelerating discovery across multiple domains, including drug development and materials science.

To evaluate how realistic our synthetic molecules appear to domain experts, we developed a molecular Turing-like test. Experts in organic chemistry, biochemistry and related fields were presented with pairs of molecules sharing the same molecular formula—one real molecule from our original dataset and one generated by CoCoGraph (Extended Data Fig. 4). By matching molecular formulas, this design

https://doi.org/10.1038/s42256-026-01229-5

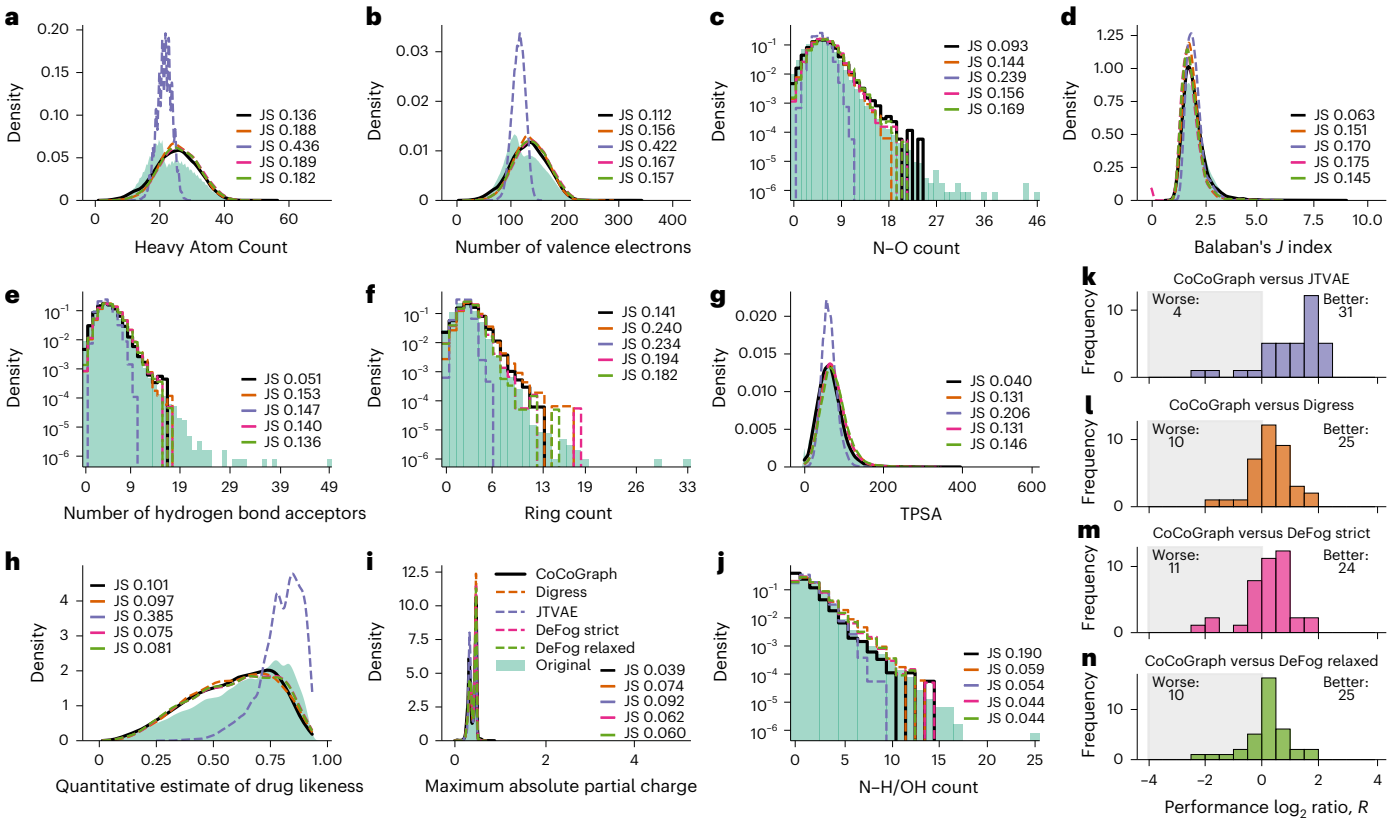


Fig. 3 | Detailed performance comparison of CoCoGraph FPS on a subset of 36 chemical properties. a–**j**, Distributions of ten molecular properties: heavy atom count (**a**); number of valence electrons (**b**); N–O count (**c**); Balaban's *J* index (**d**); number of H acceptors (**e**); ring count (**f**); TPSA (**g**); quantitative estimate of drug likeness (**h**); maximum absolute partial charge (**i**); and NH/OHCount (**j**). For each property, the distributions for molecules generated by the CoCoGraph FPS model (black line) is compared with that of the molecules in the PubChem reference database (green distribution) and with those of molecules generated by JTVAE (purple dashed line), DiGress (orange dashed line), DeFoG strict (pink dashed line) and DeFoG relaxed (light green dashed line). We show JS distance values between each model and the original distribution. **k**–**n**, Performance $\log_2$ ratio of JS distances between CoCoGraph FPS and comparator models for the 36 the properties considered: JTVAE (**k**); Digress (**l**); DeFog strict (**m**); DeFog relaxed (**n**). Positive values indicate that CoCoGraph FPS outperforming the comparator model, whereas negative values indicate poorer performance.

controlled for variables such as molecular size and atom composition, forcing participants to rely on chemical knowledge about structural plausibility and physicochemical properties to distinguish between real and generated molecules. Participants were asked to identify which molecule was the original across 20 rounds and provide their level of expertise.

We collected responses from 121 experts, totalling 2,420 individual molecule pair assessments. The results reveal that experts achieved an overall accuracy of 62% at distinguishing real molecules from those generated by CoCoGraph (Fig. 4a). This accuracy is significantly different but close to the 50% baseline that would be expected from random guessing. Breaking down the results by level of expertise (Fig. 4b) shows a slight improvement in discrimination ability with increased expertise—undergraduate participants achieved 60% accuracy, whereas graduate participants achieved 64% accuracy.

To further understand the strengths and potential biases of CoCoGraph, we break down the analysis along several dimensions (Fig. 4c–g). Larger synthetic molecules are slightly easier for experts to identify, suggesting increased generation difficulty with molecular complexity. Molecules with fewer rings are harder to classify, with acyclic molecules showing expert performance statistically compatible with random guessing (Fig. 4d). Similarly, predominantly aliphatic molecules show performance compatible with random guessing (Fig. 4f). Although the null hypothesis of real and generated molecules being indistinguishable cannot be rejected, this does not prove that they are indeed indistinguishable. However, a Bayesian model selection analysis confirms that, for these cases, a model assuming that real and generated molecules are indistinguishable to participants is favoured over alternative models (Supplementary Section 5). We found no clear tendency for conformational flexibility or functional groups (Fig. 4e,g).

In summary, we found that: (1) even subjects with university-level training in organic chemistry failed to correctly identify real molecules in close to four out of ten attempts; (2) for some particular molecules (acyclic and predominantly aliphatic), performance was actually compatible with random guessing; and (3) there were no particular classes of molecules that were systematically wrong and easy to spot, which would indicate a clear bias. These suggest that CoCoGraph captures the underlying structural patterns and chemical relationships of real molecules with high fidelity, while still exploring new regions of chemical space.

Application to drug discovery through database search and inpainting-based conditional generation

The 8.2 million molecule database provides a large collection of new molecules with realistic physicochemical properties. As a demonstration of potential applications, we used this database to identify new molecules with properties similar to paracetamol. For this, we identified nine key molecular properties and ranked all molecules in the database based on their Euclidean distance to paracetamol in the space defined by these nine properties (conveniently normalized). The top six candidates (Fig. 5a) were structurally diverse but maintained property profiles similar to paracetamol.

https://doi.org/10.1038/s42256-026-01229-5

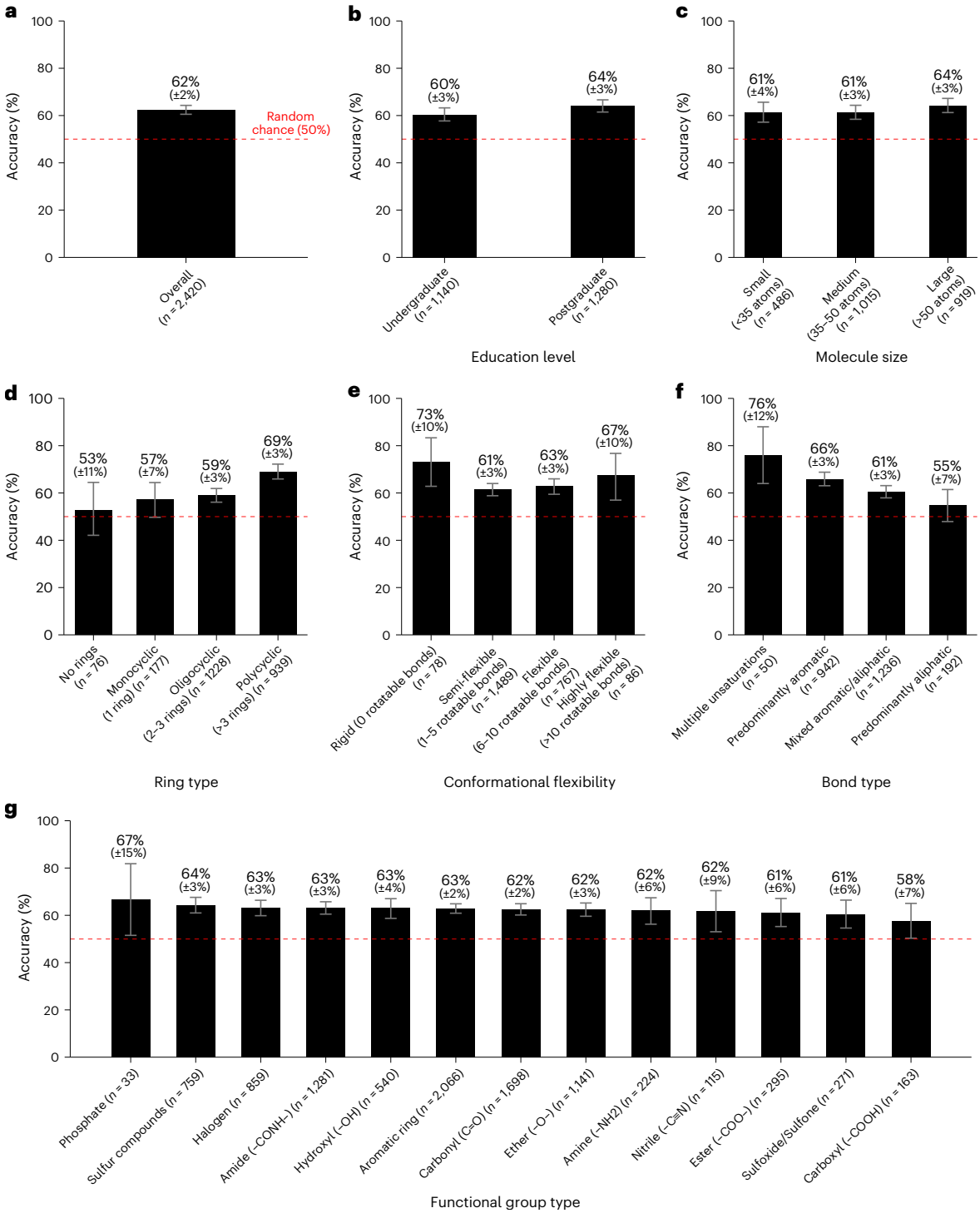


Fig. 4 | Performance in the Turing-like test. We assessed the performance of participants in the Turing-like test by computing their accuracy at correctly identifying the original, non-generated molecule over all attempts. Error bars represent the standard error of the mean calculated by bootstrapping. **a**, Overall accuracy of participants in the Turing-like test. **b**, Accuracy by level of education in organic chemistry. **c**, Accuracy by molecular size in terms of the number of atoms. **d**, Accuracy by ring structure. **e**, Accuracy by conformational flexibility of the molecules. **f**, Accuracy by bond type. **g**, Accuracy by functional group.

