Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

REVIEW

Published on 04 August 2026

Cite this: DOI: 10.1039/d6dd00025h

Generative virtual screening: from search to generate and back

Morgan Thomas, *[a] Raul Astudillo,[b] Yasir S. Raouf,[c] Carles Navarro,[d] Gianni De Fabritiis[def] and Andreas Bender*[aghi]

A key challenge in drug discovery is the efficient search of chemical space to identify small molecules with desirable biological activity and developability. Computational approaches address this challenge by applying *in silico* evaluators to prioritise compounds for experimental validation. Traditionally, this has been achieved through brute-force virtual screening of large, predefined compound libraries. More recently, advances in generative molecular design (GMD) have enabled models that directly propose molecules optimised for specific objectives. GMD offers favourable computational cost but often at the expense of progressability, evaluability, and synthetic feasibility. In this review, we analyse molecular search strategies across a spectrum ranging from brute-force search to *de novo* generative design, focusing on how chemical space is specified and explored. We emphasise three practical axes—computational cost, progressability, and evaluability—and discuss how different paradigms trade off these properties. In particular, we highlight Generative Virtual Screening (GVS) as a hybrid approach in which generative optimisation is constrained to a fixed, explicit compound library. GVS combines the computational efficiency of generative methods with the progressability and retrospective evaluability of library-based search. This enables scalable exploration of chemical space while providing ground-truth reference points for algorithmic assessment. We argue that GVS offers both immediate practical benefits for hit discovery and a controlled framework for developing and benchmarking generative search algorithms, facilitating their principled extension to broader and less constrained chemical spaces.

Received 20th January 2026
Accepted 26th June 2026

DOI: 10.1039/d6dd00025h

rsc.li/digitaldiscovery

1 Introduction

The search for new biologically active small molecules (often referred to as "hits") is a critical step in drug discovery, forming the foundation for identifying new therapeutic candidates. This process is frequently described as a "needle in a haystack" problem:[1] an astronomically large virtual chemical space must be explored to identify a small number of molecules with suitable bioactivity, selectivity, and drug-like properties for a given context.[2] Because only a minute fraction of virtual compounds are ever synthesized or experimentally tested, the central challenge lies in efficiently prioritising which molecules to pursue. In practice, prioritisation depends on three components: the chemical space considered, the search method used, and the accuracy of computational evaluators. The latter determines how well experimental outcomes can be predicted.

In early-stage computational drug discovery, *in silico* evaluators used for the third component—the computational prediction of experimental success—primarily estimate the biological activity of a molecule, for example *via* quantitative structure–activity/property relationship (QSAR/QSPR) models or molecular docking simulations. In practice, however, the criteria defining a promising drug candidate are numerous and often involve trade-offs. Consequently, the computational methods used to approximate these criteria will influence which molecules are ultimately prioritised in a given discovery context. In this review, we remain deliberately agnostic to the specific evaluator(s) used to identify favourable molecules, as these have been extensively discussed elsewhere.[3–5] Instead, we focus on the first two components: how chemical space is defined and how it is searched.

[a]*Department of Medicine and Center for Biotechnology, College of Medicine and Health Sciences, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates. E-mail: morganthomas263@gmail.com; andreas.bender@ubbcluj.ro*

[b]*Department of Machine Learning, Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates*

[c]*Department of Chemistry, United Arab Emirates University, Al Ain, United Arab Emirates*

[d]*Acellera Labs, C Dr Trueta 183, 08005 Barcelona, Spain*

[e]*Computational Science Laboratory, Universitat Pompeu Fabra, Barcelona Biomedical Research Park (PRBB), C Dr Aiguader 88, 08003 Barcelona, Spain*

[f]*Institució Catalana de Recerca i Estudis Avançats (ICREA), Passeig Lluis Companys 23, 08010 Barcelona, Spain*

[g]*Centre for Molecular Informatics, Department of Chemistry, University of Cambridge, Lensfield Road, Cambridge CB2 1EW, UK*

[h]*STAR-UBB Institute, Babeş-Bolyai University, Cluj-Napoca, Romania*

[i]*Research Center for Functional Genomics, Biomedicine and Translational Medicine, Iuliu Hatieganu University of Medicine and Pharmacy, Cluj-Napoca, Romania*

A fundamental limitation in searching virtual chemical spaces is their enormous size. Estimates of the theoretical size of "drug-like" chemical space range from approximately 10^{33} molecules[6] to as many as 10^{60}.[7] The lower estimate does not account for beyond-Rule-of-5 compounds. These made up nearly 40% of FDA-approved oral drugs between 2013 and 2019,[8] suggesting that substantially larger estimates are plausible. Even storing 10^{33} molecules would be computationally infeasible. Extrapolating from GDB-17, this would require more than 10^{21} TB of compressed disk space†. As a result, practical virtual libraries necessarily represent only small subsets of chemical space. These include vendor libraries of commercially available compounds (deliverable within days to weeks), make-on-demand libraries built from validated synthetic routes (delivered within weeks to months),[10] and proprietary in-house collections. As summarised by Warr *et al.*,[11] such libraries range in size from approximately 10^6 to reported values as large as 10^{26}; despite representing only a vanishingly small fraction of the theoretical chemical space, searching these catalogues on reasonable timescales remains a substantial challenge.

The simplest approach to searching chemical space is brute-force search over pre-specified virtual libraries, which has been a cornerstone of computational hit discovery for decades.[2,12] When this exhaustive evaluation is applied specifically to estimating biological activity—most commonly using molecular docking evaluators[13]—it is referred to as Virtual Screening (VS). Advances in computational power and *in silico* evaluators (also termed oracles or scoring functions) have enabled VS campaigns to scale to billions of compounds.[14] In contrast, Generative Molecular Design (GMD)[15,16] introduces a different paradigm: rather than filtering a fixed library, generative models construct molecules *de novo* by sampling from programmed or learned representations of chemical space. Methodologically, brute-force search is a calculate-and-rank procedure over an explicit, finite space. In contrast, GMD samples from an implicit distribution with the potential to explore much larger theoretical chemical spaces.[17] Determining which of these paradigms—or which combination of them—provides the most effective strategy for identifying virtual hit molecules remains an open question.

In this review, we compare and contrast approaches to molecular search in computational drug discovery, spanning brute-force search, generative molecular design, and a spectrum of approximate and compute-optimised methods in between. The review is organised around dominant search paradigms in the literature, each corresponding to a distinct regime defined by how chemical space is specified and explored. Although these paradigms are often treated separately, their underlying algorithmic principles are not mutually exclusive. Our focus is on practical considerations, including the progressability of chemical matter, the computational efficiency and scalability of different search strategies, and how search performance can be evaluated. In particular, we examine Generative Virtual Screening (GVS) as a hybrid approach that balances progressability with computational scalability (illustrated in Fig. 1). Rather than providing exhaustive coverage of individual algorithms or model architectures, we highlight unifying principles, shared challenges, and points of divergence across paradigms, as illustrated in Section 2 and summarised in Table 1. In doing so, we seek to clarify the current strengths and limitations of each approach and their implications for the future of computational search in small-molecule discovery (Fig. 2).

† Assuming linear scaling from GDB-17, which requires approximately 400 GB of gzipped storage.[9] In practice, storage requirements would be considerably higher due to larger molecular representations beyond 17 heavy atoms.

2 Brute-force search

Brute-force search consists of computing properties for all molecules in a virtual library and subsequently ranking them. In drug discovery, these properties range from fast-to-evaluate descriptors, such as physicochemical properties or topological similarity to a query compound, to substantially slower simulations of molecule–protein binding *via* molecular docking. Virtual Screening (VS) typically targets the latter, focusing on the estimation of biological activity. It is these slow evaluations that constitute the principal computational bottleneck in VS: even a 1 s calculation applied to 1 billion molecules amounts to approximately 31.7 CPU years. Docking simulations commonly require 15–60 s per molecule. Assuming an average docking time of 30 s, screening 1 billion molecules with 1000 CPUs would still require approximately 1.1 years. Although such campaigns are increasingly tractable with large-scale compute resources and optimised implementations, current commercial libraries are already an order of magnitude larger,[10] and continue to grow.

2.1 Computational cost

The approximate computational cost for brute-force search is $C_{\text{total}} \approx Nc_{\text{f}}$, where c_{f} is the per-molecule evaluation cost, reflecting linear scaling with virtual library size N. While linear scaling is generally favourable in algorithmic terms, the

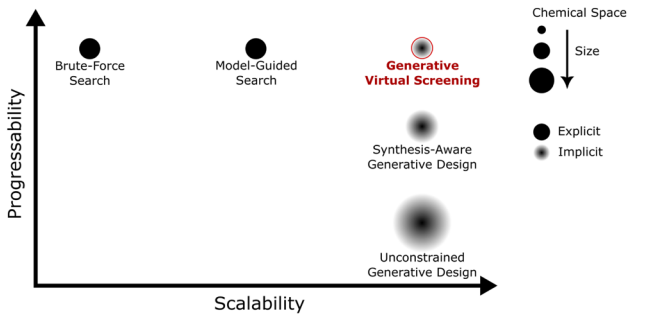

Fig. 1 High-level illustration of the trade-off between computational scalability and progressability in chemical space search. Approaches that operate over explicit compound libraries tend to offer high progressability but scale poorly with increasing library size, whereas methods that sample from implicit chemical spaces achieve favourable scaling at the cost of reduced progressability. Hybrid strategies aim to balance these competing objectives by combining elements of both regimes.

