nature computational science

Article

https://doi.org/10.1038/s43588-023-00532-0

Guided diffusion for inverse molecular design

In the format provided by the authors and unedited

Supporting Information

Contents

1 Detailed Formulation of Diffusion Models

Here, we provide a detailed review of the formulation of Gaussian diffusion models from Ho *et al.*[1]. We start by defining our data distribution $\mathbf{z}_0 \sim q(\mathbf{z}_0)$ and a Markovian noising process q which gradually adds noise to the data to produce noised samples $\mathbf{z}_1$ through $\mathbf{z}_T$. In particular, each step of the noising process adds Gaussian noise according to some variance schedule given by β_t,

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \sqrt{1-\beta_t}\mathbf{z}_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

Ho *et al.*[1] note that in lieu of t repeated applications of single-step transitions, $q(\mathbf{z}_t|\mathbf{z}_0)$ can directly be expressed as a Gaussian distribution

$$q(\mathbf{z}_t|\mathbf{z}_0) = \mathcal{N}(\mathbf{z}_t; \sqrt{\bar{\alpha}_t}\mathbf{z}_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{2}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$. Alternatively, the distribution can be viewed as the following affine transformation of the normal distribution,

$$q(\mathbf{z}_t|\mathbf{z}_0) = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \boldsymbol{\epsilon}\sqrt{1-\bar{\alpha}_t},\ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{3}$$

We note that $1 - \bar{\alpha}_t$ expresses the noise variance at an arbitrary time step, and can be used as an alternative way to define the noise schedule instead of β_t.

Using Bayes' theorem, one finds that the posterior $q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0)$ also has Gaussian distribution

$$q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0) = \mathcal{N}(\mathbf{z}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{z}_t, \mathbf{z}_0), \tilde{\beta}_t\mathbf{I}) \tag{4}$$

with mean $\tilde{\boldsymbol{\mu}}_t(\mathbf{z}_t, \mathbf{z}_0)$ and isotropic variance $\tilde{\beta}_t$ defined as

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{z}_t, \mathbf{z}_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{z}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{z}_t,\ \ \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{5}$$

In order to sample from the data distribution $q(\mathbf{z}_0)$, one can first sample from $q(\mathbf{z}_T)$ and then proceed in reversed time direction sampling from $q(\mathbf{z}_{t-1}|\mathbf{z}_t)$ until $\mathbf{z}_0$. Under reasonable settings for β_t and T, the distribution $q(\mathbf{z}_T)$ is nearly an isotropic Gaussian distribution, so sampling $\mathbf{z}_T$ is trivial. All that is left is to approximate $q(\mathbf{z}_{t-1}|\mathbf{z}_t)$ using a neural network since it cannot be computed exactly when the data distribution is unknown. To this end, Dickstein *et al.* note that $q(\mathbf{z}_{t-1}|\mathbf{z}_t)$ approaches a diagonal Gaussian distribution as $T \to \infty$ and correspondingly $\beta_t \to 0$, so it is sufficient to train a neural network to predict a mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}$ and a diagonal covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$,

$$q(\mathbf{z}_{t-1}|\mathbf{z}_t) \approx p_{\boldsymbol{\theta}}(\mathbf{z}_{t-1}|\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_t, t), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)). \tag{6}$$

To train this model such that $p(\mathbf{z}_0)$ learns the true data distribution $q(\mathbf{z}_0)$, we can optimize the following variational lower-bound $L = L_0 + \cdots + L_T$ for $p_\theta(\mathbf{z}_0)$, where

$$L_0 = -\log p_\theta(\mathbf{z}_0|\mathbf{z}_1), \tag{7}$$

$$L_{t-1} = D_{KL}(q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{z}_0) \,||\, p_{\boldsymbol{\theta}}(\mathbf{z}_{t-1}|\mathbf{z}_t)), \tag{8}$$

$$L_T = D_{KL}(q(\mathbf{z}_T|\mathbf{z}_0) \,||\, p(\mathbf{z}_T)). \tag{9}$$

While the above objective is well-justified, Ho *et al.* found that a different objective produces better samples in practice. In particular, they do not directly parameterize $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)$ as a neural

network, but instead train a model $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)$ to directly predict $\boldsymbol{\epsilon}$ from Equation 3. The simplified training objective is defined as

$$L_{\text{simple}} = \mathbb{E}_{t\sim[1,T],\mathbf{z}_0\sim q(\mathbf{z}_0),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)\|^2 \tag{10}$$

