Chemical Language Modeling with Structured State Spaces

Rıza Özçelik[1,2], Sarah de Ruiter[1], Emanuele Criscuolo[1], and Francesca Grisoni[1,2*]

[1]Eindhoven University of Technology, Institute for Complex Molecular Systems and Dept. Biomedical Engineering, Eindhoven, Netherlands.
[2]Centre for Living Technologies, Alliance TU/e, WUR, UU, UMC Utrecht, Netherlands.
*e-mail: f.grisoni@tue.nl

Abstract

Generative deep learning is reshaping drug design. Chemical language models (CLMs) – which generate molecules in the form of molecular strings – bear particular promise for this endeavor. Here, we introduce a recent deep learning architecture, termed Structured State-Space Sequence (S4) model, into *de novo* drug design. In addition to its unprecedented performance in various fields, S4 has a remarkable capability to capture the global properties of long sequences. This aspect is key for chemical language modeling, where complex molecular properties like bioactivity can 'emerge' from distant positions in the molecular strings. This observation gives rise to the following question: *Can S4 advance chemical language modeling for de novo design*? To provide an answer, we systematically benchmark S4 with state-of-the-art CLMs on an array of drug discovery tasks, such as the identification of bioactive compounds, and the design of drug-like molecules and natural products. S4 showed a superior capacity to learn complex molecular properties, while at the same time exploring diverse scaffolds. Finally, when applied prospectively to kinase inhibition, S4 designed eight of out ten molecules that were predicted as highly active by molecular dynamics simulations. Taken together, these findings advocate for the introduction of S4 into chemical language modeling – uncovering its untapped potential in the molecular sciences.

1 Introduction

Designing molecules with desired properties from scratch is a '*needle in the haystack*' problem. The chemical universe – estimated to comprise up to 10^{60} small molecules [1] – remains largely uncharted. Generative deep learning offers unprecedented opportunities to explore the chemical universe in a time- and cost-efficient manner [2], by enabling the production of desirable molecules without the need for handcrafted design rules. In particular, chemical language models (CLMs) have yielded experimentally-validated bioactive designs [3–7] and stood out as powerful molecular generators [2, 8].

CLMs adapt algorithms developed for sequence processing to learn the 'chemical language', that is, how to generate molecules that are chemically valid (*syntax*) and possess desired properties (*semantics*) [7]. This is achieved by representing molecular structures as string notations, such as the Simplified Molecular Input Line Entry Systems (SMILES [9], Figure 1a), among others [10, 11]. These molecular strings are then used for model training and subsequent generation of molecules in textual form. Compared to generative methods based on molecular graphs [12], CLMs can learn more complex molecular properties better [8], and generate increasingly larger molecules more efficiently [13, 14]. These aspects have made CLMs become one of the *de facto* approaches for *de novo* drug design.

Several CLM architectures have been proposed for *de novo* design [15], the most popular of which are

long short-term memory (LSTM) [3–5, 16, 17] models. LSTMs are trained to produce molecular strings element-by-element and have fast generation capabilities. However, the iterative structure forces those models to compress the sequence into an information bottleneck and challenges the learning of global sequence properties [18–20]. Transformers [21], a more recent architecture, overcome this bottleneck by processing the entire input molecular string at once [22, 23] and can learn certain molecular properties better than LSTMs [20, 24–27]. While this holistic look at the input is promising, it makes Transformers computationally intensive for string generation, thereby potentially limiting their chemical space exploration capabilities. These aspects make it necessary to stretch the boundaries of current CLM approaches further, to chart the chemical space more effectively in search for bioactive molecules [20].

Structured state-space sequence models (S4s) are a recent member of the fast-growing family of state-space architectures [28–31], which are gathering increasing attention in the deep learning community [32–35]. S4s showed outstanding performance in audio, image, and text generation [30] and have a 'dual nature': they (a) are trained over the entire input sequences to learn complex global properties and (b) generate one string element at a time – thereby combining some respective strengths of Transformers and LSTMs. Motivated by such *'best of two worlds'* behavior, here we ask the following question: Can S4 advance the current state-of-the-art in chemical language modeling? We find evidence that it can.

Here, we apply S4 to chemical language modeling on SMILES strings and benchmark it on various tasks relevant to drug design – from learning bioactivity to chemical space exploration and natural-product design. Moreover, we further corroborate the promise of S4 via the prospective *de novo* design of kinase inhibitors, validated using molecular dynamics simulations. Our results show the promise of S4 for chemical language modeling, especially in capturing bioactivity and complex molecular properties. To the best of our knowledge, this is the first time that state space models have been applied to molecular tasks, and we expect their relevance for chemical language modeling to increase in the future.

2 Structured state-space sequence model (S4)

S4s are an extension of discrete state-space models, which are widely adopted in control engineering [36]. Discrete state-space models map an input sequence u to an output sequence y, through the learnable parameters $\overline{\boldsymbol{A}} \in \mathbb{R}^{N\times N}$, $\overline{\boldsymbol{B}} \in \mathbb{R}^{N\times 1}$, $\overline{\boldsymbol{C}} \in \mathbb{R}^{1\times N}$, and $\overline{\boldsymbol{D}} \in \mathbb{R}^{1\times 1}$, as follows:

$$\begin{aligned} x_k &= \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k \\ y_k &= \overline{\boldsymbol{C}} x_k \quad + \overline{\boldsymbol{D}} u_k. \end{aligned} \tag{1}$$

In other words, discrete state-space models define a *linear recurrence*: at any step k, the k-th element of the input sequence u_k is fed into the model and used to update the hidden state x_k and to generate an output, y_k. The matrices $\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}, \overline{\boldsymbol{C}}$, and $\overline{\boldsymbol{D}}$ control how the input and the hidden state are combined to provide an output (Figure 1b).

Besides their *recurrent formulation*, discrete state-space models can be formulated as a *convolution* with the same set of parameters. It can be demonstrated that, by 'unrolling' the linear recurrence (equation (1)), the output sequence y can be obtained via a learnable convolution over the input sequence u:

$$y = u * \overline{\boldsymbol{K}}, \tag{2}$$

where $\overline{\boldsymbol{K}}$ is the convolution filter, parameterized via $\overline{\boldsymbol{A}}$, $\overline{\boldsymbol{B}}$, and $\overline{\boldsymbol{C}}$ (*see* Supplementary Material for a detailed derivation). This *convolutional representation* reveals a key aspect of state-space models: they learn explicitly from the entire sequence (via global convolution) while preserving recurrent generation capabilities (Figure 1b).

Learning the optimal parameters of a discrete state-space system, however, introduces vanishing gradient problems both in recurrent and convolutional formulations. Structured state-space sequence models, (S4s) [30], tackle those issues by introducing additional structure to the model parameters (via the so-called high-order polynomial projection operators [28]) and reducing the unstable computations to the stable Cauchy kernel [37] computation (*see* [30] for more detail). Ablation studies [30] have

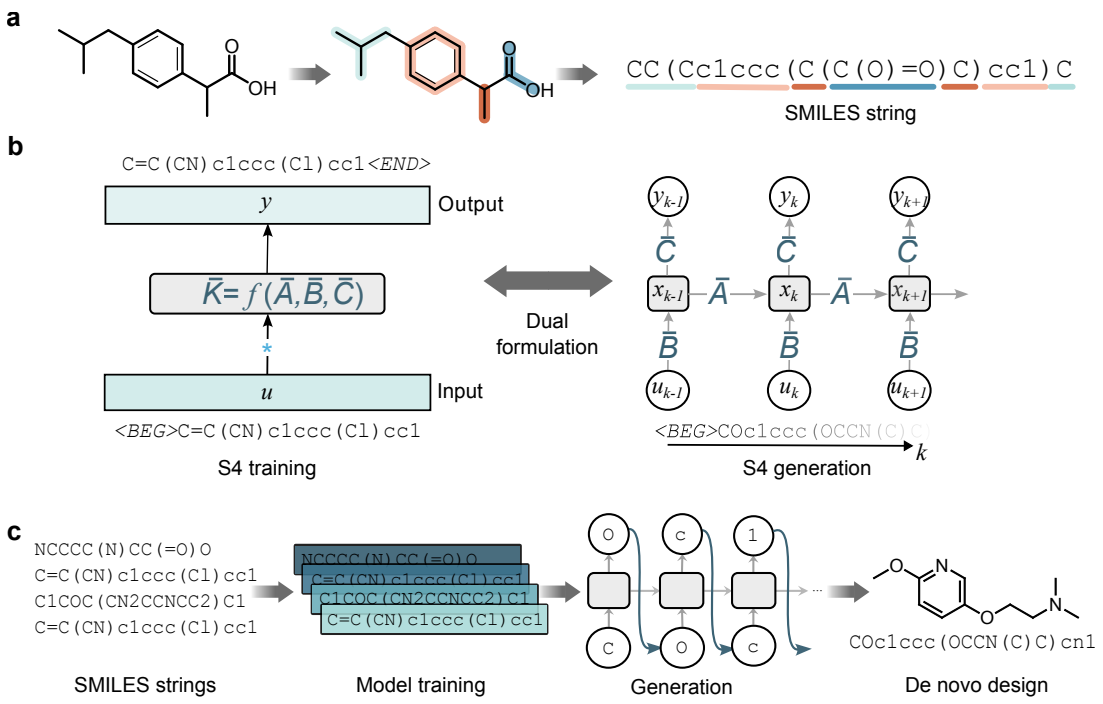

Figure 1: *Key concepts of structured State Space Sequence (S4) models for chemical language modeling.* **(a)** Simplified Molecular Input Line Entry System (SMILES) strings [9], used as the 'chemical language'. SMILES strings are obtained by traversing the molecular graph and annotating atom types, rings, and bond types in the form of text. **(b)** S4 for *de novo* SMILES design. During training, S4 is formulated as a global convolution and processes the whole molecular string simultaneously to better capture the properties of the molecule. The global convolution filter $\overline{\boldsymbol{K}}$ is parameterized via the matrices $\overline{\boldsymbol{A}}$, $\overline{\boldsymbol{B}}$, and $\overline{\boldsymbol{C}}$ (equation (2)). During the generation, S4 switches to the recurrent formulation (via the same parameters, (equation (1)) and produces SMILES strings element-by-element for more efficient and effective chemical space exploration. **(c)** Computational pipeline, where S4 was used to learn from known SMILES strings and generate new molecules *de novo*.

shown the relevance of the added structure to achieve computational feasibility and performance on long sequences. Moreover, such reduction allows S4 to address numerical instabilities encountered in model training and made S4 state-of-the-art in several generative tasks that require learning long-distance relationships [28–30]. Motivated by its performance in other domains and the potential benefits of its dual structure, here we introduce S4 to the molecular sciences for the first time.