Table 1 Comparative overview of major paradigms for molecular search and generation in virtual chemical space. Approaches are compared qualitatively in terms of progressability, computational cost, and evaluability, reflecting practical considerations in early-stage hit discovery. Molecular properties of interest (*e.g.*, binding affinity) are approximated using an evaluator $f(m)$, which may be replaced or approximated by a surrogate model $s(m)$ in model-guided or generative settings. Qualitative indicators (+ to +++) denote relative performance within each category

Approach	Brute-force search	Model-guided search	Generative virtual screening (GVS)	Synthesis-aware GMD	Generative molecular design (GMD)
Progressability	+++	+++	+++	++	+
Scalability	+	++	+++	+++	+++
Evaluability	+++	+++	+++	+	+
Description	Brute-force calculation and ranking of molecular properties for a full virtual library	Fast approximation of slower calculations of molecular properties, to screen subsets of a virtual library	Generative design constrained to compound presence within an existing vendor library (or enumerated vendor library)	Generative design constrained to a set of synthetic rules, or guided by accurate measures of synthetic feasibility	Generative design according to molecular representation and generative algorithm
Progressability	Screening pre-specified libraries means compounds are accessible in days/weeks	Identical to brute-force	Identical to brute-force	Compounds still need to undergo synthesis, requiring weeks to months and many resources	Many compounds are commercially novel, and often not stable or synthesizable
Computational cost	Nc_f Evaluator cost (c_f) scales with size of library (N), where $N \gtrsim 10^9$	$kNc_s + Sc_f + C_{train}(S)$ Surrogate cost (c_s) scales with kN, where $kNc_s \ll Nc_f$	$n(c_g + c_f) + C_{update}(n)$ Identical to GMD	$n(c_g + c_f) + C_{update}(n)$ Identical to GMD	$n(c_g + c_f) + C_{update}(n)$ Sampling (c_g) and evaluation cost (c_f) scales with molecules sampled n, where $n \sim 10^{2-3}$
Retrospective evaluation	Accuracy/ranking enrichment achieved by $f(m)$ with respect to experimental values	Recall achieved by using $s(m)$ to approximate $f(m)$	Often evaluated the same as GMD, but can be evaluated identical to model-guided search	Identical to GMD	Assuming that $f(m)$ is 100% accurate, by how much and how fast $f(m)$ has been improved

combination of ultra-large library sizes (typically $N > 10^9$) and computationally expensive evaluators $f(m)$ renders brute-force search prohibitively costly in practice. As a result, ultra-large VS campaigns require extensive parallelisation across many CPUs or GPUs and are typically executed on high-performance computing (HPC) systems or cloud computing platforms. Access to shared HPC resources must be coordinated with other users, often limiting throughput. Cloud-based screening can also be costly: for example, screening 12.7 million molecules on commercial cloud infrastructure was reported to cost $20 300,[18]

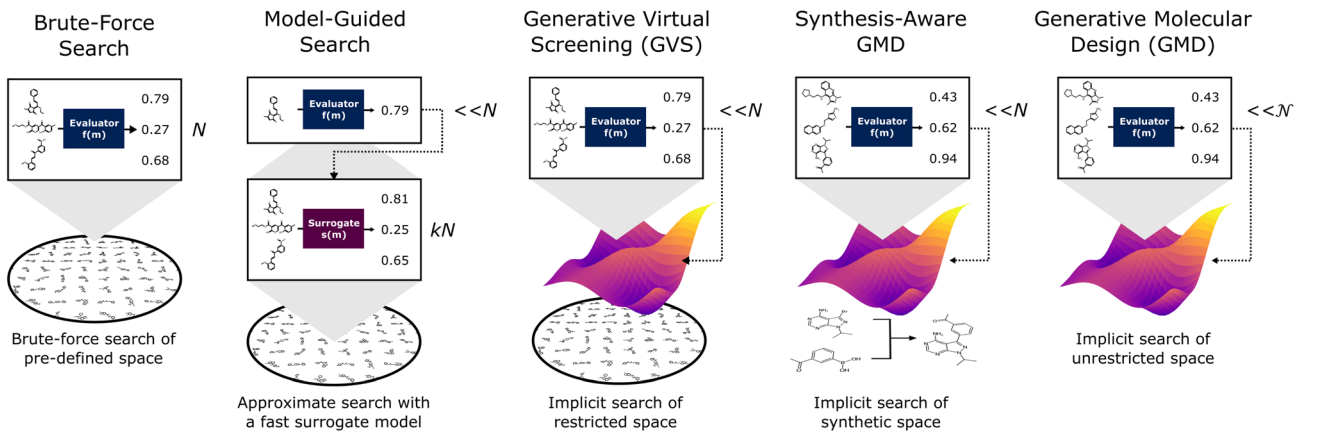

Fig. 2 Illustrative comparison of molecular search paradigms arranged from left to right by decreasing explicitness of chemical space. Brute-force search operates over an explicitly enumerated library of known size N by exhaustively evaluating all compounds. Model-guided search retains this fixed library but improves efficiency by prioritising subsets for evaluation using surrogate models. Generative virtual screening (GVS) applies generative optimisation within a fixed, explicitly defined reference library, enabling efficient traversal of chemical space while preserving library membership and retrospective evaluability. Synthesis-aware generative molecular design further relaxes explicit library constraints by restricting generation through synthetic rules or feasibility models, expanding the reachable space while improving progressability relative to unconstrained generation. Finally, generative molecular design (GMD) samples molecules from an implicit chemical space of unknown size $\mathcal{N}$, offering lower computational cost at the expense of reduced progressability and retrospective evaluability.

not accounting for potentially low hit rates due to evaluator inaccuracy. Beyond raw compute, operational constraints further compound these challenges:[19] storing 1.56 billion molecules in compressed SMILES format requires approximately 89 GB of disk space, while conversion to 3D representations and preparation for molecular docking expanded this requirement to 10.3 TB of compressed data.[20] Consequently, brute-force search of ultra-large virtual libraries presents substantial computational and logistical barriers, particularly for academic groups and small organisations.

2.2 Progressability

Searching an existing virtual library represents the most practically progressable computational hit-finding strategy, as selected molecules are either already available in proprietary collections or can be ordered directly from commercial vendors. Nevertheless, practical limitations remain: some compounds may be out of stock, and make-on-demand libraries do not always deliver successfully (with large catalogues such as Enamine REAL reporting synthesis success rates of approximately 80% (ref. 10)). In addition, freedom-to-operate considerations from an intellectual property (IP) perspective can influence the downstream value of identified hits. Compounds sourced from widely accessible commercial libraries may be more difficult to patent or less suitable for IP-protected discovery campaigns.

2.3 Evaluability

From a retrospective evaluation perspective, brute-force search is a search-complete method over the virtual library, as all molecules are explicitly evaluated by the same evaluator $f(m)$. Consequently, methodological assessment only reduces to validating how accurately the evaluator reflects the experimentally measurable endpoint of interest.

2.4 Further discussion

In Table 2, we summarise examples of ultra-large ($N > 10^9$) brute-force VS campaigns for which sufficient details on computational time and resources have been reported. These studies illustrate that, depending on the complexity of the evaluator $f(m)$, ultra-large VS can require enormous computational resources, often beyond the reach of typical academic or industrial settings. To reduce wall time to one or two days, some campaigns have relied on specialised supercomputers, utilising up to 20 000 CPUs[18] or 27 600 GPUs.[21] In contrast, more modest—though still high-performance—compute clusters typically require weeks to months to complete docking-based brute-force VS. Beyond evaluation itself, molecular preparation constitutes a non-negligible overhead: for example, preparing 1.56 billion molecules for docking was reported to take up to 30 days using 640 CPUs.[20] Although such preparation is performed only once, continued library growth exacerbates downstream storage requirements for 3D-ready formats. Importantly, VS exhibits a strong scale dependence: larger libraries yield more high-scoring compounds according to $f(m)$, leading to higher hit rates and stronger binding affinities.[14] This "bigger is better" dynamic further motivates the search of ever-larger virtual chemical spaces, amplifying the computational burden already inherent to brute-force approaches.

2.4.1 Compute-optimised search. Owing to the computational demands of brute-force search, substantial effort has been devoted to developing compute-optimised strategies tailored to specific evaluators. These approaches aim to reduce the effective cost of exhaustive search by exploiting structure in either the evaluator or the virtual library itself, rather than by altering the underlying search paradigm.

The most common examples target molecular similarity to a query compound, using fingerprint-based similarity or substructure search.[30–35] The importance of this use case is such that specialised GPU implementations have been developed to accelerate similarity calculations, achieving throughputs of up to approximately 45 billion pairwise comparisons per second on modern hardware.[36] Other approaches exploit the combinatorial structure of vendor libraries by comparing molecules at the level of building blocks or fragments prior to full enumeration,[35] thereby drastically reducing the number of required comparisons.

Related strategies have also been applied to more computationally expensive docking workflows by iteratively assembling molecules from building blocks according to vendor-specific reaction rules, while enforcing compatibility with a protein binding site.[37,38] By avoiding enumeration of pocket-incompatible chemistry, these methods substantially reduce computational waste and and therefore depart pure brute-force search. Finally, machine-learning-based indexing approaches

Table 2 Representative examples of ultra-large brute-force virtual screening (VS) campaigns reported in the literature. For studies involving multiple targets (*e.g.*, different query ligands or protein receptors), reported compute time and resource usage are given per target where possible, together with any non-negligible molecule preparation costs. The symbol '%3c' indicates cases in which only the available compute capacity was reported, rather than actual resource utilisation

References	Library size (N)	Evaluator ($f(m)$)	Compute cost	Resources used	Wall time
Grebner *et al.*[18] (2019)	12.7B	3D similarity (FastROCS[22])	∼200K CPU hours & ∼200 GPU hours	29K CPUs & 200 GPUs	∼2.4 days
Liu *et al.*[14] (2025)	1.7B	Docking (DOCK3.8 (ref. 23))	∼243.7 CPU years	3K CPUs	∼30 days
Sivula *et al.*[20] (2023)	1.56B	Docking (Glide-HTVS[24])	∼173.4 CPU years	640 CPUs	∼99 days
Acharya *et al.*[21] (2020)	1.4B	Docking (AutoDock-GPU[25])	<112.3 GPU years	<27.6K GPUs	∼1 day
Gorgulla *et al.*[26] (2020)	1.3B	Docking (QuickVina 2 (ref. 27))	∼615.4 CPU years	∼8K CPUs	∼28 days
Michino *et al.*[28] (2023)	1.12B	3D similarity (GPU shape[29])	156 GPU hours	216 GPUs	∼45 min

embed molecules into vector spaces trained on "ground-truth" similarity measures, enabling approximate nearest-neighbour retrieval at scale.[39] Collectively, these techniques can reduce search times for specific evaluators—most commonly molecular similarity—to seconds or less, even for multi-billion-compound libraries. However, a key limitation is that such acceleration is highly bespoke: each approach is tightly coupled to a particular evaluator $f(m)$ and does not provide a general-purpose speed-up applicable across different properties or search objectives.