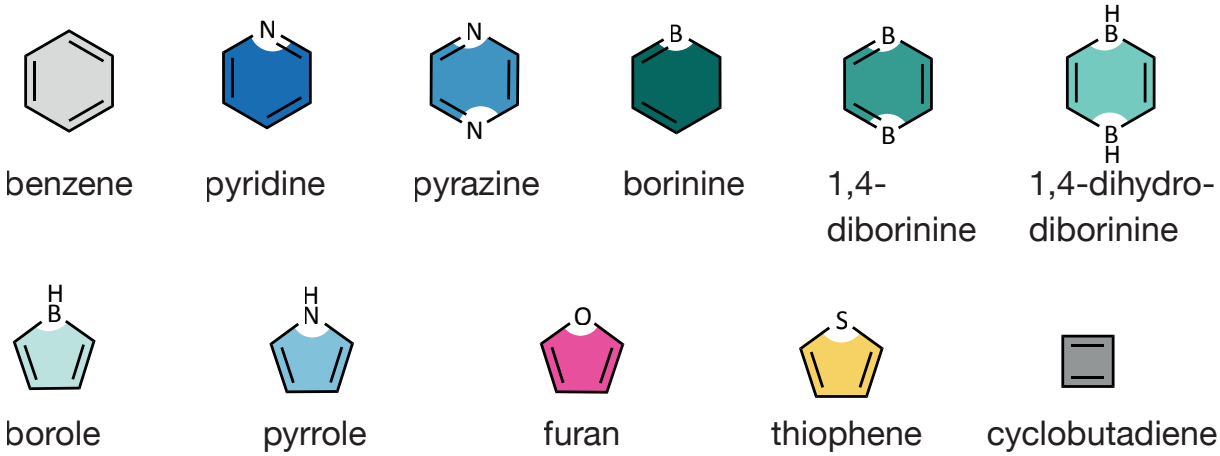
Note that training with the above loss never requires backpropagating through time of the diffusion process; therefore, backpropagation depth is bounded by the depth of the model $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ which poses no numerical challenges such as vanishing or exploding gradients.

During sampling, we can straightforwardly derive the location parameter $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)$ from $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)$:

