Supporting Information

ChemSpaceAL: An Efficient Active Learning Methodology Applied to Protein-Specific Molecular Generation

Gregory W. Kyro, Anton Morgunov, Rafael I. Brent, Victor S. Batista

Department of Chemistry, Yale University, New Haven, Connecticut 06511-8499

List of Sections

- Section 1: ADMET and Functional Group Filters.
- Section 2: Similarity between FDA-Approved Inhibitors of c-Abl Kinase.
- Section 3: Comparing the Generations to c-Abl Kinase Inhibitors for Different Methods.
- Section 4: Radar Plots Showing Evolution of ADMET Metrics.
- Section 5: Scores of Molecules across Five Iterations of Active Learning for HNH.
- Section 6: Implementation Details of t-Distributed Stochastic Neighbor Embedding (t-SNE).
- Section 7: Choosing the Number of Clusters to Use for k-means.
- Section 8: Details and Parameters Used for Running DiffDock.
- Section 9: t-SNE Visualization of the Evolution of the Generated Molecular Ensembles.
- Section 10: Vocabulary Composition of the Combined Dataset.
- Section 11: Frequencies of Block Sizes, Molecular Weights and Tokens in Pretraining Sets.
- Section 12: Details of the GPT Architecture.
- Section 13: Training the GPT Model.
- Section 14: Pretrained GPT Model Performance on the MOSES Benchmark.
- Section 15: RDKit Descriptors Used to Construct the Chemical Space Proxy.
- Section 16: Wall Times of Each Step in the Complete Pipeline.
- Section 17: Evaluating the Methodology with Lower-Dimensional MQN Filters.
- Section 18: Logits for the End-of-Sequence Token as the Molecular Descriptor Vector.
- Section 19: Frequency as a Function of Cluster Size for Alignment to c-Abl Kinase.
- Section 20: Evaluation of Scoring Function Compared to PDBbind v2020 Refined Set.
- Section 21: Alternative Methods for Converting Mean Cluster Scores to Sampling Fractions.
- Section 22: Distributions of Mean and Median Cluster Scores.
- Section 23: Additional Evaluation of Generations across Active Learning Iterations.
- References

Section 1: ADMET and Functional Group Filters.

Table S1.1. Upper and lower bounds applied to each ADMET metric used for generation filter. We use upper bound for logP of 6.5 because one of the FDA-approved inhibitors of c-Abl kinase, nilotinib, has a value of 6.356 as calculated by RDKit. All other bounds are taken from ADMETlab 2.0.¹

ADMET Property	Lower Bound	Upper Bound
Molecular Weight	100	600
Number of Hydrogen Bond Acceptors	0	12
Number of Hydrogen Bond Donors	0	7
Number of Rotatable Bonds	0	11
Number of Rings	0	6
Number of Heteroatoms	1	15
Formal Charge	-4	4
Topological Polar Surface Area	0	140
LogP	-0.4	6.5

Table S1.2. List of functional groups excluded by generation filter.

- 'fr_azide'
- 'fr isocyan'
- 'fr isothiocyan'
- 'fr_nitro'
- 'fr_nitro_arom'
- 'fr nitro arom nonortho'
- 'fr nitroso'
- 'fr_phos_acid'
- 'fr phos ester'
- 'fr sulfonamd'
- 'fr sulfone'
- 'fr_term_acetylene'
- 'fr thiocyan'
- 'fr prisulfonamd'
- 'fr C S'
- 'fr azo'
- 'fr diazo'
- 'fr epoxide'
- 'fr ester'
- 'fr COO2'
- 'fr Imine'
- 'fr N O'
- 'fr \overline{SH} '
- 'fr aldehyde'

- 'fr dihydropyridine'
- 'fr hdrzine'
- 'fr hdrzone'
- 'fr ketone'
- 'fr_thiophene'
- 'fr_phenol'



Section 2: Similarity between FDA-Approved Inhibitors of C-Abl Kinase.

Figure S2.1. Tanimoto Similarity between RDKit fingerprints of the FDA-approved inhibitors of c-Abl kinase: imatinib, nilotinib, dasatinib, bosutinib, ponatinib, bafetinib, and asciminib.

Section 3: Comparing the Generations to c-Abl Kinase Inhibitors for Different Methods.



M model

imatinib nilotinib dasatinib bosutinib ponatinib bafetinib asciminib

Figure S3.1. Visualizing the evolution of the generated molecular ensemble from the model pretrained on the MOSES dataset with the generation filtered based on ADMET metrics and functional group restrictions, and comparing it to the FDA-approved small-molecule inhibitors of c-Abl kinase. The average Tanimoto similarities between the RDKit fingerprints of all generated molecules at each iteration of the pipeline and each inhibitor are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases



C model Random Selection

imatinib nilotinib dasatinib bosutinib ponatinib bafetinib asciminib

Figure S3.2. Visualizing the evolution of the generated molecular ensemble from the model utilizing random selection (i.e., 1,000 molecules are randomly selected from the generated ensemble and scored, and those that satisfy the score threshold and replicated *N* times where *N* is the smallest integer to achieve a total of 5,000 datapoints to be in the active learning set), pretrained on the combined dataset with the generation filtered based on ADMET metrics and functional group restrictions, and comparing it to the FDA-approved small-molecule inhibitors of c-Abl kinase. The average Tanimoto similarities between the RDKit fingerprints of all generated molecules at each iteration of the pipeline and each inhibitor are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases



imatinib nilotinib dasatinib bosutinib ponatinib bafetinib asciminib

Figure S3.3. Visualizing the evolution of the generated molecular ensemble from the model utilizing random selection with random sampling (i.e., random selection with the addition of randomly sampling 5,000 molecules from the generated ensemble that have not been scored to be in the active learning training set), pretrained on the combined dataset with the generation filtered based on ADMET metrics and functional group restrictions, and comparing it to the FDA-approved small-molecule inhibitors of c-Abl kinase. The average Tanimoto similarities between the RDKit fingerprints of all generated molecules at each iteration of the pipeline and each inhibitor are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



C model Uniform Sampling

imatinib nilotinib dasatinib bosutinib ponatinib bafetinib asciminib

Figure S3.4. Visualizing the evolution of the generated molecular ensemble from the model utilizing uniform sampling (i.e., cluster-based sampling where each cluster is assigned a sampling fraction f = 0.01 to generate the active learning set), pretrained on the combined dataset with the generation filtered based on ADMET metrics and functional group restrictions, and comparing it to the FDA-approved small-molecule inhibitors of c-Abl kinase. The average Tanimoto similarities between the RDKit fingerprints of all generated molecules at each iteration of the pipeline and each inhibitor are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases





Figure S4.1. Radar charts for c-Abl kinase depicting the mean and 95th percentile values for each ADMET metric with respect to the lower and upper bounds enforced for the generated molecular ensembles from the model pretrained on the combined dataset with the generations filtered based on ADMET metrics are shown in (A), the charts for the ensembles from the model pretrained on the combined dataset with the generational group restrictions are shown in (B), and the ensemble for the model pretrained on the MOSES dataset with the generations filtered based on ADMET metrics are shown in (C). Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Figure S4.2. Radar charts for the HNH domain of Cas9 depicting the mean and 95th percentile values for each ADMET metric with respect to the lower and upper bounds enforced for the distribution for the model pretrained on the combined dataset with generation conditioned on ADMET filters are shown in (A), the distribution for the model pretrained on the combined dataset with generation conditioned on ADMET and functional group filters are shown in (B), and the distribution for the model pretrained on the MOSES dataset with generation conditioned on ADMET and functional group filters are shown in (C). Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Section 5: Scores of Molecules across Five Iterations of Active Learning for HNH.

Figure S5.1. Attractive interaction scores of scored molecules across five iterations of active learning for the HNH domain of Cas9. The distribution for the model pretrained on the combined dataset with the generations filtered based on ADMET metrics are shown in (A). The distributions for the model pretrained on the combined dataset with the generations filtered based on ADMET metrics and functional group restrictions are shown in (B). The distributions for the model pretrained on the MOSES dataset with the generations filtered based on ADMET metrics and functional group restrictions are shown in (C). Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.

Section 6: Implementation Details of t-Distributed Stochastic Neighbor Embedding (t-SNE).

To create a standard t-SNE space which involves a constant coordinate system, we proceed as follows. Firstly, we collect scored molecules from all iterations (6,000 molecules). Secondly, we add the molecules from all active learning training sets that employed either softmax or uniform selection methods. Thirdly, we add a random sample of 10,000 molecules from the set of generations at each iteration. We perform this sampling to have the same number of molecules from the active learning training sets and generated sets, which enables us to fairly compute the difference in distributions. Note that our training sets usually contain slightly more than 10,000 molecules, so we sample exactly 10,000 for consistency. After combining all molecules and dropping all duplicates, we perform a t-SNE reduction.

Section 7: Choosing the Number of Clusters to Use for k-means.