3 Model-guided search

Model-guided search is a general strategy for allocating evaluations of $f(m)$ with high compute cost c_f to a learned or heuristic surrogate model $s(m)$ with low cost c_s. The surrogate is used to prioritise which molecules should be evaluated by the high-cost evaluator, thereby improving search efficiency under a limited evaluation budget. Importantly, this principle is agnostic to how candidate molecules are generated. Model-guided search can be applied to explicitly enumerated libraries, combinatorial reagent spaces, or implicitly defined generative spaces, supporting both selection-based and generation-based procedures. In molecular discovery, this framework is closely related to active search and active learning methodologies.[40–42]

In this section, we focus on model-guided search applied to a fixed, finite virtual library. This setting enables direct comparison with brute-force virtual screening, as the underlying chemical space is identical and search performance can be evaluated retrospectively against a well-defined reference ranking. Although the same model-guided principles also underpin many generative molecular design and generative virtual screening workflows—where surrogates guide candidate construction or generator updates[43,44]—those settings are considered under Section 4.

In the fixed-library setting, model-guided search ranks compounds using the surrogate model $s(m)$. This ranking is then used to select subsets for evaluation by the true oracle $f(m)$. This prioritisation can be achieved using similarity-based heuristics or classifier models,[40,45] or *via* regression-based surrogate models that directly approximate the true oracle score.[20,42,46,47]

3.1 Computational cost

Model-guided search fundamentally reduces the computational cost relative to brute-force search. Whereas brute-force evaluation requires computing the evaluator for all N compounds in a library, model-guided search replaces this with a combination of low cost surrogate inference across the library and a much smaller number of high cost evaluations. The approximate computational cost is $C_{\text{total}} \approx kNc_s + Sc_f + C_{\text{train}}(S)$, where k is the number of surrogate inference cycles, c_s is the per-molecule surrogate inference cost, c_f is the per-molecule oracle evaluation cost, $S \ll N$ is the number of oracle evaluations, and $C_{\text{train}}(S)$ is the surrogate training cost. In practice, S often corresponds to only 0.1–1% of the library, yielding orders-of-magnitude reductions in wall time and overall compute requirements. However, surrogate modelling itself introduces non-negligible overhead due to its iterative nature. Although this cost is frequently overlooked relative to the savings in oracle evaluations, it can be substantial. For example, Sivula *et al.*[20] reported that surrogate model training and inference alone required 203–335 GPU hours. As virtual libraries continue to expand, the computational burden associated with surrogate inference and retraining is therefore likely to become increasingly significant.

3.2 Progressability

A key advantage of model-guided search is that it preserves the progressability of brute-force search over specified virtual libraries. In this setting, model-guided methods alter only the computational strategy used to prioritise molecules, while leaving downstream experimental workflows unchanged and avoiding any additional synthesis requirements.

3.3 Evaluability

Because model-guided search performs an approximate search of a fixed virtual library with respect to the true oracle $f(m)$, performance is assessed retrospectively using $f(m)$ values as ground truth, typically obtained from a brute-force baseline. Evaluation focuses on how effectively the method recovers the highest-ranking molecules and how large a subset S is required relative to the full library size N. Common metrics include recall of the top-k molecules ranked by $f(m)$, enrichment relative to brute-force screening, the subset size S needed to achieve a target recall, and the false-negative rate. Minimising false negatives is particularly important, as any high-quality compound incorrectly excluded early by a surrogate model cannot be recovered in later stages of the workflow.

3.4 Further discussion

Model-guided search encompasses a spectrum of strategies that differ in both surrogate model formulation and selection strategy (often referred to as the acquisition strategy), as summarised in Table 3. Early large-scale applications employed binary classification surrogates to rapidly exclude regions of chemical space unlikely to contain promising molecules. Screening approximately 1% of a library over multiple iterations, these methods achieved up to 2500-fold enrichment of the top 100 compounds relative to random selection.[48] More recently, Luttens *et al.*[47] demonstrated that a one-shot surrogate-based selection of 5 million compounds (0.17% of the initial library) was sufficient to identify experimentally confirmed single-digit μM hits. Regression-based surrogate models instead approximate the evaluator's continuous scoring landscape, enabling finer-grained prioritisation of candidate molecules. Lightweight iterative frameworks such as HASTEN[49,50] have shown that simple greedy selection strategies can recover up to 90% of the top 1000 molecules while evaluating only 1% of a billion-scale library.

More sophisticated surrogate models additionally estimate predictive uncertainty, enabling more informed selection strategies grounded in Bayesian decision theory,[51] such as

Table 3 Representative literature examples of model-guided search campaigns over fixed virtual libraries. For studies conducted on multiple targets (*e.g.*, query ligands or protein receptors), reported compute time and resource usage are shown as per-target averages where possible, together with any non-negligible preparation or surrogate-model overhead. Selection strategies include greedy acquisition and active learning approaches that account for predicted sample value and uncertainty. Unreported or indeterminable values are denoted by '—', while '%3e' indicates partially reported quantities

References	Library size (N)	Surrogate model	Surrogate cycles (k)	Strategy	Evaluator ($f(m)$)	Library subset (S)	Top-k recall	Compute cost	Resources used	Wall time
Zhou *et al.*[53] (2024)	5.5B	FFNN	10	Greedy	Docking (RosettaVS[54])	6M (0.11%)	—	—	3000 CPUs & 1 GPU	7 days
Luttens *et al.*[47] (2025)	3.5B	CatBoost	1	Greedy	Docking (DOCK 3.7 (ref. 23))	6M (0.17%)	—	>12.8K CPU hours	—	—
Sivula *et al.*[20] (2023)	1.56B	MPNN	10	Greedy	Docking (Glide-HTVS[24])	15.6M (1%)	90% (k = 1K)	631 CPU days & 13.6 GPU days	640 CPUs & 10 GPUs	10–14 days
Gentile *et al.*[48] (2020)	1.36B	MLP	10	Greedy	Docking (FRED[55])	13M (1%)	—	—	60 CPUs & 4 GPUs	—
Yang *et al.*[56] (2021)	118M	GCNN	2	Greedy (+AL)	Docking (DOCK 3.7 (ref. 23); Glide-SP[24])	5.9M (0.5%)	86% (k = 10K)	8.7K CPU hours	100 CPUs & 1 GPU	~4.9 days
Graff *et al.*[46] (2021)	118M	MLP; MPNN; RF	5	Greedy (+AL)	Docking (DOCK 3.7 (ref. 23))	2.4M (2%)	90% (k = 50K)	—	—	—
Graff *et al.*[52] (2022)	98M	MPNN	5	AL + pruning	Docking (Glide-SP[24])	1.2M (1.2%)	68% (k = 10K)	—	—	—

Bayesian optimisation, or more generally within active learning (AL) frameworks (Table 3). These uncertainty-aware strategies explicitly balance exploration and exploitation during search. For example, Graff *et al.*[46] evaluated multiple acquisition functions including upper confidence bound, Thompson sampling, expected improvement, and probability of improvement to incorporate uncertainty into candidate selection. Although simple greedy acquisition performed best in their study, recovering 90% of the top 50 000 molecules after evaluating 2% of the library, uncertainty-aware approaches offer important practical advantages. In particular, they help mitigate challenges arising from noisy evaluators (*e.g.*, docking scores), bias introduced by early training data, and over-emphasis on narrow chemotypes.[49] These benefits are reflected in later work by Graff *et al.*,[52] who used an upper confidence bound strategy to recover 68% of the top 10 000 molecules after evaluating only 1.2% of the library. Importantly, prediction uncertainty was also leveraged to enable library pruning—iteratively removing compounds confidently predicted to be inactive—thereby reducing surrogate inference overhead by approximately half.

Overall, model-guided search has demonstrated substantial computational efficiency gains over brute-force search, enabling multi-billion-compound screening campaigns to be completed within days. This while retaining strong recovery of top-ranking candidates with respect to the true oracle of interest.[50,53]

4 Generative molecular design

Recent advances in machine learning and generative artificial intelligence have led to a rapid expansion of research in GMD. Broadly, GMD algorithms construct molecules de novo—atom-wise, fragment-wise, or *via* explicit synthetic pathways (discussed in Section 5).[57] In this review, GMD is an umbrella term encompassing heuristic approaches based on explicit chemical rules (*e.g.*, genetic algorithms[58]) and machine-learning-based models that learn discrete or continuous molecular distributions (*e.g.*, chemical language models[59] and autoencoders[60]). When paired with goal-directed GMD optimisation frameworks (such as Bayesian optimisation,[44,60,61] reinforcement learning,[62] or particle swarm optimisation[63]) these models generate molecules optimised for evaluator-defined objectives.

GMD can also be viewed as a form of chemical space search, with the key distinction that the full space being explored is typically not explicitly defined. As a result, GMD operates over implicit, open chemical spaces that are often substantially larger[17] than explicitly specified libraries that can be indexed and searched *via* brute-force or model-guided approaches. Although these implicit spaces may possess theoretical or empirical bounds—for example, the search space induced by a graph-based genetic algorithm could in principle be fully enumerated given fixed seed molecules and chemical operations, or the probability distribution of a chemical language model could be sampled until near exhaustion[64]—such bounds are rarely characterised in practice. Consequently, GMD explores chemical space by constructing focused sets of candidate molecules without access to a well-defined ground-truth

search space in the traditional sense, with the additional advantage that such exploration may circumvent human and chemical biases embedded in explicitly specified virtual libraries.

4.1 Computational cost

In contrast to brute-force search, GMD scales with the number of molecules sampled from the model, n, rather than the size of an explicitly enumerated library. The approximate computational cost is $C_{\text{total}} \approx n(c_g + c_f) + C_{\text{update}}(n)$, where c_g is the per-molecule generation cost, c_f is the evaluation cost, and $C_{\text{update}}(n)$ captures any model update steps. In practice, n is typically orders of magnitude smaller than the size of specified virtual libraries N, while the implicit chemical space explored by GMD is of unknown size, denoted $\mathcal{N}$. Here, n scales with the number of virtual hits desired (h) dependent on the probability that a sampled molecule is a hit (p) *i.e.*, $n \approx h/p$.‡ Although h can be fixed in advance, variability in computational efficiency is largely governed by p, which reflects how effectively the model optimises the evaluator objective. For example, Renz *et al.*[74] report that strong methods can generate approximately 100–1000 high-scoring molecules per 10 000 evaluated samples, corresponding to $0.01 \leq p \leq 0.1$ depending on the task and evaluator. This average behaviour, however, masks important dynamics. First, p is typically low early in optimisation and may increase with training. Second, mode collapse can reduce effective yield by repeatedly sampling similar molecules. Third, chemically implausible or non-drug-like structures should not be counted as hits,[75,76] further reducing p. Finally, some GMD approaches incur substantial up-front pre-training costs, ranging from hours to days for large foundation models.[77] However, this cost is usually paid once and amortised across downstream optimisation runs.