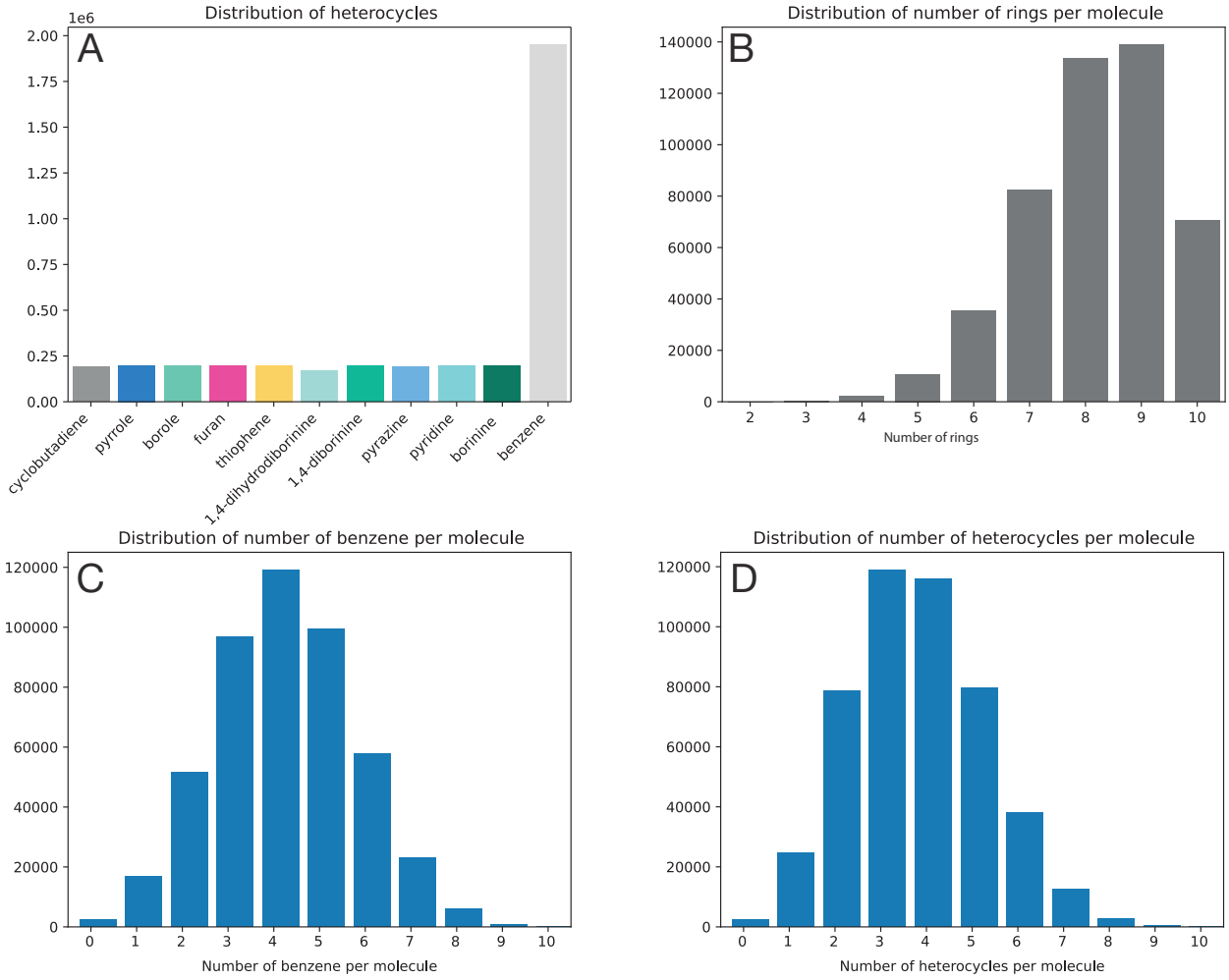
$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)\right) \tag{11}$$

Note that L_{simple} does not provide any learning signal for $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)$. Ho *et al.* find that instead of learning $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{z}_t, t)$, a constant sequence of isotropic covariances $\beta_t\mathbf{I}$ or $\tilde{\beta}_t\mathbf{I}$ can be used. These scales correspond to the upper and lower bounds, respectively, for the true reverse step variance.

