

Using GNN property predictors as molecule generators

Corresponding Author: Professor Oleksandr Voznyy

This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.

Version 0:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

I co-reviewed this manuscript with one of the reviewers who provided the listed reports. This is part of the Nature Communications initiative to facilitate training in peer review and to provide appropriate recognition for Early Career Researchers who co-review manuscripts.

(Remarks on code availability)

Reviewer #2

(Remarks to the Author)

In this work, the authors propose DIDgen, a framework to generate molecules with targeted properties. The framework relies on a trained proxy model and applies gradient ascent to optimize the input molecular representations (ie, adjacency matrix). Though the work brings up an interesting way to generate molecules, I have following major concerns.

1. As mentioned by the authors, the performance of the model heavily relies on the performance of the proxy models. Figure 2 shows that the predicted properties are actually significantly off when compared with oracle methods (eg, DFT). It's known that neural networks struggle to predict out of distribution data (like initial random graph in generation progress). Such flawed proxy models can generate unreliable results.

2. In DIDgen, feature vector is determined directly by adjacency matrix, which works on dataset contains limited element types like QM9. However, when applied to more general and broader problems, such method fails to decide the correct atom.

3. The framework can be applied to various GNN proxy models. However only GCN and CrippenNet are applied for energy gap and logP respectively. I wonder whether the authors explored different GNN models.

4. Since the framework requires an explicit adjacency matrix to represent molecules, it restricts the application to many powerful architectures, especially those with equivariant message-passing.

5. It is mentioned that in some cases no conformation could be obtained from the generated molecules. And these molecules are discarded. What's the ratio of such failure cases?

(Remarks on code availability)

N/A

Reviewer #3

(Remarks to the Author)

This research explores the use of a graph neural network as a tool for generative molecular discovery with desired electronic properties. Overall, the paper is well-written. However, some contractions (e.g., "don't") should be corrected to their full forms (e.g., "do not") for a more formal tone. More discussion and clarification are needed to make it impactful. I believe it should be published after addressing the following concerns:

- This method allows for greater flexibility and control over the diversity of the generated molecules, as you can directly adjust the characteristics of the input to explore different regions of chemical space. However, it is unclear whether diversity is a fair evaluation metric for benchmarking. The author could incorporate a correlation metric such as Pearson or R2.
- The author highlighted the current limitations due to the computational cost associated with this method. It would have been beneficial to see a comparison of the computation times between the proposed method and other methods, as this was not compared or discussed in the paper.
- The authors need to clarify the error (RMSE or MAE) associated with the pretrained models in the property prediction. Ultimately, the final molecule prediction will have two sources of error: one from the initial models trained and one from the proposed approach optimization. The authors could explain how they address this degree of uncertainty.
- I was wondering about the property distribution of molecules in the datasets used for training, validation, and testing. This should be clarified.
- It is not clear what motivated the use of the average composition in QM9 as an additional term to guarantee real molecules. How does this term address "realistic" problems?

(Remarks on code availability)

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)

I co-reviewed this manuscript with one of the reviewers who provided the listed reports. This is part of the Nature Communications initiative to facilitate training in peer review and to provide appropriate recognition for Early Career Researchers who co-review manuscripts.

(Remarks on code availability)

Reviewer #2

(Remarks to the Author)

I thank the authors for addressing my concerns with extra experiments and revised manuscript. I have no further questions.

(Remarks on code availability)

Reviewer #3

(Remarks to the Author)

We thank the authors for their revisions and think these are sufficient.

(Remarks on code availability)

Open Access This Peer Review File is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

In cases where reviewers are anonymous, credit should be given to 'Anonymous Referee' and the source.

The images or other third party material in this Peer Review File are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/>

Answers to Reviewer #2

In this work, the authors propose DIDgen, a framework to generate molecules with targeted properties. The framework relies on a trained proxy model and applies gradient ascent to optimize the input molecular representations (ie, adjacency matrix). Though the work brings up an interesting way to generate molecules, I have following major concerns.