For each implementation of k-means, we utilize the scikit-learn Python package,²¹ which employs the k-means++ initialization algorithm, where the first centroid is selected randomly and subsequent centroids are iteratively chosen with a probability proportional to their squared distance from the nearest existing centroid



Figure S7.1. Attractive interaction scores for molecules generated by the pretrained model (iteration 0) and by the model after each of the five iterations of active learning where, prior to sampling for docking, molecules in the chemical space are grouped into 10 clusters. Cluster scores are converted into sampling fractions using the *softsub* approach.



Figure S7.2. Attractive interaction scores for molecules generated by the pretrained model (iteration 0) and by the model after each of the five iterations of active learning where, prior to sampling for docking, molecules in the chemical space are grouped into 100 clusters. Cluster scores are converted into sampling fractions using the *softsub* approach. This figure occurs in the main text (Figure 4C), but is also shown here for comparison.

Table S7.3. Statistics of the distribution of attractive interaction scores, when molecules are clustered into 10 groups and cluster scores are converted into sampling fractions using the *softsub* method.

Iteration	Percent ≥ 11	Q1	Q2	Mean	Q3	Max	Std
0	14.29	8.26	8.91	8.75	10.28	11.25	2.25
1	42.86	8.96	10.27	10.67	11.49	16.09	2.62
2	57.14	10.32	11.51	13.31	15.34	21.00	3.91
3	37.50	6.80	10.45	8.88	12.51	14.94	5.44
4	37.50	7.33	9.10	9.03	11.74	14.12	3.83
5	25.00	7.50	8.75	8.67	10.99	12.90	2.96

^a The percentage of generated molecules with attractive interaction scores equal to or above our score threshold is shown (Percent ≥ 11), as well as the score at the first quartile (Q1), second quartile (Q2), Mean, third quartile (Q3), maximum (Max), and standard deviation (Std) of the distribution.

^b Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.

Table S7.4. Statistics of the distribution of attractive interaction scores, when molecules are clustered into 100 groups and cluster scores are converted into sampling fractions using the *softsub* method.

Iteration	Percent ≥ 11	Q1	Q2	Mean	Q3	Max	Std
0	28.10	5.50	8.00	8.46	11.50	31.50	4.89
1	37.00	6.00	9.00	9.76	12.50	39.50	5.63
2	49.70	7.50	10.50	12.22	16.00	51.00	7.93
3	62.60	8.00	13.50	15.14	20.63	54.00	9.70
4	72.90	10.00	16.50	18.25	25.00	55.50	10.90
5	76.00	11.00	19.00	20.08	27.63	59.00	11.90

^b Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.

Section 8: Details and Parameters Used for Running DiffDock.

DiffDock handles all of the ligand preparation; we simply provide it with a protein structure and ligand SMILES string. It should be noted that since DiffDock is a diffusion generative model, it is inherently stochastic in nature During the docking inference stage, we utilize 20 inference steps, 10 samples for each complex, and a batch size of 6. Utilizing RDKit, DiffDock uses the MolFromSmiles module to process ligands, adds hydrogen atoms, and retrieves the 3D atomic coordinates with the AllChem.EmbedMolecule module employing the ETKDGv2 methodology.



Figure S8.1. Generated molecule docked to the c-Abl kinase (**A**) with the corresponding proteinligand fingerprint (**B**).



Figure S8.2. Generated molecule docked to the HNH domain of Cas9 (**A**) with the corresponding protein-ligand fingerprint (**B**).

Section 9: t-SNE Visualization of the Evolution of the Generated Molecular Ensembles.



Figure S9.1. Generated molecules and active learning training sets across each iteration of our pipeline, visualized in two dimensions after performing t-distributed stochastic neighbor embedding (t-SNE). The generated molecules and active learning training sets are shown in (A) and (B), respectively. Changes in the generated molecules and active learning training sets relative to the molecules generated at iteration 0 are shown in (C) and (D), respectively. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.

Section 10: Vocabulary Composition of the Combined Dataset.

Table S10.1. List of unique tokens that occur in the unfiltered combined dataset less than 1,000 times.

-	'%10'	-	'[125IH]'	-	'[32P]'	-	'[Be+2]'
-	'%11'	-	'[125I]'	-	'[35S]'	-	'[Bi+3]'
-	'%12'	-	'[127I]'	-	'[3H]'	-	'[BiH3]'
-	'%13'	-	'[127Xe]'	-	'[42K+]'	-	'[Bi]'
-	'%14'	-	'[129Xe]'	-	'[45Ca+2]'	-	'[Br+2]'
-	'%15'	-	'[131Cs]'	-	'[47Ca+2]'	-	'[Br]'
-	'%16'	-	'[131I-]'	-	'[4H]'	-	'[C+]'
-	'%17'	-	'[1311]	-	'[73Se]'	-	'[CH+]'
-	'%18'	-	'[133Xe]'	-	'[75Se]'	-	'[CH-]
-	'%19'	-	'[135I]'	-	'[76BrH]'	-	'[CH2+]'
-	'%20'	-	'[13CH2]'	-	'[76Br]'	-	'[CH2-]'
-	'%21'	-	'[13CH3]'	-	'[81Kr]'	-	'[CH2]'
-	'%22'	-	'[13CH]'	-	'[82Rb+]'	-	'[CH]'
-	'%23'	-	'[13C]'	-	'[82Rb]'	-	'[C]'
-	'%24'	-	'[13NH3]'	-	'[85Sr+2]'	-	'[Ca++]'
-	'%25'	-	'[13cH]'	-	'[85SrH2]'	-	'[Ca+2]'
-	'%26'	-	'[13c]'	-	'[89Sr+2]'	-	'[CaH2]'
-	'%27'	-	'[14C@@H]'	-	'[Ag+]'	-	'[Ca]'
-	'%28'	-	'[14C@@]'	-	'[Ag-4]'	-	'[Cl+2]'
-	'%29'	-	'[14C@H]'	-	'[Ag-]'	-	'[Cl+3]'
-	'%30'	-	'[14CH2]'	-	'[Ag]'	-	'[Cl+]'
-	'%31'	-	'[14CH3]'	-	'[Al+3]'	-	'[C1]'
-	'%32'	-	'[14CH]'	-	'[Al-3]'	-	'[Co]'
-	'*'	-	'[14C]'	-	'[Al]'	-	'[Cs+]'
-	':'	-	'[14cH]'	-	'[Ar]'	-	'[Cs]'
-	'[*]'	-	'[14c]'	-	'[As+]'	-	'[Cu-]'
-	'[10B]'	-	'[15NH]'	-	'[As-]'	-	'[Cu]'
-	'[11C-]'	-	'[15OH2]'	-	'[AsH3]'	-	'[F+]'
-	'[11C@@H]'	-	'[15nH]'	-	'[AsH]'	-	'[F-]'
-	'[11CH2]'	-	'[15n]'	-	'[As]'	-	'[Fe++]'
-	'[11CH3]'	-	'[17F]'	-	'[At]'	-	'[Fe]'
-	'[11CH]'	-	'[18F-]'	-	'[Au-]'	-	'[Fe-3]'
-	'[11C]'	-	'[18FH]'	-	'[Au]'	-	'[Fe]'
-	'[11c]'	-	'[18F]'	-	'[B@-]'	-	'[Gd-4]'
-	'[123I-]'	-	'[18OH]'	-	'[B@@-]'	-	'[Gd-5]'
-	'[123IH]'	-	'[18O]'	-	'[BH-]'	-	'[H+]'
-	'[123I]'	-	'[19F]'	-	'[BH2-]'	-	'[H-]'
-	'[123Te]'	-	'[211At]'	-	'[BH3-]'	-	'[HH]'
-	'[124I-]'	-	'[223Ra]'	-	'[B]'	-	'[He]'
-	'[124I]'	-	'[22Na+]'	-	'[Ba+2]'	-	'[Hg]'
-	'[125I-]'	-	'[32PH]'	-	'[Ba]'	-	'[I+2]'