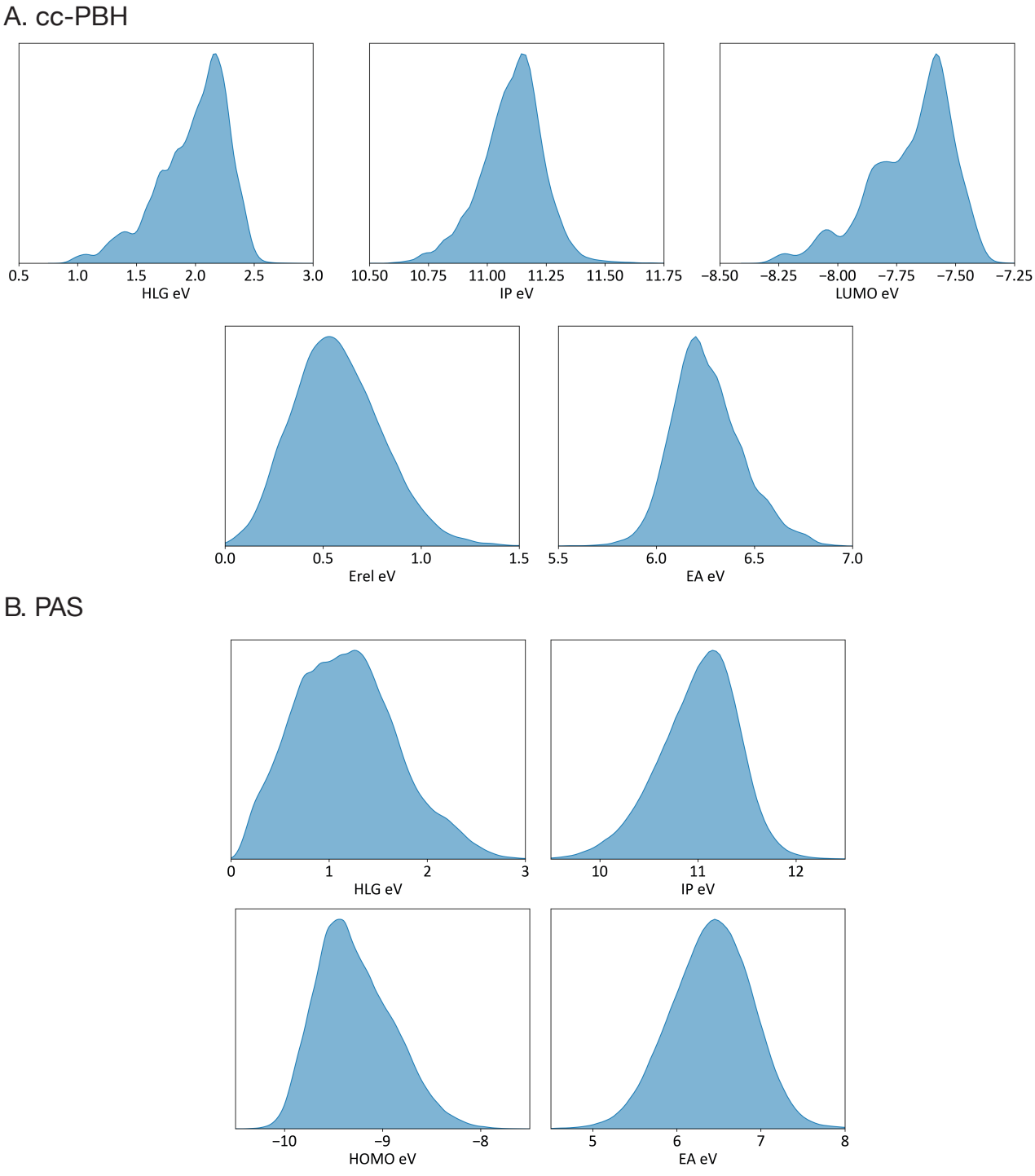
2 PASs Data Set Generation



Supplementary Figure 1: Library of aromatic/antiaromatic building blocks for the PASs data set. Top row, from left: benzene (light gray), pyridine (dark blue), pyrazine (blue), borinine (dark cyan), 1,4-diborinine (green), 1,4-dihydroborinine (teal); Bottom row, from left: borole (light teal), pyrrole (light blue), furan (magenta), thiophene (yellow), cyclobutadiene (dark gray).