Finally, to further illustrate potential uses of CoCoGraph, we address the same problem of identifying compounds with properties similar to paracetamol by modifying the sampling process to achieve targeted molecular design through inpainting-based conditional generation. In particular, the modified sampling combines a fixed original molecule (for example, paracetamol) and a random molecular graph corresponding to a fragment with arbitrary molecular formula. The fragment is connected to the original molecule, and CoCoGraph's denoising is then applied only to the fragment while keeping the original molecule fixed. This enables the generation of new molecular candidates that preserve desired structural features while exploring chemical space through controlled additions.

The inpainting-based conditional sampling generates candidates by adding small fragments (2–5 heavy atoms) and medium fragments (6–15 heavy atoms). We then ranked those candidates as before, according to their Euclidean distance from paracetamol across nine

https://doi.org/10.1038/s42256-026-01229-5

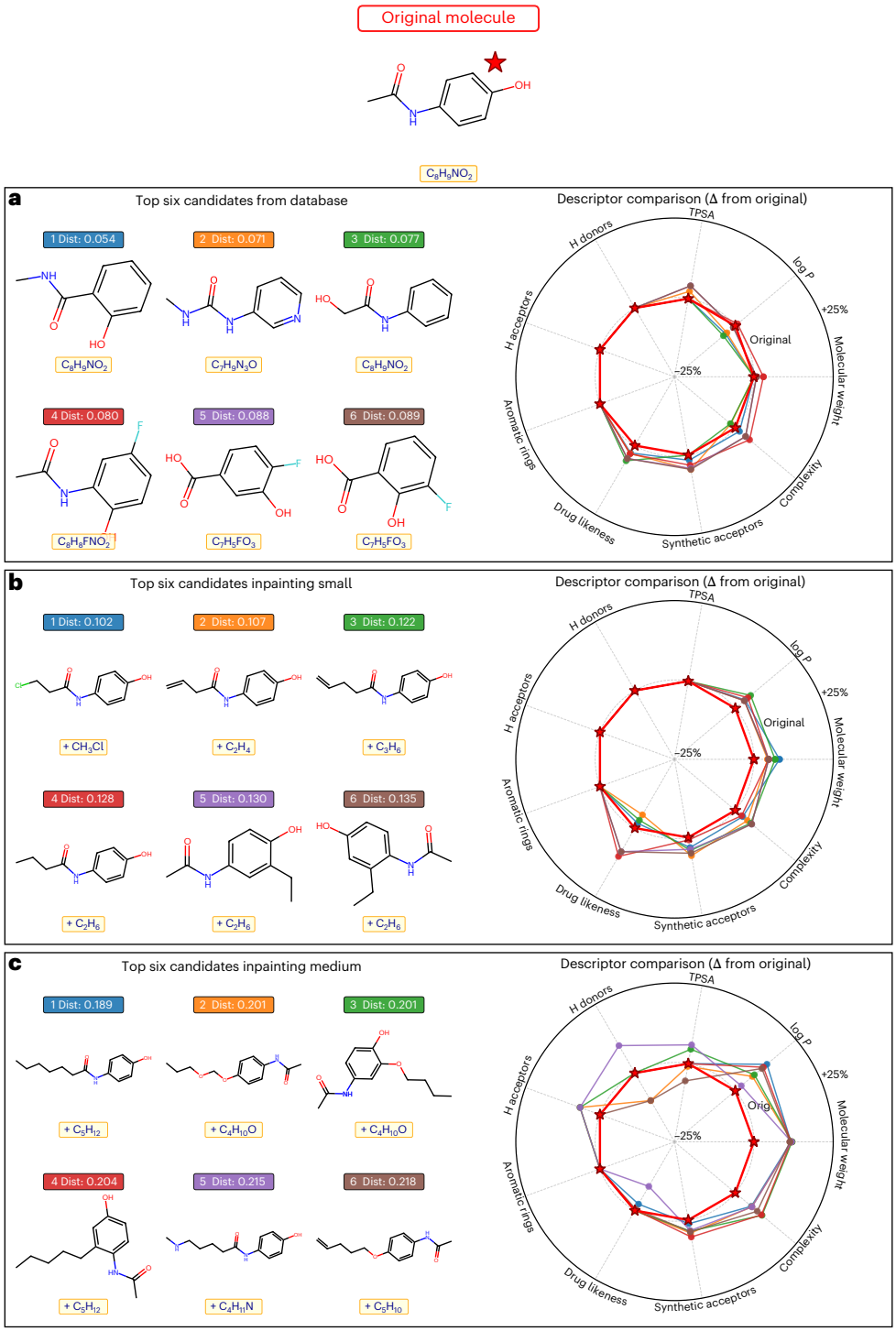


Fig. 5 | Molecular candidates similar to paracetamol obtained through database search and inpainting-based generation. a, Top six molecular candidates found by searching the 8.2 million molecule database for molecules with physicochemical properties similar to paracetamol. **b**, Top six molecular candidates generated by adding small fragments (2–5 heavy atoms) to paracetamol by means of inpainting-based conditional generation. **c**, Top six candidates generated by adding medium fragments (6–15 heavy atoms) to paracetamol by means of inpainting-based conditional generation. For each candidate, we show its rank, Euclidean distance (dist) to paracetamol across nine physicochemical properties and a radar chart of its properties.

physicochemical properties. In Fig. 5b,c we show the top candidates for each fragment size. This demonstrates that CoCoGraph can generate plausible candidates through inpainting-based conditional generation, which could be valuable for lead optimization in drug discovery, where specific molecular scaffolds must be preserved while exploring structural variations.

Discussion

CoCoGraph is a collaborative constrained discrete diffusion model that achieves perfect chemical validity by design and generates diverse and realistic molecular structures, as verified through comprehensive benchmarks and expert evaluation. Our approach addresses key challenges that have limited previous molecular generative approaches.

Whereas previous models attempt to learn chemical rules into their parameters, CoCoGraph imposes constraints directly into the generative process. By building chemical validity into the diffusion process, our approach is more efficient and allows our model to focus entirely on capturing the subtle structural patterns of real molecules. As a result, when comparing the distributions of the physicochemical properties of the generated molecules, CoCoGraph outperforms state-of-the-art models such as Digress and DeFog.

Another relevant innovation is the collaborative mechanism between our diffusion and time models. Ablation experiments using just the diffusion model with the actual time step as input show lower performance because the denoising process progresses at different rates for different molecules. The time model resolves this by learning how far a molecular graph is from complete denoising, providing a more informed measure of progress. Unlike other approaches that use predetermined schedules or revision steps, our model adapts its predictions to the actual state of the molecule, resulting in a more precise generation of molecules.

Finally, to show CoCoGraph's efficient exploration of the chemical space, we generated a database of 8.2 million synthetic molecules, with 7.1% redundancy and 95.7% novelty. This database will be a valuable resource for the community, as we have illustrated in the application to the exploration of new drug-like chemical spaces and ranking of possible drug candidates.

Importantly, CoCoGraph can be tailored to different situations. By design, the molecular formula is fixed, which could be limiting in applications in which the molecular formula is unknown. However, in unseeded mode, CoCoGraph can start directly from any user-specified molecular formula and generate molecules with that composition. Therefore, when exploration of multiple formulas is required, one would just need to solve the simpler task of generating valid molecular formulas that CoCoGraph can use as input.

Another issue is how to extend CoCoGraph to generate molecules that have more than $n = 70$ atoms. The computational complexity of computing possible double edge swaps among four bonds is $O(n^4)$ but, for applications requiring larger molecules, the model can be retrained with adjusted dimensions and additional computational resources, as this limitation is not inherent to the constrains-by-design principle or the collaborative mechanism.

Our results open the widow to new research directions. A better understanding of how the model generates molecules would be useful to assess how it explores the chemical space[44]. Our synthetic molecule database could be used to explore chemical space and try to discover functionalities for the molecules generated by the model[45]. CoCoGraph could be extended for conditional molecule generation based on desired properties or from observed molecular data. In particular, CoCoGraph may be useful for inverse problems in mass spectrometry (MS)[46], where precursor mass can often constrain molecular formula and thus restrict the search space; conditioning denoising on features derived from MS/MS spectra could enable spectrum-to-structure reconstruction. Finally, our inpainting-based conditional generation could address drug optimization by enabling fragment growing or replacement while preserving a fixed scaffold. These directions would build upon our constrained, collaborative approach that guarantees valid molecule generation with property distributions closely matching real molecules.

Methods

Molecular data processing

We used for training a curated dataset of 2.25 million molecules derived from several established molecular databases including PubChem, ChEMBL, ZINC and NIST. The process of curating and processing our molecular database involved multiple steps to ensure quality and consistency.

Initially, all molecules were represented in simplified molecular input line entry system (SMILES) format. Using RDKit, we canonicalized these SMILES strings based on their InChI keys to establish a standardized representation. During this process, we eliminated duplicates and molecules that could not be properly converted. We deliberately chose to work with a reduced but stable set of molecules that are consistently represented, rather than incorporating a larger quantity of molecules from different datasets with inconsistent representations. This approach ensures that our model learns from a clean, uniform dataset rather than having to accommodate representation inconsistencies that might exist across different molecular collections.

After canonicalization, the SMILES strings were converted to molecular graphs composed of nodes (atoms) and edges (bonds). For each molecule, we extracted the explicit heavy atoms from the RDKit molecule object. In addition, implicit hydrogen atoms were derived and represented as explicit nodes in our graph. This approach treated all atoms, including hydrogens, as first-class entities in the molecular graph.

To ensure fair comparison across all models and manage computational resources, we restricted our dataset to molecules containing between 5 and 70 atoms. This upper limit represents a practical choice for our experimental validation rather than an architectural limitation—the constraints-by-design principle and collaborative architecture of CoCoGraph can accommodate larger molecules if needed. The 70-atom threshold was selected because: (1) it enables comprehensive evaluation while maintaining computational efficiency during training and validation; (2) the GuacaMol benchmark database, on which other baseline models are typically trained, contains 90% of molecules with 70 atoms or fewer (median: 48 atoms); and (3) our initial dataset of 2.25 million molecules similarly has 80% of compounds with 70 atoms or fewer (median: 50 atoms). Thus this threshold did not substantially restrict the chemical diversity of our training data. After applying this size filter, approximately 1.67 million molecules remained in our dataset. We note that the distributions of physicochemical properties between complete and size-restricted datasets were highly similar (KL divergence scores >96%), confirming that this filtering does not introduce bias in model evaluation.

Benchmark evaluation and baseline models

To provide a comprehensive and fair comparison of CoCoGraph against existing generative models, we established a common evaluation framework using the PubChem database as a reference dataset and systematically selected baseline models based on reproducibility criteria.

PubChem reference dataset. To ensure fair comparison across all models trained on different datasets, we evaluated all models against the PubChem database, which is the largest publicly available database of molecules and provides the most comprehensive representation of the known chemical universe. We processed the entire PubChem database through the following steps: (1) downloaded all molecules in SMILES format; (2) canonicalized all SMILES strings using RDKit based on their InChI keys to establish standardized representations; (3) removed duplicate molecules after canonicalization; (4) filtered molecules to include only those with 5 to 70 atoms to enable fair comparison across all baseline models; and (5) removed all molecules present in any of the training datasets used in this work (our dataset, GuacaMol training set and ZINC250K training set). This processing resulted in a reference dataset of 94.7 million unique molecules.

For the evaluation metrics reported in Table 1, we calculated novelty by comparing generated molecules against the complete processed PubChem reference dataset (94.7 million molecules). For the KL divergence score, which is the primary metric in the GuacaMol benchmark for assessing distribution matching quality, we randomly selected five independent groups of 1 million molecules each from the processed PubChem reference dataset (restricted to molecules with fewer than 70 atoms). We used the same five groups for evaluating all models to ensure fair comparison.