4.2 Progressability

The principal challenge facing GMD algorithms lies in their exploration of large, implicitly defined chemical spaces. In practice, more than 90% of molecules sampled from GMD models are not present in the training data[78,79] and, although rarely reported, are typically not found in commercial catalogues, rendering them non-orderable (approximately 80% commercially novel in one reported case[80]). This introduces a major practical bottleneck: synthesis. To progress GMD-proposed molecules, they must be synthesised and experimentally tested; however, many models struggle to generate compounds that are synthetically accessible.[81] As a result, many industrial and academic discovery efforts are reluctant to allocate substantial resources to *de novo* synthesis without extensive prior validation of both the GMD algorithm and the evaluator, particularly at early, high-attrition stages of hit finding. At the same time, this novelty can be advantageous: *de novo* compounds are more likely to occupy unencumbered intellectual property space, affording greater freedom to operate in downstream development.

4.3 Evaluability

Despite the proliferation of benchmarks for generative molecular design, retrospective evaluation of GMD performance remains fundamentally challenging. Because the underlying chemical space being searched is not explicitly defined, proposed molecules lack a ground-truth reference and cannot be unambiguously classified as optimal or suboptimal in retrospect.[82] In practice, GMD methods are therefore typically evaluated either by the degree to which a chosen property is maximised,[78,83] or by the number and chemical diversity of generated solutions exceeding a predefined threshold.[74,75] However, such criteria are vulnerable to Goodhart's Law:[84] when a metric becomes the optimisation target, it may cease to reflect the intended objective. Indeed, GMD models can exploit weaknesses in evaluators.[85,86] For example, some methods overfit to artefacts such as the random seed of an ML-based evaluator,[85] while others generate disproportionately large and lipophilic molecules to inflate docking scores, which often reflect additive interaction terms.[87] While counting the number of diverse high-scoring solutions partially captures search behaviour, it enables only relative comparisons between optimisation strategies and provides no absolute measure of search optimality. Recent work such as MolExp[88] defines a finite objective with a known number of optimal solutions. This has highlighted that many GMD methods perform poorly in multi-solution settings, indicating that exploration efficiency in implicit chemical spaces remains a major open challenge.

4.4 Further discussion

GMD is now well established in the literature, with hundreds of methods and case studies reported in peer-reviewed work,[16,89] and an increasing number of prospectively validated campaigns. For example, several studies from the Chemistry42 platform[90] have demonstrated successful application of GMD in real discovery settings, including the identification of a TNIK inhibitor that has progressed to Phase II clinical trials.[91] Munson *et al.*[92] reported biological activity for 19 of 32 experimentally tested molecules after sampling $n = 1\,638\,300$ compounds from a variational autoencoder model designed to optimise polypharmacology. In related work, three novel nanomolar adenosine A_{2A} receptor ligands (8 experimental hits from 9 tested compounds) were identified by sampling $n = 12\,800$ molecules per search campaign. The authors used a chemical language model coupled to a docking evaluator;[80] each campaign required approximately 6 hours using around 30 CPUs and one GPU. Similarly, Korablyov *et al.*[93] experimentally validated 24 of 35 GMD-proposed compounds, all close analogues, exhibiting micromolar activity against soluble epoxide hydrolase. This campaign sampled $n = 100\,000$ molecules over 1.6 days using 672 CPUs, with optimised evaluator inner and outer loops, and operated within a fragment-based GMD framework. In this case, with a known theoretical

‡ Note that for brute-force search p can be interpreted as fixed. Whereas for model-guided search, the surrogate model learns an estimator $\hat{p}(\mathbf{x})$ to approximate the true probability p.

On human biases when searching chemical space

Drug hunters are rarely neutral explorers of chemical space.

Although drug discovery is often framed as a rational and data-driven discipline, human judgement plays a decisive role in determining which molecules are prioritised for experimental validation.[65] In practice, this manifests as implicit and explicit biases about what constitutes an acceptable or "drug-like" molecule. These biases are shaped by decades of historical data on *in vivo* safety, synthetic tractability, and clinical success. Analyses of such data have given rise to average-driven heuristics that are widely reinforced through canonical guidelines, including Lipinski's rule-of-five,[5] pan-assay interference substructures (PAINS),[66] structural alerts, and composite measures of chemical desirability.[67]

Beyond formal guidelines, additional biases are embedded through influential discovery campaigns and widely cited literature that implicitly define acceptable chemical space. These paradigms—often successful in retrospect—become codified and transmitted across generations of medicinal chemists, shaping how molecules are judged long before biological or clinical evidence is available. Individual scientists further contribute their own experiential biases, often favouring chemical familiarity based on prior successes. Notably, such preferences are sufficiently systematic that machine learning models can often predict which chemist designed a given molecule.[68]

However, real-world drugs frequently defy these learned expectations.[69] Many clinically and commercially successful therapeutics violate long-standing assumptions about drug-likeness and would likely be deprioritised—or rejected outright—if assessed solely using contemporary structural heuristics.

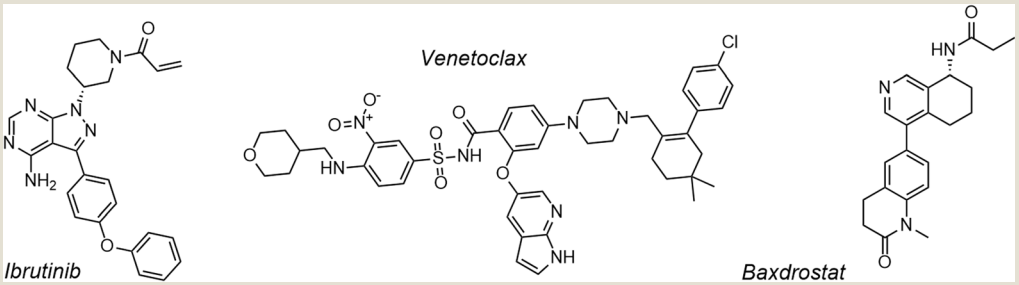

Fig. 3: Examples of clinically successful drugs that challenge conventional medicinal chemistry heuristics for oral drug-likeness.

Ibrutinib (Imbruvica®), a first-in-class covalent Bruton's tyrosine kinase inhibitor, illustrates this tension. From a design perspective, its reactive warhead and known kinase promiscuity would likely trigger concern or exclusion in many modern screening workflows.[70] Nevertheless, ibrutinib transformed the BTK inhibitor landscape, achieved regulatory approval across multiple indications, and has generated more than $20 billion USD in cumulative revenue.

Venetoclax (Venclexta®), a selective BCL-2 inhibitor, presents an even more striking example. In addition to its large molecular weight (868.44 g mol^{-1}), venetoclax contains several features often flagged as structural liabilities, including an aromatic nitro group, a cyclic alkene, and a terminal tetrahydropyran ring. Despite this, venetoclax received regulatory approval in 2016 and has become a multi-billion-dollar therapeutic that fundamentally altered treatment paradigms for haematological malignancies.[71]

Baxdrostat represents the opposite extreme in terms of molecular complexity. With a modest molecular weight (363.45 g mol^{-1}) and a scaffold resembling a fragment commonly found in screening libraries, it could easily be overlooked in conventional campaigns. Yet baxdrostat demonstrated compelling *in vivo* efficacy in treatment-resistant hypertension, prompting AstraZeneca's $1.8 billion acquisition of CinCor Pharma in 2023.[72]

Together, these examples (Fig. 3) highlight a central tension in medicinal chemistry: neither structural complexity, heuristic drug-likeness, nor perceived chemical "appeal" reliably predict clinical or commercial success.[73] Ultimately, patients do not benefit from molecular elegance; they benefit from efficacy, safety, and access. Search strategies that overly constrain chemical space—whether by human judgement or algorithmic bias—risk excluding precisely the kinds of molecules that go on to make the greatest impact.

chemical space of 10^{15} molecules, corresponding to exploration of approximately 10^{-8}% of that space.

Despite the prospective success of some GMD algorithms, assessing their overall search capability remains difficult, if not impossible. The size of the implicit chemical space explored by a given model is rarely known, and even more rarely is it accompanied by evaluator values $f(m)$ that could serve as a ground-truth reference baseline. Moreover, many studies do not report the computational resources or wall time, further limiting even high-level comparison across different molecular search paradigms.

4.4.1 Substructure constrained generative methods. Substructure-constrained GMD enables targeted design tasks such as fixed-scaffold decoration and fragment linking. A range of approaches have been proposed: some require specialised model architectures, molecular grammars, or training datasets,[94–96] whereas other models support both *de novo* and substructure-constrained generation within a single framework,[97] or can be used as plug-ins to steer existing *de novo* GMD models toward user-specified constraints.[98] In general, substructure constraints tend to improve progressability because user-defined scaffolds can correspond to synthetically accessible intermediates, enabling focused exploration of local structure–activity relationships (SAR), which is particularly useful in hit-to-lead and lead optimisation. These workflows can also be combined with reaction rules to encourage scaffold decoration through specific, chemically plausible transformations.[94,98]

5 Synthesis-aware generative molecular design

Incorporating synthetic awareness into generative molecular design (GMD) primarily aims to improve the progressability of

generated molecules by increasing the likelihood that proposed structures can be synthesised and experimentally validated. Synthetic awareness can be introduced into GMD in multiple ways;[81] however, each strategy entails distinct limitations and trade-offs that affect computational cost, search efficiency, and evaluability.

Broadly, existing approaches fall into three categories: (1) synthesizability-biased pre-training,[81] in which models are trained on datasets such as ZINC[79] or ChEMBL;[99] (2) synthesizability evaluators, used either as post-hoc filters or as optimisation objectives *via* computer-aided synthesis planning (CASP) tools or faster surrogate metrics such as SAScore[100] or RAScore;[101] and (3) synthesis-constrained generation, which restricts the generative process to predefined reaction rules and building blocks.