3 Results and Discussion

We evaluated S4 for its ability to learn from and generate drug-like molecules and natural products in an array of tasks, and in terms of multiple molecular properties. LSTMs and Generative Pretrained Transformers (GPTs) were used as benchmarks, since they are the *de facto* approaches in chemical language modeling for *de novo* design [2, 7, 8, 20]. Furthermore, LSTM (recurrent training and generation) and GPT (holistic training and generation) constitute the ideal benchmarks for S4, due to S4's dual formulation (convolution during training and recurrence during generation), which allows inspecting the effect of each of these aspects on the overall performance. Finally, the prospective *de novo* design of putative MAPK1 inhibitors, corroborated by molecular dynamics simulations, was performed to test the potential of S4 in real-world drug discovery scenarios.

3.1 Designing drug-like molecules

S4 was analyzed for its ability to design drug-like small molecules (SMILES length lower than 100 tokens) extracted from ChEMBL database [38], by focusing on its ability to (a) learn the chemical syntax, (b) capture structural features relevant for bioactivity, and (c) designing structurally diverse molecular entities.

3.1.1 Learning the SMILES syntax

All investigated CLMs were trained on 1.9M canonical SMILES strings extracted from ChEMBL v31 [38]. The generated strings were evaluated according to their (a) *validity*, *i.e.*, the number (and frequency) of SMILES corresponding to chemically valid molecules; (b) *uniqueness*, which captures the number (and frequency) of structurally-unique molecules among the designs; and (c) *novelty*, corresponding to the number (and frequency) of unique and valid designs that are not included in the training set. A high number of 'chemically-valid' designs suggests that the model has learned how to generate plausible molecules, while high uniqueness and novelty values indicate little redundancy among the designs and with the training set, respectively. Although these metrics are vulnerable to trivial baselines [39], they provide insights into a model's capacity to learn the SMILES 'syntax'.

All CLMs generated more than 91% valid, 91% unique and 81% novel molecules (Table 1). Their designs approximated the training and test sets in terms of selected properties (Figure S1). These results agree with the literature on CLMs [2, 40] and demonstrate the robustness of the model training procedure. S4 designs the most valid, unique, and novel molecules, by generating more novel molecules than the benchmarks (from approximately 4,000 to 12,000 more), and displays a good ability to learn the 'chemical syntax' of SMILES strings.

To shed additional light on the strengths and limitations of S4 in comparison with the benchmarks, we analyzed the sources of invalid molecule generation for all methods in terms of branching and ring errors, erroneous bond assignment, and other (miscellaneous) syntax issues (Figure 2). Interestingly, each method seems to show different types of errors leading to SMILES invalidity. LSTM struggles the most with branching, and performs the best with bond assignment, while GPT struggles the most with rings and bond assignment, and has intermediate performance otherwise. S4 struggles more than LSTM with bond assignment, and generates remarkably fewer errors than both benchmarks in branching and ring design. Our hypothesis is that bond assignment indicates good learning of 'short-range' dependencies, while branching and ring opening and closure require better capturing of the 'long-range' relationships. This suggests that S4 captures long-distance

Table 1: *Designing drug-like molecules de novo with S4.* The results of LSTM and GPT models on the same tasks are also reported for comparison. Each model was trained on 1.9M SMILES strings from ChEMBL and used to generate 102,400 SMILES strings *de novo*. The number and percentage of valid, unique, and novel molecular designs are reported. The best value per metric is highlighted in boldface.

Model	Valid	Unique	Novel
S4	**99,268 (97%)**	**98,712 (96%)**	**95,552 (93%)**
LSTM	97,151 (95%)	96,618 (94%)	82,988 (81%)
GPT	93,580 (91%)	93,263 (91%)	91,590 (89%)

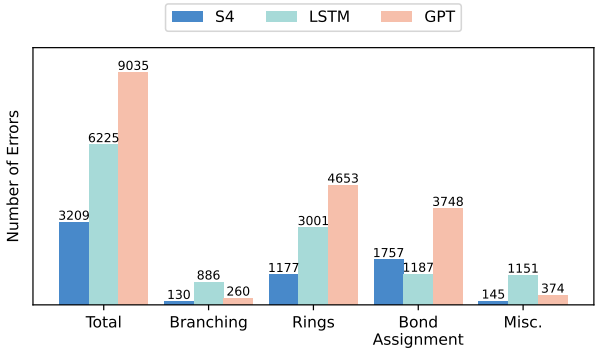

Figure 2: *SMILES design errors, grouped by category and CLM architecture.* Each CLM trained on ChEMBL was used to design 102,400 SMILES strings and the invalid designs are categorized per error. The values indicate the number of errors in each category.

relationships well, in agreement with existing evidence in other domains [28–30].

3.1.2 Capturing bioactivity

We evaluated S4 for its ability to learn elements of bioactivity. With CLMs this is often achieved with transfer learning [41], which allows transferring knowledge acquired from one task to another task with fewer available data. Via transfer learning, after pre-training a CLM on a large corpus of SMILES strings, the model can be then 'fine-tuned' on a smaller, and task-focused set (*e.g.*, bioactive molecules) by additional training [17]. Here, we performed five fine-tuning campaigns, focusing on distinct macromolecular targets from the LIT-PCBA [42] dataset: (1) pyruvate kinase muscle isoform 2 (PKM2), (2) mitogen-activated protein kinase 1 (MAPK1), (3) glucocerebrosidrase (GBA), (4) mechanistic target of rapamycin (mTORC1), and (5) cellular tumor antigen p53 (TP53).

Evaluating the bioactivity of *de novo* designs (besides synthesis and wet-lab testing) is non-trivial, since this property cannot be fully captured by traditional molecular descriptors, and might not be accurately predicted by quantitative structure-activity relationship models [43, 44]. Hence, we used experimentally-tested molecules to evaluate the capacity of a CLM to learn elements of bioactivity retrospectively. Several studies have shown that the likelihoods learned by a CLM during fine-tuning can be used to prioritize designs with high chances of being bioactive [45–47]. Based on the same principle, here we used the likelihoods learned by the CLMs to rank existing molecules and evaluate their capacity to prioritize bioactive compounds over inactive ones.

For each of the selected targets, bioactive molecules from LIT-PCBA (Table S1) were used for fine-tuning, with ten random training-validation-test splits (Figure S2). After fine-tuning the CLMs on each target, we proceeded as follows (for all training-test splits):

1. With each fine-tuned model and per each target, we predicted the likelihoods (equation (4)) of the SMILES strings in the respective test set. The considered test sets resemble a real-world scenario in terms of hit-rate, and they comprise 9 (mTORC) to 54 active molecules (PKM2) and 10,240 inactive molecules (except for TP53, containing 3,301 inactive molecules, Table S1);
2. We ranked the molecules of the test set according to the predicted likelihoods (equation (5));
3. For each target and each test set, we computed the fraction of actives ranked among the top 10, top 50, and top 100 molecules. The higher the number of active molecules ranked in early portions of the test set by a CLM, the better the

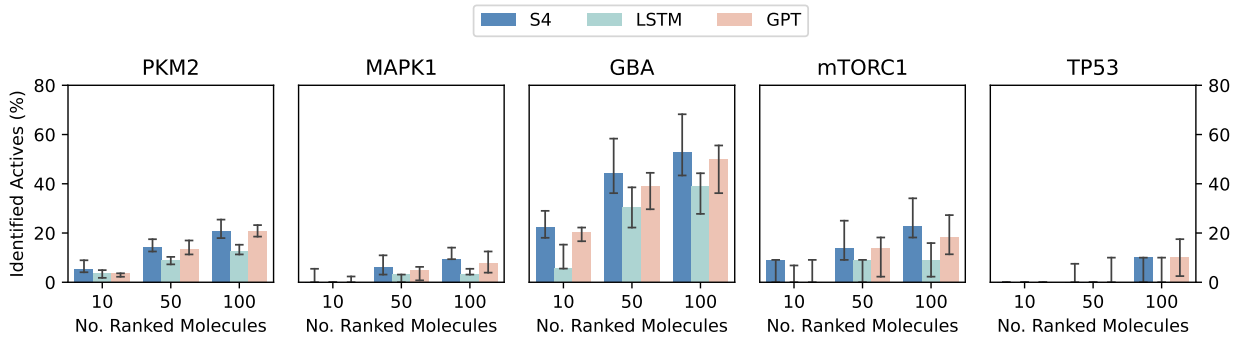

Figure 3: *Retrospective enrichment analysis for all models across five selected macromolecular targets.* The fine-tuned models were used to rank (*see* Section 5.1.4) held-out actives and inactives of the respective protein targets. The ratio of the test set actives in increasing list sizes (10, 50, 100) is computed across 10 runs of the models. Bar heights report the median across runs and error bars report the first and third quartiles.

model has learned what is relevant for bioactivity on the investigated target.

Our results show variable performance depending on the target (Figure 3). The most challenging target is TP53, on which no model could consistently retrieve actives among the top 10 scoring molecules. Notably, this target has the most challenging test set, where inactive molecules are similar to the actives of both the training and the test sets (Figure S2), potentially indicating the presence of activity cliffs [48]. MAPK1 and mTORC1 also challenge the CLMs; here, S4 retrieved more active molecules than the benchmarks, especially in the early portions of the test set. PKM2 and GBA are the easiest datasets; here, all CLMs identified bioactive molecules in their top 10, with S4 achieving the highest median across the board. A Wilcoxon signed-ranked test [49] on the pooled scores across datasets supports the superior performance of S4 compared to the benchmarks ($p < 0.05$), and of GPT compared to LSTM ($p < 0.05$).

Under the constraints of the study design, these results indicate that processing the input SMILES 'holistically' (as GPT and S4 do) leads to capturing complex properties like bioactivity better, with a better performance obtained by S4.

