Journal Name

ARTICLE TYPE

Cite this: DOI: 00.0000/xxxxxxxxxx

Materials Funnel 2.0 - Data-driven hierarchical search for exploration of vast chemical spaces

Raul Ortega Ochoa,[a] Bardi Benediktsson,[a] Renata Sechi,[a] Peter Bjørn Jørgensen,[a] and Arghya Bhowmik[a*]

Received Date
Accepted Date

DOI: 00.0000/xxxxxxxxxx

Innovating ways to explore the materials phase space accelerates functional materials discovery. For breakthrough materials, faster exploration of larger phase spaces is a key goal. High-throughput computational screening (HTCS) is widely used to rapidly search for materials with the desired functional property. This article redefines the HTCS methods to combine multiple deep learning models and physics-based simulation to explore much larger chemical spaces than possible by pure physics-driven HTCS. Deep generative models are used to autonomously create materials libraries with a high likelihood of desired properties, inverting the standard design paradigm. Additionally, machine-learned surrogates enable the next layer of screening to prune the set further so that high-quality quantum-mechanical simulations can be performed. With organic photovoltaic (OPV) molecules as a test bench, the power of this redesigned HTCS approach is shown in the inverse design of OPV molecules with very limited computational expense using only ∼1% of the original physics-based screening dataset.

1 Introduction

Accelerating the search for energy materials with properties, stability, and reliability beyond the state of the art is a complex but important step to overcome technical obstacles in the green energy revolution, as material properties often limit what can be achieved at the system level. Since its inception, two decades ago [1], high throughput computational screening (HTCS) has been the key to groundbreaking discoveries in energy materials such as batteries [2–5], thermoelectrics [6,7], photo/electrocatalysts [8–12] solar cell materials [13–15] etc.

The discovery of functional energy materials has been limited by our capacity to computationally and experimentally explore newer and larger phase spaces. Thus HTCS methods need to be rethought to search for novel materials with desired properties in exponentially larger phase spaces. We introduce a novel HTCS paradigm that integrates data-driven methods with traditional physics-based models, illustrating effective strategies for exploring new energy materials and molecules throughout the chemical universe. This approach allows us to explore practically infinite phase space; therefore, it can establish a new era in computational materials design by letting us search through material spaces that were inaccessible before.

[a]*Technical University of Denmark, DTU Energy, Anker Engelunds Vej, Building 301 2800 Kgs. Lyngby, Denmark, E-mail: arbh@dtu.dk.*

The challenge of *inverse design* is to find the structure manifold [16], the collection of materials that satisfy multiple desired properties out of the incredibly large space of possible materials. Inverse design is a one-to-many problem since there can be multiple materials, with different structures, satisfying a set of properties.

In this article, we propose a novel HTCS accelerated inverse design in a very large materials space combining the benefits of generative modeling, computationally efficient machine learning surrogate and high-quality physics-based simulation.

Our unique combination of generative and regressive machine learning can be roughly divided into two steps: (a) creating meaningful candidate libraries and (b) filtering based on the target properties. To search in a larger chemical space, these two steps need to be redefined and accelerated. The selection of better candidates from the chemical universe reduces the number of materials that need to be tested, searching *smarter*. Reducing the prediction time allows testing more candidates per unit of time, searching *faster*.

The traditional approach, which is optimizing the pre-selection step, relies on expert, pre-existing knowledge, for example, knowledge of common substructures often found in materials with similar properties. Using correlations of commonly found substructures, topological fingerprints, or other human-extracted correlation patterns can be a restrictive assumption, since the

correlations may lie in a complex non-linear space that cannot be easily captured. However, data-driven approaches like deep-generative models can be used to learn complex non-linear correlations, without domain expertise[17–22].

The pruning step, which predicts the properties of candidate materials, often relies on computationally expensive and time-consuming calculations such as density functional theory (DFT) simulations[23]. To increase the speed of property prediction, meta-models, or surrogate models, can be used as a prefilter to expensive computations[24,25], reducing the computational cost by exchanging some accuracy for faster predictions. Deep learning (DL)-based surrogate models are a popular choice for this task, because they achieve a great computation speedup with minimal loss of accuracy[26].

In summary, our novel HTCS 2.0 funnel increases the vastness of the chemical space that is accessible by employing a Conditional Deep Generative Model to pre-select meaningful candidates, a Deep Learning surrogate property predictor model to filter the pre-selected candidates, and finally DFT calculations as the last screening step. Fig. 1 displays the structure of our tool.

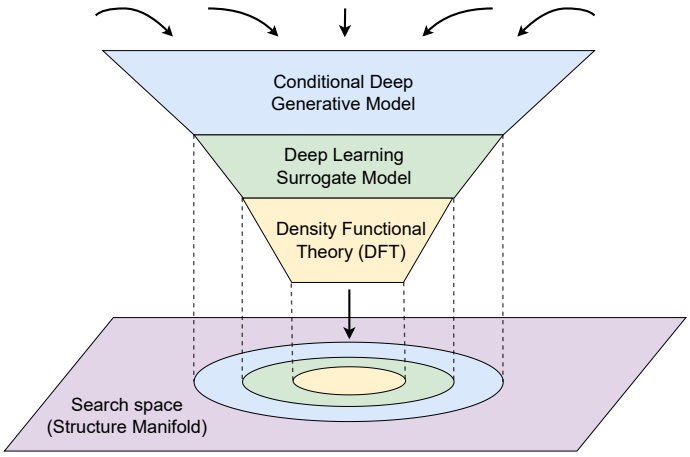

Fig. 1 Funneling samples from the large search space into final materials with the sought-after property. Our hybrid data and physics model-based funnel maintain high accuracy while enabling search in a much wider space.

2 Methods

The proposed approach is tested on an extended version of *Harvard Clean Energy Project (CEP) dataset*[27]. The dataset was produced from a DFT based large-scale screening of organic photovoltaic (OPV) candidates and contains the lowest unoccupied molecular orbital (LUMO) and highest occupied molecular orbital (HOMO) energies. The key molecular property for harvesting solar energy is the HOMO-LUMO gap, the energy difference between the LUMO and HOMO.

In Fig. 2, a sample molecule from the dataset is shown. The distribution of the HOMO-LUMO gap in the dataset is Gaussian-like with a mean of 2.8 eV and values ranging from 1.6 to 4.1 eV. To demonstrate the data efficiency of our approach, we used only 40k molecules to train the data-driven models used in the HTCS.

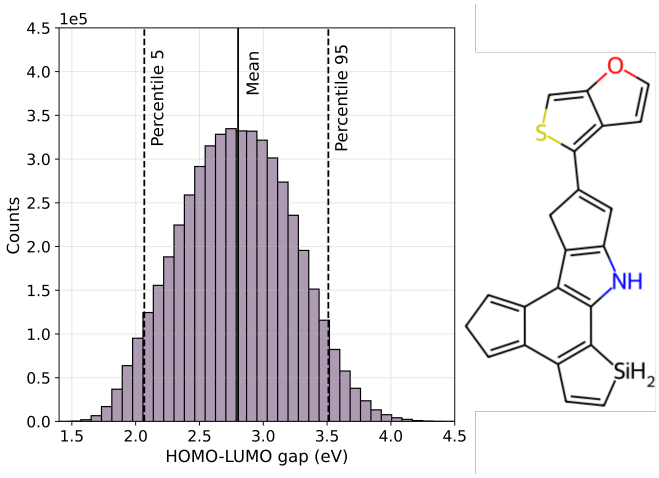

Fig. 2 Left: Distribution of the HOMO-LUMO energy gap (eV). Right: Sample molecule from the dataset rendered with RDKIT[28].

2.1 Generative model: conditional hierarchical variational autoencoder

Generative models such as the Variational Auto Encoder (VAE) learn a compact representation of the dataset by forcing a bottleneck of information between a pair of Encoder-Decoder networks[29]. The space where the learned representations lie is referred to as the *latent space*, and a sample from this space is referred to as *latent vector*. The latent vector is a representation of a molecule, and a molecular graph can be constructed from the latent vector using the decoder network. By randomly sampling a latent vector from the latent space and using the decoder network to construct a molecule, one can generate structures unseen in the training set, but that resemble it.

We employ a recently proposed VAE architecture[30] for molecules based on the hierarchical nature[31] of molecular graphs. The VAE works on different levels of resolution of the molecular structure, from *fragments* to individual atoms and atomic bonds. Working at different levels of structure resolution of the molecules helps the model scale better to larger structures. Another feature of the proposed model is the autoregressive decoder, which, starting from an initial *fragment*, builds the whole molecule by sequentially attaching fragments to a growing molecular structure instead of generating the molecule in a single step.

However, if we sample randomly from the latent space, we cannot control the properties of the resulting molecules. As discussed in the introduction, we need to pre-select candidate materials likely to have the desired properties. This requires having control over the generation process.

An autoencoder approximates the probability distribution

$p(x)$ over the sample space $x \in X$ as a latent variable model $p(x|z)p(z)dz$ with $z \in Z$ the latent representation defined over the latent space. With a *conditional* autoencoder, we directly learn the conditional distribution $p(x|c)$ of the sample x with condition c, where c is a property of x.

During training, we include the desired property to control generation by concatenating it in the decoder input. The encoder network maps the molecular graph (G) into the d-dimensional latent vector (z), $f_\theta^{enc}(G) \to z \in \mathbb{R}^d$, and the decoder maps the concatenated latent vector (z) with condition (c) into the molecular graph $f_\phi^{dec}([z;c]) \to G$ where $[z;c] \in \mathbb{R}^{d+1}$. First, we sample a latent vector from the latent space $z \sim \mathcal{N}^{(d)}(\mu, \sigma)$. Then we concatenate it with some desired value of condition $[z;c]$ and create the conditional input for the decoder. The decoding process is illustrated schematically in Fig. 3.

2.2 Surrogate molecular property predictor

To filter the pre-selected candidates, we employ a DL-based surrogate property predictor. The choice of model is motivated by recent work that investigated the equivalence of molecular graph convolutions and molecular wave functions with poor basis sets[32] and the proposed data-driven molecular property prediction method *Quantum Deep Field* (QDF)[33].

The model uses DL to learn implicit nonlinear complex functions in predicting the HOMO-LUMO energy gap. It is physics-inspired instead of relying solely on end-to-end learned mapping from the molecular graph. This hybrid approach achieves better generalization and competitive accuracy while using fewer parameters. Due to its characteristics, it was chosen as the surrogate model in the filtering step to enhance robustness and generalization in the search for novel structures.

2.3 Density Functional Theory calculations (DFT)

The last stage of the proposed funnel approach is based on computationally expensive but accurate DFT calculations[23] performed with the ORCA quantum chemistry program package[34], version 5.0.1. The DFT-predicted HOMO-LUMO energy gap depends on the selected level of theory, which depends on the chosen functional and basis set. For the predicted HOMO-LUMO gap values computed using DFT to be consistent with those of the generator and surrogate filtering steps, trained on the dataset[27], the same choice of functional and basis sets is made as in the original CEP dataset. That is, the B3LYP hybrid functional[35,36] with the Ahlrichs def2 valence triple-zeta polarization basis set (def2-TZVP keyword in ORCA)[37]. We used Orca 5.0.1 default values for the SCF convergence.

2.4 Funnel pipeline

The three building blocks are combined to form the HTCS workflow funnel. Starting from a molecular library (from the generative model) candidate molecules are pruned into a final curated set. Fig. 4 illustrates the computation steps and data type transformations.