Comment R2.1 As mentioned by the authors, the performance of the model heavily relies on the performance of the proxy models. Figure 2 shows that the predicted properties are actually significantly off when compared with oracle methods (eg, DFT). It’s known that neural networks struggle to predict out of distribution data (like initial random graph in generation progress). Such flawed proxy models can generate unreliable results.

Answer to R2.1 We fully agree with this: it is indeed a limitation of any generative model. The only way to not rely on a proxy is to condition the generation directly using the oracle (“the ground truth”). This is possible with certain ML models and with genetic algorithms. However:

1. In scientific discovery applications, there are few examples of models that are so reliable that they can be considered as the ground truth and simultaneously so fast that they can be used to condition a generative AI scheme. In the case of the energy gap task for example, using DFT directly leads to worse results with significantly more computation because even with a generous time limit the algorithm can only go through a limited number of iterations (see Table 1.).
2. As long as a generative model can hit a target with a non-zero rate, it is always possible to generate more molecules to obtain a specific number of target molecules. For example, SGDS has an 86% success rate at generating molecules with logP between -2.5 et and -2. According to their paper they generated 10,000 molecules which means 8,600 of them were in the target range. DiDgen has a 43.5% success rate on the same task which means we could also obtain about 8600 target molecules by generating 20,000 molecules. Since validating a molecule is fast, it is easy to generate molecules until the desired number of “good” molecules is reached. In other words, if the oracle is cheap, methods who do not use it directly can also take advantage of its speed.

The following passage mentions these points in the manuscript with some clarifications stemming from this discussion highlighted in blue:

Generative methods like ours that use a learned proxy rely heavily on its performance, especially its ability to generalize in order to hit target properties. Models that can directly use an oracle do not always suffer from this limitation. For example, in the logP targeting task, the oracle was not computationally limiting and thus the number of evaluations was not taken into account. In that case, oracle-based methods had superior success rates. But since **oracle** evaluations are not costly, as pointed out by other groups of authors [27], it would be easy to generate more molecules and filter out molecules that **do not** meet the criteria.

For instance, when targeting logP between -2.5 and -2, since the rate of success of DID-

gen is about half of that of SGDS, we could simply generate twice as many molecules to obtain the same number of in-target molecules. Since the oracle is inexpensive, this would be easy to verify. In fact, the success rate metric makes little sense outside the context of budgeted (expensive) evaluations: in principle one can always make an algorithm that “generates” a molecule only if it reaches the target. For example, a genetic algorithm that would iterate until the entire population is inside the target range would achieve exactly that. The success rate becomes useful when one has a budget of x oracle evaluations and wants to obtain the highest number of true positives within that budget.

~~Furthermore~~~~In reality~~, application-relevant properties are often computationally expensive and directly optimizing them is not an option, this is actually the case with the water-octanol partition (logP) which is difficult to evaluate precisely beyond Wildman and Crippen’s model [49]. The energy gap task is another example of that. In that case, we have seen that JANUS was able to find significantly more molecules in the target range when using a proxy than when using DFT directly ~~because it could not go through enough generations within a reasonable time limit~~. So, even though JANUS does not *require* the use of a proxy, it clearly benefits from its use. This means that ~~forin~~ a lot of applications, methods that can use an oracle will also end up being dependant on the performance and generalizability of an ML proxy.

–Revised manuscript, p.4

Comment R2.2 In DIDgen, feature vector is determined directly by adjacency matrix, which works on dataset contains limited element types like QM9. However, when applied to more general and broader problems, such method fails to decide the correct atom.

Answer to R2.2 DIDgen *can*, in fact, work with an arbitrary number of element types with the same valence. In the logP task we used C, O, N, F, H, S, Cl, Br, I, and P which are all the elements present in the ZINC dataset. We agree that this was not made clear enough in the manuscript. Passages in blue were added to the manuscript to address this point.

... In terms of matrices, this means that the sum of a row (or columns) of the adjacency matrix defines the element associated with that row (or column).