3.1.3 Chemical space exploration

We analyzed the ability of S4 to explore the chemical space, in terms of generating structurally diverse and bioactive molecules. To this end, we employed a commonly-used strategy with CLMs, that is, varying the sampling temperature (T) to control chemical diversity [50]. T affects which elements of a string are generated by a weighted random sampling (equation (3)). When $T \to 0$ the most likely element (based on the CLM prediction) is selected as the next element of the sequence, while the higher the T, the more random the selections. $T = 1$ corresponds to using the CLM predictions as the sampling probability of each element at each generation step.

We experimented with an increasing sampling temperatures (from $T = 1.0$ to $T = 2.0$ with a step of 0.25). Each T value was used to generate 10,240 SMILES strings per model across the five chosen targets and all training-test splits. Then, we evaluated the designs based on three metrics (Figure 4):

- *The validity* of the generated strings, which captures how robust the model is to increasing degrees of randomness in preserving a correct syntax. The higher the validity, the better.
- *Rediscovery rate. De novo* design models are of-

ten evaluated for their capacity to reproduce existing molecules with experimentally verified biological activities [40, 44]. For this purpose, we used the held-out actives previously described for each target. Moreover, to 'relax' the criterion of rediscovery, we considered held-out actives with substructure similarity higher than 60% to a *de novo* design (as computed via Tanimoto similarity on extended connectivity fingerprints [51]) to compute rediscovery. Higher rediscovery rates in increased temperatures indicate that the model can explore regions related to bioactivity despite increased randomness.

- *Scaffold diversity.* Designing molecules with novel scaffolds bears relevance in lead identification [52], and can be used as a proxy to evaluate CLMs [53]. Here, to have a better evaluation of what constitutes a novel scaffold, the novel designs were grouped in clusters based on their scaffold similarity. This was achieved via hierarchical clustering, to group designs with similar Bemis-Murcko scaffolds [54] (as computed via the Tanimoto similarity on the corresponding extended connectivity fingerprints [51] higher than 60%). Only novel and unique scaffolds were considered. We then counted the number of obtained scaffold clusters, the higher, the better.

The models display similar trends with increasing T values for all the analyzed factors across datasets, with varying magnitude (Figure 4). In general, the validity decreases with increasing temperature (as previously observed [50]), with the highest effect observed for GPT (median validity across training setups getting lower than 40%, Figure 4a).

Both S4 and LSTM show higher robustness than GPT to increasing temperature values (with LSTM performing slightly better for $T \geq 1.75$), suggesting that sequential generation can boost chemical space exploration. S4 outperforms LSTM in terms of rediscovery rate (Figure 4b), in agreement with our previous results on bioactivity (Figure 3). We also compute the exact rediscovery rate (identical molecular structure) and observe that no model can consistently generate held-out actives. When it comes to the diversity of the designs (Figure 4c), LSTM can

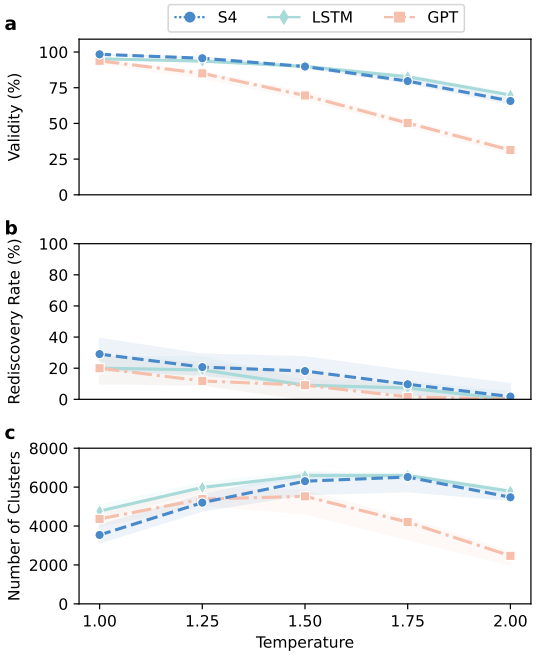

Figure 4: *Model performance when varying the temperature value.* Each model was analyzed for its performance when varying the temperature values from 1.0 to 2.0, with a step of 0.25, and sampling 10,240 molecules. **(a)** Analysis of the SMILES validity across temperature. **(b)** Variation of rediscovery rate. The models were evaluated for their capability to reconstruct (Tanimoto similarity of extended connectivity fingerprints higher than 60%) a set of bioactive molecules not used for model training. **(c)** Analysis of the number of diverse groups of scaffolds generated per method. Scaffolds were clustered together if they had a Tanimoto similarity (computed on extended connectivity fingerprints) larger than 60%. For each plot, the solid line indicates the median obtained across the five analyzed protein targets (PKM2, MAPK1, mTORC1, and TP53) and 10 runs, while the shaded area indicates the interquartile range. The statistics per individual target can be found in Figure S4.

generate the highest number of structurally unique scaffolds (median across datasets and setups: 6602, $T = 1.75$) and S4 is the close second-best model (6520, $T = 1.75$). While GPT obtains a suboptimal performance across the board, LSTM seems better for chemical space exploration when bioactivity is not the main objective, while S4 can better capture bioactivity and preserve a good chemical space exploration at the same time, combining the strengths of the two benchmarks with its dual structure. These results confirm the promise of S4 when it comes to generating structurally diverse and bioactive drug-like molecules.

3.2 Designing natural products

S4 was further tested on more challenging molecular entities than drug-like molecules. To this end, we evaluated its capacity to design natural products (NPs), which are invaluable sources of inspiration for medicinal chemistry [57, 58]. Compared to synthetic small molecules, NPs tend to possess more intricate molecular structures and ring systems, as well as a larger fraction of sp^3-hybridized carbon atoms and chiral centers [59–61]. These characteristics introduce longer SMILES sequences on average, with more long-range dependencies, and make natural products a challenging test case for CLMs [14, 62].

We trained the CLMs on large natural products (32,360 SMILES strings with length > 100, chosen to complement the previous analysis) from the COlleCtion of Open Natural ProdUcTs (COCONUT) database [55]. We then used the CLMs to design 102,400 SMILES strings *de novo* and computed the fraction of valid, unique, and novel designs (Table 2). All CLMs can design natural products, with lower performance compared to drug-like molecules. S4 designs the highest number of valid molecules by approximately 6,000 to 12,000 molecules (7% to 13% better), and LSTM achieves the highest novelty by approximately 2,000 molecules (2%) over S4.

To further investigate the characteristics of the designs, we computed the natural-product likeness [56], which captures how similar a molecule is to the chemical space covered by natural products in terms of its substructures (the higher the NP-likeness, the more similar). The novel designs of S4 have significantly higher (Mann-Whitney U test, $p < 0.01$) values of NP-likeness than the benchmarks, closer to the values of the training and test sets on average (Table 2). Moreover, the NP-likeness values better match the distribution of the COCONUT molecules in terms of Kolmogorov-Smirnov (KS) distance [63], which quantifies how much the cumulative distributions of two observations differ (between 0% and 100%; the lower, the closer the distributions).

In addition to NP-likeness, we evaluated the novel designs in terms of several structural properties important for natural products [59–61], namely: the number of sp^3-hybridized carbon atoms, aliphatic rings, spiro atoms and heavy atoms, as well as the molecular weight and the size of the largest fused ring system. These properties provide additional evidence on the molecular characteristics of the designs, and their structural complexity in comparison with the training natural products. Here, S4 achieved the lowest KS distance to the training and test sets across the board, indicating that its designs match the training natural products best. These results confirm the ability of S4 to learn complex molecular properties for *de novo* design.

Finally, we analyzed the training and generation speed of the CLM architectures when increasing the SMILES length, to test their practical applicability when designing bigger molecules. Our analysis highlighted that S4 is as fast as GPT during training (both are approximately 1.3 times faster than LSTM), and the fastest in terms of generation (Figure S5), thanks to its dual formulation. This further advocates for the introduction of S4 as an efficient approach for molecule design, that 'makes the best of both worlds' compared to GPT and LSTM.

3.3 Prospective *de novo* design

We conducted a prospective *in silico* study with S4, focused on designing inhibitors of mitogen-activated protein kinase 1 (MAPK1), a relevant target for oncological therapies [64]. The putative bioactivity of the designs was then evaluated via molecular dynamics (MD).

The S4 model previously pre-trained on ChEMBL

Table 2: *Natural product design with CLMs.* The models were trained on 32,360 natural product SMILES strings from the COCONUT database [55] and used to generate 102,400 SMILES strings *de novo*. The number and fraction of valid, unique, and novel molecular designs are calculated for each model. Mean and standard deviation of designs' (a) natural-product-likeness [56], (b) the number of sp^3 carbons, (c) the number of aliphatic rings, (d) the number of spiro atoms, (e) molecular weight, (f) size of the largest fused ring system, (g) the number of heavy atoms and the corresponding Kolmogorov-Smirnov distance to the training and test sets (KS_{train} and KS_{test}, respectively) are reported. The same statistics from train and test sets (32,360 and 5,000 natural products, respectively) are reported for comparison. For each CLM and each metric, the best value is highlighted in boldface. All descriptors were computed on valid, unique, and novel SMILES.