5.1 Computational cost

As a subset of GMD, synthesis-aware approaches share the approximate computational cost $C_{\text{total}} \approx n(c_g + c_f) + C_{\text{update}}(n)$ characteristic of the broader GMD paradigm. However, introducing synthetic constraints can substantially influence both p and the per-sample computational cost c_g or c_f. In particular, CASP-based evaluators impose significant overhead in c_f, with runtimes typically ranging from 30 seconds to several minutes per molecule.[102] As a result, synthesis-aware workflows often rely on surrogate synthesizability metrics to remain computationally tractable.

5.2 Progressability

The central motivation for synthesis-aware GMD is to mitigate the synthesizability bottleneck inherent to unconstrained generation and to increase the likelihood that proposed molecules can be acted upon experimentally. The extent to which this goal is achieved, however, varies substantially across approaches. Synthesizability-biased pre-training implicitly biases generation toward previously synthesised chemistry. However because most ML-based GMD models still generate predominantly novel molecules,[79] progressability is far from guaranteed without additional evaluation. Synthesizability evaluators provide a stronger signal by estimating synthetic feasibility and, in the case of CASP tools, proposing candidate synthesis routes. However, their high computational cost often necessitates the use of faster surrogate scores, which do not provide explicit routes and therefore only increase the likelihood of synthesizability rather than guaranteeing actionability. In contrast, synthesis-constrained GMD constructs molecules according to validated reaction rules and available building blocks, inherently providing actionable synthetic pathways. Approaches in this category are best positioned to propose routes that are robust, cost-effective, and compatible with available starting materials, offering a direct path from virtual hits to experimental validation.

5.3 Evaluability

From an evaluation perspective, synthesis-aware GMD inherits the same fundamental limitations as unconstrained GMD: search performance is assessed using optimisation-based metrics rather than against a ground-truth reference space. While improved synthesizability makes prospective validation more feasible, it does not resolve the challenge of retrospectively assessing search completeness or efficiency. A key trade-off—particularly for synthesis-constrained GMD—is a reduced effective chemical space, which can lead to fewer discoverable hits[103] and may partially explain why such methods often perform worse on standard GMD benchmarks.[83] In some cases, this reduced performance is not accompanied by a corresponding increase in predicted synthesizability.[104] Although this limitation is largely irrelevant in practice—non-synthesizable hits are not actionable—it complicates algorithmic comparison across GMD approaches, as differences in building blocks and reaction sets define non-standardised and incomparable search spaces.[105] This lack of a shared reference space remains a major obstacle to evaluating the search capability of synthesis-aware GMD methods in isolation.

5.4 Further discussion

The approaches discussed below illustrate the principal mechanisms by which synthetic awareness has been incorporated into generative molecular design. Synthesizability biased pre-training datasets is already commonplace for many ML-based GMD algorithms[78,79,83] by pre-training models on molecules that have already been made (*e.g.*, ChEMBL[99]) or molecules that are makeable (*e.g.*, ZINC[106]). However, as many ML-based GMD models produce mostly novel molecules, the aim is to learn the underlying chemical distributions that lead to synthesizable compounds. However, Gao and Coley[81] demonstrated that the use of ZINC and ChEMBL pre-training datasets lead to different predicted synthesizability of *de novo* compounds, as measured by CASP-evaluators. This despite the fact that both pre-training datasets contain synthesizable molecules. The predicted synthesizability ranged from 61–68% for these datasets, hence this is not an exact solution to synthetic-awareness.

Synthesizability evaluators can be used as post-hoc filters to remove molecules with poor predicted synthesizability, or as secondary objectives to maximise during model optimisation.[81] However, this remains an approximate GMD solution as GMD can still propose molecules that do not satisfy CASP-based evaluators. This contributes to search inefficiency as it reduces the probability of a molecule being a "hit" p. For example, Guo *et al.*[107] leveraged a highly efficient GMD setup that still resulted in approximately 30% to 44% of molecules not adhering to CASP synthesizability objectives. In a similar vein, GMD can be guided to "in-house" synthesis capability by coupling the CASP evaluator to a tailored stock of inventory-specific building blocks and reactions.[108] In this case, approximately 60% of generated molecules were not predicted to be synthesizable; however, the approach did facilitate experimental verification of three molecules (1/3 with <10 μM MGLL affinity, the primary objective).

Synthesis-constrained GMD approaches restrict the search space of the GMD model by limiting it to a set of predefined reaction rules and building blocks. Thus, the practical benefits

of being able to follow proposed validated synthetic routes provide a huge advantage for follow-up synthesis and experimental validation. Implementations can include Monte Carlo Tree Search,[109] autoregressive models,[110–112] reinforcement learning,[113] and flow networks.[114] In the latter case, Cretu *et al.*[114] demonstrated that their flow network implementation could scale to chemical space sizes of $\sim 10^{16}$ based on 221 181 building blocks and 105 reactions with trajectory lengths of 4. Subsequently this enabled the finding of better virtual hits with higher predicted binding affinity than brute-force VS of much smaller explicit libraries after $n = 400\,000$ molecules (requiring 1000 GPU hours due to more computationally expensive evaluators).[115] For a recent review on different synthesis-aware approaches see Papidocha *et al.*[105].

6 Generative virtual screening

Recently, a growing class of generative molecular design approaches has emerged[109,116–118] that combine generative optimisation with explicit constraints on chemical space, such that only molecules belonging to a specified reference library can be proposed. This paradigm is referred to as Generative Virtual Screening (GVS).[107] GVS differs from synthesis-aware GMD, which aims to propose molecules alongside feasible synthetic routes but does not necessarily guarantee membership in a predefined compound catalogue. However, when synthesis-aware GMD is restricted to the same building blocks and reaction rules used by enumerated vendor libraries, such as Enamine REAL,[10] it effectively becomes a form of GVS. More generally, GVS can be realised by any approach that constrains generative design to an explicitly defined, fixed chemical space. As such, GVS constitutes a hybrid search paradigm that applies generative optimisation methods over known libraries, bridging brute-force and model-guided search with *de novo* generative design.

At first glance, GVS may appear counter-intuitive given that model-guided approaches can already be used to efficiently search specified virtual libraries. However, GVS inherits complementary advantages from multiple paradigms: the favourable computational cost of GMD, the progressability of explicit library-based search, and the retrospective evaluability of model-guided search over fixed chemical spaces, all applied within a generative optimisation framework. These combined characteristics are discussed in more detail below.

6.1 Computational cost

GVS inherits the favourable cost behaviour of GMD, with computational cost governed by the number of molecules sampled, n, rather than the size of the full reference library N. As in GMD, the approximate computational cost is $C_{\text{total}} \approx n(c_g + c_f) + C_{\text{update}}(n)$, with n determined by h/p *i.e.*, the probability of sampling a virtual hit largely affecting scaling. Crucially, when GVS strictly enforces membership in a reference library, p is typically higher than in unconstrained GMD, as sampled molecules are guaranteed to be orderable or experimentally testable. This removes the dependence of p on the chemical validity or availability of generated structures and shifts optimisation efficiency to the ability of the generative algorithm to identify high-scoring regions within the fixed library.

6.2 Progressability

GVS retains the same prospective practicality as brute-force or model-guided search over specified virtual libraries. Proposed molecules can be ordered directly from in-stock catalogues, procured through established make-on-demand workflows used in ultra-large screening campaigns, or progressed directly to experimental validation when constrained to proprietary in-house libraries. As with other library-based approaches, freedom-to-operate and intellectual property considerations remain relevant when relying on commercial vendor catalogues.

6.3 Evaluability

Unlike unconstrained GMD, GVS is retrospectively evaluable in the same manner as model-guided search over a fixed chemical space. Because ground-truth evaluator values can be computed for the entire reference library—or a large, representative subset—GVS methods can be assessed using standard retrieval-style metrics: recall of the top-k molecules or enrichment at fixed sampling budgets n. This enables isolation and analysis of the search behaviour of generative algorithms without confounding effects arising from the unknown size and composition of implicit chemical spaces. As a result, GVS allows direct, like-for-like comparison of search efficiency with brute-force and model-guided approaches and provides an absolute reference point for assessing search completeness. Notably, despite this advantage, many existing GVS studies do not yet exploit retrospective evaluation and instead adopt GMD-style metrics, such as reporting the best-scoring molecules found as a function of the number of evaluator calls $f(m)$.

6.4 Further discussion

In practice, existing GVS methods can be broadly categorised into two classes. The first comprises approaches that strictly guarantee membership in a reference library, most commonly by employing synthesis-constrained GMD aligned with the building blocks and reaction rules of an enumerable vendor catalogue. The second consists of approximate GVS methods, which guide unconstrained GMD toward reference library membership or map unconstrained proposals back into the reference catalogue *via* projection-based strategies. A summary of representative approaches from both categories is provided in Table 4.