To deal with an arbitrary number of elements that may have the same valence, an additional weight matrix $\mathbf{w}_{\text{fea}}$ is used to differentiate between these elements. If the valence of an atom is 1, the weights of the corresponding row in $\mathbf{w}_{\text{fea}}$ specify if this atom is H, F or Cl for example.

–Revised manuscript, p.2

For each of the two target ranges ($[-2.5, -2]$, $[5, 5.5]$) we generated 1000 molecules and evaluated their diversity using the average pairwise Tanimoto distance—we evaluated the diversity of all generated molecules, not only the ones in the target range to be

consistent with ref [33]. We limited the molecule size to 85 atoms (including hydrogens), again to be consistent with ref [33]. For this task generated molecules can contain the same element types as in the ZINC250 dataset: C, O, N, F, H, S, Cl, Br, I, and P. We initialized the generation with random molecules from QM9 because CrippenNet performed significantly better on these molecules.

–Revised manuscript, p.4

The trainable weights $\mathbf{w}_{\text{fea}}$ are only used to differentiate between elements forming the same number of bonds. For example, H and F both form a single bond (valence of 1), in that case $\mathbf{w}_{\text{fea}}$ is a single weight and ϕ is the sloped rounding function of equation 1; they determinethat determines if the element is H (0) or F (1). When there is more than two elements with the same valence, e.g. H, F, Cl, Br and I, then ϕ is a sloped max function defined as:

$$(\text{sloped max } \mathbf{x})_i = \begin{cases} 1 - a(1 - x_i), & \text{if } x_i = \max \mathbf{x} \\ bx_i, & \text{otherwise} \end{cases} \quad (8)$$

That function is applied only to the indices of the elements with the same valence. Finally, to ensure that the one-hot encoding are as close as possible to binary the sloped max function (eq 8) is applied to the rows of the final feature vector ($\mathbb{F}_{ij}$). Finally, to ensure that the one-hot encoding are as close as possible to binary with normalized rows, we use a sloped maximum function defined as:

–Revised manuscript, p.10 (SI)

Comment R2.3 The framework can be applied to various GNN proxy models. However only GCN and CrippenNet are applied for energy gap and logP respectively. I wonder whether the authors explored different GNN models.

Answer to R2.3 The framework was originally applied to CrippenNet and our own GNN architecture (that we call “SimpleNet” in the online repository). To address this comment we implemented and tested the framework with: Graph Isomorphism Network (GIN), Graph Attentions Networks (GATs), GraphSAGE (SAmple and aggreGatE), Graph Convolutional Network (GCN) and Crystal Graph Convolutional Neural Networks (CGCNN). This brings the total number of working implementations to 7. We also ran the energy gap targeting task with GIN and verified the results with DFT, adding 300 new DFT calculations. We added Figure 7 and a panel to Figure 8 in the SI. We also added the following passages to the manuscript and SI.

The inversion procedure presented here can be used on any GNN architecture that uses molecular 2D graphs (like SMILES), but it requires modifying them so that they use an explicit representation of the adjacency matrix which may require retraining. This can be a limiting factor for the adoption of our procedure. To help ease this process, in addition to the implementations of our simple energy gap GNN and CrippenNet, we implemented and tested 5 commonly used GNNs: Graph Isomorphism Network (GIN)[50], Graph Attentions Networks (GATs)[51], GraphSAGE (SAmple and aggreGatE)[52], Graph Convolutional Network (GCN)[53] and Crystal Graph Convolutional

Neural Networks (CGCNN)[54]. For GIN, GAT, GraphSAGE and GCN, we started from the implementations found in ref. [55]. These models can either be used directly to train and generate, or as a starting point to implement other models. For example, we performed the energy gap targeting task with GIN and compared the properties of the generated molecule with DFT. The model predictive performance matched the literature without any hyper-parameter tuning and the generation performance was slightly worse than when using our GNN. Detailed results are presented in the SI. ~~To give an example of how this process can be done we make available on our public repository a version of the popular CGCNN model [54] using an explicit adjacency matrix.~~