Metric		**S4**	**LSTM**	**GPT**	**Training**	**Test**
Syntax	Valid	**82,633 (81%)**	76,264 (74%)	70,117 (68%)	*n.a.*	*n.a.*
	Unique	**53,293 (52%)**	51,326 (50%)	50,487 (49%)	*n.a.*	*n.a.*
	Novel	40,897 (40%)	**43,245 (42%)**	43,168 (42%)	*n.a.*	*n.a.*
NP Likeness	Value	**1.6 ± 0.7**	1.5 ± 0.7	1.5 ± 0.7	1.6 ± 0.7	1.6 ± 0.7
	KS_{train}	**4.03%**	5.89%	9.44%	0.00%	0.81%
	KS_{test}	**4.51%**	6.60%	10.13%	0.81%	0.00%
No. sp^3 Carbons	Value	42 ± 16	44 ± 17	43 ± 16	38 ± 16	37 ± 15
	KS_{train}	**13.96%**	17.31%	14.51%	0.00%	1.02%
	KS_{test}	**14.08%**	17.45%	14.34%	1.02%	0.00%
No. Aliphatic Rings	Value	6 ± 4	6 ± 4	6 ± 4	7 ± 4	6 ± 4
	KS_{train}	**5.65%**	6.91%	8.12%	0.00%	1.08%
	KS_{test}	**5.41%**	6.25%	7.56%	1.08%	0.00%
No. Spiro Atoms	Value	0.3 ± 0.9	0.3 ± 0.8	0.3 ± 0.7	0.6 ± 1.2	0.6 ± 1.2
	KS_{train}	**10.81%**	12.88%	12.71%	0.00%	0.21%
	KS_{test}	**10.87%**	12.93%	12.77%	0.21%	0.00%
Molecular Weight	Value	1114 ± 315	1180 ± 360	1119 ± 307	1061 ± 295	1063 ± 290
	KS_{train}	**9.23%**	16.97%	11.02%	0.00%	1.40%
	KS_{test}	**9.04%**	16.67%	10.75%	1.40%	0.00%
Size of the Largest Fused Ring System	Value	5 ± 2	5 ± 2	5 ± 2	5 ± 2	5 ± 2
	KS_{train}	**8.05%**	9.42%	11.19%	0.00%	0.60%
	KS_{test}	**7.93%**	9.44%	11.21%	0.60%	0.00%
No. Heavy Atoms	Value	78 ± 22	83 ± 25	79 ± 21	75 ± 20	75 ± 20
	KS_{train}	**7.76%**	15.81%	9.73	0.00%	1.24%
	KS_{test}	**7.30%**	15.34%	9.31	1.24%	0.00%

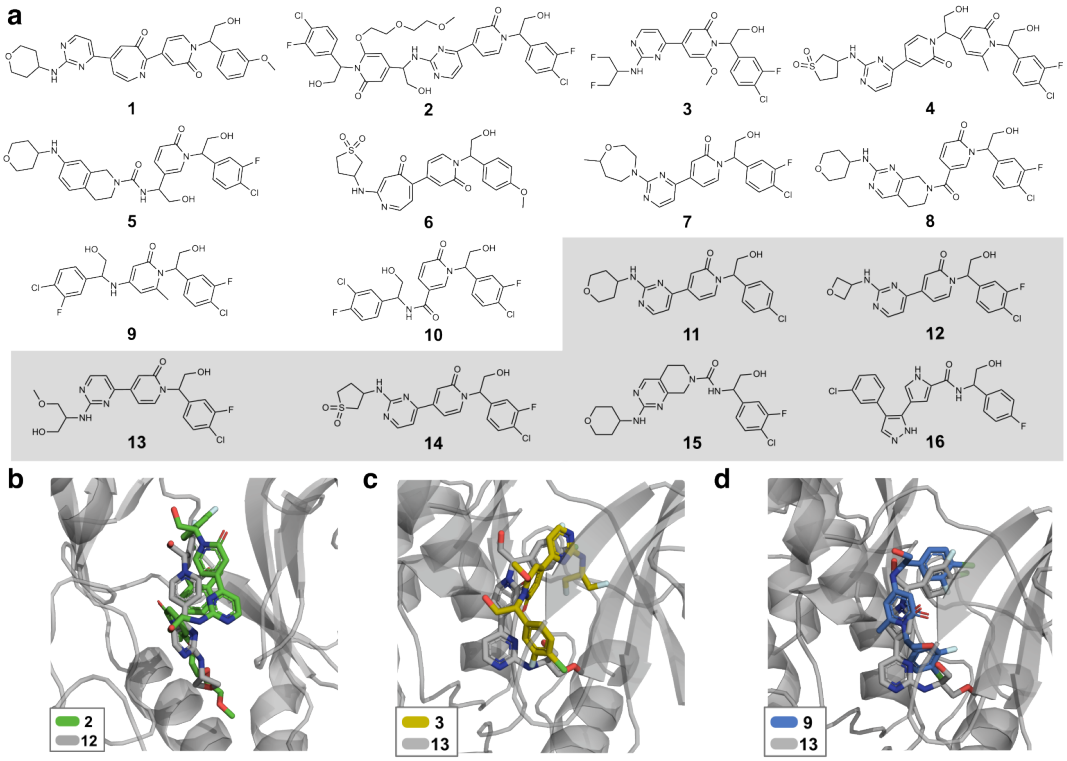

Figure 5: *Prospective de novo design of putative MAPK1 inhibitors with S4.* (**a**) Selected *de novo* designs (molecules **1** to **10**) for further characterization. For each *de novo* design, its most similar training MAPK1 inhibitor (as reported in Table 3) is depicted (compounds **11**-**16**, grey box). The ligand binding pose (obtained viaf Umbrella Sampling) of selected designs interacting with MAPK1 (PDB-ID= 2Y9Q), in comparison with their most similar bioactive molecule from the fine-tuning set is also depicted: (**b**) Design **2** (green) compared with compound **12** (grey). (**c**), Design **3** (yellow) compared with compound **13** (grey). (**d**) Design **9** (blue) compared with compound **13** (grey).

was fine-tuned with the SMILES strings of 68 manually-curated inhibitors from ChEMBL, having an experimental constant of inhibition (K_i) lower than 1 μM on MAPK1. The last five epochs of the fine-tuned model were then used to generate 256K molecules (51,200 designs per each T value, ranging from 1.0 to 2.0 with a step of 0.25).

The designs were ranked and filtered via log-likelihood score (equation (5)) and similarity to the training set (*see* Materials and Methods for further details). The ten top-scoring molecules (**1** to **10**, Figure 5a) were considered for further characterization using MD simulations. As a reference for evaluation, we performed MD simulations also for the closest fine-tuning neighbor of the considered designs (compounds **11** to **16**, selected based on scaffold Tanimoto similarity on extended connectivity fingerprints, Figure 5a). The absolute protein-ligand binding free energy (expressed as ΔG; the lower the stronger the predicted binding) for molecules **1**-**16** was computed via Umbrella Sampling [65] (Table 3). The computed ΔG values for known bioactive molecules (**11**-**16**) have a good correspondence with experimental K_i values from ChEMBL (Table 3), confirming the va-

Table 3: *In silico prospective study on designing mitogen-activated protein kinase (MAPK1) inhibitors with S4.* The absolute binding free energy of interaction (ΔG, the lower, the better) was determined via Umbrella Sampling. Values are reported as the average over three repeats, along with the corresponding standard deviation. *De novo* designs **1-10** are compared with the closest inhibitor from the fine-tuning set (selected based on Tanimoto similarity on extended connectivity fingerprints computed on scaffolds). The experimentally-determined inhibition of MAPK1 for compounds **11-16** (reported as constant of inhibition, K_i) are also reported.

S4 design		**Most similar training active**		
ID	**ΔG** [kcal/mol]	**ID**	**ΔG** [kcal/mol]	$\mathbf{K}_i$ [nM]
1	-5.6 ± 0.9	**11**	-9.1 ± 0.8	0.1
2	-23 ± 4	**12**	-12 ± 2	0.4
3	-19.6 ± 0.9	**13**	-10.5 ± 0.7	3.0
4	-13 ± 2	**14**	-11 ± 3	2.5
5	-7 ± 2	**15**	-13 ± 2	0.6
6	-11 ± 3	**14**	-11 ± 3	2.5
7	-10.3 ± 0.6	**11**	-9.1 ± 0.8	0.1
8	-11.2 ± 0.4	**15**	-13 ± 2	0.6
9	-17 ± 2	**13**	-10.5 ± 0.7	3.0
10	-15 ± 2	**16**	-9.1 ± 0.2	63.0

lidity of the chosen MD protocol.

Eight out of ten designs (except **1** and **5**) showed a high predicted affinity (Table 3), with ΔG values ranging from $\Delta G = -10.3 \pm 0.6$ kcal/mol (**7**) to $\Delta G = -23 \pm 4$ kcal/mol (**2**). Interestingly, these affinities are comparable or even surpassing those of the closest active neighbor ($\Delta G = -9.1 \pm 0.8$ kcal/mol to $\Delta G = -13 \pm 2$ kcal/mol).

The most potent design based on MD predictions is molecule **2** ($\Delta G = -23 \pm 4$ kcal/mol, Table 3). This molecule – which is the largest one among the designs (Figure 5a) – engages extensively with the binding pocket of MAPK1 (Figure 5b), which explains the remarkably favorable predicted affinity. It is important to highlight, however, that the synthetic accessibility of compound **2** might be limited. Design **3** is predicted with the second highest affinity ($\Delta G = -19.6 \pm 0.9$ kcal/mol), and it differs from compound **13** due to the replacement of the ether and hydroxy moieties with two fluorine atoms, and the addition of a methoxy group (Figure 5a). Interestingly, this structural modification leads to an improvement of the predicted ΔG value (of approximately -10 kcal/mol), possibly due to the ability to penetrate deeply into the binding pocket thanks to the fluorine atoms (Figure 5c). Halogens are, in fact, favorable for MAPK1, as evident from the fine-tuning molecules (91% of them containing at least one halogen) and existing literature (*e.g.*, [66–69]). Evidence of a favorable positioning of halogens is shown on both the 'top' [66, 67]) and 'bottom' [68, 69]) of the binding pocket, further supporting the predicted affinity of compound **3**.

Design **9** ($\Delta G = -17 \pm 2$ kcal/mol) features halogens on both sides, unlike its closest neighbor, molecule **13** ($\Delta G = -10.5 \pm 0.7$ kcal/mol), from which it also differs in the moiety attached to the pyridonic ring (Figure 5a). When inspecting the predicted binding pose, it can be observed that the aromatic ring with halogen substituents, hydroxyl, and carbonyl of pyridone are situated in the same region of the binding groove (Figure 5d). The difference in ΔG values (approximately 6.5 kcal/mol, in favor of design **9**) could be ascribed, like with molecule **3**, to the presence of halogens in the lower binding pocket region. This might also explain the high predicted affinity of design **10** ($\Delta G = -15 \pm 2$ kcal/mol) – which differs from **9** by a carbonyl and a methyl group.

With 8 out of 10 designs predicted as bioactive on the intended target, showing comparable or higher predicted affinities than their closest fine-tuning neighbors, these results further corroborate the promise of S4 for *de novo* drug design.

4 Conclusions

This study pioneered the introduction of state-space models into chemical language modeling, with a focus on structured state spaces (S4s). The unique dual nature of S4s, involving convolution during training and recurrent generation, makes them particularly intriguing for *de novo* design starting from SMILES

strings.