First, the *generative* model creates an initial collection of molecules, based on connectivity maps and without information on the 3D structure. This collection is designed based on a specific HOMO-LUMO energy gap value. Given some target HOMO-LUMO gap value, it outputs a collection of molecules represented by the *Simplified Molecular Input Line Entry System* format (SMILES)[38].

The surrogate model predicts the HOMO-LUMO gap of the initial collection of molecules, for which it requires the molecular 3D structure. This 3D structure is computed using OpenBabel[39] fragment-based coordinate generation[40]; OpenBabel transforms the SMILES strings into XYZ format. Next, the geometry of the generated structures is relaxed using the Universal Force Field (UFF)[41].

We predict the surrogate QDF model to estimate the HOMO-LUMO gap for all molecules in the initial collection using their 3D molecular structure. Molecules whose absolute predicted HOMO-LUMO gap exceeds a predefined threshold are filtered out. The remaining selected molecules are then further curated by DFT simulations.

In summary, given some design conditions, the funnel selects molecules generated by the conditional generator that pass consecutively through the surrogate and DFT filters, as shown in Fig. 1.

3 Results & Discussion

3.1 Training the generator

The generator was trained on approximately 40000 samples from the CEP dataset. To regularize the model and avoid overfitting, the model was trained with a dropout[42] probability on the encoder and decoder of p=0.4. The weight of the KL divergence term[43] was set to be constant at $\beta = 1.0$ and the latent space dimension chosen was $dim_z = 30$, with the rest of the hyperparameters matching those of the original implementation. The total loss and the KL-divergence term of the loss were monitored for each epoch of the training set and the validation set and its evolution through training is shown in Fig. 5. The metrics for the training set are shown in blue, and for the validation set they are shown in orange. The values at the last epoch of training are then shown as a dashed gray line.

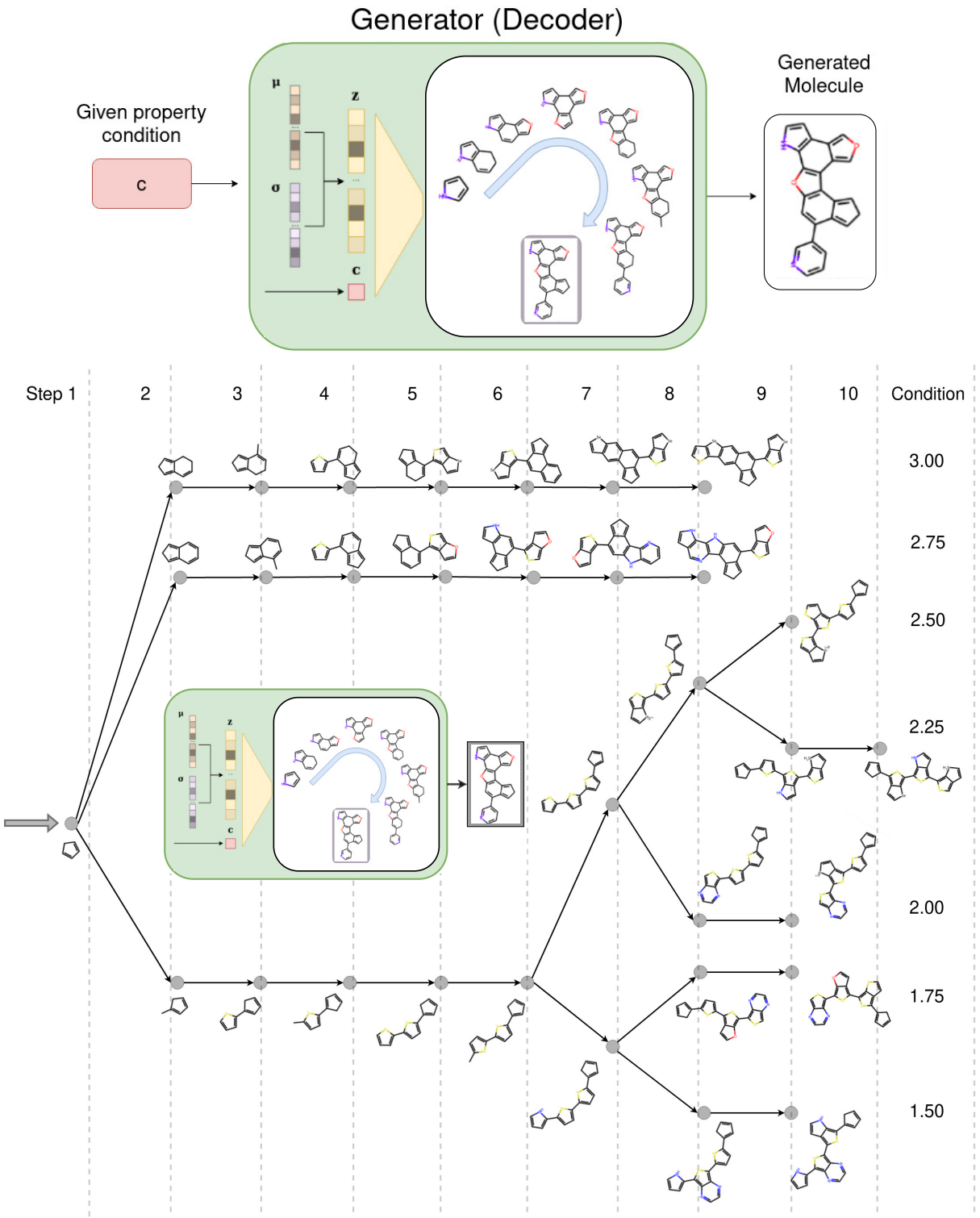

Fig. 3 Schema of the conditional decoder. Given the condition c in the red box, first a latent vector $z \sim \mathcal{N}^{(d)}(\mu, \sigma)$ is sampled, then it is concatenated with the condition $z_{ext} = [z; c]$ to form the input for the auto-regressive decoder. In the Generator (Decoder) box, we display the steps taken in growing the molecular graph into its final form, which is the molecule that is then returned as output. The lower part of the Figure shows how the autoregressive decoder evolves the graph into different structures that come from a common source tree by using the same latent vector z and altering only the concatenated condition. For all conditions, we start from the same motif (step 1), because we start from the same latent vector, then as a result of the different design conditions the molecular graph in the branches grows different motifs.

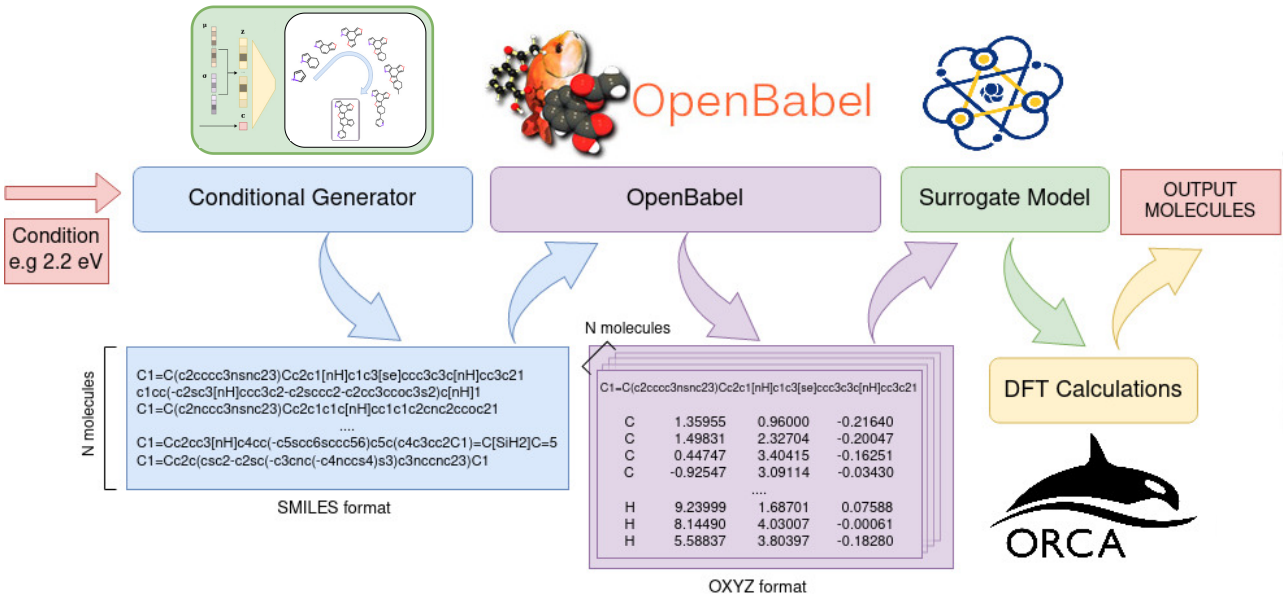

Fig. 4 Schema of the funnel process. First, the Generative Model creates an initial collection of molecules that have been generated conditioned to the target property; the molecules are described as a graph (subsequently converted to SMILES strings). Second, we optimize the structure of the selected molecules with classical force fields and predict their HOMO-LUMO gap values using the surrogate model. The generated molecules are further filtered by setting a threshold on the deviation from the target value. Third, since the number of candidates has been narrowed to a smaller amount, the collection is further curated using single-point DFT into the final candidate structures that more closely satisfy the target condition. The building blocks are described sequentially.

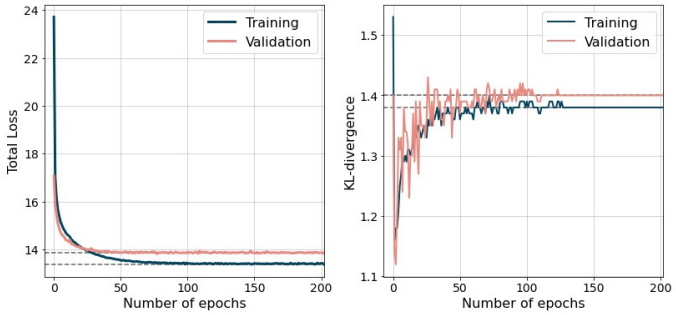

Fig. 5 Left: Evolution of the total loss of the generator during training on the training set (blue) and validation set (orange). Right: Evolution of the KL-divergence term of the loss during training. The metrics on the training set are shown in blue and for the validation set they are shown in orange. The gray dashed lines mark the values of the losses/kl-term in the final epoch.

3.2 Training the surrogate

Basis set	6-31G	# Units/layer Functional	500
Radius	0.75	# Layers Functional	6
Grid Interval	0.3	# Units/layer HK	500
# Epochs	150	# Layers HK	6
Batch size	8	Learning rate	1e-4
LR decay	0.5	Operation	Mean

Table 1 Parameters used in processing the sample molecules and Hyper-parameters of the QDF model used in training.

The surrogate model was trained on the same structures as the generator. The hyperparameters used are shown in *table* 1. The difference in the choice of hyperparameters from the original implementation is doubling the number of neurons per linear layer and doubling the number of linear layers in both the Functional map and the Hohenberg-Kohn map, the two nonlinearities approximated using neural networks.

In Fig. 6 the surrogate model is evaluated on the validation partition of the dataset. For every sample in the validation set, the predicted HOMO-LUMO gap is computed and plotted against the true value from the DFT baseline, resulting in the scatter plot in Fig. 6 (left). The sample points lie along the ideal correlation line (black) and are distributed tightly and symmetrically along it, indicating good accuracy and absence of systematic bias in error. In Fig. 6 (right) the distribution of the prediction error is shown, the mean absolute error (MAE) calculated for the validation set is 0.070 eV (HOMO), 0.077 eV (LUMO), 0.115 eV (HOMO-LUMO).