–Revised manuscript, p.5

The molecular representation was initialized either completely randomly or starting from existing molecules in QM9 and adding a small amount of random noise to their representations. We find that the later gives slightly better results because the proxy model performs better on more QM9-like molecules. ~~We also trained our explicit adjacency matrix implementation of a Graph Isomorphism Network (GIN)[50] on QM9. Its predictive performance is plotted in Figure 7. We used this model to generate 100 molecules for each target gap and calculated their DFT energy gap. Table III is a reproduction of Table I with an additional row for the random generation and an additional row for generation using DIDgen with GIN. Figure 8 illustrates the performance of DIDgen and JANUS at generating molecules with different models and initializations. Figure 8 a shows the performance of DIDgen at generating molecules with a specific energy gap starting from random noise.~~

–Revised manuscript, p.12 (SI)

Comment R2.4 Since the framework requires an explicit adjacency matrix to represent molecules, it restricts the application to many powerful architectures, especially those with equivariant message-passing.

Answer to R2.4 This is a very interesting question that touches on many aspect of our work and molecular generation in general. We broke down the implications of the question below:

1. **Can you have an equivariant message-passing network with an explicit adjacency matrix?** Yes. An explicit adjacency matrix contains the same information as a “normal” graph; any graph can be represented as a matrix.
2. **Can the current DIDgen framework be applied to equivariant message passing networks?** No. All the models we implemented are *invariant* message passing networks. Equivariance with respect to rotation requires 3D information which DIDgen currently does not handle. We would like to extend DIDgen to 3D-graphs by including 3D information as edge or node features that would be optimized along with the current features. We believe this will be possible but outside the scope of this paper.
Moreover, none of the other frameworks presented here (JANUS, SGDS, GCPN and OR-GAN) generate molecule conformations which means although some of them *could* use an equivariant message passing network for their evaluation, the conformation would have to

be inferred using a different method. The error on finding this conformation would probably outweigh the advantage conferred by equivariance.

To partly verify this claim, we trained PaiNN a popular equivariant message passing network to predict energy gap on QM9. We used the default parameters and obtained an MAE of 0.048 eV which is very close to the reported MAE of 0.046 eV. We ran this model on 1000 random QM9 molecules using (1) the DFT relaxed atomic positions and (2) the atomic positions from the RDKit embedding function. We obtained MAEs of 0.034 eV and 0.200 eV respectively; the performance is significantly reduced when the conformation is not given. The performance with the conformation inferred by RDKit is actually slightly worse than that of our invariant 2D-graph model which had an MAE of 0.173 eV on these molecules.

3. **Do more powerful equivariant message-passing networks perform better on generative tasks?** Not necessarily. Since generated molecules are often outside of distribution, the advantage of equivariant message passing networks over 2D-graphs networks might be marginal or non-existent on these molecules. To partly verify this claim, we used the same PaiNN model (an equivariant message-passing network) to make predictions on all molecules generated by DIDgen and JANUS for which we have calculated the energy gap with DFT. This forms a dataset of 2410 molecules including 522 molecules that were already in QM9 (mostly from starting populations of JANUS). Figure 9 in SI compares the performance of PaiNN and our 2D-Graph model on the generated molecules (excluding QM9 molecules) and the QM9 molecules alone as a reference. As expected PaiNN outperforms our model on the qm9 data, but our model performs slightly better on the generated molecules despite using less information (PaiNN uses the relaxed DFT positions). Table IV of the SI presents a breakdown of the different types of generated data. The 2D-model performs better on every subset.