Baseline model selection and evaluation. For baseline models, we generated molecules as follows. For JTVAE, we used the publicly available code to train the model and generated molecules. For DiGress, GDSS and GruM, we used the publicly available pretrained model weights and sampling code to generate molecules. For DeFoG, the authors provided pregenerated molecules which we used directly. For all models, we filtered to molecules with 70 atoms or fewer.

Inference time measurements (reported as molecules per second in Table 1) were calculated on identical hardware (NVIDIA RTX 4090 GPU) for all models except DeFoG. For CoCoGraph, DiGress, GDSS and GruM, we measured the wall-clock time required to generate 10,000 molecules and calculated the throughput as molecules per second. We do not report inference times for JTVAE. For DeFoG, we report the inference times as published in their original paper.

Molecular graph diffusion

As introduced in the main text, our approach uses a discrete diffusion process based on DES that preserves atomic valence constraints throughout the diffusion trajectory, ensuring chemical validity. Here we describe the mathematical formulation of the noising process and the denoising process. The details about how the collaborative interaction between our diffusion and time models enables efficient generation of valid molecules are provided in 'Sampling of new molecules'.

Noising process. Our diffusion process is built upon a valence-constrained mechanism that ensures that all molecular graphs throughout the diffusion trajectory maintain chemical validity. The core of this mechanism is the DES operation, in which we: (1) randomly select two edges $e_1 = (i,j)$ and $e_2 = (k,l)$ in the molecular graph G_t; (2) remove these edges; and (3) create two new edges $e_3 = (i,k)$ and $e_4 = (j,l)$ by cross-connecting the atoms of the original edges. This process ensures that each atom maintains its original valence because each atom loses one bond and gains another. By iterating this process, the molecular graph diffuses toward a Molloy–Reed distribution[35,36,42], which is the maximum-entropy distribution over graphs with a fixed degree sequence.

Mathematically, the DES operation can be described as a transformation $T: G_{t-1} \to G_t$. Since each DES operation affects four bonds (two bonds are removed and two new bonds are created), the number of DES operations needed to completely randomize a molecule is approximately 25% of the total number of bonds. Unlike existing discrete diffusion models, which use a constant transition matrix, our diffusion approach is a Markov process where the transition probability from a molecular graph G_{t-1} to a noisier graph G_t depends on G_{t-1}. Therefore, there is no general closed-form expression for the t-step transition matrix. The multidimensional transition matrix Q_t has elements $[Q_t]_{ijkl}$, which represent the probability that an edge (i,j) and an edge (k,l) are removed (note that there might be other edges remaining between (i,j) and/or (k,l) if the original multiplicity of those edges was larger than one: that is, if the bonds were double or triple) and edges (i,k) and (j,l) are created (note that edges (i,k) and/or (j,l) may already exist, in which case we simply increase the multiplicity of such edges) and are given by

$$[Q_t]_{ijkl} = \frac{F_t(i,j,k,l)}{\sum_{i',j',k',l'} F_t(i',j',k',l')}$$

Where $F_t(i,j,k,l)$ is function that determines the feasibility of removing edges (i,j) and (k,l) and creating valid edges (i,k) and (j,l) given molecular graph G_t. This function is defined as

$$F_t(i,j,k,l) = \begin{cases} 1 & \text{if removing } (i,j) \text{ and } (k,l) \text{ and creating } (i,k) \\ & \text{and } (j,l) \text{ results in a valid molecular graph} \\ 0 & \text{otherwise} \end{cases}$$

For $F_t(i,j,k,l)$ to equal 1, the following conditions must be met.

(1) All four nodes must be distinct, $i \neq j \neq k \neq l$.
(2) Edges (i,j) and (k,l) must exist in G_t.
(3) The new edges must give rise to, at most, triple bonds.
(4) The connectivity of the resulting graph G_{t+1} must be maintained.

Denoising process. The denoising process in our molecular graph diffusion is mathematically formalized as an optimization process that seeks to reverse the structural modifications introduced during the noising process. More precisely, let G_t be the molecular graph at time t during the diffusion process and let G_0 be the original molecular graph. The objective of the denoising process is to find a sequence of transformations $T^{-1}: G_t \to G_{t-1}$ that reverse the structural modifications and recover a chemically valid molecular graph with properties similar to those of real molecules. This process is implemented through our constrained collaborative mechanism, which employs two specialized models that work in tandem—a diffusion model and a time model.

Each denoising step takes as input the molecular graph G_t and the normalized time step t, and selects a suitable DES. To do this, the diffusion model learns three probability distributions: (1) the probability $[Q_t^{-1}(\theta,\theta_f,\theta_b)]_{ijkl} = \mathrm{Prob}_{\theta,\theta_f,\theta_b}(\text{select } (i,j)\,\&\,(k,l)|G_t,t)$ of selecting (i,j) and (k,l) for the next denoising DES; (2) the probability $[P_t^{\text{form}}(\theta,\theta_f)]_{ij} = \mathrm{Prob}_{\theta,\theta_f}((i,j) \text{ exists}|G_t,t)$ of forming an edge (i,j); and (3) the probability $[P_t^{\text{break}}(\theta,\theta_b)]_{ij} = \mathrm{Prob}_{\theta,\theta_b}((i,j) \text{ does not exist}|G_t,t)$ of breaking an edge (i,j).

Some parameters θ of these distributions are shared among all models, whereas others (θ_f, θ_b) are specific to different distributions. These parameters are learned so as to minimize three corresponding binary cross-entropy (BCE) loss functions. For DES prediction, we minimize

$$\mathcal{L}_{\text{BCE-DES}} = -\frac{1}{N_q}\sum_{(i,j,k,l)}\Big[y_{ijkl}^{t-1}\log[Q_t^{-1}(\theta,\theta_f,\theta_b)]_{ijkl} + \left(1-y_{ijkl}^{t-1}\right)\log\left(1-[Q_t^{-1}(\theta,\theta_f,\theta_b)]_{ijkl}\right)\Big] \tag{1}$$

where N_q is the number of feasible quadruplets (i,j,k,l) and y_{ijkl}^{t-1} is the binary label indicating whether DES (i,j) and (k,l) is the one that actually led from G_{t-1} to G_t during the (forward) noising process.

For bond formation prediction we minimize

$$\mathcal{L}_{\text{BCE-form}} = -\frac{1}{N_P}\sum_{(i,j)}\Big[y_{ij}^{f0}\log[P_t^{\text{form}}(\theta,\theta_f)]_{ij} + \left(1-y_{ij}^{f0}\right)\log\left(1-[P_t^{\text{form}}(\theta,\theta_f)]_{ij}\right)\Big] \tag{2}$$

where N_P is the number of pairs of nodes in the molecular graph and y_{ij}^{f0} indicates whether edge (i,j) should be formed with respect to the original molecule (time $t = 0$).

Finally, for bond breakage prediction we minimize

$$\mathcal{L}_{\text{BCE-break}} = -\frac{1}{N_E}\sum_{(i,j)}\Big[y_{ij}^{b0}\log[P_t^{\text{break}}(\theta,\theta_b)]_{ij} + \left(1-y_{ij}^{b0}\right)\log\left(1-[P_t^{\text{break}}(\theta,\theta_b)]_{ij}\right)\Big] \tag{3}$$

where N_E is the number of pairs of edges in the molecular graph and y_{ij}^{b0} indicates whether edge (i,j) should be broken with respect to the original molecule (time $t = 0$).

The denoising process also requires a time model, which learns to predict the normalized time step of the diffusion process. This model takes as input the molecular graph G_t, node features X, edge features E and the graph features g, and estimates how far the current molecular graph is from the original molecule, providing a normalized time

value between 0 and 1. The time model is trained using mean squared error (MSE) loss

$$\mathcal{L}_{\mathrm{MSE}} = (t_{pred} - t_{real})^2 \quad (4)$$

In 'Sampling of new molecules', we describe how these models are used in the actual process of sampling, that is, of generating new molecules.

Diffusion model and time model architectures

Our collaborative constrained diffusion approach is implemented through two separate neural network architectures that process molecular graph features and work together during inference—the diffusion model and the time model. These architectures are designed to efficiently handle the graph-structured data while keeping the parameter count low. The valence constraints are enforced by the DES mechanism rather than by the neural architectures, which just make the predictions for pairs of edges to choose during the denoising process. Extended Data Fig. 5 provides a comprehensive overview of our model architectures for the BASE CoCoGraph model.

Diffusion model architecture. The diffusion model employs a GNN architecture consisting of two main components: (1) a message-passing component that processes node, edge and graph features and the diffusion time; and (2) a prediction component that estimates edge probabilities from processed features (Extended Data Fig. 5a).

The most important component of the diffusion model is the message-passing component (Extended Data Fig. 5c), which uses a sequence of three enhanced graph isomorphism networks[47] with edge features (EnhancedGINE) layers. These layers extend the standard GINE layers by incorporating global graph features directly into the message-passing mechanism. Each EnhancedGINE layer transforms node features into 124-dimensional embeddings, with the final layer outputs capturing comprehensive atomic environments. The architecture can be summarized as follows.

(1) Initial feature processing: Node features X, edge features E, graph features g and the diffusion time t are processed by the first EnhancedGINE layer.
(2) Hidden representations: The output embeddings are passed through a nonlinear activation function followed by a feedforward layer. This is repeated for the second and third EnhancedGINE layers, with residual connections between layers to preserve information flow.
(3) Node embedding aggregation: After the EnhancedGINE layers, we obtain embeddings for each node that capture its local and global context within the molecular graph.
(4) Edge probability: For each pair of nodes, the corresponding node embeddings are concatenated to each other and to edge and graph features, and processed through two different feedforward networks, each with 256 hidden units (Extended Data Fig. 5d), to predict: (1) the probability of bond formation between each pair of atoms; (2) the probability of bond breakage for existing bonds.
(5) Double edge swap probability: These individual bond probabilities are combined to compute the probability of each possible DES operation.

The probabilities for the DES are computed outside of the model by combining the probabilities of bond formation and bond breakage for each possible DES operation. This architecture efficiently processes molecular graphs with approximately 471,000 parameters for the BASE model. The neural network makes no assumptions about chemical validity—it focuses solely on learning structural patterns from the training data, whereas the valence constraints are handled externally by the diffusion process.

Time model architecture. The time model shares a similar GNN backbone with the diffusion model but serves a different purpose. It estimates the progress of the diffusion process by predicting how far the current molecular graph is from a valid molecule. Its architecture consists of (Extended Data Fig. 5b).

(1) EnhancedGINE layers: Three layers that process node, edge and graph features similarly to the diffusion model, maintaining the same 124-dimensional embeddings.
(2) Graph-level embedding: Node embeddings are aggregated using mean pooling to obtain a fixed-dimensional representation (124-dimensional) of the entire molecular graph.
(3) Time prediction: The graph embedding is processed through a simple feedforward network (64 hidden units) that outputs a scalar value between 0 and 1, representing the normalized diffusion time.

With approximately 63,000 parameters, the time model is significantly smaller than the diffusion model while still providing crucial guidance during the denoising process.

FPS model variant. The FPS variant of CoCoGraph extends the BASE models by incorporating molecular fingerprints that capture substructural information. This architecture, depicted in Extended Data Fig. 6, processes the 2,048-dimensional Morgan fingerprints (ECFP3) by the following methods.

(1) Fingerprint processing: The binary fingerprint vector is passed through a dedicated feedforward neural network (with layers of 1,024, 512 and 256 units) that reduces its dimensionality while preserving substructure information.
(2) Feature integration: The processed fingerprint features (256-dimensional) are concatenated with the graph embeddings before the final prediction layers of the diffusion and time models.
(3) Enhanced prediction: The combined representations enable the models to make predictions informed by both graph structure and specific molecular substructures.

This enhancement increases the parameter count to approximately 3.1 million for the diffusion model and 1.3 million for the time model, but improves performance by incorporating explicit substructural information.