Our systematic analysis against GPT and LSTM on a variety of drug discovery tasks revealed S4's remarkable strengths: while recurrent generation (LSTM and S4) is superior in learning the chemical syntax and exploring diverse scaffolds, learning holistically on the entire SMILES sequence (GPT and S4) excels in capturing certain complex properties, like bioactivity. S4 with its dual nature, makes '*the best of both worlds*': it demonstrated comparable or better performance than LSTM in designing valid and diverse molecules, and systematically outperformed both benchmarks in capturing complex molecular properties – all while maintaining computational efficiency.

The application of S4 to MAPK1 inhibition, validated by MD simulations, further showcases its potential to design potent bioactive molecules. In the future, we will apply S4 prospectively in combination with wet-lab experiments to enhance its impact in the field. Strategies to increase the structural diversity of the considered designs, such as SMILES augmentation [70] and improved ranking protocols, could further boost its potential in medicinal chemistry.

Several aspects of S4 await to be explored in the molecular sciences, such as its potential with even longer sequences (*e.g.*, macrocyclic peptides and protein sequences) and on additional molecular tasks (*e.g.*, organic reaction planning [71] and structure-based drug design [72]).

In the future, we envision the relevance of S4 for molecule discovery to increase, and to potentially replace widely established chemical language models like LSTM and GPT. We believe that the provided open-access code will contribute to the adoption and expansion of S4, to further stretch the boundaries of chemical language modeling.

5 Methods

5.1 Designing drug-like molecules

5.1.1 Dataset creation

The pre-training set was generated starting from ChEMBL v31 [38]. Fine-tuning datasets were extracted from LIT-PCBA [42]. All sets were generated by (a) retaining molecules containing selected atoms (C, H, O, N, S, P, F, Cl, Br, and I), (b) removing salts and disconnected structures, as well as stereochemistry annotations and charge, (c) retaining molecules whose canonicalized SMILES strings contained 100 tokens or less. After sanitization, canonicalization, label encoding, and padding (to 100), molecules were randomly split into training, validation, and test sets. For ChEMBL, this led to a training set of 1,900,000, and a validation and a test set of 100,000 and 23,680 molecules, respectively. The number of compounds for each fine-tuning campaign is reported in Table S1.

5.1.2 Training

Pretraining. The hyper-parameters of the LSTM and GPT were tuned with random search for five days on a single NVIDIA A100 40GB GPU. The defined hyper-parameter space is based on previous work [22, 46, 50, 73] (Table S2). 40 LSTM and 35 GPT models were optimized within a five-day limit. Hyper-parameter search was conducted to maximize the validity during pre-training.

To account for the lack of previous information on optimal hyper-parameters for molecule generation with S4, we implemented a two-step procedure for hyper-parameter tuning. First, 242 models were trained to prioritize hyper-parameters (*see* supporting information, Table S2). High-performing hyper-parameter values in terms of validation accuracy were advanced to the second phase, where 108 experiments were conducted. Hyper-parameter search was conducted for 10 days on multiple NVIDIA A100 40GB GPUs to maximize the validity during pre-training.

Fine-tuning. Five fine-tuning campaigns were conducted on five proteins: PKM2, MAPK1, GBA,

mTORC1, and TP53. Ten runs with different training, validation, and test splits are run for each protein and early stopping on the validation cross-entropy is adopted with a patience of five epochs and a tolerance of 10^{-5}.

5.1.3 Temperature sampling

The sampling probability (p) of each i-th element at any step of the sequence was computed as follows:

$$p_i = \frac{e^{(y_i/T)}}{\sum_j e^{(y_j/T)}} \tag{3}$$

where y_i is the predicted probability of the i^{th} element, T is the sampling temperature, and j runs over all tokens in the vocabulary.

5.1.4 Molecule ranking with log-likelihoods

The molecules were ranked based on the joint likelihood of the tokens (*i.e.,* SMILES characters) they contain [73]. For each test molecule, the joint log-likelihood ($\mathcal{L}$) by a model ($\mathbf{M}$) was computed as:

$$\mathcal{L}(\mathbf{M}) = \sum_i \log p(t_i) \tag{4}$$

where t_i is the i^{th} token of the SMILES string of a given test molecule and $p(t_i)$ is the probability of that token as predicted by the model $\mathbf{M}$; i runs over all the elements in the molecular string.

To only consider the fine-tuning information and remove potential pre-training bias (as previously observed [73]), the pre-training log-likelihood was subtracted from the fine-tuning likelihood, to obtain a final score:

$$\mathcal{L}_{score}(\mathbf{M}) = \mathcal{L}(\mathbf{M_{ft}}) - \mathcal{L}(\mathbf{M_{pt}}) \tag{5}$$

where $\mathbf{M_{ft}}$ is the fine-tuned model and $\mathbf{M_{pt}}$ is the pre-trained model. The obtained $\mathcal{L}_{score}$ was used to rank each test molecule, the higher the $\mathcal{L}_{score}$, the better the rank.

5.2 Natural product design

The COlleCtion of Open Natural ProdUcTs (COCONUT) [55] database was used for model training. Salts, disconnected structures, stereochemistry, and charge annotations were removed. Molecules with canonical SMILES strings longer than 100 characters were used to train the models. A random search strategy was adopted to tune the hyper-parameters of all models, as previously explained. The models were given a five-day limit on a cloud NVIDIA A100 GPU and 1,024 strings were generated by each model. The models with the highest SMILES validity were selected for further evaluation.

5.3 Prospective *de novo* design

5.3.1 Data curation

Fine-tuning data were collected from ChEMBL v33 [38]. All annotations for MAPK1 were retained (target ID: CHEMBL4040). Available assay descriptions were manually inspected and analyzed. Molecules whose inhibitory constant (K_i) was lower than 1 μM on reliable inhibition assays (CHEMBL3412886, CHEMBL917079) were retained. SMILES canonicalization and removal of stereochemistry and duplicates led to a set of 68 unique SMILES strings for fine-tuning (SMILES available in the dedicated GitHub repository).

5.3.2 Model fine-tuning and *de novo* design

The fine-tuning dataset was split into ten train and validation splits (80% to 20%) to find the optimal number of fine-tuning epochs. Early stopping on validation loss was used, with patience of five epochs and tolerance of 10^{-5}. The experiments suggested 45 epochs to be optimal and then the pre-trained model was fine-tuned on the whole dataset accordingly.

The models of the last five fine-tuning epochs were used to design molecules. 10,240 designs for temperature values ranging from 1.0 to 2.0 (step size 0.25) were generated per temperature and model, totaling $5 \times 5 \times 10,240 = 256K$ designs. The novel and unique molecules among those designs were ranked by their fine-tuning log-likelihood (equation (4)) and

the top 5000 molecules were selected for further analysis.

The 5000 top-scoring molecules were divided into two groups, based on their similarity to the fine-tuning set (measured via Tanimoto similarity on extended connectivity fingerprints [51] of scaffolds) using a similarity threshold of 60%. The designs in the lists were grouped by their most similar training molecule and ranked by the log-likelihood score (equation (5)). The highest-scoring molecule in each group was picked. The top five molecules of the design lists (*i.e.*, **1-5** for higher similarity, and **6-10** for lower similarity) and their most similar actives (**11** to **16**) were selected for molecular dynamics simulations.

5.3.3 Molecular dynamics simulation

The three-dimensional structures of MAPK1 were sourced from the Protein Data Bank under the accession code 2Y9Q, characterized by a Resolution of 1.55Å and an R-Value Free of 0.177. Initial complex structures resulted from Docking simulations using Vina [74], establishing the binding pose for subsequent investigation via Umbrella Sampling. The setup of the simulation system was facilitated by the CHARMM-GUI web-based graphical interface [75]. A cubic water box with an edge distance of 13Å encapsulated the system, supplemented by a 0.15 M ionic NaCl solution for solvation neutralization. Gromacs software version 2021 [76], operationalized on the Dutch supercomputer Snellius, facilitated all simulations. The energy minimization of solvated systems involved a sequence of steps utilizing the steepest descent method and the conjugate gradient algorithm. Subsequently, equilibration occurred through 5 ns NPT (constant Number of atoms, Pressure, Temperature) ensemble after the first 1 ns NVT (constant Number of atoms, Volume, Temperature) ensemble.

5.3.4 Binding free energy calculation

The last conformations of the equilibration phase were used as the starting structures of ligand unbinding simulation. The distance-based Steered MD simulation (center-of-mass-pulling method) was used to pull the ligand away from the protein by approximately 30Å over the course of 4 ns by using a $1000, \mathrm{kJ/(mol\ nm^2)}$ force along the reaction coordinate (ξ), with a pulling speed (ν) set at $0.001, \mathrm{nm/ps}$. Snapshot intervals of 10 ps generated 400 configurations from these pulling simulations. Different ligands prompted the extraction of varying conformations, ranging from 22 to 28, along the reaction coordinate (ξ) at approximately 0.1 nm intervals. These distinct configurations were then employed as the initial points for individual Umbrella Sampling simulations, differing in quantity depending on the specific ligand under study. Each conformation underwent independent NPT equilibration for 5 ns, followed by a 20 ns MD run in triplicate for each ligand. The potential mean force (PMF) was determined via the weighted histogram analysis method (WHAM) [77], a component of Gromacs. The resultant PMF graphs depicted force in kcal/mol, representing the force required to dissociate the ligand from the binding pocket, against the corresponding distance. The computation of the binding free energy (ΔG) for each ligand involved comparing the plateau region of the PMF curve to the energy minimum obtained from each simulation. In total, the Umbrella Sampling simulations spanned 400 to 560 ns, comprising 3 replicates, thereby accumulating simulation times ranging from 1.2 μs to 1.6 μs for each ligand.

5.4 Software and code

Data preprocessing, molecular fingerprint, descriptor and scaffold calculation were conducted using RDKit v2020.09.01 in a Python environment. LSTM and GPT were implemented in Keras v2.7.0 (Tensorflow v2.7.1). The S4 code was extracted from the existing Pytorch-lightning v1.15.0 implementation [30] and simplified to rely solely on Pytorch v1.13.1.

Data and code availability

The Python code and data to replicate and extend our study are available on GitHub at the following URL: https://github.com/molML/s4-for-de-novo-drug-design.