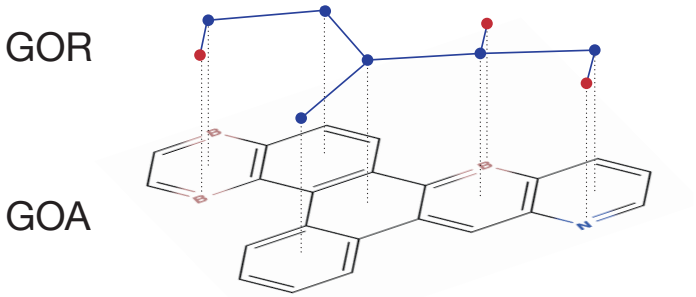
Supplementary Figure 2: Overview of the structural diversity of the PAS data set. a) Histogram of the prevalence of the respective ring types across all molecules in the PAS data set. b) Histogram of the total number of rings per molecule in the data set. c) The number of benzene moieties per molecule. d) The number of non-benzene moieties per molecule.



Supplementary Figure 3: Distributions of the molecular properties in the (A) cc-PBH data set and in the (B) PAS data set. The properties displayed: HOMO-LUMO gap, Ionization potential, LUMO, E_{rel}, Electron affinity.

3 GOR Representation

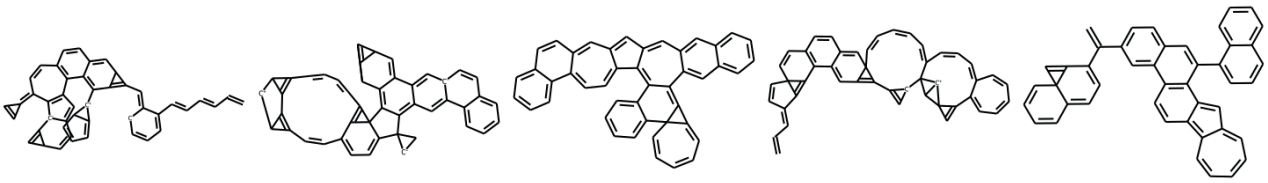
As described in the main text, in this work we implemented a more sophisticated version of our graph of rings (GOR) representation, taking into account the location of the heteroatom(s) in the molecules. In Supplementary Figure 4, we present an illustration of the correspondence between the GOR and GOA representations, with the additional nodes for the heteroatoms.



Supplementary Figure 4: Example of the graph of rings (GOR) representation. Blue nodes represent rings and red nodes are the orientation nodes.

4 Unguided Generation - Graph of Atoms Representation

As can be seen in the main text, the EDM[2] was able to generate almost 100% valid molecules when trained using the graph of rings (GOR) representation. To compare, we trained the same model on the same data, this time using the graph of atoms (GOA) representation. The results were significantly poorer: the percentage of valid molecules, as measured by RDKit,[3] was only 31.3% (note: RDKit determined the validity of a molecule by checking the valency of each atom). In other words, only 31.3% of the generated structures obeyed chemical bonding rules. However, visual inspection of the generated molecules revealed that most of these formally valid molecules are not, in fact, within the PASs chemical space. Indeed, some of the proposed structures are entirely unfeasible and illogical, from a chemical perspective. Some visual examples of non-PASs molecules that were marked as valid by RDKit are available in Supplementary Figure 5. When using the GOR representation – all the valid generated molecules are PASs molecules, by definition. Nevertheless, we recognize that the structures generated using the GOR may not be currently synthetically accessible, either.



Supplementary Figure 5: Visual examples of non-PAS molecules marked as valid by RDKit.