Molecule featurization

Molecule featurization transforms molecular information into structured numerical data that can be processed effectively by our models. By extracting relevant features at the node (atom), edge (bond) and graph (molecule) levels, we allow CoCoGraph models to accurately capture the molecular properties that characterize valid chemical structures. These features provide a comprehensive representation of molecular characteristics at multiple levels of granularity. Although additional topological and structural features could be extracted, our experiments suggest that this feature set strikes a balance between model performance and computational efficiency. In Extended Data Table 3, we summarize the molecular features used by CoCoGraph, which we describe in more detail next.

Node-level features. At the node level, we extract features X that encode both chemical and structural properties of each atom: (1) one-hot encoded representation of the atom's element (15 dimensions covering the most common elements in organic chemistry; namely, boron, nitrogen, carbon, oxygen, fluorine, phosphorus, sulfur, chlorine, bromine, iodine, calcium, potassium, sodium, magnesium and hydrogen); (2) binary indicators showing presence in cycles of sizes 3 to 14, and larger than 14; (3) number of non-hydrogen neighbouring atoms; and (4) number of bridges the atom participates in. These node

features enable the model to understand the chemical environment around each atom when predicting valid DES operations.

Edge-level features. For edges, we extract features E that describe bond properties and their structural role in the molecular graph: (1) one-hot encoding of bond multiplicity (single, double or triple); (2) binary features indicating participation in cycles of sizes 3 to 14 and more than 14; (3) binary indicator of whether the bond is a bridge; (4) number of distinct paths between the two atoms of the bond; and (5) two-dimensional distance between the atoms in the molecular graph. These edge features allow the diffusion model to identify bonds that can be validly swapped while maintaining chemical constraints.

Graph-level features. At the whole-graph level, we capture global structural properties g: (1) number of cycles between sizes 3 to 14 and above 14; (2) a measure of whether the molecular graph is planar; (3) number of connected components; (4) fraction of edges that are bridges and simplified bridges; and (5) proportion of each bond multiplicity in the molecule. In addition, the diffusion time step is stored as a normalized feature between 0 and 1, providing temporal context during the diffusion process.

Molecular fingerprints in FPS models. For our enhanced CoCoGraph FPS model, we incorporate Morgan fingerprints[48] with a dimensionality of 2,048. These fingerprints capture the presence of specific substructures within the molecule, providing a rich representation of molecular motifs that may not be explicitly captured by the other features. The inclusion of these fingerprints allows the FPS CoCoGraph model to identify patterns of substructural arrangements that correlate with valid chemical transformations, enhancing its ability to predict realistic DES operations.

Feature normalization. All extracted features are normalized to ensure balanced contribution to the model's learning process. For categorical features such as element type and bond multiplicity, we use one-hot encoding. Count-based features are normalized to appropriate ranges, whereas binary features remain as 0/1 indicators.

Model training

Our training dataset consisted of 2.25 million molecules derived from established molecular databases as described in 'Molecular processing'. For computational efficiency, we processed the dataset in slices of approximately 100,000 molecules, with 80% allocated for training and 20% for validation. Molecules were filtered based on our established criteria (5–70 atoms, valid chemical elements and a single connected component).

The training process leveraged an implicit form of data augmentation arising from our diffusion methodology. Although the dataset contained a finite number of molecules, our constrained diffusion process generated a different random intermediate graph for each molecule during each training iteration through the application of random DES operations. This approach effectively expanded the training distribution, enhanced generalization capabilities and prevented overfitting to specific diffusion trajectories.

Diffusion model training. We trained the diffusion model using a stepwise approach with batch size 12. For each step in the diffusion trajectory, we computed features as detailed in 'Feature extraction'. We trained the model by processing pairs of consecutive diffusion steps, which yielded better performance compared with batchwise or molecule-wise training.

The training used three weighted binary cross-entropy loss components, as described above. Each loss was weighed and masked and applied to the model normalized by the number of diffusion steps processed in each backward pass to ensure stable training regardless of molecular size or diffusion trajectory length.

Time model training. We trained the time model using a similar data processing approach to the diffusion model, with identical batch sizes and dataset slicing. Unlike the diffusion model, which predicts edge swapping operations, the time model was trained to predict the normalized diffusion time (ranging from 0 to 1) corresponding to each graph state in the diffusion trajectory.

We employed a simple mean squared error loss between the predicted normalized time and the actual time step of the diffusion process. This model provides crucial guidance during the denoising process by informing the diffusion model of the actual progress of denoising, as illustrated in Fig. 1.

Optimization and computational implementation. The BASE variants of the diffusion and time models were trained for three epochs with an initial learning rate of 10^{-4}. The FPS models were initialized with one epoch pretrained BASE weights and fine-tuned for two more epochs using differential learning rates: 10^{-5} for pretrained parameters and 10^{-4} for the new fingerprint-related parameters, which preserved learned knowledge while allowing fingerprint-specific parameters to adapt more rapidly.

To manage the computational load, we implemented several efficiency measures, including distributed dataset processing with checkpoints saved every 1,000 batches, state preservation across slice boundaries and parallel feature computation using a process pool executor with 24 workers. The complete training process for all model variants required approximately 60 days on a single NVIDIA RTX 4090 GPU.

Sampling of new molecules

For the sampling of new molecules, we employ our collaborative constrained diffusion model. During the sampling process, both the diffusion and time models work together to generate the final molecule. Initially, a molecular formula is selected directly from the database. This molecular formula determines the atoms (nodes) and their valences (degree constraints) for the generation process. A random molecular graph G_T that satisfies these constraints is then constructed using the noising process as described above.

During the denoising process, a cycle is repeated where the diffusion and time models are applied iteratively to generate the final molecule. At step t, the molecular graph features are extracted, and the diffusion model is applied to obtain the probabilities of DES operations, bond breakage and bond formation. Based on these probabilities, a random selection is made from the possible DES operations with a probability greater than a threshold (initially set at 95%). If the threshold prevents any operation from being selected, the threshold is reduced by 5%.

After a DES operation is selected and applied, the resulting graph is verified to ensure that it has not been previously encountered in the sampling process and that it remains fully connected—a critical requirement for valid molecules. If these conditions are met, the time model is applied to predict the normalized diffusion time t_{pred} of the current graph state. This cycle is repeated for a predetermined number of diffusion steps, calculated based on the number of bonds in the molecular formula.

Finally, from the trajectory of molecular graphs generated from G_T along with their diffusion time predictions, the output molecule is selected as the graph with the normalized diffusion time t_{pred} closest to 0. This selection mechanism, which prioritizes graphs that the time model identifies as being closest to valid molecules rather than simply taking the final graph in the sequence, is a key advantage of our collaborative approach. The result is a chemically valid molecule with the specified molecular formula, generated through a controlled

and constrained diffusion process, and validated by the collaboration between our diffusion and time models.

Inpainting-based conditional generation

Fragment library generation. To enable inpainting-based conditional generation, we first constructed a library of molecular fragments by systematically breaking bonds in molecules from our training database. For each molecule, we identified all single bonds not participating in ring structures as potential breaking points. Breaking each bond produces two fragments, and we retained those within specified size ranges, namely, 2–5 heavy atoms for small fragments and 6–15 heavy atoms for medium fragments.

For each size category, we calculated the molecular formula of all extracted fragments and ranked them by frequency of occurrence across the training database. We retained the top 500 most common fragments for each size category, ensuring that the fragment library represents chemically relevant and frequently occurring formulas of substructures. This frequency-based selection ensures that inpainting operations use realistic molecular building blocks rather than rare or fragment formulas.

Sampling procedure for inpainting-based conditional generation. Inpainting-based conditional generation combines an existing molecule (which will be preserved) with a new molecular fragment (which will be added and refined). The process consists of four main steps: (1) fragment selection and random graph generation; (2) molecular connection; (3) constrained denoising; and (4) candidate evaluation.

First, a fragment molecular formula is randomly selected from the appropriate fragment library (small or medium). We construct a random molecular graph for this fragment that satisfies all chemical constraints (Supplementary Information Section 1, for details on the random graph construction procedure). This noisy fragment graph serves as the starting point for the inpainting process.

Second, we connect the fragment to the original molecule at a randomly selected attachment point. The attachment point must be a heavy atom in the original molecule that is bonded to at least one hydrogen atom. We break the bond between this heavy atom and one of its hydrogen atoms, and similarly identify a heavy atom in the fragment that is bonded to hydrogen. By removing one hydrogen from both the molecule and the fragment, and creating a new bond between the two heavy atoms, we ensure that valence constraints are preserved throughout the connection process. The result is a single connected molecular graph containing both the original molecule and the noisy fragment.

Third, we apply the constrained denoising process to refine the fragment while keeping the original molecule fixed. To achieve this, we modify the diffusion model's prediction step by masking all DES operations that would involve bonds within the original molecule. Specifically, for any potential DES operation that would remove or create bonds between atoms that were both present in the original molecule, we set the corresponding probability to zero. This masking ensures that only bonds involving at least one atom from the fragment can be modified during denoising, effectively freezing the original molecular structure while allowing the fragment to be refined into a chemically valid and contextually appropriate addition.

Finally, the denoising process proceeds as described in 'Sampling of new molecules', with the diffusion and time models collaborating to generate a trajectory of molecular graphs. The molecule with the smallest predicted time is selected as the final generated structure.

Paracetamol candidates. For the paracetamol candidate evaluation presented in Fig. 5, we implemented a property-based search approach. For a target molecule (in our demonstration, paracetamol), we first identified nine key physicochemical properties: molecular weight, molecular log *P* (lipophilicity), TPSA, number of hydrogen-bond donors, number of hydrogen-bond acceptors, number of aromatic rings, quantitative estimate of drug likeness, synthetic accessibility score and molecular complexity (Bertz index). We calculated these properties for the target molecule, for all 8.2 million molecules in the database using RDKit and for 3,000 generated candidate molecules for each fragment size inpainting category (small and medium additions).

To enable meaningful comparison across properties with different scales, we normalized each property value to the range [0, 1] based on typical ranges for drug-like molecules. We then calculated the Euclidean distance from each molecule to the target molecule in this nine-dimensional normalized property space. Molecules were ranked on three different ranks (database search, small inpainting, medium inpainting) by their Euclidean distance, with smaller distances indicating greater similarity to the target molecule. The top-ranked candidates (presented in Fig. 5) represent molecules from each set that most closely match the target's property profile.

Development of Turing-test web and analysis of results

To evaluate the plausibility of our synthetic molecules, we developed a web-based molecular Turing-like test platform. This platform was designed to present human experts with pairs of molecules and challenge them to distinguish between real molecules from our database and new molecules generated by CoCoGraph. The application was hosted on our laboratory server and made accessible at http://cocograph.seeslab.net.

The molecule pairs were drawn from our database of 8.2 million generated molecules and their corresponding original molecules used as seed structures. For each testing round, the application randomly selected a generated molecule uniformly at random from the database, identified its molecular formula and then randomly selected an original molecule with the same molecular formula from the original dataset. Importantly, the original molecule paired with a generated molecule was not necessarily the seed molecule used to generate it—molecules were matched solely by molecular formula. This pairing strategy ensured that participants could not rely on trivial differences such as molecular size or atom composition, which would have been immediately apparent and bias their choices. Instead, by presenting molecules with identical formulas, the test forced participants to evaluate structural plausibility and physicochemical properties based on their chemical knowledge, providing a more stringent and meaningful assessment of the generated molecules realism.

Regarding potential duplicate pairs: if a generated molecule happened to match an original molecule (which was very rare given the 95.7% novelty rate against PubChem), and by chance that same original molecule was selected as the pair (which would again be rare, even conditioned on the generated molecule not being new), we still presented it as a pair. This was an extremely rare event and was not problematic, as it indicated that the generated molecule successfully replicated a valid existing molecule. Molecules were rendered as two-dimensional structural diagrams using RDKit's drawing utilities to ensure clarity and consistency in presentation.

Test protocol and participant recruitment. Participants were presented with an information page explaining the nature of the experiment without revealing specific details about the generative model or selection criteria that might bias their choices. After consenting to participate and providing their expertise level (high school, undergraduate or postgraduate training in organic chemistry), each participant proceeded through 20 rounds of molecule comparison. In each round the following occurred.