3.3 Funnel pipeline validation

To substantiate the efficacy of the screening pipeline, we need to validate each of the individual blocks: generator, surrogate model, and DFT calculations. The objective is to study how each of the blocks behaves when assembled together (and alone) and to report their limitations in terms of the range of condition values where the funnel is operational.

For example, the standard procedure to operate the funnel is first to provide the desired HOMO-LUMO gap, such as 2.8

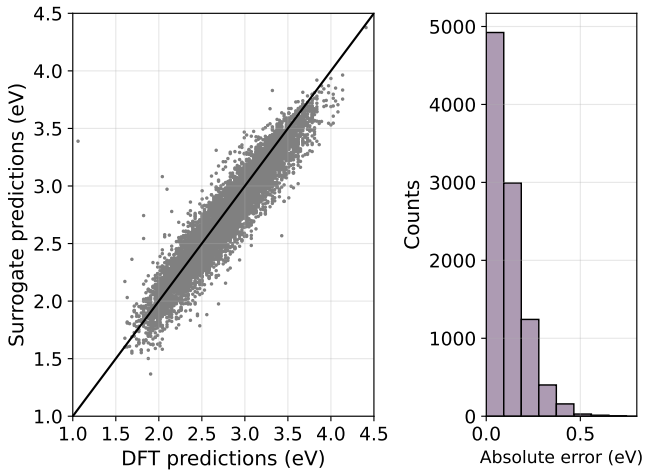

Fig. 6 Left: Scatter plot of sample molecules from the validation set, predicted vs. correct HOMO-LUMO gap (eV). Right: Distribution of the HOMO-LUMO gap error. The Mean Absolute Error (MAE) computed is 0.101 eV.

eV; using this condition, the generator creates an initial set of molecules (e.g., 100 structures). Then, using the HOMO-LUMO predictions from the surrogate model, this set is reduced, e.g., to 20 molecules. As the final filtering step, the funnel calculates the HOMO-LUMO gap directly with DFT for these 20 molecules. Based on the predictions, the funnel further narrows down the number of candidates from 20 structures to, for example, 6 structures. These 6 structures are then the output of the funnel.

It is worth noting that some molecules returned by the funnel can be present in the CEP dataset, but not in the subset used in training. It can be said that these molecules are novel because the models did not see them during training. However, to highlight the potential of our approach in generating novel molecules outside of databases, we consider a molecule novel if it is not present in the CEP dataset, which is ∼100 times bigger than the training dataset.

To determine the range of values for which the funnel is operational, we repeat the process for different HOMO-LUMO gap values. This example pertains to a specific condition value of 2.8 eV. The distribution of the HOMO-LUMO gap of the dataset used can be seen in Fig. 2. The range of values span approximately from 1.6 to 4.1 eV with a mean at 2.8 eV while 5 and 95 percentiles annotated at 2.1 and 3.5 eV respectively. To assess the capacity of the funnel to conditionally generate samples for given HOMO-LUMO gap values, we query it for different HOMO-LUMO gap values spanning from 1.0 eV to 5.5 eV. This range of values allows exploring the behavior of the funnel for HOMO-LUMO gap values *in-distribution* and *out-of-distribution* since the range (1.0-5.5eV) exceeds the range of values in the dataset (∼ 1.6-4.1 eV).

The funnel is queried for HOMO-LUMO gap values ranging from 1.0-5.5 eV in steps of 0.1 eV. For each of these values, the generator creates 100 initial structures and then prunes them into curated lists. A total of 4600 structures were generated to cover 46 different HOMO-LUMO gap conditions. Some of these 4600 structures are not *novel*, i.e. they are present in the CEP dataset, and we consider their tabulated HOMO-LUMO gap value in the dataset as the reference value for validation metrics.

3.3.1 Validating the generator

As stated, some of the 4600 structures initially generated are not *novel* and we can use their HOMO-LUMO gap value as it appears in the dataset as the ground truth for validation metrics. *Novel* molecules cannot be validated by this approach, as they do not appear in the dataset, and they will be thoroughly tested by different means. To validate the performance of the generator, we compare the HOMO-LUMO gap given as input and the HOMO-LUMO gap value from the dataset, that is, we compare the *prompted condition* with a DFT calculation.

There is no filtering at this point, so these structures are the ones purely created by the conditional generator. Ideally, the *prompted condition* would match exactly the DFT-calculated HOMO-LUMO gap; in practice, we can expect some spread around the target condition. We are interested in evaluating this spread and its dependence on the *prompted condition*.

In Fig. 7 the *prompted condition* is shown versus the DFT calculation for the generated structures, together with a histogram of the absolute errors defined as $abs(prompted - predicted)$. On the scatter plot, the ideal correlation line is superposed as the black diagonal line. To aid in visualization, the mean of the DFT calculated values for every *prompted condition* is shown as the blue line. Finally, the vertical black dashed lines represent the percentile 5 and 95 of the HOMO-LUMO gap values in the dataset, as referenced in Fig. 2.

The blue line, representing the mean of the DFT calculated values, and the black diagonal line, representing the ideal correlation, overlap in the range (2.0-3.9 eV), coinciding approximately with the percentile 5 to percentile 95 range of values in the dataset. This shows that the generator can consistently generate structures with certain HOMO-LUMO gap values *in-distribution*. The overlap of the blue and black line is present also beyond percentiles 5 and 95, so the generator model is capable of consistent and conditional generation even in the boundary of *in-distribution* values, see the extremes of the HOMO-LUMO gap distribution shown in Fig. 2.

When prompted for HOMO-LUMO gap conditions beyond those observed in the training dataset, the generator model becomes inconsistent. It generates structures with DFT predicted gaps that are centered on the same value regardless of the prompted condition. This effect starts to occur near

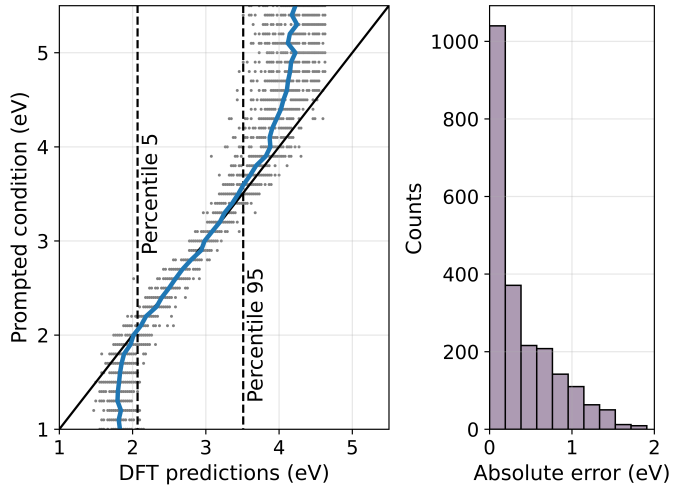

Fig. 7 Validation of the generator model for *in-* and *out-of-distribution* prompted HOMO-LUMO conditions. Left: *Prompted condition* is shown versus the DFT calculated value. The black diagonal line represents the ideal correlation. The mean of the DFT calculated values for every distinct *prompted condition* is shonw in blue. The vertical lines represent the percentiles 5, 95 of the values in the CEP, in connection with Fig. 2. Right: Histogram of the absolute errors, defined as $abs(condition.prompted - DFT.Predicted)$.

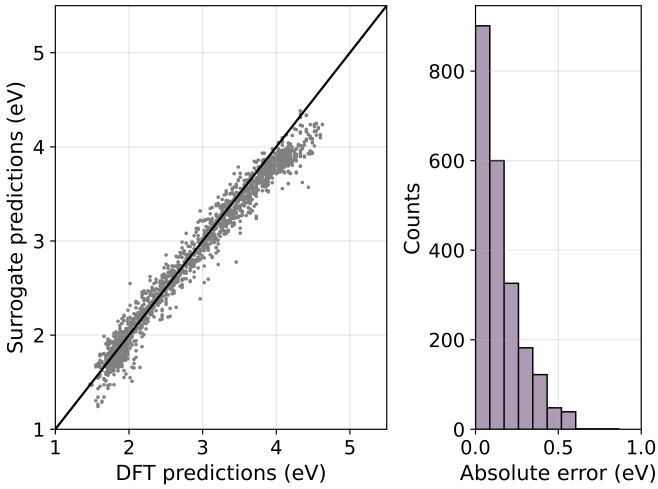

Fig. 8 Left: Surrogate vs DFT HOMO-LUMO energy gap (eV) for the generated molecules. The black diagonal line represents the ideal correlation. Right: Distribution of the absolute errors $abs(Surogate.Predicted - DFT.Predicted)$. The Mean Absolute Error (MAE) value is 0.149 eV.

the distribution edges. The generator model can still produce candidates with the desired condition slightly beyond the distribution boundary. But such generation is inefficient as the mean of the predictions departs from the ideal correlation and only the tails of the distribution will obey the prompted condition.

3.3.2 Validating the surrogate

Using the 4600 structures generated, the surrogate HOMO-LUMO gap predictor can be validated by comparing its predictions for the HOMO-LUMO versus the DFT calculated values from the dataset.

This validation is different from the one performed when training the surrogate model. We are evaluating the performance of the surrogate model on sample structures created by the generator, so the distribution of samples is different. Notably, it has more relative representation of structures in the tails of the HOMO-LUMO gap distribution of the dataset, for which the surrogate model is more likely to have worse accuracy.

In Fig. 8 the surrogate model predictions of the HOMO-LUMO gap are shown versus the DFT calculated values. A histogram of the absolute errors defined as $abs(Surrogate.Predicted - DFT.Predicted)$ are also presented. On the scatter plot, the ideal correlation line is superposed as the black diagonal line.

The scatter of the surrogate predictions appears to be symmetric and clustered alongside the ideal correlation line, indicating good calibration of the surrogate model. We note, however, that there is a slight tendency for the surrogate model to underestimate the HOMO-LUMO for molecules on the higher-end gap values. This effect is more apparent in this validation than in the one performed during training (s. Fig. 6), due to the higher relative representation of structures in the tails of the HOMO-LUMO gap distribution. This effect also explains the increase in the MAE reported for validation, which during training was 0.101 eV and in this validation it is reported as 0.149 eV. In turn, the newly reported MAE provides a more reliable accuracy metric for the surrogate model in the context of the funnel pipeline. It serves as the accuracy threshold for the filtering task performed by the surrogate: Samples whose surrogate-predicted HOMO-LUMO gap is further than 0.149 eV from the target value are filtered out in the funnel.

3.3.3 Validating DFT calculations from UFF minimized structures

The final step in the funnel performs DFT single-point calculations from the UFF minimized structures to obtain the HOMO-LUMO gap value. These calculations would only be performed on the samples that have been prefiltered by the surrogate model, so they would not need to be performed on all the 4600 structures generated.

However, there is a need to assess whether the DFT calculations from UFF minimized structures return HOMO-LUMO gap values consistent with those in the dataset. To perform this check, we performed single-point DFT calculations from the UFF minimized structures of all the 4600 generated structures.

For samples generated that are not *novel*, we can use their DFT-computed HOMO-LUMO gap value in the dataset as

reference values. Then we plot the single-point DFT HOMO-LUMO as a function of the *Dataset DFT HOMO-LUMO*. In Fig.9 the *single-point DFT HOMO-LUMO* values are shown versus the *Dataset DFT HOMO-LUMO* reference values, together with a histogram of the absolute errors defined as $abs(Single.Point.DFT.Predicted - DFT.Predicted)$. On the scatter plot, the ideal correlation line is superposed as the black diagonal line.