Following this discussion we decided to include the dataset of 2410 xyz files and corresponding dipole moments, HOMO energies, LUMO energies and energy gaps with the paper such that other models can be benchmarked on this dataset of generated molecules. We also added Figure 9, Table IV as well as the following text to the manuscript and SI:

... We demonstrate the application of this method by generating molecules with specific DFT-verified energy gaps and octanol-water partition coefficients (logP). Our approach hits target properties with rates comparable to or better than state-of-the-art generative models while consistently generating more diverse molecules. [Moreover, while validating our framework we created a dataset of 1617 new molecules and their corresponding DFT-calculated properties that could serve as an out-of-distribution test set for QM9-trained models.](#)

—Abstract, p.1

... In the future, our method could be used as part of an active learning loop to iteratively improve the model generalizability, in this case, by retraining on a set of molecules containing O-O bonds, for example.

[This drop in model performance is observed across all generated molecules including molecules generated with JANUS using DFT and with DIDgen using the Graph Isomorphism Network \(GIN\)\[50\]. To verify if this drop in performance would also affect more complex models, we trained, PaiNN \[51\], an equivariant message passing network](#)

on QM9 and tested its performance on generated molecules. Despite having a much lower MAE on QM9 data (0.048 eV vs. 0.12 eV), PaiNN performed worse than our model on generated molecules with an MAE of 1.16 eV versus 0.96 eV for our model (see Figure 9 and Table IV). This suggests that generating molecules might be a good way to obtain out of distribution data that could be used for benchmarking and fine-tuning ML models. As a first step in that direction, we provide on our public repository a dataset of 1617 unique generated molecules with their corresponding DFT relaxed 3D coordinates as well as their dipole moments, HOMO energy, LUMO energy and energy gap. This data was generated to obtain Table I and Table III.

–Revised manuscript, p.5

DFT Dataset

When gathering data for the energy gap task, we performed 2410 DFT calculations with the parameters described in the Methods section. These calculations were done on 2017 unique molecular graphs (conformation may differ), 1617 of which were new molecules that cannot be found in QM9. We make these 2410 calculations and corresponding xyz files available on our public repository with the following labels: dipole moment (mu) in Debye, lowest unoccupied molecular orbital energy (E_L) in eV, highest occupied molecular orbital (E_H) in eV, HOMO-LUMO energy gap (gap) in eV, the generation method, the qm9 ID number if available, the initial smiles string before relaxation (smiles_init) and the final smiles string after relaxation (smiles_final).

We compared our DFT calculations of the energy gap with the reported value for the 522 molecules that are also in QM9. We found a mean average difference of 0.221 eV. To eliminate the effect of conformation on that difference, we also recalculated the energy gap for 100 of these molecules using the same atomic coordinates as in QM9 and found a similar MAE of 0.212 eV. The difference is probably caused mostly by the use of a slightly different basis set 6-31g(2df,2p) rather than 6-31g(2df,p) used in QM9. In Table IV we benchmarked two models, PaiNN and SimpleNet (ours) on different splits of this dataset. Despite having much better performance on QM9 molecules, PaiNN performs slightly worse than our model on generated molecules (see also Figure 9). The values presented in Table IV exclude predictions above 12 eV for PaiNN. For both models, molecules generated with DIDgen are harder to predict than molecules generated with JANUS.

–Revised manuscript, p.15 (SI)

Comment R2.5 It is mentioned that in some cases no conformation could be obtained from the generated molecules. And these molecules are discarded. What’s the ratio of such failure cases?

Answer to R2.5 The ratio of molecules for which the RDKit embedding function could not find a conformation was 19% when targeting 4.1 eV, 2% when targeting 6.8 eV and 1% when targeting 9.3 eV. Molecules with a smaller gap tend to be larger which makes finding the conformation more difficult. To address this comment we now state these numbers explicitly in the manuscript:

For each generated molecule, we obtained the conformation (3D atomic positions) using the RDKit. In some cases no conformation could be obtained from the generated molecule. In those cases, the molecule was discarded and a new molecule was generated until a total of 100 molecules with conformation were obtained for each target. About 19% of initially generated molecules were discarded when targeting 4.1 eV, 2% when targeting 6.8 eV and 1% when targeting 9.3 eV

–Revised manuscript, p.6

Answers to Reviewer #3

This research explores the use of a graph neural network as a tool for generative molecular discovery with desired electronic properties. Overall, the paper is well-written. However, some contractions (e.g., "don't") should be corrected to their full forms (e.g., "do not") for a more formal tone. More discussion and clarification are needed to make it impactful. I believe it should be published after addressing the following concerns:

Answer to R3.0 We appreciate that the reviewer thought the paper was well written and should be published after addressing their concerns. We have revised the text to eliminate contractions. They are indicated in blue throughout the revised manuscript.