Author Contribution

Conceptualization: RÖ and FG. *Data curation*: RÖ. *Formal Analysis*: all authors. *Investigation*: all authors. *Methodology*: all authors. *Software*: RÖ and SdR. *Visualization*: RÖ, FG and EC. *Writing – original draft*: RÖ and FG. *Writing – review and editing*: all authors.

Acknowledgements

This research was co-funded by the European Union (ERC, ReMINDER, 101077879). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. The authors also acknowledge support from the Irene Curie Fellowship, the Centre for Living Technologies, and SURF (NWO grant EINF-5406). The authors thank Selen Parlar and the Molecular Machine Learning team (H. Brinkmann, C. Izquierdo-Lozano, M. Reksoprodjo, L. Rossen, Y.G. Nana Teukam, D. van Tilborg, L. van Weesep) for their feedback on the manuscript.

References

[1] Bohacek, R. S., McMartin, C. & Guida, W. C. The art and practice of structure-based drug design: a molecular modeling perspective. *Medicinal research reviews* **16**, 3–50 (1996).

[2] Skinnider, M. A., Stacey, R. G., Wishart, D. S. & Foster, L. J. Chemical language models enable navigation in sparsely populated chemical space. *Nature Machine Intelligence* **3**, 759–770 (2021).

[3] Yuan, W. *et al.* Chemical space mimicry for drug discovery. *Journal of chemical information and modeling* **57**, 875–882 (2017).

[4] Merk, D., Friedrich, L., Grisoni, F. & Schneider, G. De novo design of bioactive small molecules by artificial intelligence. *Molecular informatics* **37**, 1700153 (2018).

[5] Grisoni, F. *et al.* Combining generative artificial intelligence and on-chip synthesis for de novo drug design. *Science Advances* **7**, eabg3338 (2021).

[6] Ballarotto, M. *et al.* De novo design of nurr1 agonists via fragment-augmented generative deep learning in low-data regime. *Journal of Medicinal Chemistry* (2023).

[7] Grisoni, F. Chemical language models for de novo drug design: Challenges and opportunities. *Current Opinion in Structural Biology* **79**, 102527 (2023).

[8] Flam-Shepherd, D., Zhu, K. & Aspuru-Guzik, A. Language models can learn complex molecular distributions. *Nature Communications* **13**, 3293 (2022).

[9] Weininger, D. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences* **28**, 31–36 (1988).

[10] Krenn, M. *et al.* Selfies and the future of molecular string representations. *Patterns* **3** (2022).

[11] O'Boyle, N. & Dalke, A. Deepsmiles: an adaptation of smiles for use in machine-learning of chemical structures. *ChemRxiv* (2018).

[12] Atz, K., Grisoni, F. & Schneider, G. Geometric deep learning on molecular representations. *Nature Machine Intelligence* **3**, 1023–1032 (2021).

[13] Abate, C., Decherchi, S. & Cavalli, A. Graph neural networks for conditional de novo drug design. *Wiley Interdisciplinary Reviews: Computational Molecular Science* e1651 (2023).

[14] Ochiai, T. *et al.* Variational autoencoder-based chemical latent space for large molecular structures with 3d complexity. *Communications Chemistry* **6**, 249 (2023).

[15] Wang, M. *et al.* Deep learning approaches for de novo drug design: An overview. *Current Opinion in Structural Biology* **72**, 135–144 (2022). URL `https://www.sciencedirect.com/science/article/pii/S0959440X21001433`.

[16] Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural computation* **9**, 1735–1780 (1997).

[17] Segler, M. H., Kogej, T., Tyrchan, C. & Waller, M. P. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science* **4**, 120–131 (2018).

[18] Bahdanau, D., Cho, K. H. & Bengio, Y. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015* (2015).

[19] Gómez-Bombarelli, R. *et al.* Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science* **4**, 268–276 (2018).

[20] Chen, Y. *et al.* Molecular language models: Rnns or transformer? *Briefings in Functional Genomics* elad012 (2023).

[21] Vaswani, A. *et al.* Attention is all you need. *Advances in neural information processing systems* **30** (2017).

[22] Bagal, V., Aggarwal, R., Vinod, P. & Priyakumar, U. D. Molgpt: molecular generation using a transformer-decoder model. *Journal of Chemical Information and Modeling* **62**, 2064–2076 (2021).

[23] Yang, L. *et al.* Transformer-based generative model accelerating the development of novel braf inhibitors. *ACS omega* **6**, 33864–33873 (2021).

[24] Wang, S., Guo, Y., Wang, Y., Sun, H. & Huang, J. Smiles-bert: large scale unsupervised pre-training for molecular property prediction. In *Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics*, 429–436 (2019).

[25] Honda, S., Shi, S. & Ueda, H. R. Smiles transformer: Pre-trained molecular fingerprint for low data drug discovery. *arXiv preprint arXiv:1911.04738* (2019).

[26] Lim, S. & Lee, Y. O. Predicting chemical properties using self-attention multi-task learning based on smiles representation. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 3146–3153 (IEEE, 2021).

[27] Jiang, J. *et al.* Trangru: focusing on both the local and global information of molecules for molecular property prediction. *Applied Intelligence* **53**, 15246–15260 (2023).

[28] Gu, A., Dao, T., Ermon, S., Rudra, A. & Ré, C. Hippo: Recurrent memory with optimal polynomial projections. *Advances in neural information processing systems* **33**, 1474–1487 (2020).

[29] Gu, A. *et al.* Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems* **34**, 572–585 (2021).

[30] Gu, A., Goel, K. & Ré, C. Efficiently modeling long sequences with structured state spaces. In *The International Conference on Learning Representations (ICLR)* (2022).

[31] Fu, D. Y. *et al.* Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052* (2022).

[32] Lu, C. *et al.* Structured state space models for in-context reinforcement learning. *arXiv preprint arXiv:2303.03982* (2023).

[33] Nguyen, E. *et al.* Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *arXiv preprint arXiv:2306.15794* (2023).

[34] Gu, A. & Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752* (2023).

[35] Ma, J., Li, F. & Wang, B. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722* (2024).

[36] Hamilton, J. D. State-space models. *Handbook of econometrics* **4**, 3039–3080 (1994).

[37] Pan, V. Fast approximate computations with cauchy matrices and polynomials. *Mathematics of Computation* **86**, 2799–2826 (2017).

[38] Gaulton, A. *et al.* The chembl database in 2017. *Nucleic acids research* **45**, D945–D954 (2017).

[39] Renz, P., Van Rompaey, D., Wegner, J. K., Hochreiter, S. & Klambauer, G. On failure modes in molecule generation and optimization. *Drug Discovery Today: Technologies* **32**, 55–63 (2019).

[40] Brown, N., Fiscato, M., Segler, M. H. & Vaucher, A. C. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling* **59**, 1096–1108 (2019).

[41] Weiss, K., Khoshgoftaar, T. M. & Wang, D. A survey of transfer learning. *Journal of Big data* **3**, 1–40 (2016).

[42] Tran-Nguyen, V.-K., Jacquemard, C. & Rognan, D. Lit-pcba: an unbiased data set for machine learning and virtual screening. *Journal of chemical information and modeling* **60**, 4263–4273 (2020).

[43] van Tilborg, D., Alenicheva, A. & Grisoni, F. Exposing the limitations of molecular machine learning with activity cliffs. *Journal of Chemical Information and Modeling* **62**, 5938–5951 (2022).

[44] Weng, G. *et al.* Rediscmol: Benchmarking molecular generation models in biological properties. *Journal of Medicinal Chemistry* (2024).

[45] Laban, P., Wu, C.-S., Liu, W. & Xiong, C. Near-negative distinction: Giving a second life to human evaluation datasets. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2094–2108 (2022).

[46] Moret, M., Helmstädter, M., Grisoni, F., Schneider, G. & Merk, D. Beam search for automated design and scoring of novel ror ligands with machine intelligence. *Angewandte Chemie International Edition* **60**, 19477–19482 (2021).

[47] Ballarotto, M. *et al.* De novo design of nurr1 agonists via fragment-augmented generative deep learning in low-data regime. *Journal of Medicinal Chemistry* **66**, 8170–8177 (2023). URL `https://doi.org/10.1021/acs.jmedchem.3c00485`. PMID: 37256819, `https://doi.org/10.1021/acs.jmedchem.3c00485`.

[48] Maggiora, G. M. On outliers and activity cliffs why qsar often disappoints (2006).

[49] Woolson, R. F. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials* 1–3 (2007).

[50] Moret, M., Friedrich, L., Grisoni, F., Merk, D. & Schneider, G. Generative molecular design in low data regimes. *Nature Machine Intelligence* **2**, 171–180 (2020).

[51] Rogers, D. & Hahn, M. Extended-connectivity fingerprints. *Journal of chemical information and modeling* **50**, 742–754 (2010).

[52] Schneider, G., Schneider, P. & Renner, S. Scaffold-hopping: how far can you jump? *QSAR & Combinatorial Science* **25**, 1162–1171 (2006).

[53] Polykovskiy, D. *et al.* Molecular sets (moses): a benchmarking platform for molecular generation models. *Frontiers in pharmacology* **11**, 565644 (2020).

[54] Bemis, G. W. & Murcko, M. A. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry* **39**, 2887–2893 (1996).

[55] Sorokina, M., Merseburger, P., Rajan, K., Yirik, M. A. & Steinbeck, C. Coconut online: collection of open natural products database. *Journal of Cheminformatics* **13**, 1–13 (2021).

[56] Ertl, P., Roggo, S. & Schuffenhauer, A. Natural product-likeness score and its application for prioritization of compound libraries. *Journal of chemical information and modeling* **48**, 68–74 (2008).

[57] Harvey, A. L., Edrada-Ebel, R. & Quinn, R. J. The re-emergence of natural products for drug discovery in the genomics era. *Nature reviews drug discovery* **14**, 111–129 (2015).

[58] Atanasov, A. G., Zotchev, S. B., Dirsch, V. M. & Supuran, C. T. Natural products in drug discovery: advances and opportunities. *Nature reviews Drug discovery* **20**, 200–216 (2021).

[59] Lee, M.-L. & Schneider, G. Scaffold architecture and pharmacophoric properties of natural products and trade drugs: application in the design of natural product-based combinatorial libraries. *Journal of combinatorial chemistry* **3**, 284–289 (2001).