(1) Two molecular structures with identical molecular formulas were displayed side by side.
(2) The order of real and generated molecules was randomized for each pair.

(3) Participants selected which molecule they believed was from an established chemical database.
(4) No time limit was imposed, allowing participants to make thorough evaluations.
(5) No feedback was provided until completion of all 20 rounds to prevent learning effects.

We recruited participants primarily from the University Rovira i Virgili, targeting departments of chemical engineering, mechanical engineering, and the faculties of chemistry and biochemistry. The recruitment process involved email invitations to departmental mailing lists and direct outreach to research groups. The test was accessible for a two-week period, allowing participants to complete it at their convenience.

Data analysis methodology. For each participant, we recorded their level of expertise and their selections for each molecule pair. The primary metric calculated was accuracy—the percentage of molecule pairs for which participants correctly identified the real molecule. We analysed this metric globally and stratified by: (1) level of expertise; (2) molecular size; and (3) chemical properties, including functional groups, aromaticity and other structural features calculated using RDKit. Confidence intervals (95%) for accuracy metrics were calculated using bootstrap with 1,000 resampling iterations to ensure robust statistical inference.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The minimum datasets required to interpret, verify and extend this study are publicly available. The training dataset used in this work is available via Zenodo at https://doi.org/10.5281/zenodo.18939751 (ref. 49). The dataset of 8.2 million molecules generated in this work is available via Zenodo at https://doi.org/10.5281/zenodo.18939448 (ref. 50). No access restrictions apply.

Code availability

Our code for training our constrained collaborative model, sampling new molecules and evaluating the results is available via GitHub at https://github.com/manurubo/CoCoGraph/tree/main. The archived, citable release associated with this work is available via Zenodo at https://doi.org/10.5281/zenodo.18940151 (ref. 51). Model weights are available in the same repository.

References

1. Alakhdar, A., Poczos, B. & Washburn, N. Diffusion models in de novo drug design. *J. Chem. Inf. Model*. https://pubs.acs.org/doi/10.1021/acs.jcim.4c01107 (2024).
2. Menon, D. & Ranganathan, R. A generative approach to materials discovery, design, and optimization. *ACS Omega* **7**, 25958–25973 (2022).
3. Alkhatib, I. I. I., Albà, C. G., Darwish, A. S., Llovell, F. & Vega, L. F. Searching for sustainable refrigerants by bridging molecular modeling with machine learning. *Ind. Eng. Chem. Res.* **61**, 7414–7429 (2022).
4. Young, A., Röst, H. & Wang, B. Tandem mass spectrum prediction for small molecules using graph transformers. *Nat. Mach. Intell.* **6**, 404–416 (2024).
5. Ackloo, S. et al. Cache (critical assessment of computational hit-finding experiments): a public-private partnership benchmarking initiative to enable the development of computational methods for hit-finding. *Nat. Rev. Chem.* **6**, 287–295 (2022).
6. Polishchuk, P. G., Madzhidov, T. I. & Varnek, A. Estimation of the size of drug-like chemical space based on gdb-17 data. *J. Comput. Aided Mol. Des.* **27**, 675–679 (2013).
7. Kutchukian, P. S. & Shakhnovich, E. I. *De novo*design: balancing novelty and confined chemical space. *Expert Opin. Drug Discov.* **5**, 789–812 (2010).
8. Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O. & Walsh, A. Machine learning for molecular and materials science. *Nature* **559**, 547–555 (2018).
9. Blaschke, T. et al. Reinvent 2.0: an AI tool for de novo drug design. *J. Chem. Inf. Model.* **60**, 5918–5922 (2020).
10. Bilodeau, C., Jin, W., Jaakkola, T., Barzilay, R. & Jensen, K. F. Generative models for molecular discovery: recent advances and challenges. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **12**, e1608 (2022).
11. Jin, W., Barzilay, R. & Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. In *Proc. 35th International Conference on Machine Learning* https://proceedings.mlr.press/v80/jin18a (PMLR, 2018).
12. Cao, N. D. & Kipf, T. MolGAN: an implicit generative model for small molecular graphs. Preprint at https://arxiv.org/abs/1805.11973 (2022).
13. Mercado, R. et al. Graph networks for molecular design. *Mach. Learn. Sci. Technol.* **2**, 025023 (2021).
14. Dai, H., Nazi, A., Li, Y., Dai, B. & Schuurmans, D. Scalable deep generative modeling for sparse graphs. *In Proc. 34th International Conference on Machine Learning* 2302–2312 (PLMR, 2020).
15. Liao, R. et al. Efficient graph generation with graph recurrent attention networks. In *Proc. 33rd International Conference on Neural Information Processing Systems* 4255–4265 (Curran Assoc., 2019).
16. You, J., Ying, R., Ren, X., Hamilton, W. L. & Leskovec, J. GraphRNN: generating realistic graphs with deep auto-regressive models. In *Proc. 35th International Conference on Machine Learning* 5708–5717 (PLMR, 2018).
17. Lee, S., Jo, J. & Hwang, S. J. Exploring chemical space with score-based out-of-distribution generation. In *Proc. 40th International Conference of Machince Learning* 18872–18892 (PMLR, 2023).
18. Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N. & Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proc. 32nd International Conference on Machine Learning* 2256–2265 (PMLR, 2015).
19. Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems* (eds Larochelle, H., et al.) **33**, 6840–6851 (NeurIPS, 2020).
20. Song, Y. et al. Score-based generative modeling through stochastic differential equations. *Int. Conf. Learn. Represent.* (2021).
21. Yang, L. et al. Diffusion models: a comprehensive survey of methods and applications. *ACM Comput. Surv*. https://doi.org/10.1145/3626235 (2023).
22. Ramesh, A. et al. Zero shot text-to-image generation. In *Proc. 38th International Conference on Machine Learning* 8821–8831 (PMLR, 2021).
23. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. & Chen, M. Hierarchical text-conditional image generation with clip latents. Preprint at https://arxiv.org/abs/2204.06125 (2022).
24. Austin, J., Johnson, D. D., Ho, J., Tarlow, D. & van den Berg, R. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems* (eds Ranzato, M., et al.) **34**, 17981–17993 (NeurIPS, 2021).
25. Hoogeboom, E., Satorras, V. G., Vignac, C. & Welling, M. Equivariant diffusion for molecule generation in 3D. In *Proc. 39th International Conference on Machine Learning* 8867–8887 (PMLR, 2022).

26. Jo, J., Lee, S. & Hwang, S. J. Score-based generative modeling of graphs via the system of stochastic differential equations. In *Proc. 39th International Conference on Machine Learninger* 10362–10383 (PMLR, 2022).
27. Haefeli, K. K., Martinkus, K., Perraudin, N. & Wattenhofer, R. Diffusion models for graphs benefit from discrete state spaces. Preprint at https://arxiv.org/abs/2210.01549 (2023).
28. Qin, Y., Vignac, C. & Frossard, P. Sparse training of discrete diffusion models for graph generation. *Trans. Mach. Learn. Res.* (2023).
29. Vignac, C. et al. DiGress: discrete denoising diffusion for graph generation. *Int. Conf. Learn. Represent.* (2023).
30. Huang, H., Sun, L., Du, B. & Lv, W. Conditional diffusion based on discrete graph structures for molecular graph generation. In *Proc. AAAI Conference on Artificial Intelligence* **37**, 4302–4311 (AAAI Press, 2023).
31. Yang, L. et al. Graphusion: latent diffusion for graph generation. *IEEE Trans. Knowl. Data Eng.* **36**, 6358–6369 (2024).
32. Du, Y., Fu, T., Sun, J. & Liu, S. MolGenSurvey: a systematic survey in machine learning models for molecule design. Preprint at https://arxiv.org/abs/2203.14500 (2022).
33. Jo, J., Kim, D. & Hwang, S. J. Graph generation with diffusion mixture. In *Proc. 41th International Conference on Machine Learning* 22371–22405 (PMLR, 2024).
34. Qin, Y., Madeira, M., Thanou, D. & Frossard, P. DeFoG: discrete flow matching for graph generation. In *Proc. 42nd International Conference on Machine Learning* 50269–50326 (PMLR, 2025).
35. Maslov, S. & Sneppen, K. Specificity and stability in topology of protein networks. *Science* **296**, 910–913 (2002).
36. Milo, R., Kashtan, N., Itzkovitz, S., Newman, M. E. J. & Alon, U. On the uniform generation of random graphs with prescribed degree sequences. Preprint at https://arxiv.org/abs/cond-mat/0312028 (2004).
37. Kharel, S. R., Mezei, T. R., Chung, S., Erdös, P. L. & Toroczkai, Z. Degree-preserving network growth. *Nat. Phys.* **18**, 100–106 (2022).
38. Brown, N., Fiscato, M., Segler, M. H. & Vaucher, A. C. GuacaMol: benchmarking models for de novo molecular design. *J. Chem. Inf. Model.* **59**, 1096–1108 (2019).
39. Madeira, M., Vignac, C., Thanou, D. & Frossard, P. Generative modelling of structurally constrained graphs. In *Advances in Neural Information Processing Systems* (eds Globerson, A., et al.) **37**, 137218–137262 (NeurIPS, 2024).
40. Fishman, N., Klarner, L., De Bortoli, V., Mathieu, E. & Hutchinson, M. Diffusion models for constrained domains. *Trans. Mach. Learn. Res.* (2023).
41. Fishman, N., Klarner, L., Mathieu, E., Hutchinson, M. & De Bortoli, V. Metropolis sampling for constrained diffusion models. In *Advances in Neural Information Processing Systems* (eds Oh, A., et al.) **36**, 62296–62331 (NeurIPS, 2023).
42. Molloy, M. & Reed, B. A critical point for random graphs with a given degree sequence. *Random Struct. Alg.* **6**, 161–180 (1995).
43. Landrum, G. et al. RDKit: open-source cheminformatics. *Zenodo* https://www.rdkit.org (2025).
44. Amaral, L. A. N. Artificial intelligence needs a scientific method-driven reset. *Nat. Phys.* **20**, 523–524 (2024).
45. Ren, P. et al. A survey of deep active learning. *ACM Comput. Surv.* **54**, 180 (2021).
46. Bohde, M., Manjrekar, M., Wang, R., Ji, S. & Coley, C. W. DiffMS: diffusion generation of molecules conditioned on mass spectra. In *Proc. 42nd International Conference on Machine Learning* 4737–4756 (PMLR, 2025).
47. Hu, W. et al. Strategies for pre-training graph neural networks. *Int. Conf. Learn. Represent.* (2020).
48. Morgan, H. L. The generation of a unique machine description for chemical structures—a technique developed at Chemical Abstracts Service. *J. Chem. Doc.* **5**, 107–113 (1965).
49. Ruiz-Botella, M. Training dataset for 'A collaborative constrained graph diffusion model for the generation of realistic synthetic molecules. *Zenodo* (https://arxiv.org/abs/2505.16365) https://doi.org/10.5281/zenodo.18939751 (2026).
50. Ruiz-Botella, M. Generated 8.2M molecules from 'A collaborative constrained graph diffusion model for the generation of realistic synthetic molecules'. *Zenodo* (https://arxiv.org/abs/2505.16365) https://doi.org/10.5281/zenodo.18939448 (2026).
51. Ruiz-Botella, M. CoCoGraph code release for 'A collaborative constrained graph diffusion model for the generation of realistic synthetic molecules'. *Zenodo* https://doi.org/10.5281/zenodo.18940151 (2026).

Acknowledgements

This research was supported by project PID2022-142600NB-I00 (M.S.-P. and R.G.) from MCIN/ AEI/ 10.13039/ 501100011033, by the Government of Catalonia (2021SGR-633) (M.S.-P. and R.G.) and by the EU Next Generation, URV and SEPE (2022PMF-INV-10) (M.R.-B.). We thankfully acknowledge the computer resources at MareNostrum5 and the technical support provided by BSC (RES-IM-2025-1-0021).