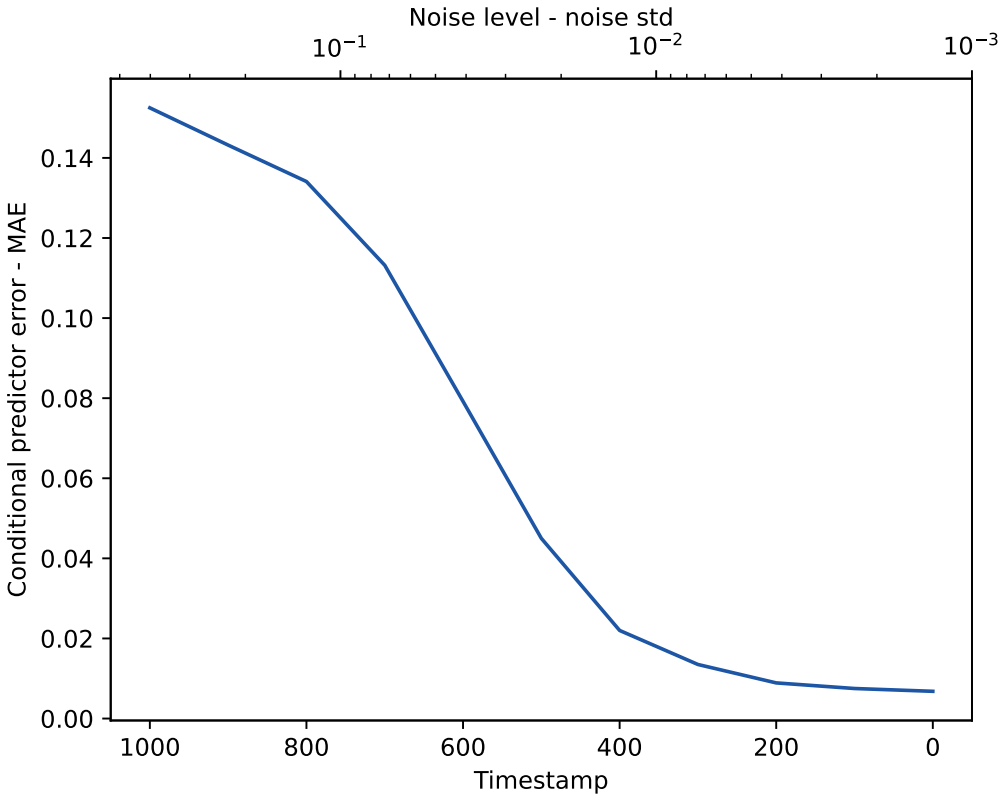
5 Guided Diffusion Sampling Algorithm

Supplementary Algorithm 1 Guided diffusion sampling, given a diffusion model $(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_t), \boldsymbol{\Sigma}_t)$, $f(\mathbf{z}, t)$, and gradient scale s.

$\mathbf{z}_T \leftarrow$ sample from $\mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, T-1, \ldots, 1$ **do**
  $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{z}_t)$
  $\mathbf{g} \leftarrow -\nabla_{\mathbf{z}_{t-1}} f(\mathbf{z} = \boldsymbol{\mu}, t)$
  $\mathbf{z}_{t-1} \leftarrow$ sample from $\mathcal{N}(\boldsymbol{\mu} + s\boldsymbol{\Sigma}_t\mathbf{g}, \boldsymbol{\Sigma}_t)$
end for
return x_0

6 Prediction Model Performance

As described in the main text, we used the EGNN[4] as a conditional predictor to predict the desired properties during the generation process. As can be seen in Supplementary Figure 6, in the beginning of the denoising (generation) process, the error is high; as the samples become cleaner, the prediction error decreases. In order to verify the EGNN model, we tested the trained model on clean samples only, and obtained a 0.0078 eV MAE when tested on the cc-PBH data set. These results are very similar to the performance achieved in our previous work. [6] Thus, we deemed the predictor satisfactory. Additionally, we wanted to validate the GOR representation, which we modified from our previous implementation by adding an additional node to denote the location of the heteroatom(s). We obtained an MAE of 0.0071 eV for the GOR and 0.0078 eV for the GOA. This demonstrates that using the GOR does not results in loss of information. Notably, training the model with the GOR reduced the running time from 270 hours to only 54 hours.



Supplementary Figure 6: Prediction error (MAE) when predicting the HOMO-LUMO gap as a function of the timestamp of the inverse diffusion process. An equivalent noise level axis is shown for convenience.