-	'[I+3]'	-	'[Ni++]'	-	'[S-]'	-	'[TeH]'
-	'[I+]'	-	'[Ni]'	-	'[S@+]'	-	'[Te]'
-	'[IH2]'	-	'[O+]'	-	'[S@@+]'	-	'[V]'
-	'[IH]'	-	'[O-2]'	-	'[S@]'	-	'[W]'
-	'[I]'	-	'[OH+]'	-	'[SH+]'	-	'[Xe]'
-	'[KH]'	-	'[OH-]'	-	'[SH-]'	-	'[Zn++]'
-	'[K]'	-	'[OH]'	-	'[SH2]'	-	'[Zn+2]'
-	'[Kr]'	-	'[O]'	-	'[SH]'	-	'[Zn+]'
-	'[Li+]'	-	'[Os]'	-	'[S]'	-	'[Zn-2]'
-	'[LiH]'	-	'[P-]'	-	'[Sb]'	-	'[Zn]'
-	'[Li]'	-	'[P@+]'	-	'[Se+]'	-	'[b-]'
-	'[Mg+2]'	-	'[P@@+]'	-	'[Se-2]'	-	'[c+]'
-	'[Mg+]'	-	'[P@@]'	-	'[Se-]'	-	'[c-]'
-	'[MgH2]'	-	'[P@]'	-	'[SeH2]'	-	'[cH+]'
-	'[Mg]'	-	'[PH+]'	-	'[SeH]'	-	'[cH-]'
-	'[Mn]'	-	'[PH2+]'	-	'[Si-]'	-	'[c]'
-	'[Mo]'	-	'[PH2]'	-	'[Si@]'	-	'[n-]'
-	'[N@+]'	-	'[PH]'	-	'[SiH-]'	-	'[nH+]'
-	'[N@@+]'	-	'[P]'	-	'[SiH2]'	-	'[n]'
-	'[N@@H+]'	-	'[Pd]'	-	'[SiH3-]'	-	'[o+]'
-	'[N@@]'	-	'[Pd]'	-	'[SiH3]'	-	'[o]'
-	'[N@H+]'	-	'[Pt]'	-	'[SiH4]'	-	'[s+]'
-	'[N@]'	-	'[Pt]'	-	'[SiH]'	-	'[s]'
-	'[NH-]'	-	'[Ra]'	-	'[Sn]'	-	'[se+]'
-	'[NH2+]'	-	'[Rb+]'	-	'[Sr++]'	-	'[te+]'
-	'[NH4+]'	-	'[Rb]'	-	'[Sr+2]'	-	'[te]'
-	'[NH]'	-	'[Re-]'	-	'[SrH2]'	-	'b'
-	'[N]'	-	'[Re]'	-	'[Tc]'	-	'p'
-	'[NaH]'	-	'[Ru-]'	-	'[Te+]'		
-	'[Na]'	-	'[Ru]'	-	'[Te-]'		
-	'[Nb]'	-	'[S-2]'	-	'[TeH2]'		

Table S10.2. List of unique tokens that occur in the filtered combined dataset.

_

_

_

-

-

-

_

-

-

-

_

-

_

-

-

_

'[Na+]'

'[O-]'

'[P+]'

'[S+]'

'[Se]'

'[Si]'

'[n+]'

'[nH]'

'[se]'

'\'

'c'

'n'

'o'

's'

'~'

'[S@@]'

- '[NH3+]' _
- -
- '[NH+]'
- '[N-]' -
- '[N+]' -
- '[K+]' -

'!'

'#'

'('

') '

·_'

'.'

'/'

'1'

'2'

'3'

'4'

'5'

'6'

'7'

'8'

'9'

'<'

'='

'B'

'Br'

'C'

'Cl'

'F'

Ί'

'N'

'O'

'P'

'S'

'[2H]'

'[B-]'

'[Br-]' '[C-]'

'[C@@H]'

'[C@@]'

'[C@H]'

'[C@]'

'[Cl-]'

'[H]'

_

-

_

-

_

-

_

_

_

-

-

-

_

-

-

-

_

_

_

-

-

-

_

-

-

-

-

-

_

--

-

-

-

-

-

-

-

- '[I-]' _





Figure S11.1. Frequency as a function of block size (A) and token (B). Vertical dotted lines are positioned at 133 in (A) and serves as our block size cutoff. In (B), our cutoff, illustrated with the vertical dotted lines, is positioned at the first token where the frequency is less than 1,000 times.



Figure S11.2. Frequency as a function of molecular weight of molecule in the (A) MOSES and (B) combined datasets.

Section 12: Details of the GPT Architecture.

The GPT model that we employ is based on the transformer architecture introduced in the revolutionary paper, "Attention is All You Need". Within the framework of the transformer architecture, the encoder processes input data into a sequence of context-rich vectors, while the decoder utilizes this contextual information to generate output data. Both of these components utilize a self-attention mechanism, which enables the model to selectively focus on distinct parts of the input sequence at each computational step. The technical difference between the encoder and decoder parts of the transformer model is that the decoder ensures that the prediction for a particular token only depends on the preceding tokens, while each token in the encoder can attend to all other tokens in the sequence. Our GPT model is constructed as a series of transformer decoder blocks. This approach is appropriate for tasks that require generating novel sequential data such as SMILES strings.

The forward pass of our GPT model begins by dividing each SMILES string into distinct units known as tokens, processing each token with embedding layers, and combining these embeddings to form a vector representation of each token. These embedded vectors are then sequentially passed through a series of transformer decoder blocks, each comprised of a self-attention layer and a feed-forward network, with additional structural elements to enhance learning. The final result is a sequence of vectors, each corresponding to a position in the output SMILES string, where the elements of each vector represent probabilities for each token in the vocabulary. This high-level overview sets the stage for a more detailed discussion of the individual components.

Embeddings: Initially, a vocabulary comprising all of the unique tokens in the training data is constructed. For any given SMILES string in the input data, the input tokens undergo three distinct processing methods: token, positional, and type embeddings. The token embedding maps each token in the input sequence to a learnable vector representation, allowing the model to learn an optimal high-dimensional characterization for each token. Similarly, the positional embedding maps each input token to a learnable vector based on its position in the sequence. The type embedding layer uniformly assigns a constant bias to all embeddings of each input sequence. The sum of these three embeddings is passed through a dropout layer, setting 10% of its scalar components to 0. This embedding process transforms the input tokens into a form more suitable for the downstream modeling process.

Transformer Decoder Stack: For each token in the input sequence, the resulting embedding is passed to the first transformer decoder block, which begins with layer normalization, a process that adjusts and scales each embedding to have a mean of 0 and a standard deviation of 1. A self-attention mechanism is then applied to the normalized embedding, using learned matrices to linearly transform the embedding into three different vectors known as the query, key, and value vectors:

$$\mathbf{q}_i = \mathbf{W}_q \times \mathbf{e}_i \tag{1}$$

$$\mathbf{k}_i = \mathbf{W}_k \times \mathbf{e}_i \tag{2}$$

$$\mathbf{v}_i = \mathbf{W}_{\boldsymbol{v}} \times \mathbf{e}_i \tag{3}$$

where \mathbf{W}_q , \mathbf{W}_k , and \mathbf{W}_v are learned weight matrices that transform each input embedding, represented by \mathbf{e}_i , into the corresponding query, key, and value vectors. The dot products of the query and each key vector are then scaled according to the dimensionality of the key vectors and

passed through a softmax function, transforming them into a probability distribution to serve as attention weights. Finally, the attention scores are used to generate a weighted sum of the value vectors, as shown in the following equation:

$$\mathbf{e'}_{i} = \mathbf{V} \times \operatorname{softmax} \begin{pmatrix} \frac{\mathbf{q}_{i} \cdot \mathbf{k}_{1}}{\sqrt{d_{k}}} \\ \cdots \\ \frac{\mathbf{q}_{i} \cdot \mathbf{k}_{L}}{\sqrt{d_{k}}} \end{pmatrix}$$

Here, $\mathbf{e'}_i$ represents the output of the attention mechanism at position *i* in the sequence, **V** is the value matrix whose *j*th column is the value vector corresponding to the embedding at position *j* in the sequence, d_k denotes the dimensionality of the key vectors, and *L* represents the length of the entire sequence. This operation amplifies the information from value vectors corresponding to higher attention weights (i.e., tokens that are more relevant to the current query), while suppressing the information from less relevant value vectors.

In practice, the self-attention mechanism is executed multiple times in parallel through what is known as *multi-head* attention. Each head (i.e., execution) uses its own set of learned linear transformations to generate query, key, and value vectors for all tokens in the sequence for each item in the batch, allowing the model to simultaneously focus on different aspects of the input across the various heads. The outputs from all attention heads are then concatenated and passed through a learned linear transformation to generate the final output of the multi-head attention mechanism.

A residual connection is a shortcut that skips one or more layers and allows the original input to be added directly to the output of those layers. This technique aids in training deeper networks by mitigating the vanishing gradient problem, where the gradients become too small for the network to learn effectively. In the context of GPT models, a residual connection is made by adding the input of the attention mechanism to the output. This sum is then processed using layer normalization, and the transformed embeddings are passed through a feed-forward network using the equation:

$\mathbf{H} = \text{Dropout}(\mathbf{W}_2 \times \text{GELU}(\mathbf{W}_1 \times \mathbf{E}' + \mathbf{b}_1) + \mathbf{b}_2)$ (5)

(4)

where **H** is the output of the feed-forward network, **E**' represents the matrix whose columns are the transformed embeddings, and **W**₁ (shape: 1024×256), **b**₁ (shape: 1024), **W**₂ (shape: 256×1024), and **b**₂ (shape: 256) represent the weight matrices and bias vectors of the two linear layers. GELU, or Gaussian Error Linear Unit, is an activation function used to introduce non-linearity into the model. A residual connection is established by summing the input to this feed-forward network with the output.

This entire process is repeated for additional decoder blocks, and the output of the final decoder block is processed with layer normalization. The normalized output is then passed through a learned linear transformation with bias to map the embeddings to the output vocabulary size, and the resulting vectors are processed with softmax to generate the output probabilities at each position in the sequence.





Figure S13.1. Training loss (A), validation loss (B), and learning rate (C) during the 30 epochs of pretraining of our model on the combined dataset.



Figure S13.2. Training step losses (evaluated after each batch) during 5 rounds of active learning for the HNH domain of Cas9 (10 epochs each) with different conversion methods.