The most common approach to GVS with guaranteed library membership relies on synthesis-constrained GMD implemented using the same building blocks and reaction rules as ultra-large enumerable vendor libraries, such as Enamine REAL.[10] A representative example is SyntheMol[109] and its reinforcement-learning variant, SyntheMol-RL,[117] which combine Enamine REAL and WuXi GalaXi enumeration rules with Monte Carlo tree search or reinforcement learning. In the reinforcement-learning implementation, evaluating only $n = 10\,000$ molecules (2×10^{-7}% of the 46-billion-compound library)

Table 4 Representative examples of GVS approaches reported in the literature. For each study, the size of the reference library defining the search space (N) and the number of molecules sampled or evaluated by the method (n) are reported where available. Additional columns summarise library membership guarantees, evaluator types, and evaluation metrics. Unreported or indeterminable values are denoted by '—'

References	Library size (N)	Molecules sampled (n)	Library membership	Evaluator ($f(m)$)	Top-k recall	Compute cost	Resources used	Wall time
P de Oliveira *et al.*[116] (2024)	3T	1.8M (6×10^{-7}%)	63–95%	Docking surrogate (MLP)	—	90 CPU/GPU hours	1 GPU	—
Guo *et al.*[107] (2025)	142B	15K (1×10^{-7}%)	42%	Docking (QuickVina2-GPU[119]); CASP (MEGAN[120])	—	—	1 GPU	—
Swanson *et al.*[117] (2025)	46B	10K (2×10^{-7}%)	65%	QSAR (MPNN; MLP)	—	—	8 CPUs & 1 GPU	0.5–7 days
Swanson *et al.*[109] (2024)	30B	24K (8×10^{-5}%)	46%	QSAR (MPNN; RF)	—	—	—	8.5 hours
Klarich *et al.*[118] (2024)	334.8M	3.35M (1%)	100%	Docking (FRED[55])	58% ($k = 100$)	16 K CPU hours	—	—

found 11 of 38 tested compounds as hits (29% hit rate), including six novel molecules sharing a conserved 4-chloro-2,3-dihydroxybenzamide motif. Notably, although library membership is theoretically guaranteed through alignment with vendor enumeration rules, only 65% of generated compounds were reported to be orderable, likely reflecting additional, undocumented filtering applied by library providers. In contrast, P de Oliveira *et al.*[116] adopted an encoder–decoder approach trained with normalizing flows, in which the decoder selects from available building blocks and reaction rules to map latent molecular representations to viable products. This enabled the largest reported GVS campaign to date, implicitly spanning approximately 3 trillion Enamine unREAL compounds. Evaluating 1.8 million molecules using a single GPU, the method yielded purchasable hits later experimentally validated at a hit rate of 12% (15/123). Finally, Klarich *et al.*[118] proposed a more foundational strategy based on Thompson sampling over library building blocks and a restricted reaction set to enable efficient approximate search of large reference libraries. Crucially, this approach was evaluated against known ground-truth labels for the fully enumerated library, recovering 58% of the top-100 compounds as ranked by the evaluator while sampling $n = 3.35$ million molecules (1% of the library), approaching the efficiency gains observed in model-guided search (Table 3). To our knowledge, this remains the only GVS study that reports top-k recall relative to an available ground-truth baseline.

Approximate GVS has also been demonstrated by Guo *et al.*,[107] who guided unconstrained GMD toward membership in reference libraries by formulating library inclusion as an optimisation objective. In this framework, molecules are rewarded if they are predicted to belong to a target library, as assessed using tailored CASP evaluators. Despite the absence of strict constraints on generation, this approach achieved approximately 42% library membership when optimising for synthesizability within the Freedom Space 4.0 building block inventory. Alternatively, approximate GVS can be implemented by projecting unconstrained generative proposals onto a reference library *via* nearest-neighbour retrieval, for example using fingerprint-, embedding-, or shape-based similarity. However, even when employing highly compute-optimised similarity search methods (subsubsection 2.1.1), this projection step introduces additional computational overhead to map generated structures to purchasable analogues. As a result, the

Table 5 Examples of publicly available datasets that define explicit, fixed chemical spaces with associated evaluator labels calculated with open-source tools, and which can therefore serve as reference benchmarks for retrospective evaluation of GVS methods

Ref.	Description
Lyu *et al.*[121]	A dataset of 99 million compounds docked against AmpC and 138 million compounds docked against D_4 receptor using DOCK3.7.[23] Baseline used by Graff *et al.*[46] and can be found at https://doi.org/10.6084/m9.figshare.7359626 and https://doi.org/10.6084/m9.figshare.7359401
Hall *et al.*[122]	A database of molecules docked against 11 different targets with DOCK3.7/3.8 (ref. 23) totalling 6.3 billion molecules. Library and associated labels can be found at https://lsd.docking.org/targets
García-Ortegón *et al.*[87]	A smaller dataset of 260 155 molecules docked against 58 DUD-E protein targets using Autodock Vina[123] *via* DOCKSTRING. Library and associated labels can be found at https://doi.org/10.6084/m9.figshare.16511577

theoretical scaling advantages of GVS are diminished. More importantly, this approach creates a mismatch: evaluator values used to update the generative model correspond to the originally generated molecules, not their projected library analogues. This discrepancy complicates optimisation and reduces learning efficiency.

7 Conclusion

Across this review, we have described foundational paradigms for searching chemical space that trade off computational efficiency, evaluability, and progressability. At one extreme, classical brute-force search over a specified virtual library provides a well-defined, reproducible, progressable, and search-complete framework, but suffers from unfavourable scaling as library sizes continue to grow. At the other extreme, generative molecular design (GMD) offers substantial gains in computational efficiency by sampling molecules directly—often from learned distributions—with cost scaling primarily with the number of desired virtual hits rather than the size of chemical space. However, this efficiency comes at the expense of progressability and evaluability: without a fixed reference space, it becomes difficult both to advance proposed molecules experimentally and to quantify how effectively an optimisation algorithm explores toward all optimal solutions. Generative virtual screening (GVS) therefore emerges as a natural middle ground, retaining the favourable scaling properties of GMD while remaining anchored to an explicit, finite library. By operating within such a reference space, GVS enables scalable chemical space exploration without sacrificing immediate progressability or retrospective evaluability.

In the near term, GVS offers a practical advantage: it identifies progressable solutions with sampling budgets on the order of $n = 10\,000$—far fewer than the subsets exceeding $S > 10^6$ typically required in model-guided search—while enabling direct acquisition and experimental testing from vendor or in-house libraries. From an algorithmic perspective, however, the more significant contribution of GVS lies in the availability of ground-truth evaluator labels over a fixed chemical space, which provides a concrete reference baseline unavailable in unconstrained GMD. This enables precise assessment of how many of the best compounds in a reference library are recovered, for example *via* top-k recall, in direct analogy to model-guided search. Such baselines allow algorithm developers to explicitly analyse the balance between exploitation and exploration, a central challenge in generative optimisation. This is particularly important given the strong tendency of many GMD methods to over-exploit local optima at the expense of broader coverage of chemical space. At present, this opportunity remains largely underexploited, as only a single GVS study has reported top-k recall against a ground-truth baseline.[118] We therefore encourage future work to adopt reference libraries with available ground-truth labels, such as those listed in Table 5, to enable rigorous and comparable evaluation of GVS. More broadly, the lack of consistent reporting of computational resources and wall times across GMD studies further hampers like-for-like comparison between search paradigms, obscuring our understanding of current progress in chemical space search.

Finally, GVS should not be viewed as an endpoint, but rather as a methodological scaffold for improving the measurability and understanding of implicit chemical space search methods, while retaining immediate progressability benefits. The constraints imposed by fixed reference libraries are methodological rather than philosophical: as discussed in the boxout, such libraries inevitably encode historical biases about what constitutes acceptable chemistry. However, by operating under controlled and evaluable conditions, GVS enables optimisation algorithms to be developed and stress-tested with well-defined ground truth, allowing their search behaviour to be robustly and quantitatively characterised. Once such behaviour is established, these algorithms can be applied with greater confidence to progressively less constrained—or fully unconstrained—generative settings. This supports principled exploration of broader chemical space beyond the limits of existing libraries.

Author contributions

MT conceptualised the review topic. MT, RA, YSR, and CN prepared the original draft which was reviewed by GDF and AB.

Conflicts of interest

There are no conflicts to declare.

Data availability

No primary research results, software or code have been included, and no new data were generated or analysed as part of this review. Presented data is an approximate summary of previously published data which has been referenced accordingly.

Acknowledgements

MT and AB were funded by Khalifa University grant KU-INT-FSU-2025-022. This research was supported by the Center for Biotechnology, Khalifa University of Science and Technology (KU-BTC). YSR acknowledges the support of UAEU through the UAEU-SQU collaborative research grant (#12S244) and the UAEU-KAUST A-TRIP grant (#12S295).

References

1 A. Raghuraman, P. D. Mosier and U. R. Desai, *J. Med. Chem.*, 2006, **49**, 3553–3562.
2 B. K. Shoichet, *Nature*, 2004, **432**, 862–865.
3 R. L. van den Broek, S. Patel, G. J. P. van Westen, W. Jespers and W. Sherman, *J. Chem. Inf. Model.*, 2025, **65**, 9383–9397.
4 S. Seal, M. Mahale, M. García-Ortegón, C. K. Joshi, L. Hosseini-Gerami, A. Beatson, M. Greenig, M. Shekhar, A. Patra, C. Weis, *et al.*, *Chem. Res. Toxicol.*, 2025, **38**, 759–807.