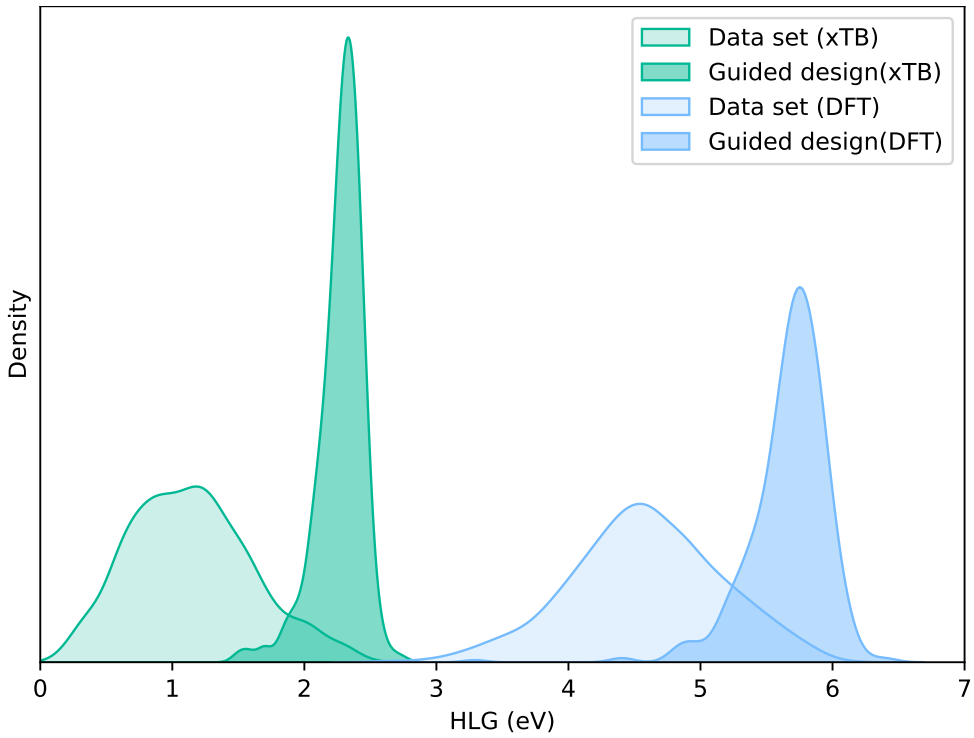
7 Training on Inexpensive Data

One of the greatest challenges in data-driven molecular design is the paucity of existing data. This is especially true for PASs, for which no database has been curated yet. Therefore, for the current work, we undertook to prepare a new data set containing almost half a million molecules with high-throughput computational chemistry. To complete the data generation in a feasible timeline, we performed the calculations with GFN2-xTB, a semi-empirical method. In recent years, xTB has become a commonly used computational tool and is considered to be reliable. Indeed, we have recently shown that xTB captures the same structure-property relationships as the much more expensive hybrid functional, B3LYP.[5] Nevertheless, the numerical values for xTB-calculated properties are in completely different ranges than those found by DFT. This can be seen in Supplementary Figure 7, where the distribution of the HLG for the data set is given, calculated with these two methods.

While this does not impede identification of trends in the data, it makes it harder to use the data with models that require some high-accuracy reference (e.g., for calculating charge transport, power conversion efficiency, or relaxation energy), and makes it impossible to task generative models with specific target values. For example, it is meaningless to task a model with finding a molecule with a HOMO of -4.5 eV, when all of the values of the training data are approximately 4-5 eV lower in value.

One possible solution is to correct all the xTB values to DFT-level values with a correction scheme. A different approach is to leverage the ability of GaUDI to be tasked with a min/max function is extremely useful. Thus, we trained GaUDI on the 15,000 data points of inexpensive (and lower accuracy) xTB data and tasked it with generating molecules with a maximal HLG. We then sampled a batch of 512 molecules generated by GaUDI and calculated the HLG using xTB. While the highest HLG in the training data set was 2.5 eV, the highest HLG of the molecules designed by GaUDI was 2.75 eV. Supplementary Figure 7 shows the shifting of the distribution of the HLG for the generated molecules versus the original data. We then calculated the HLG for the same molecules using DFT (CAM-B3LYP/def2-svp). As seen in Supplementary Figure 7, the distribution of the DFT data also shifted to higher HLG values. The highest HLG from DFT calculations was 6.43 eV (highest in the data set was 6.12).