Section 14: Pretrained GPT Model Performance on the MOSES Benchmark.

Model	Validity	Unique@1K	Unique@10K	Novelty	IntDiv ₁	IntDiv ₂	Filters
Our Model	0.996	1.000	0.999	0.730	0.856	0.850	0.998
MolGPT ²	0.994	N/A	1.000	0.797	0.857	0.851	N/A
LatentGAN ³	0.897	1.000	0.997	0.949	0.857	0.850	0.973
JT-VAE ⁴	1.000	1.000	1.000	0.914	0.855	0.849	0.976
CharRNN ⁵	0.975	1.000	0.999	0.842	0.856	0.850	0.994
MolecularRNN ⁶	1.000	N/A	0.994	1.000	0.881	0.876	N/A
iPPIgAN ⁷	0.989	1.000	0.999	0.990	N/A	N/A	N/A
DNMG ⁸	0.999	1.000	0.998	0.936	0.856	0.850	0.996
CogMol ⁹	0.955	1.000	1.000	N/A	0.857	0.851	0.989
TransVAE ¹⁰	0.567	NA	N/A	0.996	N/A	N/A	N/A
ShapeProb ¹¹	0.969	1.000	0.995	N/A	0.865	N/A	0.865
GENTRL ¹²	0.850	N/A	N/A	N/A	N/A	N/A	N/A
TransAntivirus ¹³	1.000	0.999	0.999	0.999	0.895	N/A	N/A
CRTmaccs ¹⁴	1.000	1.000	1.000	1.000	N/A	N/A	N/A
MolGCT ¹⁵	0.985	1.000	0.998	0.814	0.853	N/A	0.996
cMolGPT ¹⁶	0.988	1.000	0.999	N/A	N/A	N/A	N/A
GraphINVENT ¹⁷	0.964	1.000	0.998	N/A	0.857	0.851	0.950
cTransformer ¹⁸	0.988	1.000	0.999	N/A	N/A	N/A	N/A
GMTransformer ¹⁹	0.829	1.000	1.000	0.883	0.856	N/A	0.980

Table S14.1. Primary results of our pretrained model on the MOSES benchmark compared to topperforming models in the field.

^a Validity (ratio of generated molecules deemed valid by RDKit's molecular structure parser), Unique@1K and @10K (fraction of valid generated molecules with no duplicates), Novelty (fraction of valid and unique generated molecules that are not in the training set), IntDiv_i (internal diversity within the generated set for power mean *i*), and Filters (fraction of generated molecules that pass filters that check for specific fragments) are shown. See MOSES benchmark for more details on how these metrics are calculated.²⁰

^b The top value for each metric is shown in bold.

^c Values not reported are shown as N/A.

Model	FCD/Test	FCD/TestSF	Frag/Test	Frag/TestSF	SNN/Test	SNN/TestSF	Scaff/Test	Sca
Our Model	0.038	0.450	1.000	0.999	0.633	0.585	0.970	
MolGPT ²	0.067	0.507	N/A	N/A	N/A	N/A	N/A	
LatentGAN ³	0.296	0.824	0.999	0.998	0.538	0.514	0.886	
JT-VAE ⁴	0.395	0.938	0.997	0.995	0.548	0.519	0.896	
CharRNN ⁵	0.073	0.520	1.000	0.998	0.601	0.565	0.924	
MolecularRNN ⁶	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
iPPIgAN ⁷	5.879	6.171	N/A	N/A	N/A	N/A	N/A	
DNMG ⁸	0.373	0.631	0.999	0.998	0.472	0.579	0.784	
CogMol ⁹	0.166	0.603	0.999	0.997	0.560	0. 533	0.905	
TransVAE ¹⁰	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
ShapeProb ¹¹	1.332	1.850	0.984	0.980	0.446	0.432	0.459	
GENTRL ¹²	N/A	N/A	N/A	N/A	N/A	N/A	N/A	
TransAntivirus ¹³	10.947	N/A	N/A	N/A	N/A	N/A	N/A	
CRTmaccs ¹⁴	13.565	13.999	N/A	N/A	0.334	0.330	N/A	
MolGCT ¹⁵	0.402	0.803	0.997	0.995	0.618	0.577	0.891	
cMolGPT ¹⁶	N/A	N/A	1.000	0.998	0.619	0.578	N/A	
GraphINVENT ¹⁷	0.682	1.223	0.986	0.986	0.569	0.539	0.885	
cTransformer ¹⁸	N/A	N/A	1.000	0.998	0.619	0.578	N/A	
GMTransformer ¹⁹	0.199	0.760	0.998	0.996	0.578	0.546	0.913	

Table S14.2. Additional results of our pretrained model on the MOSES benchmark compared to top-performing models in the field.

^a FCD (Fréchet ChemNet Distance that is calculated using activation of the penultimate layer of ChemNet), Frag (compares molecular fragments between generated and training sets), SNN (average Tanimoto similarity between molecules in the generated set and the corresponding nearest molecule in the training set), and Scaff (compares molecular scaffolds between generated and training sets) are shown. Test (similarity from the training set to the test set) and TestSF (similarity from the training set to the scaffold test set) are shown for each metric. See MOSES benchmark for more details on how these metrics are calculated.²⁰

^b The top value for each metric is shown in bold.

^c Values not reported are shown as N/A.

Section 15: RDKit Descriptors Used to Construct the Chemical Space Proxy.

Table S15.1. List of RDKit descriptors discarded.

- BCUT2D CHGHI -
- BCUT2D CHGLO -
- BCUT2D LOGPHI -
- BCUT2D LOGPLOW -
- BCUT2D MRHI
- -BCUT2D MRLOW
- _ BCUT2D MWHI

- BCUT2D MWLOW -
- Ipc _
- MaxAbsPartialCharge -
- MaxPartialCharge -
- MinAbsPartialCharge
- MinPartialCharge -

Table S15.2. List of RDKit descriptors included and used to construct the chemical space proxy.

- AvgIpc _
- BalabanJ _
- **BertzCT** _
- Chi₀ _
- Chi0n _
- Chi_{0v} -
- Chi1
- Chi1n _
- Chi1v -
- Chi2n _
- Chi₂v -
- -Chi3n
- Chi3v _
- Chi4n _
- Chi4v -
- EState VSA1 _
- EState_VSA10 -
- _ EState VSA11
- EState VSA2 -
- EState_VSA3
- EState VSA4 -
- EState_VSA5 -
- EState VSA6 -
- EState VSA7 -
- EState VSA8 -
- _ EState_VSA9
- ExactMolWt -
- FpDensityMorgan1 _
- FpDensityMorgan2 -
- FpDensityMorgan3 -FractionCSP3 _

- HallKierAlpha
- -HeavyAtomCount
- HeavyAtomMolWt
- Kappa1 _
- Kappa2 _
- Kappa3 _
- LabuteASA
- MaxAbsEState _ Index
- MaxEStateIndex
- MinAbsEState -
- Index
- MinEStateIndex
- MolLogP _
- MolMR -
- _ MolWt
- NHOHCount -
- _ NOCount
- NumAliphatic -Carbocycles
- NumAliphatic -Heterocycles
- NumAliphatic -
- Rings
 - NumAromatic Carbocycles
 - NumAromatic -_ Heterocycles _ -
 - NumAromatic RingCount -Rings SMR_VSA1 -_ SMR VSA10
 - NumHAcceptors

- NumHDonors _
- -NumHeteroatoms
- NumRadical -
- Electrons
- -NumRotatable Bonds
- NumSaturated Carbocycles
- NumSaturated -Heterocycles
- NumSaturated -
- Rings NumValence
- Electrons
- PEOE VSA1
- _ PEOE VSA10
- PEOE_VSA11 -
- _ PEOE VSA12
- PEOE VSA13 -
- PEOE_VSA14
- PEOE VSA2 _
- PEOE_VSA3 -
- PEOE VSA4
- PEOE VSA5 _
- PEOE VSA6 _
- PEOE VSA8
- PEOE VSA9
- PEOE VSA7