Comment R3.1 This method allows for greater flexibility and control over the diversity of the generated molecules, as you can directly adjust the characteristics of the input to explore different regions of chemical space. However, it is unclear whether diversity is a fair evaluation metric for benchmarking. The author could incorporate a correlation metric such as Pearson or R2.

Answer to R3.1 Although we agree that diversity is not a perfect metric, we include it because it does not depend on the performance of the proxy. Other metrics such as MAE or Pearson correlation depend on the generator *and* the proxy (as pointed out in R3.3). However we think it is worthwhile to add an additional metric that accounts for all three objectives. So, as suggested, we added Pearson correlation coefficients to Table I and III.

Comment R3.2 The author highlighted the current limitations due to the computational cost associated with this method. It would have been beneficial to see a comparison of the computation times between the proposed method and other methods, as this was not compared or discussed in the paper.

Answer to R3.2 In response to this comment, we added the following discussions about compute times at the end of the "Energy gap targeting" and "logP targeting" sections:

DIDgen takes on average 12.0 s to generate an in-target molecule for 4.1 eV, 2.1 s for 6.8 eV and 10.4 s for 9.3 eV whereas JANUS takes about 100 s to generate 100 in-target molecules (1 s per molecule) for all targets using the same computer (4-CPU, 3.40 GHz). Timing varies significantly depending on the task and parameters used for both methods. DIDgen generates all molecules completely independently; all 100 molecules could be generated simultaneously. We did not implement batch generation with the use of a GPU, but it might significantly enhance performance in future versions.

–Revised manuscript, p.3

DIDgen takes on average 5.6 s to generate a molecule in the -2.5 to -2 range and it

takes on average 3.4 s to generate a molecule in the 5 to 5.5 range on a local machine (4-CPU, 3.40GHz). The authors of LIMO report generating 33 molecules per second within the target range on 2 GTX 1080 Ti GPUs. Other authors did not report their compute times. Again, the performance of DIDgen could be improved with trivial parallelization and batch generation on GPUs.

–Revised manuscript, p.4

Comment R3.3 The authors need to clarify the error(RMSE or MAE) associated with the pretrained models in the property prediction. Ultimately, the final molecule prediction will have two sources of error: one from the initial models trained and one from the proposed approach optimization. The authors could explain how they address this degree of uncertainty.

Answer to R3.3 We completely agree that it is difficult to differentiate between the contribution of the proxy and the contribution of the generator on the performance metrics especially MAE and success rate. Ideally all generation methods should be compared with each other using the same proxy, but this is often not possible since most methods use their own proxy. That is why, in the energy gap targeting task, we compared DIDgen and JANUS using exactly the same proxy such that the only difference was the generation scheme. We also discussed the importance of the proxy on performance with Reviewer #2 (see R2.1). We discussed how, when an oracle is slow, using a proxy model is advantageous despite the added uncertainty while, when an oracle is fast, it can be used to quickly sort through generated molecules. Here are a few passages of the revised manuscript that are relevant to this comment:

When compared to other methods that use a trained proxy model as a predictor for logP, as opposed to the ground truth empirical model which we termed “oracle” in Table II, DIDgen shows the highest performance by a factor of x4 and x2 respectively. However, it does not have a higher success rate than GCPN and SGDS which both use the oracle directly. This is not surprising since the success rate relies heavily on the proxy model performance and generalizability. All methods except JT-VAE could technically use CrippenNet which would offer a way to compare the generation schemes themselves separately of the proxy model’s performance (like we did for the energy gap task).