Author contributions

M.R.-B., M.S.-P. and R.G. designed the research. M.R.-B. developed the methodology, implemented the code for processing the data, training the models, sampling molecules and developing the web application for the Turing-like test. M.R.-B. performed the analyses and prepared the figures and tables with input from M.S.-P. and R.G. M.S.-P. and R.G. supervised the research. All authors interpreted the results, contributed to writing and revising the paper and approved the final version.

Competing interests

The authors declare no competing interests.

Additional information

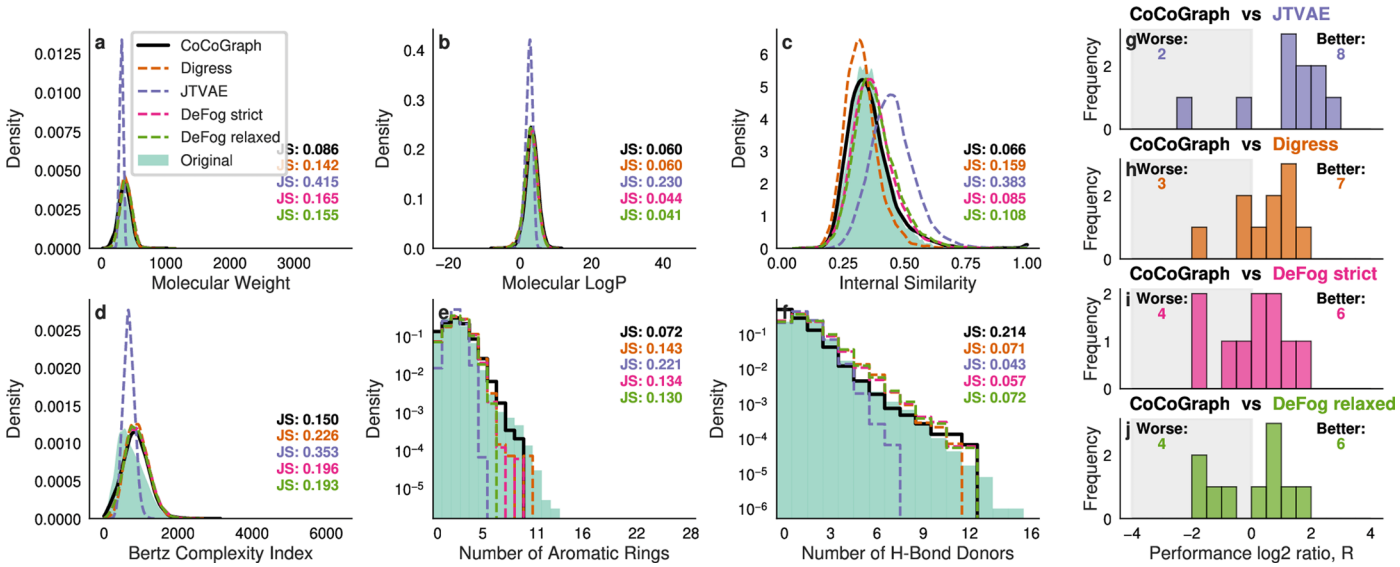
Extended data is available for this paper at https://doi.org/10.1038/s42256-026-01229-5.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-026-01229-5.

Correspondence and requests for materials should be addressed to Marta Sales-Pardo or Roger Guimerà.

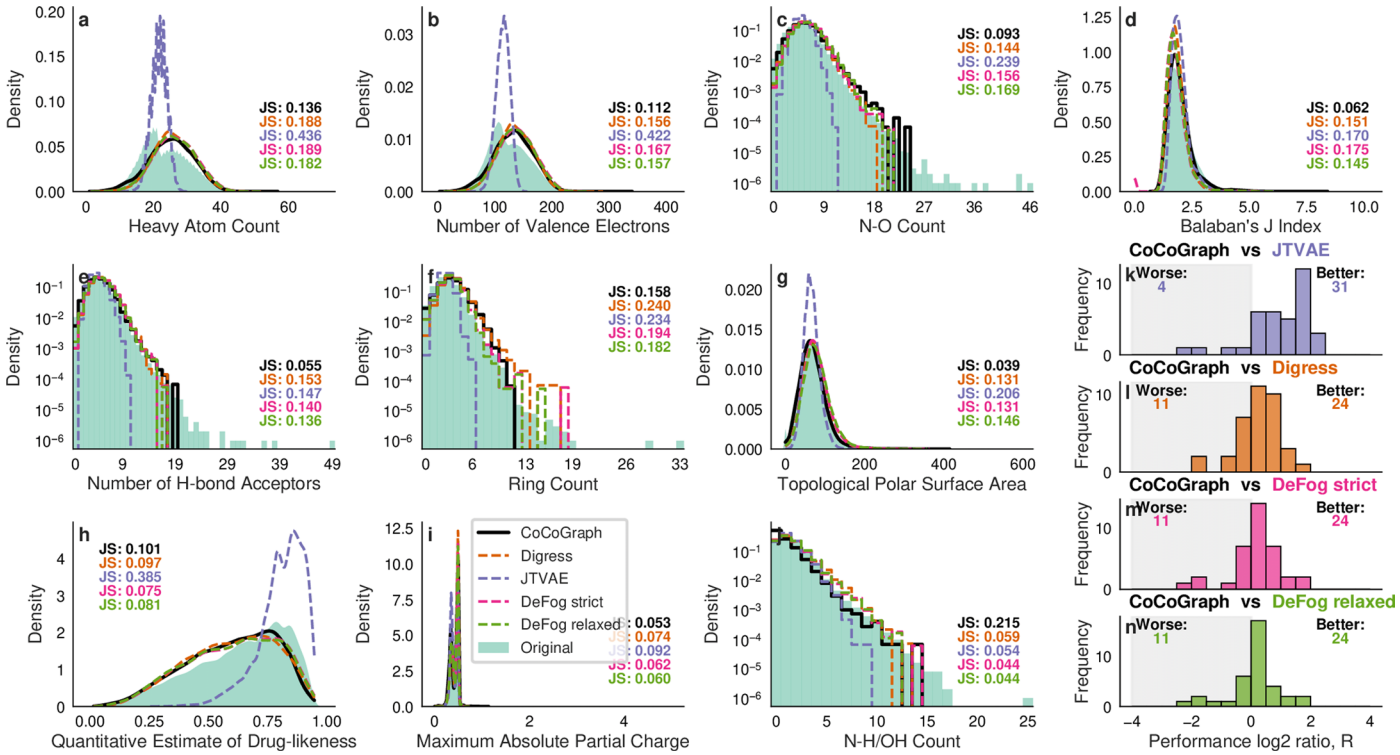
Peer review information *Nature Machine Intelligence* thanks the anonymous reviewers for their contribution to the peer review of this work. Peer reviewer reports are available.

https://doi.org/10.1038/s42256-026-01229-5



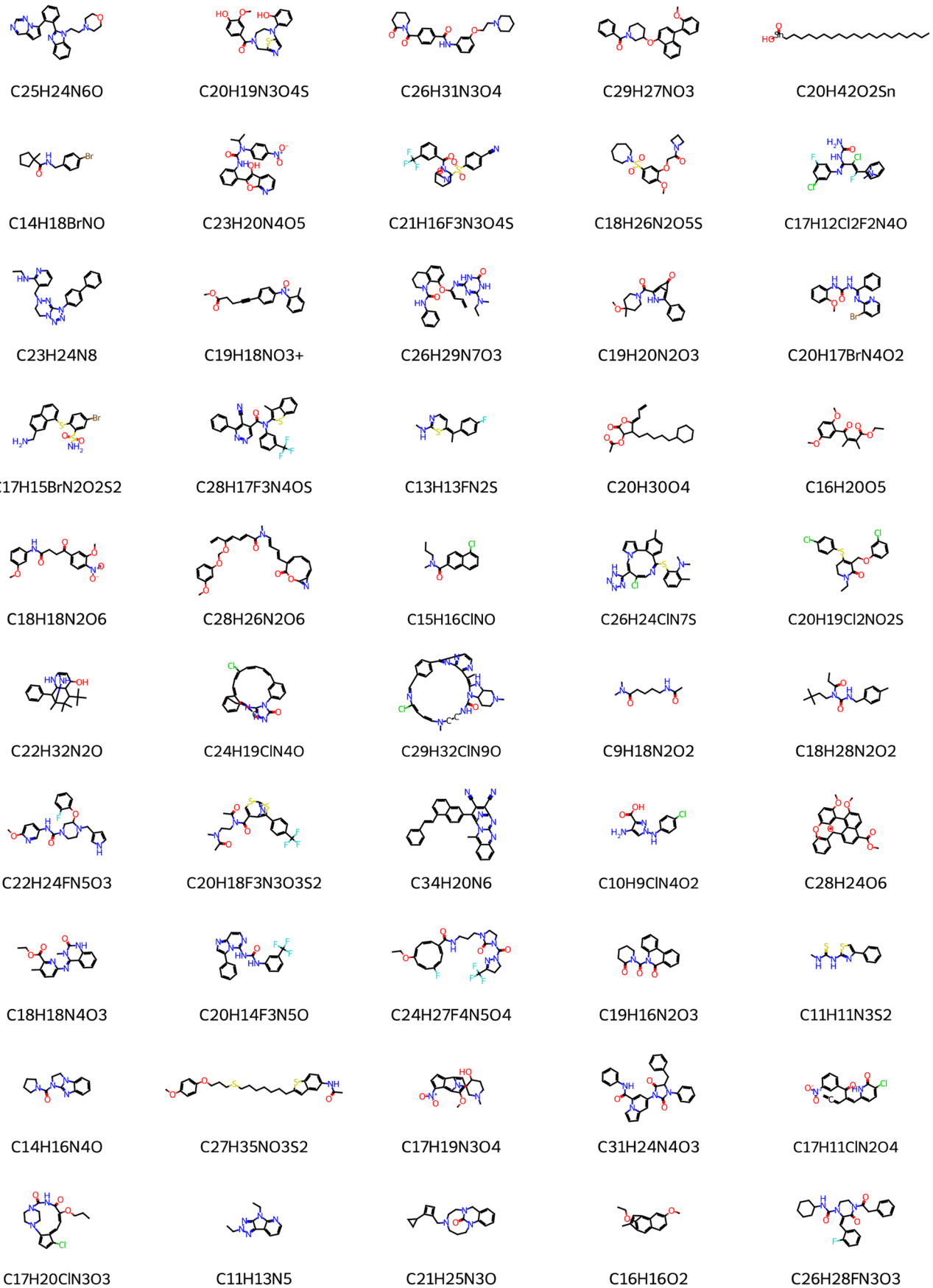
Extended Data Fig. 1 | Performance comparison of CoCoGraph BASE on GuacaMol benchmark properties. a, molecular weight; **b**, molecular logP; **c**, internal similarity; **d**, Bertz complexity index; **e**, number of aromatic rings; and **f**, number of H-bond donors. For each property, the distribution of values calculated for molecules generated by CoCoGraph (black line) is compared to that of the original molecules (green distribution), and to those of molecules generated by JTVAE (purple dashed line), DiGress (orange dashed line), DeFoG strict (pink dashed line), and DeFoG relaxed (light green dashed line). Jensen-Shannon (JS) distance values between each model and the original distribution are shown. **g-j**, Comparison based on the log2 ratio of JS distances between CoCoGraph BASE and comparator models for the properties in (**a-f**). Positive values indicate CoCoGraph BASE outperforming the comparator model, whereas negative values indicate poorer performance.