-	SMR_VSA2
-	SMR_VSA3
-	SMR_VSA4
-	SMR_VSA5
-	SMR_VSA6
-	SMR_VSA7
-	SMR_VSA8
-	SMR_VSA9
-	SlogP_VSA1
-	SlogP_VSA10
-	SlogP_VSA11
-	SlogP_VSA12
-	SlogP_VSA2
-	SlogP_VSA3
-	SlogP_VSA4
-	SlogP_VSA5
-	SlogP_VSA6
-	SlogP_VSA7
-	SlogP_VSA8
-	SlogP_VSA9
-	TPSA
-	VSA_EState1
-	VSA_EState10
-	VSA EState?
	VDA_LDIale2
-	VSA_EState3
-	VSA_EState3 VSA_EState4
- - -	VSA_EState2 VSA_EState3 VSA_EState4 VSA_EState5
- - -	VSA_EState2 VSA_EState3 VSA_EState4 VSA_EState5 VSA_EState6
- - - -	VSA_EState2 VSA_EState3 VSA_EState4 VSA_EState5 VSA_EState6 VSA_EState7
- - - -	VSA_EState2 VSA_EState3 VSA_EState4 VSA_EState5 VSA_EState6 VSA_EState7 VSA_EState8
- - - - -	VSA_EState2 VSA_EState3 VSA_EState4 VSA_EState5 VSA_EState6 VSA_EState7 VSA_EState8 VSA_EState9
- - - - -	VSA_EState2 VSA_EState3 VSA_EState4 VSA_EState5 VSA_EState6 VSA_EState7 VSA_EState8 VSA_EState9 fr_A1_COO
- - - - - -	VSA_EState2 VSA_EState3 VSA_EState4 VSA_EState5 VSA_EState6 VSA_EState7 VSA_EState8 VSA_EState9 fr_A1_CO0 fr_A1_OH
- - - - - -	VSA_EState2 VSA_EState3 VSA_EState4 VSA_EState5 VSA_EState6 VSA_EState7 VSA_EState8 VSA_EState9 fr_A1_CO0 fr_A1_OH fr_A1_OH_noTert
- - - - - - - -	VSA_EState2 VSA_EState3 VSA_EState4 VSA_EState5 VSA_EState6 VSA_EState7 VSA_EState8 VSA_EState9 fr_A1_CO0 fr_A1_OH fr_A1_OH_noTert fr_ArN
- - - - - -	VSA_EState2 VSA_EState3 VSA_EState4 VSA_EState5 VSA_EState6 VSA_EState7 VSA_EState8 VSA_EState9 fr_A1_CO0 fr_A1_OH fr_A1_OH_noTert fr_ArN fr_Ar_COO
	VSA_EState2 VSA_EState3 VSA_EState4 VSA_EState5 VSA_EState6 VSA_EState7 VSA_EState9 fr_A1_CO0 fr_A1_OH fr_A1_OH_noTert fr_ArN fr_Ar_CO0 fr_Ar_N

- fr_Ar_OH
- fr_COO _

fr COO2 _ fr_C_O _

- fr_C_O_noCOO -
- fr_C_S -
- fr_HOCCN _
- fr_Imine -
- fr_NH0 _
- fr_NH1 -
- fr_NH2 _
- fr_N_O _
- fr_Ndealkylation1 _
- fr_Ndealkylation2 -
- fr_Nhpyrrole -
- fr_SH
- -
- fr_aldehyde -
- fr alkyl carbamate -
- fr_alkyl_halide _
- fr_allylic_oxid -
- fr amide _
- fr_amidine _
- fr aniline _
- fr_aryl_methyl -
- fr_azide _
- fr azo _
- fr_barbitur _
- fr benzene -
- fr_benzodiazepine _
- fr bicyclic _
- fr diazo -
- fr_dihydropyridine -
- fr_epoxide _
- fr_ester -
- _ fr ether
- fr furan -
- fr_guanido
- fr_halogen -
- fr_hdrzine -
- fr hdrzone -
- fr imidazole
- fr imide _

- fr_isocyan _
- fr_isothiocyan _
- fr_ketone -
- fr_ketone_Topliss -
- fr_lactam _
- fr_lactone _
- fr_methoxy _
- fr_morpholine -
- fr nitrile -
- fr_nitro _
- fr_nitro_arom _
- fr_nitro_arom_ nonortho
- fr nitroso -
- fr_oxazole _
- fr oxime -
- fr_para_ _
 - hydroxylation
- fr_phenol _
- fr_phenol_
- noOrthoHbond
- fr_phos_acid -
- fr_phos_ester -
- fr_piperdine -
- fr_piperzine _
- fr priamide -
- fr_prisulfonamd _
- fr pyridine _
- fr_quatN _
- fr sulfide _
- fr sulfonamd _
- fr_sulfone -
- _ fr_term_acetylene
- fr tetrazole -
- fr_thiazole _
- fr thiocyan
- fr_thiophene -
- fr unbrch alkane -
- _ fr urea
- qed _



Figure S15.3. Cumulative fraction of variance explained by the first *N* principal components. Our chemical space proxy (i.e., the first 120 principal components) explains 99.3% of the variance in the hyperspace of 196 RDKit descriptors.



Section 16: Wall Times of Each Step in the Complete Pipeline.

Figure S16.1. Wall times of each step in the complete pipeline. Steps include pretraining, generation, calculating descriptors, clustering and sampling, docking, scoring, and active learning fine-tuning. Pretraining is performed only once, while each other step is performed once per iteration.





RDKit Descriptors

Figure S17.1. Attractive interaction scores of scored molecules across five iterations of active learning. Results for the methodology applied to the HNH domain of Cas9 using 196 RDKit descriptors, and using 42 MQN descriptors are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Section 18: Logits for the End-of-Sequence Token as the Molecular Descriptor Vector.

Figure S18.1. Attractive interaction scores of scored molecules across five iterations of active learning for the HNH domain of Cas9, utilizing the logits for the end-of-sequence token as the molecular descriptor vector for each generated molecule in the complete pipeline. The distribution for the model pretrained on the combined dataset is shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Figure S18.2. Attractive interaction scores of scored molecules across five iterations of active learning for the HNH domain of Cas9, utilizing the logits for the end-of-sequence token combined with the 196 RDKit descriptors as the molecular descriptor vector for each generated molecule in the complete pipeline. The distribution for the model pretrained on the combined dataset is shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Figure S18.3. Mean cluster attractive interaction scores across five iterations of active learning for the HNH domain of Cas9, utilizing the logits for the end-of-sequence token as the molecular descriptor vector for each generated molecule in the complete pipeline. The distribution for the model pretrained on the combined dataset is shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Figure S18.4. Median cluster attractive interaction scores across five iterations of active learning for the HNH domain of Cas9, utilizing the logits for the end-of-sequence token as the molecular descriptor vector for each generated molecule in the complete pipeline. The distribution for the model pretrained on the combined dataset is shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Section 19: Frequency as a Function of Cluster Size for Alignment to c-Abl Kinase.

Figure S19.1. Frequency as a function of cluster size for each iteration of the methodology. Results are shown for the model pretrained on the combined dataset with the generations filtered based on ADMET metrics, aligned to c-Abl kinase.



Figure S19.2. Frequency as a function of cluster size for each iteration of the methodology. Results are shown for the model pretrained on the combined dataset with the generations filtered based on ADMET metrics and functional group restrictions, aligned to c-Abl kinase.



Figure S19.3. Frequency as a function of cluster size for each iteration of the methodology. Results are shown for the model pretrained on the MOSES dataset with the generations filtered based on ADMET metrics and functional group restrictions, aligned to c-Abl kinase.



Figure S19.4. Frequency as a function of cluster size for each iteration of the methodology shown for the clustering that we select as well as the clustering with the lowest loss. Results are shown for the model pretrained on the combined dataset with the generations filtered based on ADMET metrics and functional group restrictions, aligned to c-Abl kinase.





Figure S20.1. Evaluation of our scoring function with the protein-ligand complexes in the PDBbind v.2020 refined set. (A) Binding affinity (pK_d) plotted as a function of score. There is a corresponding Pearson correlation of 0.32. (B) The relative frequency of different scores is shown. 99.6% of the complexes exceed our score threshold of 11.

Section 21: Alternative Methods for Converting Mean Cluster Scores to Sampling Fractions.

Prior to constructing the active learning training set, we need to convert the attractive interaction scores s_i obtained by using the prolif software on docked molecules into sampling fractions f_i , which will be used to calculate the number of molecules that we need to sample from each cluster. A simple way to do that is to normalize the sum of all scores to unity:

$$f_i^{\text{linear}}(s_i) = \frac{s_i}{\sum_i s_i}$$

We call this approach *linear* conversion. Because one could interpret sampling fractions as effective probabilities of sampling from a given cluster, it is natural to consider the use of a softmax function:

$$f_i^{\text{softmax}}(s_i) = \frac{e^{s_i}}{\sum_i e^{s_i}}$$

which, for computational stability purposes, is often implemented with the maximum value among a set of arguments subtracted from each individual argument. To contrast with a modification of a softmax function introduced later, we refer to this as *softsub* conversion. In the main text of our paper, we implement the *softsub* approach and refer to it as *softmax* because this is the common implementation of the softmax function.

$$f_i^{\text{softsub}}(s_i) = \frac{e^{s_i - s_{max}}}{\sum_i e^{s_i - s_{max}}}$$

For a pretrained model, cluster scores range from 0 to 16. Because exponential functions increase rapidly, the *softsub* approach will effectively favor the 1-5 clusters with largest scores. We conjecture that a smoother function may lead to better model behavior during active learning, and implement a *softdiv* conversion approach, in which, instead of subtracting the maximum cluster score, we divide by it:

$$f_i^{\text{softdiv}}(s_i) = \frac{e^{s_i/s_{max}}}{\sum_i e^{s_i/s_{max}}}$$

Empirically, this approach leads even to a narrower distribution of sampling fractions than that obtained with the *linear* conversion approach. We introduce a hyperparameter $div f \in (0,1]$ by which we multiply the s_{max} value prior to dividing by it:

$$f_i^{\text{softdivf}}(s_i) = \frac{e^{\frac{S_i}{divf \times s_{max}}}}{\sum_i e^{\frac{S_i}{divf \times s_{max}}}}$$

By visualizing the distribution of *softdiv* values with different values of the hyperparameter (Figure S4.1), we pick divf = 0.25, as it maximizes the spread in sampling fractions. In what follows, the *softdiv* conversion will refer to *softdiv* with divf = 0.25.