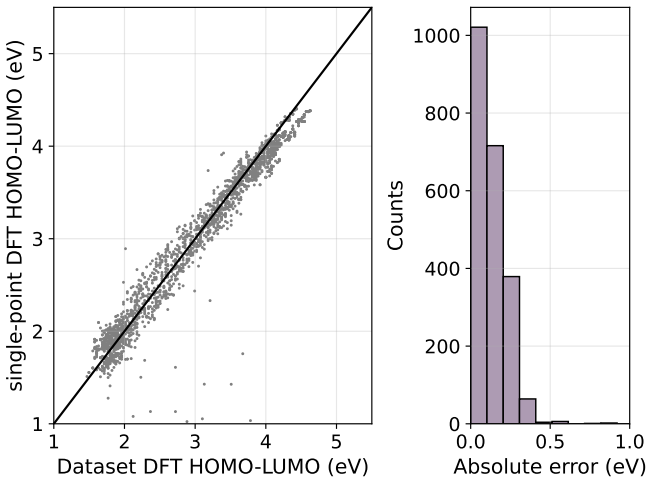

Fig. 9 Left: Single-Point DFT (based on UFF optimized structure) computed HOMO-LUMO gaps vs. reference values in the CEP dataset. The distribution of errors is tightly clustered alongside the ideal correlation line and is symmetric alongside it. The MAE computed is 0.141 eV. This value of the MAE is then used as a threshold on the DFT filter: Molecules whose DFT-calculated HOMO-LUMO gap is within 0.141 eV of the target value will pass the filter. Right: Histogram of the distribution of the absolute errors on the predictions.

The scatter of the single-point versus the DFT HOMO-LUMO gap (from the dataset) appears to be symmetric and clustered alongside the ideal correlation line. This indicates that we can trust the single-point DFT calculations from UFF minimized structures when we perform them on *novel* molecules, and it provides an accuracy metric for these calculations with respect to the values in the dataset, the mean absolute error of 0.141 eV, which is then used as the threshold for the DFT predictions filter.

Using the MAE value of 0.141 eV as the threshold for the DFT calculation-based filter, during normal operation of the funnel any molecule whose (single-point) DFT-predicted property is further than 0.141 eV from the *prompted condition* is filtered out.

3.4 Discovery of novel molecules with desired HOMO-LUMO gap values

The funnel approach filters structures in sequential steps into a final curated list of candidates. See the detailed procedure below:

1. Given some *prompted condition* value of the HOMO-LUMO gap, the generator creates M molecules (for example, M = 100).
2. The surrogate model predicts the HOMO-LUMO gap of the M molecules, then filters out the molecules whose predicted HOMO-LUMO gap value is more than 0.149 eV away from the *prompted condition* value. Narrowing down the set of candidates from M to N (e.g., from M=100 to N=20)
3. Single-point DFT calculations are then performed on the N molecules to obtain the DFT-calculated HOMO-LUMO gap values. Then any molecule whose DFT-predicted HOMO-LUMO gap value is greater than 0.141 eV away from the *prompted condition* value is filtered out. Narrowing down the set of candidates from N to N_{out} (for example, from N = 20 to N_{out} = 6). The final N_{out} candidates are the output of the funnel.

By applying the described procedure for all the 4600 structures generated in the aforementioned validation study, we can demonstrate the power of our data driven HTCS funnel in discovering *novel* structures with desired properties, i.e. a desired HOMO-LUMO gap. Here, we demonstrate the discovery of novel structures with desired HOMO-LUMO gap for a range of 46 distinct HOMO-LUMO gaps (1.0-5.5 eV range in steps of 0.1 eV).

The initial set of 4600 structures is first narrowed down to 1056 using the HOMO-LUMO predicted by the surrogate model; then these 1056 structures are narrowed down to 497 based on the HOMO-LUMO gap from DFT calculations. Of the final 496 structures produced by the funnel, 83 were *novel* (16.73%). We note that the novelty discovery ratio depends heavily on the number of initial molecules created for each of the *prompted condition* and the *prompted condition* itself. As discussed in *Validating the generator* (see Fig. 7) the generator struggles to conditionally generate structures for *out-of-distribution* HOMO-LUMO gap values, so naturally the funnel will not be able to discover many *novel* molecules for those values.

To visualize the HOMO-LUMO gap values of the *novel* structures generated by the funnel, in Fig. 10 the DFT calculated value of the HOMO-LUMO gap is shown versus the *prompted condition* for all *novel* molecules returned by the funnel as a scatter plot. The diagonal black line represents the ideal correlation, where the *prompted condition* (the desired HOMO-LUMO gap) is equal to the DFT-computed HOMO-LUMO gap of the generated structure. On the top Fig., a bar plot is shown with the number of *novel* molecules returned by the funnel for the different *prompted conditions*. Vertical dashed black lines represent the percentiles 5, 95 of the distribution of HOMO-LUMO gap values in the dataset, with reference to Figs. 2, 7.

Fig. 10 demonstrates that the proposed funnel approach is capable of discovering *novel* molecules for desired values of the HOMO-LUMO gap for the range of HOMO-LUMO gap values *in-distribution*. In other words, given any target HOMO-LUMO gap within the range of values in the CEP dataset[27], the funnel is capable of returning *novel* molecules with the desired HOMO-LUMO value.

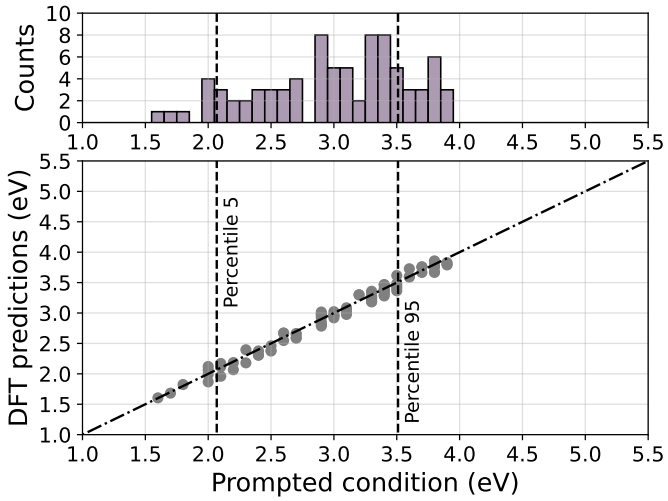

Fig. 10 *Novel* molecules. DFT calculated value of the HOMO-LUMO gap versus the *prompted condition* for all *novel* molecules returned by the Funnel. The diagonal black line represents the ideal correlation, where the *prompted condition* (the desired HOMO-LUMO gap) is equal to the DFT-computed HOMO-LUMO gap of the generated structure. On the top figure, a bar plot is shown with the number of *novel* molecules returned by the funnel for the different *prompted conditions*. The vertical dashed black lines represent the percentiles 5, 95 of the HOMO-LUMO gap values distribution in the dataset, in reference to figures 2, 7

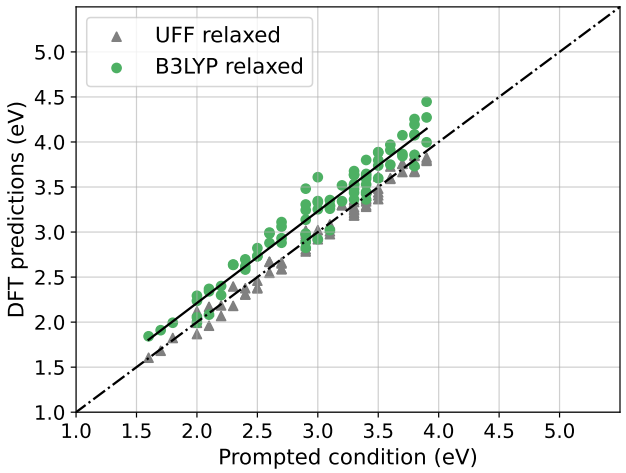

Fig. 11 Comparison of HOMO-LUMO gap for UFF relaxed and B3LYP relaxed structures for the 83 *novel* molecules discovered, as a function of the *prompted condition*. The green dots represent the values calculated using B3LYP relaxed geometries, the gray dots represent the results from the UFF relaxed geometries. A linear fit is applied to the results from the B3LYP relaxed geometries, resulting in the linear fit shown in the black continuous line with a determination coefficient of 0.943, slope coefficient of 1.018, and intercept of 0.175 eV.

3.4.1 Analysis of the *novel* molecules

In this work, we present and describe a novel approach to material discovery. To demonstrate its potential to discover *novel* molecules with desired properties, the funnel was used for 46 different values of the HOMO-LUMO gap spanning the range of values 1.0-5.5 eV in steps of 0.1 eV, finding a total of 83 *novel* structures. The objective of this section is to validate these *novel* structures in terms of stability and synthetic accessibility.

To assess the stability of *novel* structures, geometry optimization is performed using DFT for the spin multiplicity of singlet, triplet, and quintet. We exclude duplets and quartets because of the number of electrons in the molecules. For the 83 molecules, the ground state configuration found was with singlet spin-multiplicity. The predictions of the HOMO-LUMO gap by UFF relaxed B3LYP single point calculations (output of the funnel) and B3LYP relaxed values are compared to the *prompted condition* (input of the funnel) in Fig. 11.

Furthermore, in Fig. 11, HOMO-LUMO gap calculated for the B3LYP relaxed (green circle) and UFF relaxed (gray triangle) geometries are shown as a function of the *prompted condition* as a scatter plot. Ideal correlation is represented by the black dash-dotted diagonal line. A linear fit is made to the data from the B3LYP relaxed calculations and is shown as the solid black line. The coefficient of determination of the fit is R^2 =0.943, the slope coefficient is 1.018 and the intercept is 0.175 eV.

In Fig. 11 the linear fit indicate a small and systematic error between the values of the HOMO-LUMO gap from the B3LYP relaxed geometry and the *prompted condition*. It can be quantified by the value of the intercept of the linear fit at 0.175 eV. This means that our funnel results (UFF relaxed) are a good approximation to the B3LYP relaxed ones.

In order to assess the synthetizability of the *novel* molecules we use the Synthetic Accessibility Score (SAscore) [44]. SAscore is a method for estimating the ease of synthesis of molecules based on a combination of fragment contributions and a complexity penalty. The contributions of the fragments are calculated from an analysis of one million representative molecules from PubChem, capturing historical synthesis knowledge. The SAscore ranges from 1 (easy synthesis) to 10 (hard synthesis).

The objective of the analysis of the ease of synthesis for the *novel* molecules is twofold: (a) assess whether there is any pattern to the ease of synthesis metric of the structures found; (b) compare the SAscore of the *novel* structures with the SAscore distribution of the CEP dataset, to determine if there is any particular bias in the ease of synthesis for the *novel* molecules with respect to the dataset.

The SAscore is calculated for all samples in the CEP dataset and the *novel* molecules with the distributions shown in the histograms in Fig. 12. The green histogram corresponds to the SAscores distribution for *novel* molecules, the gray histogram corresponds to the SAscores distribution for molecules in the CEP dataset. For easier visualization, the histograms are normalized dividing the raw counts from each bin by the total number of counts and the bin width, so the area under the histograms is equal to one.