[60] Henkel, T., Brunne, R. M., Müller, H. & Reichel, F. Statistical investigation into the structural complementarity of natural products and synthetic compounds. *Angewandte Chemie International Edition* **38**, 643–647 (1999).

[61] Chen, Y., Rosenkranz, C., Hirte, S. & Kirchmair, J. Ring systems in natural products: structural diversity, physicochemical properties, and coverage by synthetic compounds. *Natural Product Reports* **39**, 1544–1556 (2022).

[62] Merk, D., Grisoni, F., Friedrich, L. & Schneider, G. Tuning artificial intelligence on the de novo design of natural-product-inspired retinoid x receptor modulators. *Communications Chemistry* **1**, 68 (2018).

[63] Smirnov, N. On the estimation of the discrepancy between empirical distribution for two independent samples. *Bull. Math. Univ. Moscow* **2**, 2 (1939).

[64] Braicu, C. *et al.* A comprehensive review on mapk: a promising therapeutic target in cancer. *Cancers* **11**, 1618 (2019).

[65] Kästner, J. Umbrella sampling. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **1**, 932–942 (2011).

[66] Aronov, A. M. *et al.* Flipped out: structure-guided design of selective pyrazolylpyrrole erk inhibitors. *Journal of medicinal chemistry* **50**, 1280–1287 (2007).

[67] Chaikuad, A. *et al.* A unique inhibitor binding site in erk1/2 is associated with slow binding kinetics. *Nature chemical biology* **10**, 853–860 (2014).

[68] Blake, J. F. *et al.* Discovery of 5, 6, 7, 8-tetrahydropyrido [3, 4-d] pyrimidine inhibitors of erk2. *Bioorganic & medicinal chemistry letters* **24**, 2635–2639 (2014).

[69] Liu, F. *et al.* Structure-based optimization of pyridoxal 5'-phosphate-dependent transaminase enzyme (bioa) inhibitors that target biotin biosynthesis in mycobacterium tuberculosis. *Journal of medicinal chemistry* **60**, 5507–5520 (2017).

[70] Bjerrum, E. J. Smiles enumeration as data augmentation for neural network modeling of molecules. *arXiv preprint arXiv:1703.07076* (2017).

[71] Schwaller, P. *et al.* Molecular transformer: a model for uncertainty-calibrated chemical reaction prediction. *ACS central science* **5**, 1572–1583 (2019).

[72] Özçelik, R., van Tilborg, D., Jiménez-Luna, J. & Grisoni, F. Structure-based drug discovery with deep learning. *ChemBioChem* e202200776 (2023).

[73] Moret, M., Grisoni, F., Katzberger, P. & Schneider, G. Perplexity-based molecule ranking and bias estimation of chemical language models. *Journal of chemical information and modeling* **62**, 1199–1206 (2022).

[74] Eberhardt, J., Santos-Martins, D., Tillack, A. F. & Forli, S. Autodock vina 1.2. 0: New docking methods, expanded force field, and python bindings. *Journal of chemical information and modeling* **61**, 3891–3898 (2021).

[75] Lee, J. *et al.* Charmm-gui input generator for namd, gromacs, amber, openmm, and charmm/openmm simulations using the charmm36 additive force field. *Biophysical journal* **110**, 641a (2016).

[76] Abraham, M. J. *et al.* Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **1**, 19–25 (2015).

[77] Hub, J. S., De Groot, B. L. & van der Spoel, D. g_wham: A free weighted histogram analysis implementation including robust error and autocorrelation estimates. *Journal of chemical theory and computation* **6**, 3713–3720 (2010).

[78] Wildman, S. A. & Crippen, G. M. Prediction of physicochemical parameters by atomic contributions. *Journal of chemical information and computer sciences* **39**, 868–873 (1999).

[79] Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S. & Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nature chemistry* **4**, 90–98 (2012).

[80] Bertz, S. H. The first general index of molecular complexity. *Journal of the American Chemical Society* **103**, 3599–3601 (1981).

[81] Ertl, P. & Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics* **1**, 1–11 (2009).

[82] Levenshtein, V. I. *et al.* Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, vol. 10, 707–710 (Soviet Union, 1966).

[83] Coley, C. W., Rogers, L., Green, W. H. & Jensen, K. F. Scscore: synthetic complexity learned from a reaction corpus. *Journal of chemical information and modeling* **58**, 252–261 (2018).

Supplementary Material

Dual formulation of discrete state-space models

Discrete state space models (SSMs) are typically defined with the following equations:

$$\begin{aligned} x_k &= \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k \\ y_k &= \overline{\boldsymbol{C}} x_k \quad + \overline{\boldsymbol{D}} u_k, \end{aligned} \tag{S1}$$

where u_k is the k^{th} element of the input sequence, x_k is the state vector after processing the input u_k, and y_k is the k^{th} element of the output sequence. The matrices, $\overline{\boldsymbol{A}} \in \mathbb{R}^{N\times N}$, $\overline{\boldsymbol{B}} \in \mathbb{R}^{N\times 1}$, $\overline{\boldsymbol{C}} \in \mathbb{R}^{1\times N}$, and $\overline{\boldsymbol{D}} \in \mathbb{R}^{1\times 1}$ are the (learnable) parameters of the model. Overall, a discrete SSM defines a sequence-to-sequence mapping, where the mapping is defined via the matrices.

So, SSMs are recurrence relations and can be used for auto-regressive tasks (*e.g.,* generation), where x_k is used as the state of the recurrent computation. Figure 1c illustrates the computation diagram of the SSM defined by equation (S1), which resembles the computation graph of a recurrent neural network with a skip connection.

Setting the initial state (x_0) to 0 and writing out the equation (S1) yields the following formula for the state variable x_k:

$$\begin{aligned} x_1 &= \overline{\boldsymbol{A}} x_0 \quad + \overline{\boldsymbol{B}} u_1 = \overline{\boldsymbol{B}} u_1 \\ x_2 &= \overline{\boldsymbol{A}} x_1 \quad + \overline{\boldsymbol{B}} u_2 = \overline{\boldsymbol{A}\boldsymbol{B}} u_1 + \overline{\boldsymbol{B}} u_2 \\ x_3 &= \overline{\boldsymbol{A}} x_2 \quad + \overline{\boldsymbol{B}} u_3 = \overline{\boldsymbol{A}}^2 \overline{\boldsymbol{B}} u_1 + \overline{\boldsymbol{A}\boldsymbol{B}} u_2 + \overline{\boldsymbol{B}} u_3 \\ &\vdots \\ x_k &= \overline{\boldsymbol{A}} x_{k-1} + \overline{\boldsymbol{B}} u_k = \overline{\boldsymbol{A}}^{k-1} \overline{\boldsymbol{B}} u_1 + \overline{\boldsymbol{A}}^{k-2} \overline{\boldsymbol{B}} u_2 + \cdots + \overline{\boldsymbol{B}} u_k. \end{aligned} \tag{S2}$$

Setting $\overline{\boldsymbol{D}} = 0$ (*i.e.,* disabling the skip connection from input to output), the output variable $y_k = \overline{\boldsymbol{C}} x_k + \overline{\boldsymbol{D}} u_k$ becomes:

$$y_k = \overline{\boldsymbol{C}\boldsymbol{A}}^{k-1} \overline{\boldsymbol{B}} u_1 + \overline{\boldsymbol{C}\boldsymbol{A}}^{k-2} \overline{\boldsymbol{B}} u_2 + \cdots + \overline{\boldsymbol{C}\boldsymbol{B}} u_k. \tag{S3}$$

Defining $\overline{\boldsymbol{K}}^i = \overline{\boldsymbol{C}\boldsymbol{A}}^i \overline{\boldsymbol{B}}$, equation (S3) becomes:

$$y_k = \sum_i \overline{\boldsymbol{K}}^{\boldsymbol{i}} u_{k-i} = u * \overline{\boldsymbol{K}}, \tag{S4}$$

where $*$ is the convolution operator. The derivation shows that the parameters that define the recurrence relation of an SSM can also define a convolution with an unbounded window. Therefore, an SSM can be formulated as a global convolution during training time to capture long-range dependencies and benefit from accelerated training in GPUs. During the test time, the learned matrices can be used to define a recurrence relation for fast auto-regressive generation.

Additional analysis

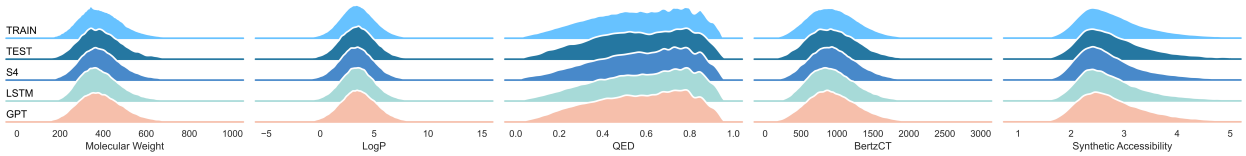

Figure S1: *Molecular descriptor distribution of pre-training designs.* All models are sampled 102,400 designs and molecular weight, octanol-water partition coefficient (Log P) [78], quantitative estimate of drug-likeness (QED) [79], Bertz complexity (BertzCT) [80], and synthetic accessibility [81] are computed.

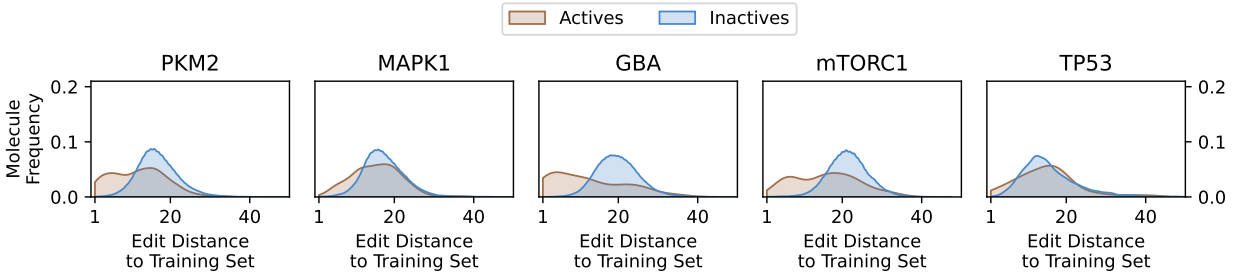

Figure S2: *Similarity of test set molecules to the training set.* Test sets across ten data splits are pooled together and the minimum edit distance [82] of each test set molecule to the respective training set is computed per actives and ianctives.