–Revised manuscript, p.4

Generative methods like ours that use a learned proxy rely heavily on its performance, especially its ability to generalize in order to hit target properties. Models that can directly use an oracle do not always suffer from this limitation. For example, in the logP targeting task, the oracle was not computationally limiting and thus the number of evaluations was not taken into account. In that case, oracle-based methods had superior success rates. But since evaluations are not costly, as pointed out by other groups of authors [27], it would be easy to generate more molecules and filter out molecules that don’t meet the criteria.

Another way to evaluate the performance of generative models that is independent of the evaluation method is to measure their ability to generate diverse molecule.

Comment R3.4 I was wondering about the property distribution of molecules in the datasets used for training, validation, and testing. This should be clarified.

Answer to R3.4 We added new panels to Figure 4 and Figure 6 showing the distribution of energy gap in QM9 and the distribution of logP in the ZINC250 dataset respectively along with the target values for both tasks.

Comment R3.5 It is not clear what motivated the use of the average composition in QM9 as an additional term to guarantee real molecules. How does this term address "realistic" problems?

Answer to R3.5 This is not a term that guarantees real molecules, it is a very minimalist way to ensure that generated molecules will not have disproportional amounts of oxygen, nitrogen and Fluorine. Molecules in QM9 (like real organic molecules) tend to have more oxygen and carbon and much less Fluorine, but "valid" molecules in terms of valence do not necessarily follow this trend. By nudging the objective towards more QM9-like proportions between different elements, the algorithm tends to generate molecules for which the ML proxy has a better chance of performing well. This is a very simple example of how one can tweak the objective of DIDgen to generate "better" molecules. A more elaborate version of this would be to use two objectives for example: the energy gap and formation energy, optimizing through two GNNs at once, or the energy gap and some graph based accessibility score. The word "composition" was misleading and we replaced it with atomic fraction. We clarified this idea in the manuscript.

We trained a simple GNN on the QM9 dataset [39] and used our direct inverse design procedure to generate 100 molecules with HOMO-LUMO gaps within 10 meV of 3 different target values: the first percentile of QM9 energy gaps, 4.1 eV, the median, 6.8 eV, and the 99th percentile, 9.3 eV. ~~As an additional soft target, we aimed for atomic fractions (the relative number of each elements in the molecule) close to the average atomic fractions in QM9. This helps guide the generation towards molecules that the proxy can predict better. As an additional soft target, we aimed for compositions close to the average composition in QM9. This helps guide the generation towards molecules that the proxy can predict better.~~

~~In order to improve generative performance, we added a simple additional term to the loss function: the average atomic fraction, C_i , in QM9 for each element i , i.e. the fraction of all atoms that are of type i . Since chemically valid molecules are not always realistic (they don't necessarily have a stable conformation), in order to guide~~

the generation towards more realistic molecules, we added an additional term to the loss function. We used the average composition, C_i , in QM9 for each element i as an objective. The total loss for this task was the following:

$$\mathbb{L} = \frac{\lambda_l \mathbb{L}_l^2 + \lambda_s \mathbb{L}_s^2 + \lambda_c \mathbb{L}_c^2}{\lambda_l + \lambda_s + \lambda_c}, \quad (2)$$

where

$$\mathbb{L}_c = \sum_i \frac{\sum_j \mathbb{F}_{ij}}{\sum_{ij} \mathbb{F}_{ij}} - C_i, \quad (3)$$

$\mathbb{L}_l$ is the root mean square error with respect to the target p , $\mathbb{L}_s$ is the loss associated with the maximum valence defined in equation 4 of the supplementary information and $\lambda_{l,s,c}$ are adjustable hyperparameters. Generated molecules with balanced valence could include molecules with disproportional amounts of certain elements (by replacing all H with F, for instance). The idea of this loss is to steer the generation away from these cases for which the proxy model would underperform and where there would likely be no valid conformation. It is a very simple example of how the loss can be used to guide generation towards molecules with desirable properties. More complex co-objectives such as accessibility scores could be used in the future.

–Revised manuscript, p.6