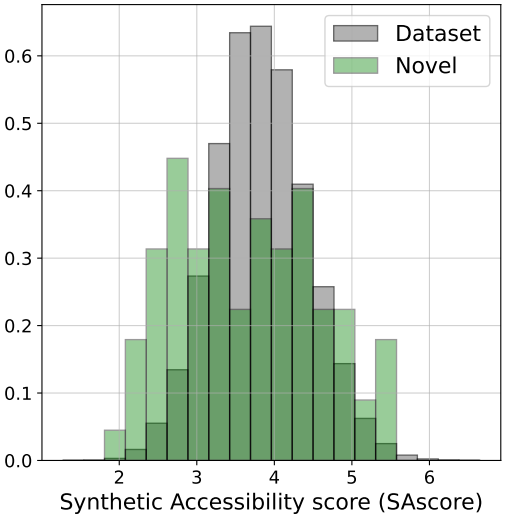

Fig. 12 Comparison of the synthetic accessibility score (SAscore)[44] for molecules in the CEP dataset[27] and *novel* molecules discovered by the proposed Funnel. The histogram bins for both the dataset and *novel* molecules are normalized by dividing the raw count of each bin by the total number of counts and the bin width.

The range of SAscore values for the set of *novel* molecules ranges from 2-5.5 corresponding to relatively easy to synthesize molecules (1 is easy to synthesize, 10 is hard). Within this range, the distribution is not symmetric, with a bias towards lower SAscore values, i.e., easier to synthesize. This is apparent when the two distributions are compared to each other. The range of values for the two distributions are approximately equal, but the set of *novel* molecules discovered with our workflow are slightly easier to synthesize than those from the CEP dataset.

4 Conclusions

We demonstrate that the proposed HTCS 2.0 approach that combines generative and supervised ML with physics-based computations can be used to efficiently discover novel solar materials with desired functionality and good synthesizability. The proposed hybrid workflow greatly expands the chemical space that can be searched compared to the traditional screening approach. Two key aspects of our proposed approach enable this:

1. It pre-selects candidates likely to produce good results using a conditional generative model (searching *smartly*)
2. It filters the initial set of candidates into a curated set manageable for expensive simulations like DFT by employing a Deep-learning-based surrogate molecular property predictor (searching *faster*).

The combination of searching *smarter* by pre-selecting meaningful candidates with generative models and searching *faster* by employing machine learning surrogates to narrow down preselected candidates opens the possibility to explore larger chemical spaces.

This approach is broadly applicable in the search for *novel* energy materials and molecules beyond solar OPVs, e.g. catalysts and batteries. The general concept of using data-driven models hierarchically within a HTCS is applicable to all other functional materials and properties. Other types of conditional generative models, or machine learning property predictors, can be used. Functional energy materials like OPVs are typically represented as large molecules or crystal structures. For that reason, learning viable structure space of a large number of atoms is difficult in conjunction with the precise condition of quantum mechanical properties. We overcome this limitation with a generative model that uses the motif-based organization of organic molecules. Cascading the generative model with a regressive model trained with the same data helps with a tight property selection in novel discovered materials. Similar approaches can also be taken for solid-state materials. Graph-based generative and regressive machine learning models similar to the ones utilized in this work can be universally used for all classes of materials.

Code and data availability

The HTCS 2.0 source code will be publicly available upon acceptance. Additionally, a web application was built as a Graphic User Interface to facilitate the use of HCTS 2.0, where the users can interact with the GUI to design OPV molecules subject to some HOMO-LUMO gap condition and visualize them in the browser. The source code for the web application will also be released for public access upon acceptance.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

The authors thank Prof. Alán Aspuru-Guzik for providing the curated dataset used in this work. The authors acknowledge financial support from DTU through the Alliance Ph.D. scholarship, from the Independent Research Fund Denmark (grant number No 0217-00326B) and the EU's Horizon 2020 research and innovation program under grants agreement No 957189 (BIG-MAP).

Notes and references

1 P. Strasser, Q. Fan, M. Devenney, W. H. Weinberg, P. Liu and J. K. Nørskov, *The Journal of Physical Chemistry B*, 2003, **107**, 11013–11021.
2 M. Aykol, S. Kim, V. I. Hegde, D. Snydacker, Z. Lu, S. Hao, S. Kirklin, D. Morgan and C. Wolverton, *Nature communications*, 2016, **7**, 13779.
3 S. Kirklin, B. Meredig and C. Wolverton, *Advanced Energy Materials*, 2013, **3**, 252–262.
4 L. Kahle, A. Marcolongo and N. Marzari, *Energy & Environmental Science*, 2020, **13**, 928–948.
5 A. Bhowmik, M. Berecibar, M. Casas-Cabanas, G. Csanyi, R. Dominko, K. Hermansson, M. R. Palacin, H. S. Stein and T. Vegge, *Advanced Energy Materials*, 2022, **12**, 2102698.
6 S. Sarikurt, T. Kocabaş and C. Sevik, *Journal of Materials Chemistry A*, 2020, **8**, 19674–19683.

7 M. Miyata, T. Ozaki, T. Takeuchi, S. Nishino, M. Inukai and M. Koyano, *Journal of Electronic Materials*, 2018, **47**, 3254–3259.

8 D. Behrendt, S. Banerjee, C. Clark and A. M. Rappe, *Journal of the American Chemical Society*, 2023, **145**, 4730–4735.

9 D. Jin, L. R. Johnson, A. S. Raman, X. Ming, Y. Gao, F. Du, Y. Wei, G. Chen, A. Vojvodic, Y. Gogotsi *et al.*, *The Journal of Physical Chemistry C*, 2020, **124**, 10584–10592.

10 J. Greeley, T. F. Jaramillo, J. Bonde, I. Chorkendorff and J. K. Nørskov, *Nature materials*, 2006, **5**, 909–913.

11 I. E. Castelli, F. Hüser, M. Pandey, H. Li, K. S. Thygesen, B. Seger, A. Jain, K. A. Persson, G. Ceder and K. W. Jacobsen, *Advanced Energy Materials*, 2015, **5**, 1400915.

12 A. K. Singh, K. Mathew, H. L. Zhuang and R. G. Hennig, *The journal of physical chemistry letters*, 2015, **6**, 1087–1098.

13 R. Ma, P. Guo, L. Yang, L. Guo, X. Zhang, M. K. Nazeeruddin and M. Grätzel, *The Journal of Physical Chemistry A*, 2010, **114**, 1973–1979.

14 T. Nakajima and K. Sawada, *The journal of physical chemistry letters*, 2017, **8**, 4826–4831.

15 R. Jacobs, G. Luo and D. Morgan, *Advanced Functional Materials*, 2019, **29**, 1804354.

16 B. Sanchez and A. Aspuru-Guzik, *Science*, 2018, **361**, 360–365.

17 H. Chen, O. Engkvist, Y. Wang, M. Olivecrona and T. Blaschke, *Drug Discovery Today*, 2018, **23**, 1241–1250.

18 B. Liu, H. He, H. Luo, T. Zhang and J. Jiang, *Stroke and Vascular Neurology*, 2019, **4**, svn–2019.

19 J. Arús-Pous, T. Blaschke, S. Ulander, J.-L. Reymond, H. Chen and O. Engkvist, *Journal of cheminformatics*, 2019, **11**, 1–14.

20 A. Bhowmik, I. E. Castelli, J. M. Garcia-Lastra, P. B. Jørgensen, O. Winther and T. Vegge, *Energy Storage Materials*, 2019, **21**, 446–456.

21 D. Diddens, W. A. Appiah, Y. Mabrouk, A. Heuer, T. Vegge and A. Bhowmik, *Advanced Materials Interfaces*, 2022, **9**, 2101734.

22 P. B. Jørgensen, M. N. Schmidt and O. Winther, *Molecular informatics*, 2018, **37**, 1700133.

23 N. Argaman and G. Makov, *American Journal of Physics*, 1998, **68**,.

24 J. Schmidt, M. R. Marques, S. Botti and M. A. Marques, *npj Computational Materials*, 2019, **5**, 83.

25 G. Pilania, C. Wang, X. Jiang, S. Rajasekaran and R. Ramprasad, *Scientific reports*, 2013, **3**, 1–6.

26 G. H. Gu, J. Noh, I. Kim and Y. Jung, *Journal of Materials Chemistry A*, 2019, **7**, 17096–17117.

27 J. Hachmann, R. Olivares-Amaya, S. Atahan-Evrenk, C. Amador-Bedolla, R. Sánchez-Carrera, A. Gold-Parker, L. Vogt, A. Brockway and A. Aspuru-Guzik, *The Journal of Physical Chemistry Letters*, 2011, **2**, 2241–2251.

28 G. Landrum, *RDKit: Open-source cheminformatics*, `https://www.rdkit.org`.

29 R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams and A. Aspuru-Guzik, *ACS central science*, 2018, **4**, 268–276.

30 W. Jin, R. Barzilay and T. Jaakkola, *Hierarchical Generation of Molecular Graphs using Structural Motifs*, 2020, `https://arxiv.org/abs/2002.03230`.

31 W. Jin, R. Barzilay and T. Jaakkola, 2018.

32 M. Tsubaki and T. Mizoguchi, *On the equivalence of molecular graph convolution and molecular wave function with poor basis set*, 2020.

33 M. Tsubaki and T. Mizoguchi, *Physical Review Letters*, 2020, **125**,.

34 F. Neese, F. Wennmohs, U. Becker and C. Riplinger, *The Journal of Chemical Physics*, 2020, **152**, 224108.

35 K. Kim and K. D. Jordan, *The Journal of Physical Chemistry*, 1994, **98**, 10089–10094.

36 P. J. Stephens, F. J. Devlin, C. S. Ashvar, C. F. Chabalowski and M. J. Frisch, *Faraday Discuss.*, 1994, **99**, 103–119.

37 M. Peintinger, D. Vilela Oliveira and T. Bredow, *Journal of computational chemistry*, 2013, **34**,.

38 D. Weininger, *Journal of Chemical Information and Computer Sciences*, 1988, **28**, 31–36.

39 N. O'Boyle, M. Banck, C. James, C. Morley, T. Vandermeersch and G. Hutchison, *Journal of cheminformatics*, 2011, **3**, 33.

40 N. Yoshikawa and G. Hutchison, *Journal of Cheminformatics*, 2019, **11**,.

41 A. K. Rappé, C. Casewit, K. S. Colwell, W. A. Goddard and W. M. Skiff, *Journal of the American Chemical Society*, 1992, **114**, 10024–10035.

42 N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, *Journal of Machine Learning Research*, 2014, **15**, 1929–1958.

43 I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed and A. Lerchner, International Conference on Learning Representations, 2017.

44 P. Ertl and A. Schuffenhauer, *Journal of cheminformatics*, 2009, **1**, 8.

Journal Name

ARTICLE TYPE

Cite this: DOI: 00.0000/xxxxxxxxxx

Supplementary Material: Materials Funnel 2.0 - Data-driven hierarchical search for exploration of vast chemical spaces

Raul Ortega Ochoa,[a] Bardi Benediktsson,[a] Renata Sechi,[a] Peter Bjørn Jørgensen,[a] and Arghya Bhowmik[a*]

Received Date
Accepted Date

DOI: 00.0000/xxxxxxxxxx

1 Supplementary section

Influence of stochasticity in the generation of 3d coordinates on the predictions of the surrogate model.

To transform SMILES[1] to XYZ coordinates, we first create an initial 3D geometry with *OpenBabel*[2,3] followed by optimization of the structure using UFF forcefield. During initial structure guessing, random numbers are used * resulting in slightly different final 3D coordinates each time we do the transformation. Given the stochastic nature of obtaining the 3D structure from the SMILES strings, it is worth studying its effect on the predictions made by the surrogate model.