Figure S21.1. Distribution of sampling fractions obtained with different conversion approaches applied to cluster scores obtained from generations of the pretrained model. A bar corresponding to sampling the same number of molecules from each cluster (i.e., uniform sampling) is shown in black.



Figure S21.2. Attractive interaction scores for molecules generated by the pretrained model (iteration 0) and by the model after each of the five iterations of active learning where, prior to sampling for docking, molecules in the chemical space are grouped into 100 clusters, and cluster scores are converted into sampling fractions using the *linear* approach.



Figure S21.3. Attractive interaction scores for molecules generated by the pretrained model (iteration 0) and by the model after each of the five iterations of active learning where, prior to sampling for docking, molecules in the chemical space are grouped into 100 clusters, and cluster scores are converted into sampling fractions using the *softdiv* approach.

Table S21.4.	Statistics of	f the distril	oution of a	tractive interac	ction scores,	when mo	plecules are
selected rand	omly (naïve	e active lea	rning), with	n no clustering			

Iteration	Percent ≥ 11	Q1	Q2	Mean	Q3	Max	Std
0	26.20	5.50	8.00	8.41	11.00	33.00	4.58
1	32.40	5.50	9.00	9.27	11.50	35.00	4.79
2	35.00	6.50	9.00	9.67	12.50	33.00	4.89
3	40.00	6.88	9.50	10.45	13.50	37.00	5.31
4	44.80	7.00	10.00	11.03	13.63	42.00	6.16
5	44.20	7.00	10.00	11.13	13.50	38.50	6.05

^b Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.

^c This table corresponds to the distribution in Figure 4A of main text.

Iteration	Percent ≥ 11	Q1	Q2	Mean	Q3	Max	Std
0	12.00	6.68	8.58	8.43	9.85	15.80	2.31
1	24.00	7.75	9.10	9.17	10.88	20.80	3.40
2	30.00	8.17	9.40	9.96	11.50	21.50	3.41
3	35.00	7.86	10.07	10.23	11.63	24.00	3.97
4	50.00	8.88	10.89	11.28	13.46	25.45	4.11
5	50.00	8.80	10.94	11.84	14.48	25.80	4.63

Table S21.5. Statistics of the distribution of attractive interaction scores, when molecules are clustered into 100 groups and cluster scores are sampled *uniformly*.

^b Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases. ^c This table corresponds to the distribution in Figure 4B of main text.

Table S21.6. Statistics of the distribution of attractive interaction scores, when molecules are clustered into 100 groups and cluster scores are converted into sampling fractions using the *linear* method.

Iteration	Percent ≥ 11	Q1	Q2	Mean	Q3	Max	Std
0	12.00	6.68	8.58	8.43	9.85	15.80	2.31
1	23.23	7.88	9.20	9.54	10.71	19.95	2.73
2	37.00	7.78	9.98	10.24	12.10	26.25	4.40
3	47.00	8.29	10.55	11.60	14.51	23.90	5.09
4	62.00	10.19	13.27	13.19	15.84	31.41	5.09
5	71.00	10.50	13.43	14.00	16.93	29.50	5.50

^a The percentage of generated molecules with attractive interaction scores equal to or above our score threshold is shown (Percent ≥ 11), as well as the score at the first quartile (Q1), second quartile (Q2), Mean, third quartile (Q3), maximum (Max), and standard deviation (Std) of the distribution.

^b Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.

^c This table corresponds to the distribution in Figure S4.1.

Iteration	Percent ≥ 11	Q1	Q2	Mean	Q3	Max	Std
0	28.10	5.50	8.00	8.46	11.50	31.50	4.89
1	34.70	5.50	8.50	9.29	12.50	42.00	5.89
2	42.20	6.50	9.50	10.54	13.50	37.00	6.64
3	54.20	7.50	11.50	13.07	18.00	55.00	8.64
4	65.90	8.50	14.25	15.82	22.00	56.50	9.99
5	71.20	9.50	16.00	17.32	24.50	51.00	10.90

Table S21.7. Statistics of the distribution of attractive interaction scores, when molecules are clustered into 100 groups and cluster scores are converted into sampling fractions using the *softdiv* method.

^b Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases. ^c This table corresponds to the distribution in Figure S4.2.

PDB Bind	44.0	131.0	209.0	3684.0	7434.0	763.0	101.0	37.0	9974.0	20.0
Iteration 0	0.0	18.0	3.0	162.0	1507.0	45.0	0.0	5.0	3701.0	17.0
Iteration 1	0.0	13.0	3.0	167.0	1751.0	60.0	0.0	4.0	3749.0	27.0
Iteration 2	0.0	24.0	5.0	142.0	1859.0	76.0	0.0	12.0	3813.0	24.0
Iteration 3	0.0	19.0	8.0	139.0	2002.0	121.0	0.0	5.0	3925.0	24.0
Iteration 4	0.0	23.0	3.0	168.0	2028.0	183.0	0.0	7.0	3881.0	22.0
Iteration 5	0.0	25.0	4.0	167.0	2053.0	172.0	0.0	8.0	3992.0	24.0
	CationPi	EdgeToFace	FaceToFace	Hydrogen-bond	Hydrophobic	Ionic	MetalAcceptor	PiCation	Van der Waals	XBDonor

Interaction Counts per 1000 Molecules (Random Sampling)

Figure S21.8. Counts of interactions of each type for 1000 scored molecules generated by the pretrained model (iteration 0) and by the model after each of the five rounds of naïve active learning with *random sampling*. A count of interactions from 1000 protein-ligand complexes randomly sampled from the refined set of PDBbind v2020 is included for comparison. These counts correspond to the score distribution in Figure 4A of main text.

PDB Bind	44.0	131.0	209.0	3684.0	7434.0	763.0	101.0	37.0	9974.0	20.0
Iteration 0	0.0	18.0	3.0	162.0	1507.0	45.0	0.0	5.0	3701.0	17.0
Iteration 1	0.0	18.0	1.0	193.0	1529.0	114.0	0.0	11.0	3890.0	10.0
Iteration 2	0.0	20.0	2.0	221.0	1543.0	155.0	0.0	9.0	4156.0	12.0
Iteration 3	0.0	7.0	2.0	231.0	1471.0	200.0	0.0	7.0	4234.0	16.0
Iteration 4	1.0	19.0	5.0	251.0	1530.0	288.0	0.0	9.0	4485.0	8.0
Iteration 5	3.0	17.0	1.0	255.0	1460.0	372.0	0.0	11.0	4584.0	16.0
	CationPi	EdgeToFace	FaceToFace	Hydrogen-bond	Hydrophobic	lonic	MetalAcceptor	PiCation	Van der Waals	XBDonor

Interaction Counts per 1000 Molecules (Diffusion-based Sampling)

Figure S21.9. Counts of interactions of each type for 1000 molecules generated by the pretrained model (iteration 0) and by the model after each of the five rounds of active learning with clustering into 100 groups and *uniform* selection from each cluster. A count of interactions from 1000 protein-ligand complexes randomly sampled from the refined set of PDBbind v2020 is included for comparison. These counts correspond to the score distribution in Figure 4B of main text.

1										
PDB Bind	44.0	131.0	209.0	3684.0	7434.0	763.0	101.0	37.0	9974.0	20.0
Iteration 0	0.0	18.0	3.0	162.0	1507.0	45.0	0.0	5.0	3701.0	17.0
Iteration 1	0.0	12.0	4.0	168.0	1630.0	116.0	0.0	9.0	3939.0	11.0
Iteration 2	0.0	14.0	2.0	207.0	1582.0	180.0	0.0	8.0	4075.0	15.0
Iteration 3	3.0	19.0	5.0	219.0	1666.0	316.0	0.0	7.0	4264.0	20.0
Iteration 4	1.0	11.0	4.0	222.0	1713.0	506.0	0.0	5.0	4579.0	13.0
Iteration 5	3.0	17.0	4.0	226.0	1691.0	618.0	0.0	10.0	4624.0	5.0
	CationPi	EdgeToFace	FaceToFace	Hydrogen-bond	Hydrophobic	lonic	MetalAcceptor	PiCation	Van der Waals	XBDonor

Interaction Counts per 1000 Molecules (Linear-based Sampling)

Figure S21.10. Counts of interactions of each type for 1000 scored molecules generated by the pretrained model (iteration 0) and by the model after each of the five rounds of active learning with clustering into 100 groups and conversion of cluster scores into sampling fractions using the *linear* method. A count of interactions from 1000 protein-ligand complexes randomly sampled from the refined set of PDBbind v2020 is included for comparison. These counts correspond to the score distribution in Figure S4.1.