https://doi.org/10.1038/s42256-026-01229-5



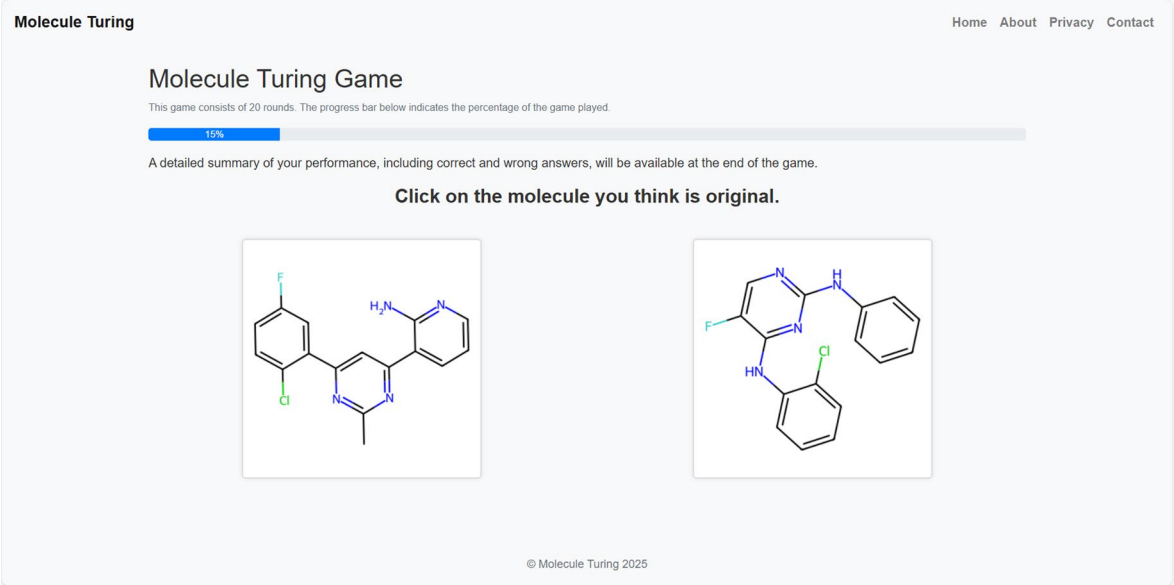
Extended Data Fig. 2 | Detailed performance comparison of CoCoGraph BASE on a subset of 36 chemical properties. a-j, Distributions of ten molecular properties: **a**, heavy atom count; **b**, number of valence electrons; **c**, NOCount; **d**, Balaban's J Index; **e**, number of H acceptors; **f**, ring count; **g**, topological polar surface area (TPSA); **h**, quantitative estimate of drug-likeness; **i**, maximum absolute partial charge; and **j**, NHOHCount. For each property, the distributions for molecules generated by the CoCoGraph BASE model (black line) is compared to that of the original molecules (green distribution), and to those of molecules generated by JTVAE (purple dashed line), DiGress (orange dashed line), DeFoG strict (pink dashed line), and DeFoG relaxed (light green dashed line). Jensen-Shannon (JS) distance values between each model and the original distribution are shown. **k-n**, Comparison based on the log2 ratio of JS distances between CoCoGraph BASE and comparator models for the properties in (**a-j**). Positive values indicate CoCoGraph BASE outperforming the comparator model, whereas negative values indicate poorer performance.

https://doi.org/10.1038/s42256-026-01229-5



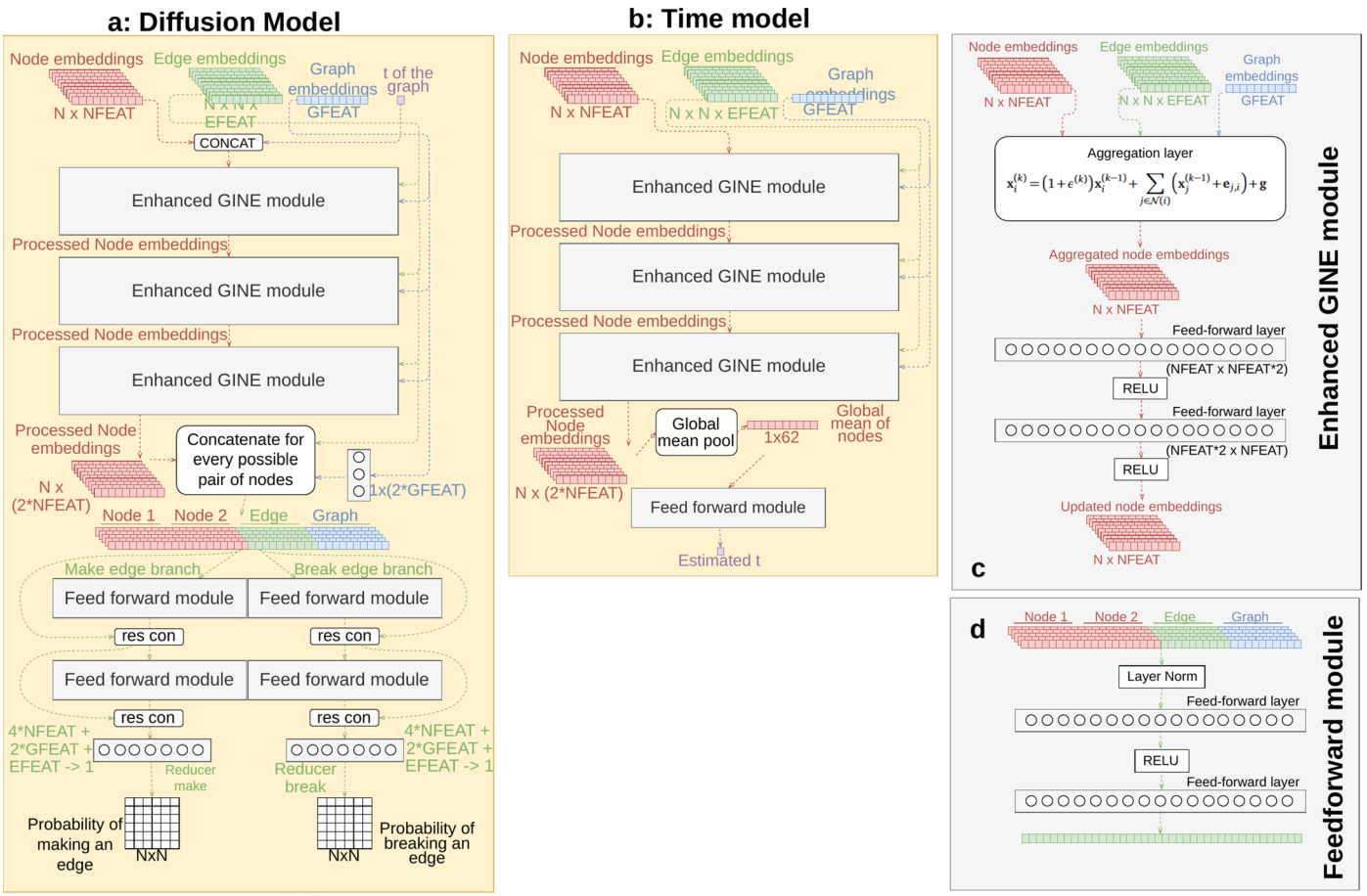
Extended Data Fig. 3 | 50 random molecules generated by CoCoGraph FPS. Molecules are sampled uniformly at random from our generated database.

https://doi.org/10.1038/s42256-026-01229-5



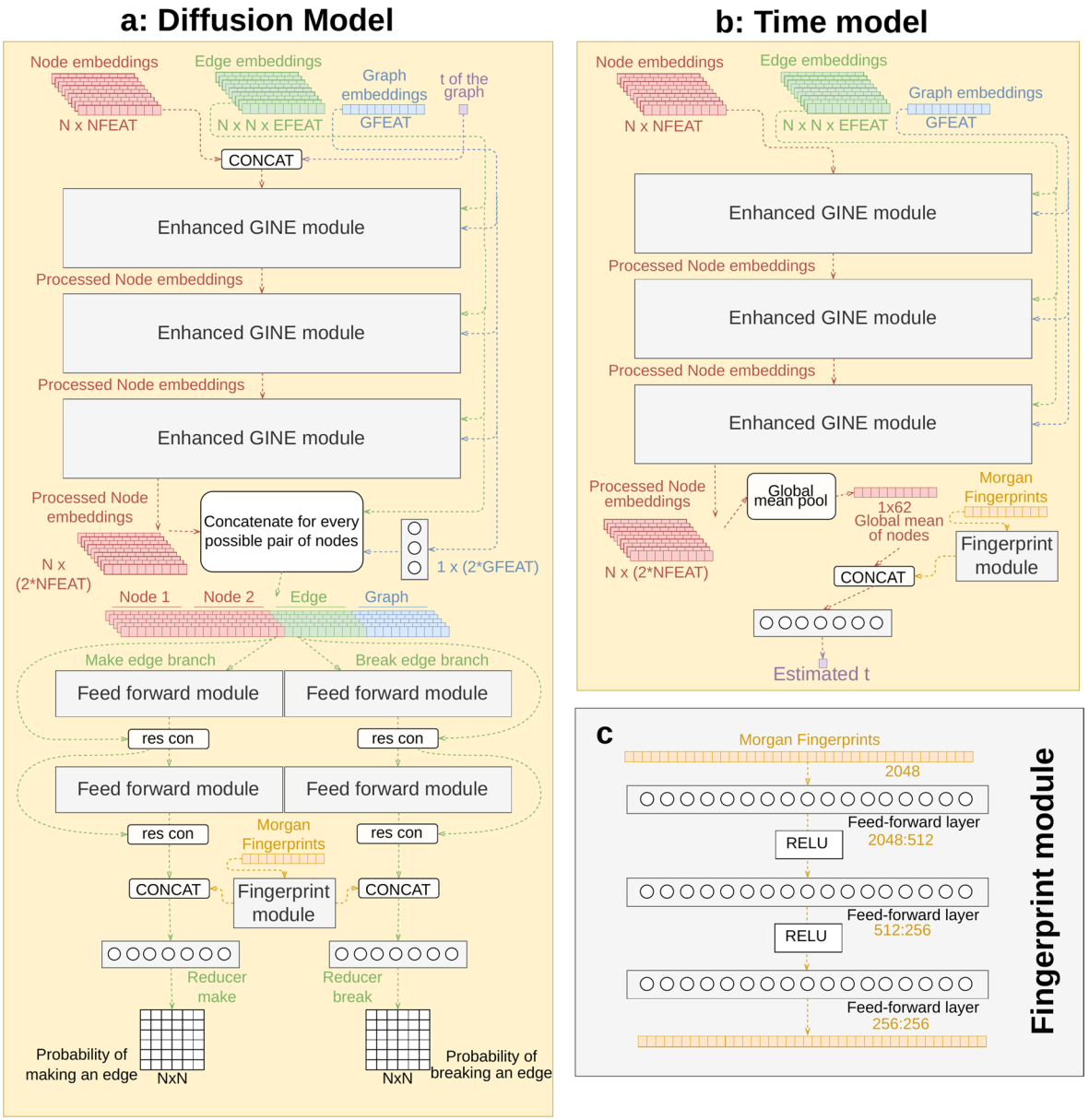
Extended Data Fig. 4 | Web user interface for the Turing-like test experiment. At each round, the web page presents two molecules with same molecular formula, one generated and one original, and the user has to guess which one is the original one. The user interface also shows a progress bar and additional information.

https://doi.org/10.1038/s42256-026-01229-5



Extended Data Fig. 5 | Architecture of CoCoGraph components. a, The diffusion model processes the molecular graph through a sequence of EnhancedGINE layers, the embedding of pairs of nodes are concatenated with edge properties and processed through two feedforward modules to predict the probability of bond formation and bond breakage for each possible DES operation. **b**, The time model estimates the diffusion timestep *t* of the current molecular graph using processed node embeddings obtained after applying the EnhancedGINE module to the features of the molecular graph. **c**, The message passing component of both models, the EnhancedGINE module. **d**, The prediction component of the diffusion model.

https://doi.org/10.1038/s42256-026-01229-5



Extended Data Fig. 6 | Architecture of CoCoGraph FPS (fingerprint-enhanced) components. a, The diffusion model processes the molecular graph through a sequence of EnhancedGINE layers. Then, the embedding of pairs of nodes are concatenated with edge and graph properties and processed through two feedforward modules, after which the fingerprint processed by the fingerprint module is concatenated. The resulting vector is processed through a feedforward network to predict the probability of bond formation and bond breakage for each possible DES operation **b**, The time model estimates the diffusion timestep t of the current molecular graph using processed node embeddings concatenated with the fingerprint processed by the fingerprint module. **c**, The fingerprint module of both models.