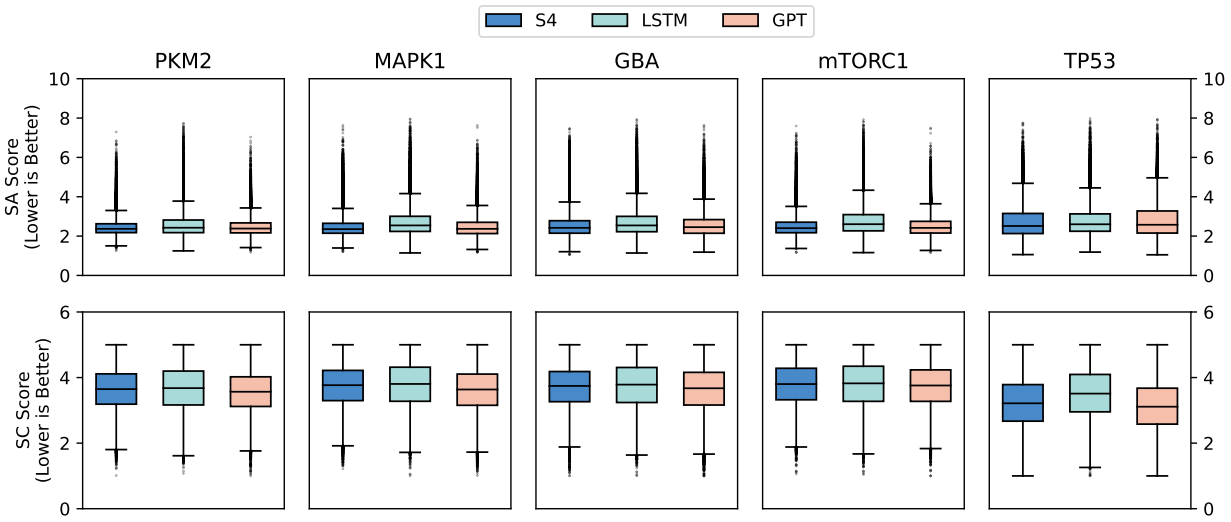

Figure S3: *Synthesizibility of the designs.* 10,240 designs were generated by each model per protein target and the synthetic accessibility (SA) score [81] and synthetic complexity (SC) score [83] were calculated.

Table S1: *Number of compounds used during the transfer-learnig phase.* Bioactive molecules were extracted from LIT-PCBA [42] database per each target and randomly divided into a training (80%), validation (10%) and test set (10%). Additionally, the test set contained 10,240 inactive molecules per target (chosen by random sampling). For TP53, all the inactive molecules available in the original dataset were considered.

Dataset	**Train**	**Valid.**	**Test**	
			Active	*Inact.*
PKM2	436	54	56	10,240
MAPK1	246	30	32	10,240
GBA	132	16	18	10,240
mTORC1	77	9	11	10,240
TP53	44	10	10	3,301

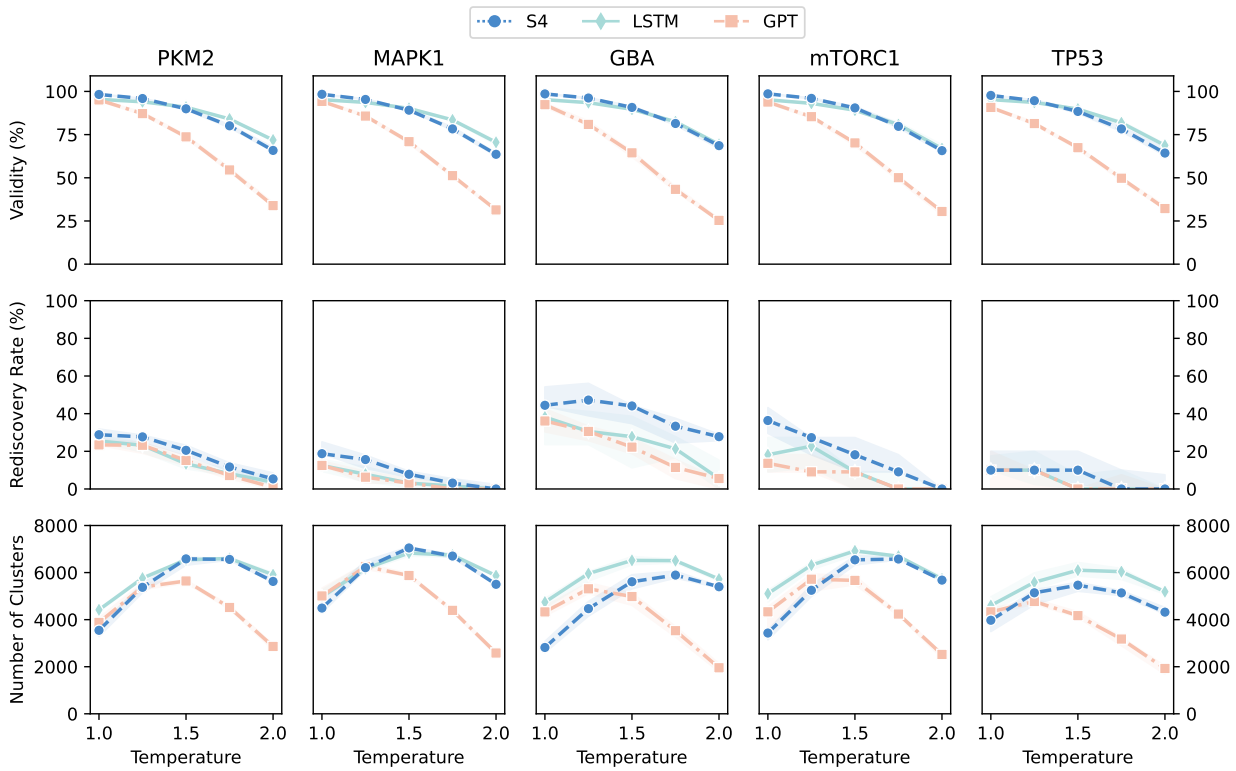

Figure S4: *Chemical space exploration per protein target.* The models were sampled in temperatures between 1 and 2 and validity, rediscovery rate (similarity above 60%), and number of scaffold clusters were computed.

Table S2: *Hyper-parameter candidates for the architectures.* The combinatorial space of hyper-parameters is defined via the values in the table. A random search strategy is used to sample the space and conduct the experiments.

Hyper-parameter	LSTM	GPT	S4 - Stage 1	S4 - Stage 2
LSTM layers	(64, 64, 64), (64, 64), (128, 64, 32), (128, 128), (128, 128, 128), (256, 128), (256, 128, 64), (256, 256), (1024, 512), (2048)	NA	NA	NA
# transformer blocks	NA	1, 2, 4, 6	NA	NA
# S4 blocks	NA	NA	1, 2, 4, 6, 8	4, 8
Model dimension	NA	NA	64, 128, 256	256, 512
Number of SSMs	NA	NA	1	1, 2, 4
Feed-forward dimension	NA	64, 128, 256	NA	NA
Embedding size	64, 128, 256	32, 64, 128, 256	64, 128, 256	256, 512
Dropout rate	0.0, 0.1, 0.2	0.0, 0.1, 0.2	0.0, 0.1, 0.2	0.0, 0.25
Learning rate	1e-2, 1e-3, 5e-4	1e-2, 1e-3, 5e-4	1e-2, 1e-3, 5e-4	1e-2, 1e-3, 5e-3
Batch size	64, 128, 512, 1024	64, 128, 512, 1024	64, 128, 512, 1024	2048
Embedding dropout	NA	NA	0.0, 0.1, 0.2	0.25
Softmax dropout	NA	NA	0.0, 0.1, 0.2	0.0

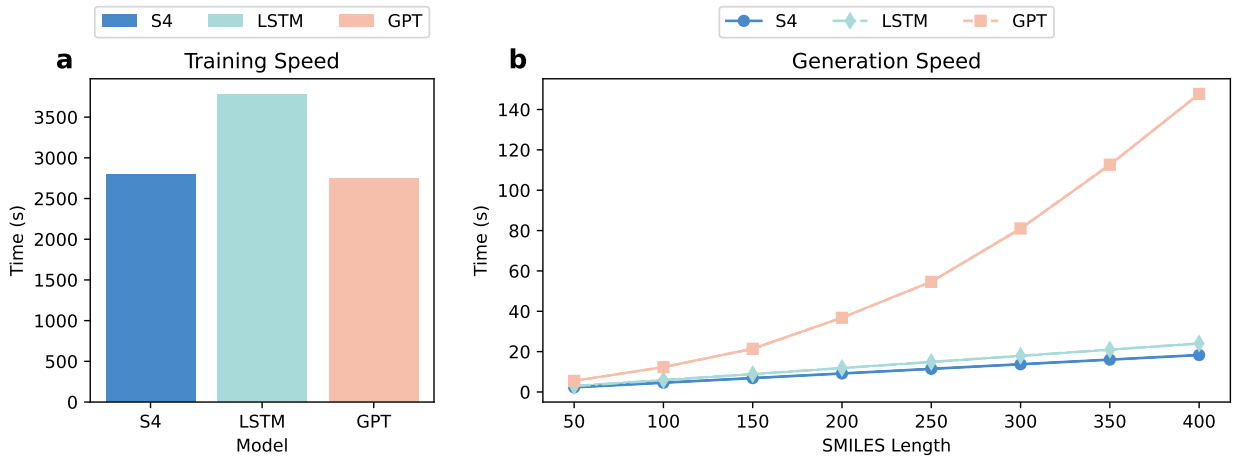

Figure S5: *Computational efficiency of training and molecule generation.* (a) 10 models are trained on sequences of length 450 and the mean training time is reported. (b) 10,240 designs are generated with 10 repetitions and mean generation time is measured in increasing lengths. A separate GPT model is trained per experimented SMILES length. Compute times are computed using an NVIDIA A100 40GB cloud GPU for both plots, and standard deviations are omitted for being negligible.