PDB Bind	44.0	131.0	209.0	3684.0	7434.0	763.0	101.0	37.0	9974.0	20.0
Iteration 0	0.0	18.0	3.0	162.0	1507.0	45.0	0.0	5.0	3701.0	17.0
Iteration 1	0.0	24.0	5.0	146.0	1269.0	93.0	0.0	6.0	3233.0	14.0
Iteration 2	0.0	16.0	4.0	213.0	1632.0	195.0	0.0	7.0	4170.0	12.0
Iteration 3	0.0	20.0	2.0	193.0	1762.0	476.0	0.0	7.0	4328.0	16.0
Iteration 4	5.0	21.0	6.0	211.0	1847.0	765.0	0.0	16.0	4589.0	14.0
Iteration 5	4.0	32.0	4.0	262.0	1708.0	974.0	0.0	21.0	4701.0	8.0
8	CationPi	EdgeToFace	FaceToFace	Hydrogen-bond	Hydrophobic	Ionic	MetalAcceptor	PiCation	Van der Waals	XBDonor

Interaction Counts per 1000 Molecules (Softdiv-based Sampling)

Figure S21.11. Counts of interactions of each type for 1000 scored molecules generated by the pretrained model (iteration 0) and by the model after each of the five rounds of active learning with clustering into 100 groups and conversion of cluster scores into sampling fractions using the *softdiv* method. A count of interactions from 1000 protein-ligand complexes randomly sampled from the refined set of PDBbind v2020 is included for comparison. These counts correspond to the score distribution in Figure S4.2.

PDB Bind	44.0	131.0	209.0	3684.0	7434.0	763.0	101.0	37.0	9974.0	20.0
Iteration 0	0.0	18.0	3.0	162.0	1507.0	45.0	0.0	5.0	3701.0	17.0
Iteration 1	0.0	17.0	2.0	185.0	1656.0	99.0	0.0	6.0	4134.0	18.0
Iteration 2	1.0	15.0	2.0	187.0	1830.0	347.0	0.0	8.0	4321.0	7.0
Iteration 3	1.0	23.0	2.0	176.0	1857.0	724.0	0.0	10.0	4352.0	15.0
Iteration 4	1.0	19.0	5.0	152.0	1954.0	1104.0	0.0	6.0	4452.0	15.0
Iteration 5	2.0	39.0	5.0	161.0	2026.0	1291.0	0.0	8.0	4653.0	12.0
8	CationPi	EdgeToFace	FaceToFace	Hydrogen-bond	Hydrophobic	Ionic	MetalAcceptor	PiCation	Van der Waals	XBDonor

Interaction Counts per 1000 Molecules (Softsub-based Sampling)

Figure S21.12. Counts of interactions of each type for 1000 molecules generated by the pretrained model (iteration 0) and by the model after each of the five rounds of active learning with clustering into 100 groups and conversion of cluster scores into sampling fractions using the *softsub* method. A count of interactions from 1000 protein-ligand complexes randomly sampled from the refined set of PDBbind v2020 is included for comparison. These counts correspond to the score distribution in Figure 4C of the main text.



Figure S21.13. Cluster scores (obtained as an average of attractive interaction scores for molecules in the cluster) for molecules generated by the pretrained model (iteration 0) and by the model after each of the five iterations of active learning where, prior to sampling for docking, molecules in the chemical space are grouped into 100 clusters, and molecules are sampled from each cluster *uniformly*. These cluster scores correspond to score distribution in Figure 4B of the main text.



Figure S21.14. Cluster scores (obtained as an average of attractive interaction scores for molecules in the cluster) for molecules generated by the pretrained model (iteration 0) and by the model after each of the five iterations of active learning where, prior to sampling for docking, molecules in the chemical space are grouped into 100 clusters, and molecules are sampled from each cluster using the *linear* method. These cluster scores correspond to score distribution in Figure SI4.1.



Figure S21.15. Cluster scores (obtained as an average of attractive interaction scores for molecules in the cluster) for molecules generated by the pretrained model (iteration 0) and by the model after each of the five iterations of active learning where, prior to sampling for docking, molecules in the chemical space are grouped into 100 clusters, and molecules are sampled from each cluster using the *softdiv* method. These cluster scores correspond to score distribution in Figure SI4.2.



Figure S21.16. Cluster scores (obtained as an average of attractive interaction scores for molecules in the cluster) for molecules generated by the pretrained model (iteration 0) and by the model after each of the five iterations of active learning where, prior to sampling for docking, molecules in the chemical space are grouped into 100 clusters, and molecules are sampled from each cluster using the *softsub* method. These cluster scores correspond to score distribution in Figure 4C of the main text.



Figure S21.17. Cluster scores (obtained as an average of attractive interaction scores for molecules in the cluster) for molecules generated by the pretrained model (iteration 0) and by the model after each of the five iterations of active learning where, prior to sampling for docking, molecules in the chemical space are grouped into 10 clusters, and molecules are sampled from each cluster using the *softsub* method. These cluster scores correspond to score distribution in Figure S10.1 of the main text.



Section 22: Distributions of Mean and Median Cluster Scores.

Figure S22.1. Mean cluster attractive interaction scores of scored molecules across five iterations of active learning for c-Abl kinase. The distribution for the model pretrained on the combined dataset with generation conditioned on ADMET filters are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Figure S22.2. Median cluster attractive interaction scores of scored molecules across five iterations of active learning for c-Abl kinase. The distribution for the model pretrained on the combined dataset with generation conditioned on ADMET filters are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Figure S22.3. Mean cluster attractive interaction scores of scored molecules across five iterations of active learning for c-Abl kinase. The distribution for the model pretrained on the combined dataset with generation conditioned on ADMET and functional group filters are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Figure S22.4. Median cluster attractive interaction scores of scored molecules across five iterations of active learning for c-Abl kinase. The distribution for the model pretrained on the combined dataset with generation conditioned on ADMET and functional group filters are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Figure S22.5. Mean cluster attractive interaction scores of scored molecules across five iterations of active learning for c-Abl kinase. The distribution for the model pretrained on the MOSES dataset with generation conditioned on ADMET and functional group filters are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Figure S22.6. Median cluster attractive interaction scores of scored molecules across five iterations of active learning for c-Abl kinase. The distribution for the model pretrained on the MOSES dataset with generation conditioned on ADMET and functional group filters are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Figure S22.7. Mean cluster attractive interaction scores of scored molecules across five iterations of active learning for the HNH domain of Cas9. The distribution for the model pretrained on the combined dataset with generation conditioned on ADMET filters are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Figure S22.8. Median cluster attractive interaction scores of scored molecules across five iterations of active learning for the HNH domain of Cas9. The distribution for the model pretrained on the combined dataset with generation conditioned on ADMET filters are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Figure S22.9. Mean cluster attractive interaction scores of scored molecules across five iterations of active learning for the HNH domain of Cas9. The distribution for the model pretrained on the combined dataset with generation conditioned on ADMET and functional group filters are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Figure S22.10. Median cluster attractive interaction scores of scored molecules across five iterations of active learning for the HNH domain of Cas9. The distribution for the model pretrained on the combined dataset with generation conditioned on ADMET and functional group filters are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Figure S22.11. Mean cluster attractive interaction scores of scored molecules across five iterations of active learning for the HNH domain of Cas9. The distribution for the model pretrained on the MOSES dataset with generation conditioned on ADMET and functional group filters are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.



Figure S22.12. Median cluster attractive interaction scores of scored molecules across five iterations of active learning for the HNH domain of Cas9. The distribution for the model pretrained on the MOSES dataset with generation conditioned on ADMET and functional group filters are shown. Iteration 0 refers to the pretraining phase, while later iterations refer to the active learning phases.

Section 23: Additional Evaluation of Generations across Active Learning Iterations.

Results in this section are regarding the model pretrained on the combined dataset with no filters applied to the generations, for alignment to HNH. It should be noted in the comparisons that poor values for the generations from the model aligned with sets curated with random sampling are likely due to memorization, since we only utilize replicas of scored molecules in this scenario.



Figure S23.1. Percentage of molecules generated by our model that are valid, unique, or novel after pretraining (iteration 0) and five rounds of active learning. Data are shown for different sampling/conversion schemes.



Figure S23.2. Memorization of training set by our model over five rounds of naïve active learning with random selection. (A) The percentage of molecules in a set of 100,000 generated at iteration *i* that occur in the training set at iteration *i*-1. (B) The percentage of molecules in a set of 100,000 generated at iteration *i* that occur in the set of scored molecules at iteration *i*-1. (C) The number of molecules from the active training set at iteration *i*-1 that occurs in generations at iteration *i* divided by the size of the active learning training set at iteration *i*-1 multiplied by 100. (D) the number of scored molecules at iteration *i*-1 that occur in generations at iteration *i* divided by number of scored molecules at iteration *i*-1 that occur in generations at iteration *i* divided by number of scored molecules at iteration *i*-1 that occur in generations at iteration *i* divided by number of scored molecules at iteration *i*-1 (i.e., 1000) multiplied by 100.