5 C. A. Lipinski, F. Lombardo, B. W. Dominy and P. J. Feeney, *Adv. Drug Delivery Rev.*, 1997, **23**, 3–25.
6 P. G. Polishchuk, T. I. Madzhidov and A. Varnek, *J. Comput.-Aided Mol. Des.*, 2013, **27**, 675–679.
7 R. S. Bohacek, C. McMartin and W. C. Guida, *Med. Res. Rev.*, 1996, **16**, 3–50.
8 S. Stegemann, C. Moreton, S. Svanbäck, K. Box, G. Motte and A. Paudel, *Drug Discovery Today*, 2023, **28**, 103344.
9 L. Ruddigkeit, R. Van Deursen, L. C. Blum and J.-L. Reymond, *J. Chem. Inf. Model.*, 2012, **52**, 2864–2875.
10 O. O. Grygorenko, D. S. Radchenko, I. Dziuba, A. Chuprina, K. E. Gubina and Y. S. Moroz, *iScience*, 2020, **23**(11), 101681.
11 W. A. Warr, M. C. Nicklaus, C. A. Nicolaou and M. Rarey, *J. Chem. Inf. Model.*, 2022, **62**, 2021–2034.
12 W. P. Walters, M. T. Stahl and M. A. Murcko, *Drug Discovery Today*, 1998, **3**, 160–178.
13 D. B. Kitchen, H. Decornez, J. R. Furr and J. Bajorath, *Nat. Rev. Drug Discovery*, 2004, **3**, 935–949.
14 F. Liu, O. Mailhot, I. S. Glenn, S. F. Vigneron, V. Bassim, X. Xu, K. Fonseca-Valencia, M. S. Smith, D. S. Radchenko, J. S. Fraser, *et al.*, *Nat. Chem. Biol.*, 2025, **21**, 1039–1045.
15 G. Schneider and U. Fechner, *Nat. Rev. Drug Discovery*, 2005, **4**, 649–663.
16 Y. Du, A. R. Jamasb, J. Guo, T. Fu, C. Harris, Y. Wang, C. Duan, P. Liò, P. Schwaller and T. L. Blundell, *Nat. Mach. Intell.*, 2024, **6**, 589–604.
17 J. Arús-Pous, T. Blaschke, S. Ulander, J.-L. Reymond, H. Chen and O. Engkvist, *J. Cheminf.*, 2019, **11**, 20.
18 C. Grebner, E. Malmerberg, A. Shewmaker, J. Batista, A. Nicholls and J. Sadowski, *J. Chem. Inf. Model.*, 2019, **60**, 4274–4282.
19 I. Pöhner, T. Sivula and A. Poso, *Computer-Aided and Machine Learning-Driven Drug Design: From Theory to Applications*, Springer, 2025, pp. 299–343.
20 T. Sivula, L. Yetukuri, T. Kalliokoski, H. Kasnanen, A. Poso and I. Pohner, *J. Chem. Inf. Model.*, 2023, **63**, 5773–5783.
21 A. Acharya, R. Agarwal, M. B. Baker, J. Baudry, D. Bhowmik, S. Boehm, K. G. Byler, S. Chen, L. Coates, C. J. Cooper, *et al.*, *J. Chem. Inf. Model.*, 2020, **60**, 5832–5852.
22 P. C. Hawkins, A. G. Skillman and A. Nicholls, *J. Med. Chem.*, 2007, **50**, 74–82.
23 R. G. Coleman, M. Carchia, T. Sterling, J. J. Irwin and B. K. Shoichet, *PLoS One*, 2013, **8**, e75992.
24 R. A. Friesner, J. L. Banks, R. B. Murphy, T. A. Halgren, J. J. Klicic, D. T. Mainz, M. P. Repasky, E. H. Knoll, M. Shelley, J. K. Perry, *et al.*, *J. Med. Chem.*, 2004, **47**, 1739–1749.
25 D. Santos-Martins, L. Solis-Vasquez, A. F. Tillack, M. F. Sanner, A. Koch and S. Forli, *J. Chem. Theor. Comput.*, 2021, **17**, 1060–1073.
26 C. Gorgulla, A. Boeszoermenyi, Z.-F. Wang, P. D. Fischer, P. W. Coote, K. M. Padmanabha Das, Y. S. Malets, D. S. Radchenko, Y. S. Moroz, D. A. Scott, *et al.*, *Nature*, 2020, **580**, 663–668.
27 A. Alhossary, S. D. Handoko, Y. Mu and C.-K. Kwoh, *Bioinformatics*, 2015, **31**, 2214–2216.
28 M. Michino, A. Beautrait, N. A. Boyles, A. Nadupalli, A. Dementiev, S. Sun, J. Ginn, L. Baxt, R. Suto, R. Bryk, *et al.*, *ACS Bio Med Chem Au*, 2023, **3**, 507–515.
29 G. M. Sastry, S. L. Dixon and W. Sherman, *J. Chem. Inf. Model.*, 2011, **51**, 2455–2466.
30 N. Software, Arthor, https://www.nextmovesoftware.com/arthor.html.
31 N. Software, SmallWorld, https://www.nextmovesoftware.com/smallworld.html.
32 A. Vardanian, *USearch by Unum Cloud*, 2023, https://github.com/ashvardanian/usearch-molecules.
33 A. Dalke, *J. Cheminf.*, 2019, **11**, 76.
34 BioSolveIT, infinisee, https://www.biosolveit.de/infiniSee.
35 L. Bellmann, P. Penner and M. Rarey, *J. Chem. Inf. Model.*, 2020, **61**, 238–251.
36 K. Boyd, *RDKit User Group Meeting*, 2025.
37 A. A. Sadybekov, A. V. Sadybekov, Y. Liu, C. Iliopoulos-Tsoutsouvas, X.-P. Huang, J. Pickett, B. Houser, N. Patel, N. K. Tran, F. Tong, *et al.*, *Nature*, 2022, **601**, 452–459.
38 P. Beroza, J. J. Crawford, O. Ganichkin, L. Gendelev, S. F. Harris, R. Klein, A. Miu, S. Steinbacher, F.-M. Klingler and C. Lemmen, *Nat. Commun.*, 2022, **13**, 6447.
39 M. Lžičař and H. Gamouh, *ChemRxiv*, 2024, preprint, DOI: 10.26434/chemrxiv-2024-cswth.
40 M. Whittle, V. J. Gillet, P. Willett, A. Alex and J. Loesel, *J. Chem. Inf. Comput. Sci.*, 2004, **44**, 1840–1848.
41 R. Garnett, T. Gärtner, M. Vogt and J. Bajorath, *J. Comput.-Aided Mol. Des.*, 2015, **29**, 305–314.
42 J. Yang, R. G. Lal, J. C. Bowden, R. Astudillo, M. A. Hameedi, S. Kaur, M. Hill, Y. Yue and F. H. Arnold, *Nat. Commun.*, 2025, **16**, 714.
43 R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams and A. Aspuru-Guzik, *ACS Cent. Sci.*, 2018, **4**, 268–276.
44 N. T. Maus, H. T. Jones, J. S. Moore, M. J. Kusner, J. Bradshaw and J. R. Gardner, *Adv. Neural Inf. Process. Syst.*, 2022, **37**, 44009–44039.
45 F. Svensson, U. Norinder and A. Bender, *J. Chem. Inf. Model.*, 2017, **57**, 439–444.
46 D. E. Graff, E. I. Shakhnovich and C. W. Coley, *Chem. Sci.*, 2021, **12**, 7866–7881.
47 A. Luttens, I. Cabeza de Vaca, L. Sparring, J. Brea, A. L. Martínez, N. A. Kahlous, D. S. Radchenko, Y. S. Moroz, M. I. Loza, U. Norinder, *et al.*, *Nat. Comput. Sci.*, 2025, 1–12.
48 F. Gentile, V. Agrawal, M. Hsing, A.-T. Ton, F. Ban, U. Norinder, M. E. Gleave and A. Cherkasov, *ACS Cent. Sci.*, 2020, **6**(6), 939–949.
49 T. Kalliokoski, *Mol. Inf.*, 2021, **40**(9), 2100089.
50 T. Sivula, L. Yetukuri, T. Kalliokoski, H. Käsnänen, A. Poso and I. Pöhner, *J. Chem. Inf. Model.*, 2023, **63**, 5773–5783.
51 R. Garnett, *Bayesian Optimization*, Cambridge University Press, 2023.