https://doi.org/10.1038/s42256-026-01229-5

Extended Data Table 1 | The 36 chemical properties used to evaluate molecules generated by each model

Physicochemical properties		
Basic Physicochemical Properties		
Molecular Weight	Exact Molecular Weight	Heavy Atom Count
Number of Valence Electrons	N-H/OH Count	N-O Count
Fraction Csp3	Quantitative Estimate of Drug-likeness	Balaban's J Index
Lipinski's Rule of Five Descriptors		
Number of H-bond Donors	Number of H-bond Acceptors	Molecular LogP
Number of Rotatable Bonds	Topological Polar Surface Area	
Ring and Aromaticity Descriptors		
Number of Aromatic Rings	Number of Aliphatic Rings	Ring Count
Number of Saturated Rings	Bertz Complexity	
Electronic Descriptors		
Molar Refractivity	Maximum Partial Charge	Minimum Partial Charge
Maximum Absolute Partial Charge	Minimum Absolute Partial Charge	IPC
EState VSA Descriptor 1		
Topological Descriptors (Chi Descriptors)		
Chi0 Index	Chi1 Index	Chi2n Index
Chi3n Index	Chi0n Index	
Van der Waals Surface Area (VSA) Descriptors		
SlogP VSA Descriptor 1	SlogP VSA Descriptor 2	SlogP VSA Descriptor 3
SlogP VSA Descriptor 4	SlogP VSA Descriptor 5	

The properties were selected to identify diverse molecular properties subdivided in 5 groups that represent interesting molecular characteristics desired for molecule generation.

https://doi.org/10.1038/s42256-026-01229-5

Extended Data Table 2 | Jensen-Shannon distances between distributions of 1M molecules from PubChem and generated molecules for each model across all 36 chemical properties

Property	CoCoGraph	DiGress	JTVAE	DeFoG Strict	DeFoG Relaxed
Basic Physicochemical Properties					
Molecular Weight	**0.0857**	0.1421	0.4146	0.1647	0.1554
Exact Molecular Weight	**0.0866**	0.1432	0.4137	0.1656	0.1563
Heavy Atom Count	**0.1359**	0.1884	0.4359	0.1889	0.1820
Number of Valence Electrons	**0.1116**	0.1564	0.4219	0.1667	0.1569
N-H/OH Count	0.1898	0.0594	0.0543	**0.0437**	0.0440
N-O Count	**0.0927**	0.1437	0.2391	0.1561	0.1685
Fraction Csp3	**0.1072**	0.1373	0.1649	0.1257	0.1374
Quantitative Estimate of Drug-likeness	0.1012	0.0969	0.3854	**0.0747**	0.0805
Balaban's J Index	**0.0633**	0.1505	0.1702	0.1752	0.1452
Lipinski's Rule of Five Descriptors					
Number of H-bond Donors	0.1870	0.0710	**0.0428**	0.0570	0.0718
Number of H-bond Acceptors	**0.0512**	0.1528	0.1474	0.1403	0.1364
Molecular LogP	0.0581	0.0601	0.2299	0.0442	**0.0407**
Number of Rotatable Bonds	0.0852	0.0760	0.2505	0.0246	**0.0242**
Topological Polar Surface Area	**0.0404**	0.1306	0.2058	0.1307	0.1458
Ring and Aromaticity Descriptors					
Number of Aromatic Rings	**0.0720**	0.1433	0.2206	0.1338	0.1301
Number of Aliphatic Rings	0.1329	0.1380	0.1161	0.0834	**0.0680**
Ring Count	**0.1408**	0.2402	0.2337	0.1944	0.1819
Number of Saturated Rings	**0.0431**	0.0908	0.0879	0.0755	0.0629
BertzCT	**0.1443**	0.2257	0.3530	0.1961	0.1930
Electronic Descriptors					
Molar Refractivity	**0.1121**	0.1493	0.3996	0.1607	0.1461
Maximum Partial Charge	0.1419	**0.1079**	0.1921	0.1332	0.1380
Minimum Partial Charge	0.0557	0.0731	0.1016	0.0530	**0.0499**
Maximum Absolute Partial Charge	**0.0395**	0.0740	0.0923	0.0619	0.0602
Minimum Absolute Partial Charge	0.1429	**0.1056**	0.2021	0.1321	0.1377
IPC	**0.0006**	**0.0006**	**0.0006**	**0.0006**	**0.0006**
EState VSA Descriptor 1	**0.0289**	0.0641	0.1198	0.1001	0.1037
Topological Descriptors (Chi Descriptors)					
Chi0 Index	**0.1207**	0.1659	0.4350	0.1718	0.1652
Chi1 Index	**0.1349**	0.1858	0.4297	0.1863	0.1783
Chi2n Index	**0.0780**	0.1358	0.3238	0.1172	0.1019
Chi3n Index	**0.0849**	0.1509	0.2786	0.1215	0.1055
Chi0n Index	**0.1047**	0.1339	0.3927	0.1397	0.1273
VSA (Van der Waals Surface Area) Descriptors					
SlogP VSA Descriptor 1	0.0555	**0.0527**	0.0780	0.0700	0.0625
SlogP VSA Descriptor 2	0.1014	**0.0965**	0.1446	0.1294	0.1275
SlogP VSA Descriptor 3	**0.0456**	0.0476	0.0524	0.0608	0.0525
SlogP VSA Descriptor 4	0.0912	**0.0507**	0.0551	0.0673	0.0649
SlogP VSA Descriptor 5	0.1057	**0.0828**	0.1059	0.0965	0.1105

Lower values indicate better distribution matching between generated and PubChem molecules. Bold formatting indicates the best (lowest) JS distance for each property.

https://doi.org/10.1038/s42256-026-01229-5

Extended Data Table 3 | Features used for molecular representation in CoCoGraph models

Feature	Type	Dimensions	Description
Node (atom) features			
Element type	Node	15	One-hot encoding of atom elements (C, H, N, O, F, etc.)
Cycle participation	Node	13	Binary indicators for presence in cycles of size 3-14 and 15+
Heavy atom neighbors	Node	1	Number of non-hydrogen connected atoms
Hydrogen neighbors	Node	1	Number of hydrogen atoms connected
Bridge count	Node	1	Number of bridges the atom participates in
Edge (bond) features			
Bond multiplicity	Edge	4	One-hot encoding of bond type (single, double, triple, aromatic)
Cycle participation	Edge	13	Binary indicators for presence in cycles of size 3-14 and 15+
Bridge status	Edge	1	Whether the bond is a bridge (1) or not (0)
Path count	Edge	1	Number of distinct paths between bonded atoms
2D distance	Edge	1	Spatial distance between atoms in 2D coordinates
Graph (molecule) features			
Cycle counts	Graph	13	Number of cycles of size 3-14 and 15+
Planarity	Graph	1	Measure of molecular graph planarity
Connected components	Graph	1	Number of distinct connected subgraphs
Bridge fraction	Graph	1	Fraction of edges that are bridges
Simplified bridge fraction	Graph	1	Fraction of edges that are simplified bridges
Bond Type Distribution	Graph	4	Proportion of each bond type in the molecule
Diffusion process features			
Normalized diffusion time	Process	1	Normalized time step in the diffusion process (0-1)
FPS model additional features			
Morgan fingerprints (ECFP3)	Fingerprint	2048	Binary representation of molecular substructures

Features are divided in 5 groups: node (atom) features, edge (bond) features, graph (molecule) features, diffusion process features (used only in the diffusion model), and FPS features (used only in CoCoGraph FPS models).

nature portfolio

Corresponding author(s): Roger Guimerà

Last updated by author(s): Mar 12, 2026

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	We use a curated dataset of 2.25M molecules from diverse establised molecular databases including Pubchem, ChEMBL, ZINC, NIST, MSDIAL, GNPS and Agilent METLIN. Our curated database is available at https://zenodo.org/records/18939448. We also use PubChem database as of 31/10/2025 for validation.
Data analysis	Data was analyzed, processed and passed to the model through python, with main modules used being PyTorch, RDKit, NumPy, Pandas, Scikit-lean and matplotlib. Code of the whole project is avaiable at https://github.com/manurubo/CoCoGraph.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

The minimum datasets required to interpret, verify, and extend this study are publicly available in Zenodo. The training dataset used in this work is available at

https://doi.org/10.5281/zenodo.18939751. The dataset of 8.2 million molecules generated in this work is available at https://doi.org/10.5281/zenodo.18939448. No access restrictions apply.

Research involving human participants, their data, or biological material

Policy information about studies with human participants or human data. See also policy information about sex, gender (identity/presentation), and sexual orientation and race, ethnicity and racism.

Reporting on sex and gender	Sex and gender were not consided in the study design. This information was not collected, and any findings apply to sex or gender differentation.
Reporting on race, ethnicity, or other socially relevant groupings	The only variable that we collected from the human participants was Organic chemistry expertise level. We use this variable to stratify the results of the Molecule Turing test-like game accuracy in order to identify if higher expertise is related to better accuracy. We did not use this variable as a proxy for any other variable. We understand Organic Chemistry expertise level as the academia level a human participant has reached, the participants self reported their expertise by selecting on a dropdown web item with their expertise. The options were "Highscool", "Undergraduate", and "Postgraduate (Master's or PhD)". We did not control for confounding variables. For our results, we stratified only between "Undergraduate" (including "Highschool" too), and "Postgraduate".
Population characteristics	Participants were workers, researchers and students from the Chemistry and Mechanical departments of the university Rovira i Virgili of Tarragona, Spain. We did not obtain any more population characteristics for them except for the Organic Chemistry expertise level, the population was divided between 57 "Undergraduate" and 64 "Postgraduate".
Recruitment	Participants were recruited via an email from the heads of department of the university. Participation was voluntary and subject to acceptance of web terms.
Ethics oversight	N/A

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	No calculation of sample size was performed. Our sample size was chosen as the result of curating different databases with diverse molecules. We consider this sample size enough as millions of diverse molecules from different databases is a big chemical space to explore and to learn from to generate novel valid molecules.
Data exclusions	Molecules excluded from the analysis were duplicates or molecules not properly represented. We decided to work with a reduced dataset so our model can learn with a stable set of molecules instead of having a very unstable big set of molecules. Molecules over 70 atoms were excluded from the study for computational resources and focus on drug like molecules. For evaluation, molecules present on any of the training datasets were excluded from Pubchem reference database.
Replication	All atempts at replications were succesfull. To verify the results we have clean installed the repository of github, followed the installation guide and training directions. After the models were trained, we generated new molecules and compared the results with our actual results.
Randomization	As we work on the objective of generating new molecules, there is not a real need to divide the molecules into sets. Nonetheless, we divided the dataset randomly by smiles, so there are not duplicate molecules between train and test sets. For evaluation against PubChem, we randomize totally the 5 sets of 1 Million molecules to compare.
Blinding	Investigators were blinded because the randomization is performed by the python code with a random seed,

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern
☒	☐ Plants

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Plants

Seed stocks

Report on the source of all seed stocks or other plant material used. If applicable, state the seed stock centre and catalogue number. If plant specimens were collected from the field, describe the collection location, date and sampling procedures.

Novel plant genotypes

Describe the methods by which all novel plant genotypes were produced. This includes those generated by transgenic approaches, gene editing, chemical/radiation-based mutagenesis and hybridization. For transgenic lines, describe the transformation method, the number of independent lines analyzed and the generation upon which experiments were performed. For gene-edited lines, describe the editor used, the endogenous sequence targeted for editing, the targeting guide RNA sequence (if applicable) and how the editor was applied.

Authentication

Describe any authentication procedures for each seed stock used or novel genotype generated. Describe any experiments used to assess the effect of a mutation and, where applicable, how potential secondary effects (e.g. second site T-DNA insertions, mosiacism, off-target gene editing) were examined.