Figure S23.3. Memorization of training set by our model over five rounds of active learning with 100 clusters and uniform selection. (A) The percentage of molecules in a set of 100,000 generated at iteration *I* that occur in the training set at iteration *i*-1. (B) The percentage of molecules in a set of 100,000 generated at iteration *i* that occur in the set of scored molecules at iteration *i*-1. (C) The number of molecules from the active training set at iteration *i*-1 that occurs in generations at iteration *i* divided by the size of the active learning training set at iteration *i*-1 multiplied by 100. (D) the number of scored molecules at iteration *i*-1 that occur in generations at iteration *i* divided by number of scored molecules at iteration *i*-1 (i.e., 1000) multiplied by 100.



Figure S23.4. Memorization of training set by our model over five rounds of active learning with 100 clusters and linear selection. (A) The percentage of molecules in a set of 100,000 generated at iteration *i* that occur in the training set at iteration *i*-1. (B) The percentage of molecules in a set of 100,000 generated at iteration *i* that occur in the set of scored molecules at iteration *i*-1. (C) The number of molecules from the active training set at iteration *i*-1 that occurs in generations at iteration *i* divided by the size of the active learning training set at iteration *i*-1 multiplied by 100. (D) the number of scored molecules at iteration *i*-1 that occur in generations at iteration *i* divided by number of scored molecules at iteration *i*-1 (i.e., 1000) multiplied by 100.



Figure S23.5. Memorization of training set by our model over five rounds of active learning with 100 clusters and softdiv selection. (A) The percentage of molecules in a set of 100,000 generated at iteration *i* that occur in the training set at iteration *i*-1. (B) The percentage of molecules in a set of 100,000 generated at iteration *i* that occur in the set of scored molecules at iteration *i*-1. (C) The number of molecules from the active training set at iteration *i*-1 that occurs in generations at iteration *i* divided by the size of the active learning training set at iteration *i*-1 multiplied by 100. (D) the number of scored molecules at iteration *i*-1 that occur in generations at iteration *i* divided by number of scored molecules at iteration *i*-1 (i.e., 1000) multiplied by 100.



Figure S23.6. Memorization of training set by our model over five rounds of active learning with 100 clusters and softsub selection. (A) The percentage of molecules in a set of 100,000 generated at iteration *i* that occur in the training set at iteration *i*-1. (B) The percentage of molecules in a set of 100,000 generated at iteration *i* that occur in the set of scored molecules at iteration *i*-1. (C) The number of molecules from the active training set at iteration *i*-1 that occurs in generations at iteration *i* divided by the size of the active learning training set at iteration *i*-1 multiplied by 100. (D) the number of scored molecules at iteration *i*-1 that occur in generations at iteration *i* divided by number of scored molecules at iteration *i*-1 (i.e., 1000) multiplied by 100.

References

¹ ADMETlab 2.0. URL: https://admetmesh.scbdd.com/explanation/index.

²Bagal, V.; Aggarwal, R.; Vinod, P. K.; Priyakumar, U. D. MolGPT: Molecular Generation Using a Transformer-Decoder Model. *J. Chem. Inf. Model.* **2022**, *62*, 2064-2076. DOI: 10.1021/acs.jcim.1c00600.

³ Prykhodko, O.; Johansson, S. V.; Kotsias, P.-C.; Arús-Pous, J.; Bjerrum, E. J.; Engkvist, O.; Chen, H. A de novo molecular generation method using latent vector based generative adversarial network. *J. Cheminf.* **2019**, *11*, 74. DOI: 10.1186/s13321-019-0397-9.

⁴ Jin, W.; Barzilay, R.; Jaakkola, T. Junction Tree Variational Autoencoder for Molecular Graph Generation. In Proceedings of the 35th International Conference on Machine Learning, Proceedings of Machine Learning Research, **2018**.

⁵ Segler, M. H. S.; Kogej, T.; Tyrchan, C.; Waller, M. P. Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Cent. Sci.* **2018**, *4*, 120-131. DOI: 10.1021/acscentsci.7b00512.

⁶ Popova, M.; Shvets, M.; Oliva, J.; Isayev, O. MolecularRNN: Generating Realistic Molecular Graphs With Optimized Properties. *ArXiv* **2019**. DOI: https://doi.org/10.48550/arXiv.1905.13372.

⁷ Wang, J.; Chu, Y.; Mao, J.; Jeon, H.-N.; Jin, H.; Zeb, A.; Jang, Y.; Cho, K.-H.; Song, T.; No, K. T. De Novo Molecular Design With Deep Molecular Generative Models for PPI inhibitors. *Briefings Bioinf.* **2022**, *23*, bbac285. DOI: 10.1093/bib/bbac285.

⁸ Song, T.; Ren, Y.; Wang, S.; Han, P.; Wang, L.; Li, X.; Rodriguez-Patón, A. DNMG: Deep Molecular Generative Model by Fusion of 3D Information for De Novo Drug Design. *Methods* **2023**, *211*, 10-22. DOI: https://doi.org/10.1016/j.ymeth.2023.02.001.

⁹ Chenthamarakshan, V.; Das, P.; Hoffman, S. C.; Strobelt, H.; Padhi, I.; Lim, K. W.; Hoover, B.; Manica, M.; Born, J.; Laino, T.; Mojsilovic, A. CogMol: Target-Specific and Selective Drug Design for COVID-19 Using Deep Generative Models. In Proceedings of the 34th International Conference on Neural Information Processing Systems, **2020**.

¹⁰ Dollar, O.; Joshi, N.; Beck, D. A. C.; Pfaendtner, J. Attention-Based Generative Models for De Novo Molecular Design. *Chem. Sci.* **2021**, *12*, 8362-8372. DOI: 10.1039/D1SC01050F.

¹¹ Skalic, M.; Jiménez, J.; Sabbadin, D.; De Fabritiis, G. Shape-Based Generative Modeling for de Novo Drug Design. *J. Chem. Inf. Model.* **2019**, *59*, 1205-1214. DOI: 10.1021/acs.jcim.8b00706.

¹²Zhavoronkov, A.; Ivanenkov, Y. A.; Aliper, A.; Veselov, M. S.; Aladinskiy, V. A.; Aladinskaya, A. V.; Terentiev, V. A.; Polykovskiy, D. A.; Kuznetsov, M. D.; Asadulaev, A.; Volkov, Y.; Zholus, A.; Shayakhmetov, R. R.; Zhebrak, A.; Minaeva, L. I.; Zagribelnyy, B. A.; Lee, L.H.; Soll, R.; Madge, D.; Xing, L.; Guo, T.; Aspuru-Guzik, A. Deep Learning Enables Rapid Identification of Potent DDR1 Kinase Inhibitors. *Nat. Biotechnol.* **2019**, *37*, 1038-1040. DOI: 10.1038/s41587-019-0224-x.

¹³ Mao, J.; Wang, J.; Zeb, A.; Cho, K.-H.; Jin, H.; Kim, J.; Lee, O.; Wang, Y.; No, K. T. Transformer-Based Molecular Generative Model for Antiviral Drug Design. *J. Chem. Inf. Model.* **2023**. DOI: 10.1021/acs.jcim.3c00536.

¹⁴ Wang, J.; Mao, J.; Wang, M.; Le, X.; Wang, Y. Explore drug-like space with deep generative models. *Methods* **2023**, *210*, 52-59. DOI: <u>https://doi.org/10.1016/j.ymeth.2023.01.004</u>.

¹⁵ Kim, H.; Na, J.; Lee, W. B. Generative Chemical Transformer: Neural Machine Learning of Molecular Geometric Structures from Chemical Language via Attention. *J. Chem. Inf. Model.* **2021**, *61*, 5804-5814. DOI: 10.1021/acs.jcim.1c01289.

¹⁶ Wang, Y.; Zhao, H.; Sciabola, S.; Wang, W. cMolGPT: A Conditional Generative Pre-Trained Transformer for Target-Specific De Novo Molecular Generation. *Molecules* **2023**, *28*. DOI: 10.3390/molecules28114430.

¹⁷ Mercado, R.; Rastemo, T.; Lindelöf, E.; Klambauer, G.; Engkvist, O.; Chen, H.; Bjerrum, E.
J. Graph Networks for Molecular Design. *ChemRxiv* 2020. DOI: 10.26434/chemrxiv.12843137.v1.

¹⁸ Wang, W.; Wang, Y.; Zhao, H.; Sciabola, S. A Transformer-Based Generative Model for De Novo Molecular Design. *ArXiv* **2022**, DOI: https://doi.org/10.48550/arXiv.2210.08749.

¹⁹ Wei, L.; Fu, N.; Song, Y.; Wang, Q.; Hu, J. Probabilistic Generative Transformer Language Models for Generative Design of Molecules. *ArXiv* **2022**, DOI: <u>https://doi.org/10.48550/arXiv.2209.09406</u>.

²⁰ Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovanov, S.; Tatanov, O.; Belyaev, S.; Kurbanov, R.; Artamonov, A.; Aladinskiy, V.; Veselov, M.; Kadurin, A.; Johansson, S.; Chen, H.; Nikolenko, S.; Aspuru-Guzik, A.; Zhavoronkov, A. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Front. Pharmacol.* **2020**, *11*, 565644. DOI: 10.3389/fphar.2020.565644.

²¹ Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O,; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825-2830. DOI: http://jmlr.org/papers/v12/pedregosa11a.html.