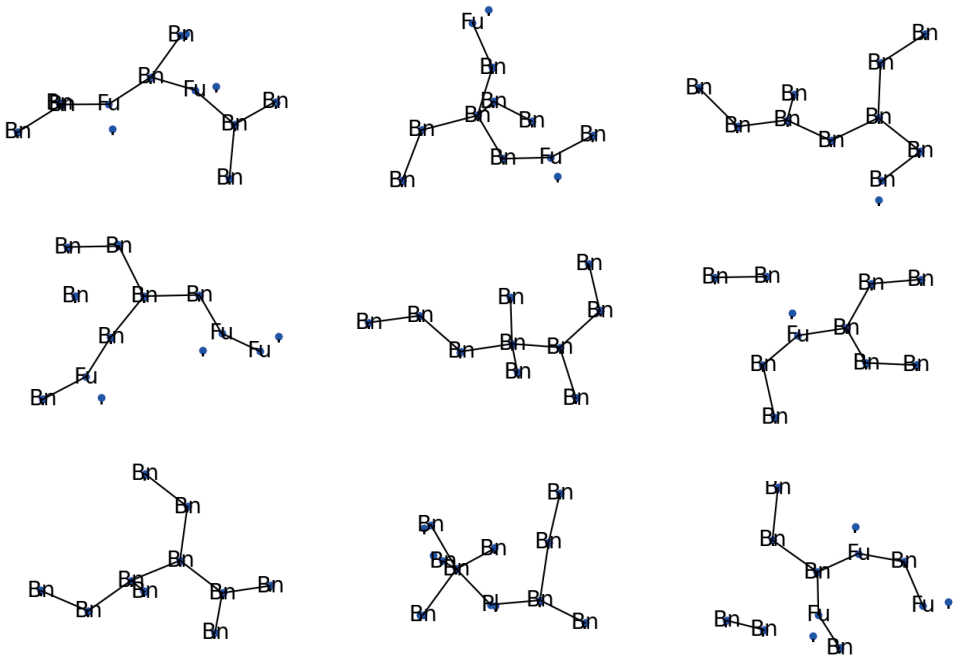
Thus, we showed that by tasking GaUDI with a max function, the model could be trained on inexpensive but less accurate data and still lead to meaningful results.



Supplementary Figure 7: Distribution of the data set versus distribution of a 512-molecule batch generated by GaUDI, calculated with xTB (green) and DFT (blue).

8 Examples of Invalid Molecules

In this section we provide some examples of the invalid molecules generated by GaUDI, for cases when the validity did not reach 100%. The molecules are presented in the GOR representation. The text represents the node type ('Bn' - benzene, 'Fu' - Furan, etc.). The most common reasons for invalidity are: a) disconnected parts, i.e., too large of a distance between rings; b) overlapping rings; c) wrong connectivity - e.g. benzene having 4 neighbors; e) infeasable spatial orientation.



Supplementary Figure 8: Examples of invalid molecules.

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[2] Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pages 8867–8887. PMLR, 2022.

[3] Greg Landrum et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 8, 2013.

[4] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pages 9323–9332. PMLR, 2021.

[5] Alexandra Wahab, Lara Pfuderer, Eno Paenurk, and Renana Gershoni-Poranne. The compas project: A computational database of polycyclic aromatic systems. phase 1: catacondensed polybenzenoid hydrocarbons. *J. Chem. Inf. Model.*, 62(16):3704–3713, 2022.

[6] Tomer Weiss, Alexandra Wahab, Alex M. Bronstein, and Renana Gershoni-Poranne. Interpretable deep-learning unveils structure–property relationships in polybenzenoid hydrocarbons. *The Journal of Organic Chemistry*, 2023.