The surrogate model was trained on a dataset which lacked stochasticity in structures. This is because re-running the structure optimization for every molecule for every epoch would be too time-consuming. To assess the effect of the random initialization of the XYZ coordinates on the surrogate predictions, 32 SMILES are selected from the 4600 generated molecules during the *Funnel Pipeline validation*, covering the range 1.6-4.6 eV of the HOMO-LUMO gap in steps of approximately 0.1 eV. For each of the 32 selected SMILES, *OpenBabel* is used to generate 100 UFF-optimized 3D structures per molecule with different random initializations, resulting in 3200 samples. For each of the 3200 samples, the surrogate model was used to predict the HOMO-LUMO gap.

The surrogate predictions are then aggregated grouping the predicted HOMO-LUMO gap values by molecule, and the mean, maximum, and minimum of the predictions for each of the 32 molecules are calculated. These aggregated results are then shown in the form of violin plots, where the y axis corresponds to the surrogate HOMO-LUMO predictions, and the x axis corresponds to the DFT-computed HOMO-LUMO values from the dataset. The mean, maximum, and minimum of the surrogate predictions are then shown in the molecule's violin plot.

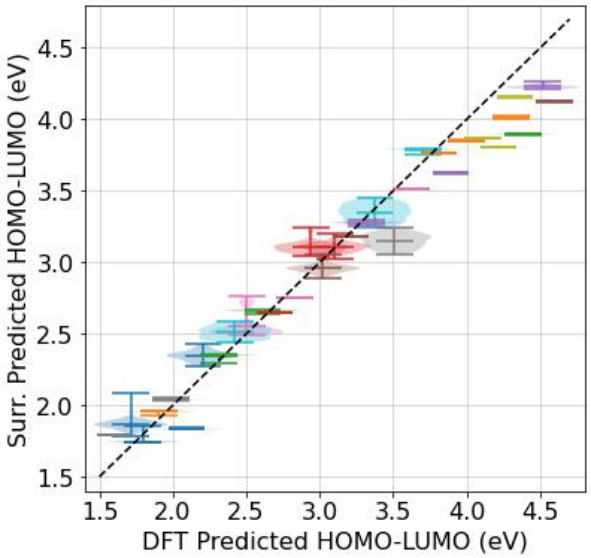

Fig. S1 Surrogate predictions of the HOMO-LUMO gap versus DFT-predicted values for 32 selected molecules from the CEP dataset[4] on 100 different random initialization of the initial 3D structure. The surrogate predictions are grouped by molecule and the mean, maximum, and minimum of the predictions are calculated. The violin plots allow for visualization of the distribution of the prediction for each molecule beyond the aggregated metrics of mean, maximum, and minimum. The black diagonal line represents the ideal correlation of the predictions.

From Fig. S1, the influence of random initialization of the 3D structure results in surrogate predictions that differ at most 0.2

[a]*Technical University of Denmark, DTU Energy, Anker Engelunds Vej, Building 301 2800 Kgs. Lyngby, Denmark, E-mail: arbh@dtu.dk.*

* See discussion in *OpenBabel* `https://github.com/openbabel/openbabel/issues/1934`

eV (min-max range in predictions). This maximum error is of the order of the mean absolute error (MAE) of the predictions of the surrogate model. Hence we can conclude that the stochastic nature of the 3D-coordinate generation affects the predictions, but it is within the accuracy of the model.

Validation of the surrogate model as a binary classifier

The objective of the surrogate model within the HTCS 2.0 is to rapidly pre-filter candidate materials into a smaller, curated set of promising materials that can be then further pruned using more expensive such as DFT into the final output of the HTCS 2.0.

To that purpose, the surrogate model must be able to compute property predictions rapidly and accurately. The accuracy of the surrogate model was thoroughly evaluated in the main work in terms of the distribution of errors in its predictions with respect to the higher-fidelity DFT calculations. In this section, we present an alternative way of quantifying the ability of the surrogate model to act as the prefilter for the more expensive DFT calculations.

The role of the surrogate model can be interpreted as a classification problem with two classes: *promising* and *non-promising* molecules. A *promising* molecule according to the surrogate is one whose predicted property is close to the desired property value, the *prompted condition*. Given some molecule and some *prompted condition* the surrogate model predicts a property of the molecule. If the prediction is close to the *prompted condition* the molecule is classified as *promising*, otherwise, it is classified as *non-promising*. The *closeness* criteria for the classification is some predefined threshold.

By interpreting the task of the surrogate model as a classification problem, we can analyze the number of molecules that are classified correctly or wrongly. The true class of some sample is the class it belongs to in accordance with its DFT value, and the predicted class of some sample is the class it belongs to by its surrogate-predicted value.

An example: Given some desired property value c (*prompted condition*), some threshold δ, some molecule $\mathscr{M}$ whose property value predicted by the surrogate is $\hat{c}_{surr}$ and its predicted value from the DFT calculation in the dataset is $\hat{c}_{dft}$, then for the predictions from the surrogate:

$$abs(c-\hat{c}_{surr}) \rightarrow \begin{cases} \text{predicted } \textit{promising} & if \leq \delta \\ \text{predicted } \textit{non-promising} & if > \delta \end{cases}$$

Similarly, based on the DFT predictions:

$$abs(c-\hat{c}_{dft}) \rightarrow \begin{cases} \text{Actual } \textit{promising} & if \leq \delta \\ \text{Actual } \textit{non-promising} & if > \delta \end{cases}$$

Note that since the predictions from DFT and the Surrogate model are not the same, there are four possible scenarios for the given molecule:

- **True *promising***: If *predicted promising* and *actual promising*.
- **False *promising***: If *predicted promising* and *actual non-promising*.
- **False *non-promising***: If *predicted non-promising* and *actual promising*.
- **True *non-promising***: If *predicted non-promising* and *actual non-promising*.

We are interested in quantifying how many of the candidates that the surrogate classifies fall on each of the four cases because it allows, for example, to obtain a measure of how many good candidates we erroneously discard in the HTCS 2.0 (False *non-promising*). To do so, we can use the data from the *Funnel Pipeline validation*. The data consist of 4600 molecules generated for a range of 46 different conditions spanning the range of the HOMO-LUMO gap from 1.0-5.5 eV. Of the 4600 molecules generated, 2221 were present in the CEP dataset (*not novel* molecules), so the DFT-calculated value of the HOMO-LUMO gap is known from the dataset. This is the value that will be taken as a reference, $\hat{c}_{dft}$.

The 2221 molecules are then divided into the four different cases described. The raw counts and the percentage of the total number of molecules are shown as a confusion matrix in *table* 1. The threshold chosen was $\delta = 0.149$, in accordance with the experiment in *Validating the Surrogate*.

Predicted/Actual	promising	non-promising
promising	628 (**28.27 %**)	128 (**5.76 %**)
non-promising	270 (**12.15 %**)	1195 (**53.80 %**)

Table 1 Confusion matrix on the classification of generated molecules based on the Surrogate model predictions of the HOMO-LUMO gap. The Actual/True values are taken from the DFT-predicted values for the molecules in the dataset.

From the confusion matrix in *table* 1 82.07% of the molecules were correctly classified by the Surrogate filter (diagonal in the confusion matrix) and approximately one-third (34.03% = 28.27%+5.76%) of the total molecules are classified by the filter as *promising*. This demonstrates that employing the Surrogate model to pre-filter molecules reliably (with a low percentage of error) reduces the load on heavier computationally expensive DFT calculations to 1/3 of its original load (without the surrogate filter).

Un-conditional versus Conditional generators for HTCS 2.0

The proposed approach for HTCS 2.0 includes a conditional generator to produce an initial set of candidate molecules subject to some design condition (*prompted* condition) that are then narrowed down in the subsequent blocks of the HTCS 2.0 funnel to form a highly curated set of candidates. One of the key characteristics of the generator used is its capacity to create molecular structures that are biased to have the target property, so that the initial set of produced molecules has a higher population of good

candidate molecules for the design condition than if the initial set were produced by randomly sampling the entire chemical space.

The objective of this section is to illustrate the advantage of employing conditional generators instead of nonconditional generators. We do it by comparing the probability of both types of generators to produce candidate molecules that would be present in the final curated list output of the HTCS 2.0 funnel.

To simplify the problem, we can directly look at the probability of some molecule generated to be within some neighborhood of the *prompted condition* c.

A nonconditional generator learns to approximate the underlying dataset distribution $p(m)$ over the sample space $m \in \mathscr{M}$ of molecules. Sampling a molecule from the learned distribution $m \sim q_\theta(m)$ we should expect for the property of the sampled molecule $\hat{c}$ to follow the distribution in the original dataset. This distribution for the HOMO-LUMO gap resembles a Gaussian distribution $\mathscr{N}(\mu_{dset} = 2.79, \sigma_{dset} = 0.44)$, so we can expect $\hat{c} \sim \mathscr{N}(\mu_{dset}, \sigma_{dset})$. The probability of some molecule to have a property value within the neighborhood δ of the *prompted condition* can be written as $P(|\hat{c} - c| \leq \delta)$. Since $\hat{c} \sim \mathscr{N}(\mu_{dset}, \sigma_{dset})$ we can then write:

$$P_{uncond}(|\hat{c} - c| \leq \delta) = \int_{c-\delta}^{c+\delta} \frac{1}{\sigma_{dset}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu_{dset}}{\sigma_{dset}}\right)^2} dx \qquad (1)$$

The conditional generator learns to sample molecules from the learned distribution while being subjected to some condition. From the analysis in *Validating the Generator*, we showed that for a *prompted condition* c *in-distribution* the conditional generator is well calibrated: For a molecule created subject to a *prompted condition* c we can expect the property value $\hat{c}$ to be following a Gaussian-like distribution centered in c with standard deviation $\sigma_{gen} \approx 0.13$. Then we can write for the conditional generator:

$$P_{cond}(|\hat{c} - c| \leq \delta) = \int_{c-\delta}^{c+\delta} \frac{1}{\sigma_{gen}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-c}{\sigma_{gen}}\right)^2} dx \qquad (2)$$

Using these two expressions for the probability that a generated molecule has a property value within $(c - \delta, c + \delta)$, we can numerically compute probabilities for different conditions c. In Fig. S2 for a condition $c = 2.3$ eV, the probability distribution of the property value for a sample molecule from the conditional (blue) and unconditional (green) model is shown. The area under the curves, delimited by the $(c - \delta, c + \delta)$ values corresponds to the probabilities $p_{cond} = P_{cond}(|\hat{c} - c| \leq \delta)$, and $p_{uncond} = P_{uncond}(|\hat{c} - c| \leq \delta)$.

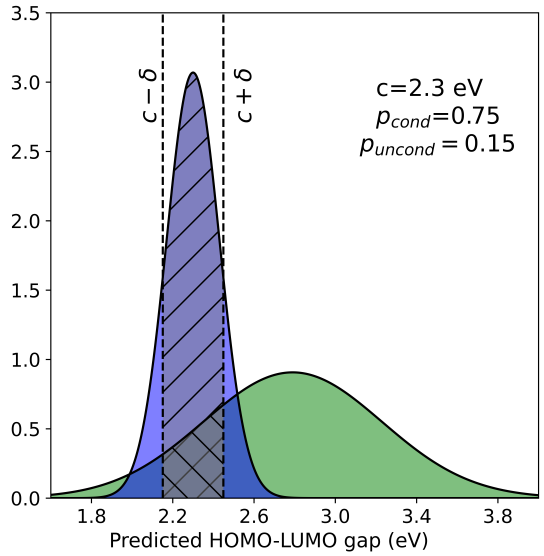

Fig. S2 Probability distribution of the property values for a sample molecule from the conditional (blue) and unconditional (green) model for $c = 2.3$ eV. The area under the curves, delimited by the $(c - \delta, c + \delta)$ values corresponds to the probabilities p_{cond}, and p_{uncond}.

Fig. S2 illustrates the advantage of the conditional model versus the unconditional model. We can then compute the ratio p_{cond}/p_{uncond} for condition values in the range of HOMO-LUMO values from percentile 5-95 (approximately 2.1-3.5 eV) to illustrate the advantage of the conditional model as a function of the condition given. Beyond percentile 5-95, the approximation $\hat{c} \sim \mathscr{N}(c, \sigma_{gen} = 0.13)$ for the conditional generator does not hold. As was shown in *Validating the Generator*, near the boundary of the 5-95 percentiles, the mean of the prediction values departs from the ideal correlation (s. Fig. 7 from the main article). For this reason, we focus the study region in the percentile 5-95 range.

Fig. S3 demonstrates that the probability of success from the conditional model is always greater than the one from the unconditional. The range of ratios between probabilities shown range from the minima at 2.82 to 10, indicating that the conditional model is at worst 280% more efficient than the unconditional, and at its best within the percentiles 5-95 range, 1000% more efficient.

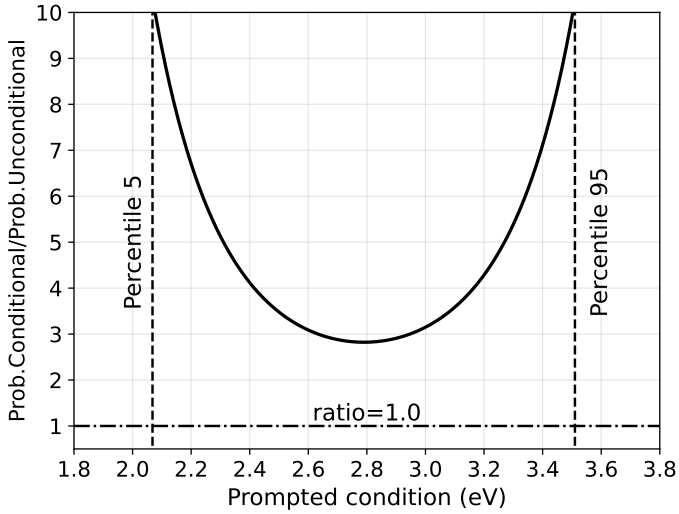

Fig. S3 Ratio between the probability of the conditional and unconditional model to produce a sample molecule whose property lies within $(c-\delta, c+\delta)$. The black curved line corresponds to $P_{cond}(|\hat{c}-c| \leq \delta))/P_{uncond}(|\hat{c}-c| \leq \delta))$ as a function of the *prompted condition* c, the HOMO-LUMO gap. The ratio is computed only for *prompted condition* c values within the percentiles 5-95 of the distribution of values in the dataset. The two dashed vertical lines indicate the percentiles 5, and 95. The dash-dotted horizontal line indicates the ratio value 1.0, where the conditional and unconditional generator would have equal probability. The minimum of the ratio function is ratio=2.82.

Spin multiplicity analysis of *novel* molecules.

id	SMILES	Singlet (Ha)	Triplet (Ha)	Quintet (Ha)
37c5448086154e6905c24e6b135d9f3eb31a0a04	C1=c2ccc3sc4c5oc(-c6ccco6)cc5c5nsnc5c4c3c2=C[SiH2]1	**-2113.61**	-2113.574	-2113.497
b1e95c76ea46696db4dc985482ac5d80914b3819	C1=CCC(c2ccc(-c3cnc(-c4scc5nccnc45)s3)c3cocc23)=C1	**-1881.638**	-1881.597	-1881.533
c5158d9277edf919c9bc15e8f614e1e8ead51287	C1=c2c(c3c([nH]c4cc(-c5cccc6c[nH]cc56)sc43)c3[se]ccc23)=C[SiH2]1	**-4045.063**	-4045.023	-4044.952
c7d2a3d94a7989ea363ff2bbd5c68e84274faa03	C1=Cc2csc(-c3scc4c3[se]c3c5cc6ccccc6cc5ccc43)c2[SiH2]1	**-4409.963**	-4409.898	-4409.82
52a61d8bc82a255fb0a4ae1be48c7af375148cde	C1=CCC(c2cnc(-c3sc(-c4scc5occc45)c4cc[se]c34)c3nsnc23)=C1	**-4680.041**	-4680.008	-4679.923
09d6f0c87a53a0f384fa534adee5d31290ff0f05	c1ccc(-c2[se]cc3c2[se]c2ccc4ccc5ccccc5c4c23)nc1	**-5741.686**	-5741.611	-5741.514
7422901f9feab557f1c44418646a659afe864004	C1=Cc2csc(-c3c4cocc4cc4c3oc3oc5[se]ccc5c34)c2[SiH2]1	**-4082.575**	-4082.515	-4082.422
a4304a4f034eaf203760d2eb25b558788db7207b	C1=Cc2c(csc2-c2sc(-c3sc(-c4nccs4)c4ccsc34)c3nccnc23)C1	**-2999.71**	-2999.684	-2999.595
1d6a6dadfcb9e1f0a8ffe4800e70abb0afb3dec9	C1=Cc2c(c3ccc4occ(-c5cccc6nsnc56)c4c3c3ccncc23)C(c2nccc3nsnc23)C1	**-2352.488**	-2352.412	-2352.34
2ca58b75dbabfae2257f3dbd2a4bb16352f39093	c1cc2cc3c(nc2cn1)oc1c3ccc2nccc(-c3cc4ccsc4s3)c21	**-1918.615**	-1918.535	-1918.438
07031182607a9dab51968858bef30495af60e79b	C1=Cc2csc(-c3c4cocc4cc4c3oc3oc5occc5c34)c2[SiH2]1	**-1756.328**	-1756.27	-1756.177
a1e4873f6840d5969e8913c85bd0f5449dbd2768	C1=c2c(c3c4[se]c(-c5cccc6cocc56)cc4[se]c3c3sccc23)=CC1	**-6159.656**	-6159.604	-6159.54
6ca23738fb6ffcdfc167c7f2da661b412b009755	C1=Cc2csc(-c3c4c(cc5c6cc[nH]c6c6nsnc6c35)=CCC=4)c2C1	**-1806.425**	-1806.393	-1806.299
a73c760e82a0fa90b74b7a1505180f0493e7b7d1	C1=Cc2c(csc2C2=Cc3[se]c4c(sc5ccc6c(c54)=C[SiH2]C=6)c3[SiH2]2)C1	**-4585.068**	-4585.034	-4584.98
96ed5acde42f06cba280569d8b24a550b8fe88c8	C1=Cc2c(-c3cc4ccsc4s3)sc(C3=CC=C4c5c(sc6[se]ccc56)SC4C3)c2[SiH2]1	**-5604.37**	-5604.324	-5604.242
ab7e6bdde5f39a6a65a8d8deea6a695735d0ca92	c1ccc2c(c1)ccc1scc(-c3cncc4nsnc34)c12	**-1613.6**	-1613.531	-1613.436
150449e2ca10ed5353e85cefdcf759b11eb33efa	C1=Cc2c(c3oc4cc(-c5cccc6c[nH]cc56)[se]c4c3c3nsnc23)C1	**-3845.672**	-3845.616	-3845.532
3c17bf295b40fa1a51ad7b5480c778f154fa1e8b	C1=Cc2csc(-c3c4cscc4cc4c3oc3[se]c5sccc5c34)c2[SiH2]1	**-4728.498**	-4728.437	-4728.344
d5947f44656c0af53b5e47e52471e76978ea5f8a	c1cc2c(cn1)c1c[nH]cc1c1cnc3c(-c4cc5ccsc5s4)c[nH]c3c21	**-1860.642**	-1860.558	-1860.456
e38b884ca81d9e5ccbfc399b52804510c7760a68	C1=Cc2csc(-c3cc4cc5c(cnc6ccccnc65)cc4o3)c2C1	**-1390.103**	-1390.028	-1389.926
0c310c805c11292df789d828c0f1b465f8927814	c1ccc(-c2cc3ccccc3c3cc4c(cc23)oc2cnccc24)nc1	**-1107.468**	-1107.381	-1107.271
4d936bdd9c0bf7c8d7934aee1df40edf8a5ee8f8	C1=C(c2scc3[se]ccc23)Cc2c1c1c(c3c4[nH]ccc4c4c[nH]cc4c23)=C[SiH2]C=1	**-4160.538**	-4160.511	-4160.441
a68707583ff08f441cc46a828977c5bb27fd348c	C1=Cc2c(-c3cccs3)sc(-c3c(-c4ccc[se]4)ccc4c[nH]cc34)c2[SiH2]1	**-4389.067**	-4388.996	-4388.919
f48c5c219dc5ff03d51e9a2c8cd5df3b535d1b02	C1=Cc2c(cnc3c2C=C2C(c4ccco4)=CNC2C3)C1	**-878.81**	-878.748	-878.638
0608dca27ce23310c8f114191593988e9ba6f787	C1=Cc2c(csc2-c2cnc(-c3sc(-c4ccco4)c4ccoc34)c3nsnc23)C1	**-2353.809**	-2353.751	-2353.672
d2592089dde92901f48d635d68c14e287717763f	c1ccc(-c2cc3ccc4c5cccnc5ccc4c3[nH]2)nc1	**-934.018**	-933.927	-933.829
1a9cbf38df65acbb706d54a69f6274424cb1ab8d	c1cc(-c2c(-c3scnc3-c3cncc4nsnc34)ccc3c[nH]cc23)c[nH]1	**-1893.895**	-1893.814	-1893.743
6d16614c6a5bc4be540bb184206fc1af5c6a56d0	C1=CC(c2occc2-c2cc3cc[se]c3s2)CC(c2ccc3c(ccc4nccc(-c5cc6nccnc6s5)c43)n2)=C1	**-4799.214**	-4799.151	-4799.056
a8cf110619a27ef5fd358cd1d0f485d9427eb3e9	C1=Cc2c(csc2-c2sc(-c3cncc4nsnc34)c3ccoc23)C1	**-2125.01**	-2124.966	-2124.863
49aa069940e471514ae968e21955d1dcd5ee1e0d	c1ccc2c(c1)ccc1[nH]cc(-c3cncc4nsnc34)c12	**-1270.792**	-1270.727	-1270.627
f3f56f9df3b56b56f55564504f65fcbfd39a746c	C1=C(c2scc3occc23)Cc2c1c1c(c3c2c2c(c4cc[se]c43)=CCC=2)=C[SiH2]C=1	**-4164.309**	-4164.289	-4164.212
0a9a29712af0fa1c0074bb2bf5dc02615172e836	C1=Cc2csc(-c3[nH]cc4c3sc3c5cccnc5ccc43)c2C1	**-1674.935**	-1674.846	-1674.754
5b69debcab20f383091d432db5954f9cacaf0a50	c1cc2c(cn1)c1cscc1c1cnc3[nH]cc(-c4cc5ccsc5s4)c3c21	**-2203.447**	-2203.36	-2203.263
e517b75a17d330c1d68371dae3d2d178b2c9d515	c1cc2cc(-c3nccc4ccsc34)ccc2cn1	**-1123.213**	-1123.123	-1123.01
488d1ff72f6be03e1b4700792c3f96465fe48dc3	c1ccc(-c2nccc3c4oc5occc5c4c4cc5sccc5cc4c23)nc1	**-1579.595**	-1579.507	-1579.424
0bfadbeb4d28675ac1de0ada05c091f8d958d96d	C1=Cc2csc(-c3cc4cc5ccc6ncccc6c5cc4o3)c2C1	**-1374.056**	-1373.98	-1373.88
cb31741b39bd869863e44c127238c18d6aafdd25	C1=Cc2csc(-c3c4cocc4cc4c3[se]c3oc5occc5c34)c2[SiH2]1	**-4082.572**	-4082.514	-4082.421
9e35cac88c6cfab2a4e65ff9dd5b41075014ed80	C1=Cc2c(-c3sccc3-c3sccc3-c3cc4c(s3)[SiH2]C=C4)sc(C3=CC=C4c5cc[se]c5SC4C3)c2C1	**-5874.623**	-5874.548	-5874.501
9fd3b1b689f45d2387c593950b7ffd0de72b1ca5	c1cc2c(cn1)cnc1c2ccc2nccc(-c3cc4ccsc4s3)c21	**-1767.192**	-1767.109	-1767.014
04058245832acce4a0a46f5b080821a541c74a14	C1=Cc2c(cnc3ccc4c(-c5cccc6cscc56)csc4c23)[SiH2]1	**-1948.357**	-1948.279	-1948.209
f2d39ead8928386ccbba494e245657a390133d0e	c1ccc(-c2ccc3cc4c(cc3c2)c2cocc2c2cc[nH]c24)nc1	**-1069.358**	-1069.279	-1069.188
ac3e3296f59d061b87690a391695d040cf340c92	C1=Cc2csc(-c3c4c[nH]cc4cc4c3[se]c3[se]c5[se]ccc5c34)c2[SiH2]1	**-8715.215**	-8715.148	-8715.056
3fb0c2dacc4da4d384487ac6c9685b740e3cdeb9	c1ccc(-c2cccc3c2ccc2c4ccccc4oc32)nc1	**-937.831**	-937.741	-937.621
782179af12ccbece2a906390eb434bb8202ed931	c1cc2c(cn1)cnc1ncc3c4cc(-c5nccs5)cnc4ccc3c12	**-1478.541**	-1478.455	-1478.359
d65f27823c7ecdea4e402f905c6386c91b972dc0	C1=C(c2ccccc2)[SiH2]c2c1sc1c2cnc2ccc3ccccc3c21	**-1627.622**	-1627.564	-1627.451
a2634d68d4d69ffdae7ce1ba719a0521bb7a2a3c	C1=Cc2csc(-c3c4cocc4cc4c3[nH]c3sc5[nH]ccc5c34)c2[SiH2]1	**-2039.571**	-2039.519	-2039.427
9102221874344fa8071112d7dc90f98786c65518	C1=Cc2c(oc3[nH]c4c(-c5scc6c5CC=C6)c5nsnc5cc4c23)C1	**-1804.229**	-1804.171	-1804.072
77a76e46d25ed2836dfc4a1872d0911ab85fe733	C1=Cc2c(-c3ncncn3)sc(-c3c(-c4ccc[nH]4)ccc4ccccc34)c2[SiH2]1	**-1792.431**	-1792.348	-1792.267
10b2f019d06e3a0b7f1a2506d3154eb03739f7cd	c1cc(-c2nccc3nsnc23)c2ccc3c4ccncc4cnc3c2c1	**-1478.536**	-1478.464	-1478.365
5a5c14ca1023f866d0e2965199e70eeae2d1dcfc	C1=c2ccc3cc4[se]c5cc(-c6cccs6)c6ccccc6c5c4cc3c2=CC1	**-3837.821**	-3837.762	-3837.688
ecc292697a2aa20d75469006034b99e23fe852b1	C1=Cc2csc(-c3[se]ccc3-c3[nH]ccc3-c3cccc4c[nH]cc34)c2[SiH2]1	**-4046.257**	-4046.179	-4046.103
ef80545efb1d7ceda6fbcad776d98d5f7f06e5b4	c1ccc(-c2ncnc3c4[nH]c5[se]ccc5c4c4cc5occc5cc4c23)nc1	**-3563.029**	-3562.936	-3562.852
2821a62a151d6c28fff11dad4315b4051a8c3c54	C1=Cc2c(-c3cccc4c3=CCC=4)sc(-c3sccc3-c3nccs3)c2[SiH2]1	**-2385.749**	-2385.709	-2385.634
239975947bc1d49f79401a21a6713b6bf7be6484	C1=Cc2c(cnc3c4c(c5c(c23)C(c2cccs2)=C[SiH2]5)=CCC=4)C1	**-1551.33**	-1551.287	-1551.193
a8b979111c91e0234527a034d3c51ff23e181035	c1cc2ccc3c(ccc4scc(-c5nccc6nsnc56)c43)c2cn1	**-1783.243**	-1783.171	-1783.073
2debb65c2b9d955f9191b79da5357199bd23bf99	c1ccc(-c2[nH]cc3ccc4cc5cnccc5cc4c23)nc1	**-933.996**	-933.917	-933.837
190984cc7ac57bd5c0f5c727a012a90def6fb98d	c1ccc(-c2[nH]cc3ccc4c5ccsc5ccc4c23)nc1	**-1238.707**	-1238.627	-1238.529
e222e28fb5a3ad90964eaa4af90a28d2352a216d	C1=c2ccc3cc4oc5cc(-c6ccccc6)sc5c4cc3c2=CC1	**-1357.988**	-1357.93	-1357.854
b7524000bcdc6f4efdd90a4ea2e17ccb2caad55d	C1=CC(c2nccs2)CC(C2=C(c3scc4c3CC=C4)Cc3[se]c4c([nH]c5c6c(ccc54)=C[SiH2]C=6)c32)=C1	**-4790.818**	-4790.783	-4790.723
604f33c37f6477f9765d89e7498624100b290c4f	c1ccc2c(c1)cc(-c1ccc3cnccc3c1)c1cncc12	**-956.053**	-955.963	-955.859
ca14174f48a96f8aa47b3e2009fa8c9fb6a990fc	c1ccc(-c2cc3c(cnc4c5cccnc5ccc34)[nH]2)nc1	**-950.065**	-949.974	-949.862
a7d4ac489f9b7f580da9137c976f2e286de4e341	C1=c2ccc3[se]c4c5sc(-c6scc7[nH]ccc67)cc5sc4c3c2=C[SiH2]1	**-4708.622**	-4708.589	-4708.526
859f5ceb74a46bfed9ddb2d5def9480437d7ff82	c1ccc(-c2cncc3ccc4ccc5occc5c4c23)nc1	**-953.848**	-953.748	-953.66
bdf318c3c633b5cc835423c89606eb65d8d06cda	c1ccc(-c2cc3c(s2)sc2c4cc5ccccc5cc4ccc32)nc1	**-1735.124**	-1735.057	-1734.971
ae7da14e63e341c6968a4f489029a5e543a6b778	c1ccc(-c2coc3cc4cnccc4cc23)c2cocc2c1	**-935.605**	-935.52	-935.451
edbe27699ed1b2b7a1faf691751af63920a05c1d	c1ccc(-c2cc3nsnc3c3cc4c(cc23)oc2ncccc24)nc1	**-1460.323**	-1460.243	-1460.139
4b263a4ac076ecff73f854b10faf96612722f6bc	C1=Cc2c(c3[nH]c4cc(-c5scc6[nH]ccc56)c5c[nH]cc5c4c3c3nsnc23)C1	**-1954.062**	-1954.006	-1953.918
adf719b366a637a837b44cdf876acd3b2be0599e	Cc1cncc(-c2ccc3cscc3c2-c2scc3c2CC=C3)c1	**-1660.065**	-1659.996	-1659.868
ed8b701ecf44c148c7af507406982e157515ea05	c1ccc(-c2scc3c2[se]c2c4cc5ccccc5cc4ccc32)nc1	**-3738.411**	-3738.346	-3738.259
ae970a4c591158c97ee34d1a624112db181e5aad	c1cc(-c2cncc3cnccc23)c2c(c1)ccc1ccncc12	**-972.083**	-971.992	-971.893
e68ba914eab011a226973a9ae52fb31a98d8ded2	C1=Cc2csc(-c3cc4cc5ccc6sccc6c5cc4o3)c2C1	**-1678.768**	-1678.692	-1678.593
bf33ff51904973a2654f8cabd8f51c30e69f7091	c1ccc2c(-c3ccc4ccncc4n3)cncc2c1	**-818.496**	-818.407	-818.3
3ae9f6262ec9ffd788347072c919b15a87cdce3b	C1=Cc2csc(-c3c4nsnc4cc4c3oc3nc5[se]ccc5cc34)c2C1	**-4204.541**	-4204.481	-4204.374
285208d821ff6672e175eb2f86ad3a7ef02937ae	c1ccc(-c2nccc3c2[se]c2cnc4ccsc4c23)nc1	**-3616.916**	-3616.82	-3616.706
37e50301c602d97862b2922753003be80a477a65	c1ccc(-c2[nH]cc3ccc4c5ccncc5ncc4c23)nc1	**-950.042**	-949.963	-949.86
ba91a07f038251055b51b896ebb386bb603f09e3	C1=Cc2c(-c3ccc(-c4scc5c4C=C[SiH2]5)cc3)sc(-c3cccnc3)c2C1	**-2065.021**	-2064.951	-2064.853
73ca42d448f3dff1931da2d2ff7ab074b21d237a	c1ccc(-c2cncc3ccc4ccc5[se]ccc5c4c23)nc1	**-3280.081**	-3279.995	-3279.898
be214f7666b71c8e4ff01ebe6deafec0b99e0410	C1=CC2C=C[Se]C2C(c2ccc3cnccc3c2)=C1	**-3111.591**	-3111.525	-3111.421
27abef5c4154f60f0c658a413ba008f7053c2abc	C1=C(c2nccc3nsnc23)[SiH2]c2c1c1c(c3c2c2c(c4cc[nH]c43)=C[SiH2]C=2)=CCC=1	**-2119.924**	-2119.9	-2119.839
675f7e4034ed552aed8ea12c32593b17df7f34de	C1=Cc2c(-c3ccc(-c4scc5sccc45)s3)sc(-c3cncc4nsnc34)c2C1	**-2999.708**	-2999.661	-2999.583
6cb2c1d7b7b33ece46c4ed0ebd576593787486d6	c1ccc2c(c1)c1cnccc1c1c3cnccc3ncc21	**-894.699**	-894.609	-894.494
a8303cb28953a6d4c2d58196b510ad724ef8fff2	Cc1cccc2c1ccc1c2ncc2c3c[nH]cc3c3cnccc3c21	**-1049.503**	-1049.412	-1049.314

Notes and references

1 D. Weininger, *Journal of Chemical Information and Computer Sciences*, 1988, **28**, 31–36.

2 N. O'Boyle, M. Banck, C. James, C. Morley, T. Vandermeersch and G. Hutchison, *Journal of cheminformatics*, 2011, **3**, 33.

3 N. Yoshikawa and G. Hutchison, *Journal of Cheminformatics*, 2019, **11**,.

4 J. Hachmann, R. Olivares-Amaya, S. Atahan-Evrenk, C. Amador-Bedolla, R. Sánchez-Carrera, A. Gold-Parker, L. Vogt, A. Brockway and A. Aspuru-Guzik, *The Journal of Physical Chemistry Letters*, 2011, **2**, 2241–2251.