52 D. E. Graff, M. Aldeghi, J. A. Morrone, K. E. Jordan, E. O. Pyzer-Knapp and C. W. Coley, *J. Chem. Inf. Model.*, 2022, **62**, 3854–3862.
53 G. Zhou, D.-V. Rusnac, H. Park, D. Canzani, H. M. Nguyen, L. Stewart, M. F. Bush, P. T. Nguyen, H. Wulff, V. Yarov-Yarovoy, N. Zheng and F. DiMaio, *Nat. Commun.*, 2024, **15**, 7761.
54 H. Park, G. Zhou, M. Baek, D. Baker and F. DiMaio, *J. Chem. Theor. Comput.*, 2021, **17**, 2000–2010.
55 M. McGann, *J. Chem. Inf. Model.*, 2011, **51**, 578–596.
56 Y. Yang, K. Yao, M. P. Repasky, K. Leswing, R. Abel, B. K. Shoichet and S. V. Jerome, *J. Chem. Theory Comput.*, 2021, **17**, 7106–7119.
57 J. Meyers, B. Fabian and N. Brown, *Drug Discovery Today*, 2021, **26**, 2707–2715.
58 N. Brown, B. McKay, F. Gilardoni and J. Gasteiger, *J. Chem. Inf. Comput. Sci.*, 2004, **44**, 1079–1087.
59 M. H. S. Segler, T. Kogej, C. Tyrchan and M. P. Waller, *ACS Cent. Sci.*, 2017, **4**, 120–131.
60 R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams and A. Aspuru-Guzik, *ACS Cent. Sci.*, 2018, **4**, 268–276.
61 J. Yang, W. Chu, D. Khalil, R. Astudillo, B. J. Wittmann, F. H. Arnold and Y. Yue, *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
62 M. Thomas, N. M. O'Boyle, A. Bender and C. de Graaf, *J. Cheminf.*, 2022, **14**, 68.
63 R. Winter, F. Montanari, A. Steffen, H. Briem, F. Noé and D.-A. Clevert, *Chem. Sci.*, 2019, **10**, 8016–8024.
64 J. Arús-Pous, S. V. Johansson, O. Prykhodko, E. J. Bjerrum, C. Tyrchan, J.-L. Reymond, H. Chen and O. Engkvist, *J. Cheminf.*, 2019, **11**, 71.
65 A. Fischer, M. Smiesko, M. Sellner and M. A. Lill, *J. Med. Chem.*, 2021, **64**, 2489–2500.
66 J. B. Baell and G. A. Holloway, *J. Med. Chem.*, 2010, **53**, 2719–2740.
67 G. R. Bickerton, G. V. Paolini, J. Besnard, S. Muresan and A. L. Hopkins, *Nat. Chem.*, 2012, **4**, 90–98.
68 A. D. Blevins and I. K. Quigley, *arXiv*, 2025, preprint, arXiv:2512.20924, DOI: 10.48550/arXiv.2512.20924.
69 D. G. Brown, *J. Med. Chem.*, 2025, **68**(22), 23751–23780.
70 M. Hopper, T. Gururaja, T. Kinoshita, J. P. Dean, R. J. Hill and A. Mongan, *J. Pharmacol. Exp. Ther.*, 2020, **372**, 331–338.
71 L. M. Juárez-Salcedo, V. Desai and S. Dalia, *Drugs Context*, 2019, **8**, 212574.
72 J. M. Flack, M. Azizi, J. M. Brown, J. P. Dwyer, J. Fronczek, E. S. Jones, D. S. Olsson, S. Perl, H. Shibata, J.-G. Wang, *et al.*, *N. Engl. J. Med.*, 2025, **393**, 1363–1374.
73 D. A. DeGoey, H.-J. Chen, P. B. Cox and M. D. Wendt, *J. Med. Chem.*, 2017, **61**, 2636–2651.
74 P. Renz, S. Luukkonen and G. Klambauer, *J. Chem. Inf. Model.*, 2024, **64**, 5756–5761.
75 M. Thomas, N. M. O'Boyle, A. Bender and C. De Graaf, *arXiv*, 2022, preprint, arXiv:2212.01385, DOI: 10.48550/arXiv.2212.01385.
76 A. Bou, M. Thomas, S. Dittert, C. Navarro, M. Majewski, Y. Wang, S. Patel, G. Tresadern, M. Ahmad, V. Moens, *et al.*, *J. Chem. Inf. Model.*, 2024, **64**, 5900–5911.
77 F. Cai, K. Zacour, T. Zhu, T.-R. Tzeng, Y. Duan, L. Liu, S. Pilla, G. Li and F. Luo, *arXiv*, 2022, preprint, arXiv:2410.21422, DOI: 10.48550/arXiv.2410.21422.
78 N. Brown, M. Fiscato, M. H. Segler and A. C. Vaucher, *J. Chem. Inf. Model.*, 2019, **59**, 1096–1108.
79 D. Polykovskiy, A. Zhebrak, B. Sanchez-Lengeling, S. Golovanov, O. Tatanov, S. Belyaev, R. Kurbanov, A. Artamonov, V. Aladinskiy, M. Veselov, *et al.*, *Front. Pharmacol.*, 2020, **11**, 565644.
80 M. Thomas, P. G. Matricon, R. J. Gillespie, M. Napiórkowska, H. Neale, J. S. Mason, J. Brown, K. Harwood, C. Fieldhouse, N. A. Swain, *et al.*, *Nat. Commun.*, 2025, **16**, 5485.
81 W. Gao and C. W. Coley, *J. Chem. Inf. Model.*, 2020, **60**, 5714–5723.
82 K. Handa, M. C. Thomas, M. Kageyama, T. Iijima and A. Bender, *J. Cheminf.*, 2023, **15**, 112.
83 W. Gao, T. Fu, J. Sun and C. Coley, *Adv. Neural Inf. Process. Syst.*, 2022, **35**, 21342–21357.
84 C. A. Goodhart, *Monetary theory and practice: The UK experience*, Springer, 1984, pp. 91–121.
85 P. Renz, D. Van Rompaey, J. K. Wegner, S. Hochreiter and G. Klambauer, *Drug Discovery Today: Technol.*, 2019, **32**, 55–63.
86 M. Langevin, R. Vuilleumier and M. Bianciotto, *J. Cheminf.*, 2022, **14**, 20.
87 M. García-Ortegón, G. N. C. Simm, A. J. Tripp, J. M. Hernández-Lobato, A. Bender and S. Bacallado, *J. Chem. Inf. Model.*, 2022, **62**, 3486–3502.
88 M. Thomas, A. Bou and G. De Fabritiis, *J. Chem. Inf. Model.*, 2025, **65**(24), 13178–13186.
89 D. D. Martinelli, *Comput. Biol. Med.*, 2022, **145**, 105403.
90 Y. A. Ivanenkov, D. Polykovskiy, D. Bezrukov, B. Zagribelnyy, V. Aladinskiy, P. Kamya, A. Aliper, F. Ren and A. Zhavoronkov, *J. Chem. Inf. Model.*, 2023, **63**, 695–701.
91 F. Ren, A. Aliper, J. Chen, H. Zhao, S. Rao, C. Kuppe, I. V. Ozerov, M. Zhang, K. Witte, C. Kruse, V. Aladinskiy, Y. Ivanenkov, D. Polykovskiy, Y. Fu, E. Babin, J. Qiao, X. Liang, Z. Mou, H. Wang, F. W. Pun, P. Torres-Ayuso, A. Veviorskiy, D. Song, S. Liu, B. Zhang, V. Naumov, X. Ding, A. Kukharenko, E. Izumchenko and A. Zhavoronkov, *Nat. Biotechnol.*, 2024, **43**, 63–75.
92 B. P. Munson, M. Chen, A. Bogosian, J. F. Kreisberg, K. Licon, R. Abagyan, B. M. Kuenzi and T. Ideker, *Nat. Commun.*, 2024, **15**, 3636.
93 M. Korablyov, C.-H. Liu, M. Jain, A. M. van der Sloot, E. Jolicoeur, E. Ruediger, A. C. Nica, E. Bengio, K. Lapchevskyi, D. St-Cyr, D. A. Schuetz, V. I. Butoi, J. Rector-Brooks, S. Blackburn, L. Feng, H. Nekoei, S. Gottipati, P. Vijayan, P. Gupta, L. Rampášek, S. Avancha, P.-L. Bacon, W. L. Hamilton, B. Paige, S. Misra, S. K. Jastrzebski, B. Kaul, D. Precup, J. M. Hernández-Lobato, M. Segler, M. Bronstein,

A. Marinier, M. Tyers and Y. Bengio, *arXiv*, 2024, preprint, DOI: 10.48550/arXiv.2405.01616.
94 V. Fialková, J. Zhao, K. Papadopoulos, O. Engkvist, E. J. Bjerrum, T. Kogej and A. Patronov, *J. Chem. Inf. Model.*, 2021, **62**, 2046–2063.
95 J. Guo, F. Knuth, C. Margreitter, J. P. Janet, K. Papadopoulos, O. Engkvist and A. Patronov, *Digital Discovery*, 2023, **2**, 392–408.
96 E. Noutahi, C. Gabellini, M. Craig, J. S. Lim and P. Tossou, *Digital Discovery*, 2024, **3**, 796–804.
97 K. Maziarz, H. Jackson-Flux, P. Cameron, F. Sirockin, N. Schneider, N. Stiefl, M. Segler and M. Brockschmidt, *ICLR*, 2022, 2022.
98 M. Thomas, M. Ahmad, G. Tresadern and G. De Fabritiis, *J. Cheminf.*, 2024, **16**, 77.
99 A. Gaulton, L. J. Bellis, A. P. Bento, J. Chambers, M. Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, *et al.*, *Nucleic Acids Res.*, 2012, **40**, D1100–D1107.
100 S. Chen and Y. Jung, *J. Cheminf.*, 2024, **16**, 83.
101 A. Thakkar, V. Chadimová, E. J. Bjerrum, O. Engkvist and J.-L. Reymond, *Chem. Sci.*, 2021, **12**, 3339–3349.
102 S. Genheden, A. Thakkar, V. Chadimová, J.-L. Reymond, O. Engkvist and E. Bjerrum, *J. Cheminf.*, 2020, **12**, 70.
103 M. Stanley and M. Segler, *Curr. Opin. Struct. Biol.*, 2023, **82**, 102658.
104 M. Thomas, A. Bou, J. C. Gómez-Tamayo, G. Tresadern, M. Ahmad and G. De Fabritiis, *J. Chem. Inf. Model.*, 2025.
105 S. M. Papidocha, A. Burger, V. Bernales and A. Aspuru-Guzik, *Curr. Opin. Chem. Eng.*, 2026, **51**, 101217.
106 J. J. Irwin and B. K. Shoichet, *J. Chem. Inf. Model.*, 2005, **45**, 177–182.
107 J. Guo, V. Sabanza-Gil, Z. Jončev, J. S. Luterbacher and P. Schwaller, *arXiv*, 2025, preprint, arXiv:2505.08774, DOI: 10.48550/arXiv.2505.08774.
108 A. K. Hassen, M. Šícho, Y. J. van Aalst, M. C. W. Huizenga, D. N. R. Reynolds, S. Luukkonen, A. Bernatavicius, D.-A. Clevert, A. P. A. Janssen, G. J. P. van Westen and M. Preuss, *J. Cheminf.*, 2025, **17**, 41.
109 K. Swanson, G. Liu, D. B. Catacutan, A. Arnold, J. Zou and J. M. Stokes, *Nat. Mach. Intell.*, 2024, **6**, 338–353.
110 J. Bradshaw, B. Paige, M. J. Kusner, M. H. S. Segler and J. M. Hernández-Lobato, *arXiv*, 2020, preprint, DOI: 10.48550/arXiv.2012.11522.
111 C. Descamps, V. Bouttier, J. Sanz García, M. Lhuillier-Akakpo, Q. Perron and H. Tajmouati, *Briefings Bioinf.*, 2025, **26**.
112 W. Gao, S. Luo and C. W. Coley, *Proc. Natl. Acad. Sci. U.S.A.*, 2025, **122**, e2415665122.
113 S. K. Gottipati, B. Sattarov, S. Niu, Y. Pathak, H. Wei, S. Liu, K. M. J. Thomas, S. Blackburn, C. W. Coley, J. Tang, S. Chandar and Y. Bengio, *arXiv*, 2020, preprint, DOI: 10.48550/arXiv.2004.12485.
114 M. Cretu, C. Harris, I. Igashov, A. Schneuing, M. Segler, B. Correia, J. Roy, E. Bengio and P. Liò, *arXiv*, 2025, preprint, arXiv:2405.01155, DOI: 10.48550/arXiv.2405.01155.
115 S. Passaro, G. Corso, J. Wohlwend, M. Reveiz, S. Thaler, V. R. Somnath, N. Getz, T. Portnoi, J. Roy, H. Stark, D. Kwabi-Addo, D. Beaini, T. Jaakkola and R. Barzilay, *bioRxiv*, 2025, preprint, DOI: 10.1101/2025.06.14.659707.
116 S. H. P de Oliveira, A. Pedawi, V. Kenyon and H. van den Bedem, *J. Med. Chem.*, 2024, **67**, 19417–19427.
117 K. Swanson, G. Liu, D. B. Catacutan, S. McLellan, A. Arnold, M. M. Tu, E. D. Brown, J. Zou and J. M. Stokes, *bioRxiv*, 2025, preprint, DOI: 10.1101/2025.05.17.654017.
118 K. Klarich, B. Goldman, T. Kramer, P. Riley and W. P. Walters, *J. Chem. Inf. Model.*, 2024, **64**, 1158–1171.
119 S. Tang, J. Ding, X. Zhu, Z. Wang, H. Zhao and J. Wu, *IEEE ACM Trans. Comput. Biol. Bioinf.*, 2024.
120 M. Sacha, M. Błaz, P. Byrski, P. Dabrowski-Tumanski, M. Chrominski, R. Loska, P. Włodarczyk-Pruszynski and S. Jastrzebski, *J. Chem. Inf. Model.*, 2021, **61**, 3273–3284.
121 J. Lyu, S. Wang, T. E. Balius, I. Singh, A. Levit, Y. S. Moroz, M. J. O'Meara, T. Che, E. Algaa, K. Tolmachova, A. A. Tolmachev, B. K. Shoichet, B. L. Roth and J. J. Irwin, *Nature*, 2019, **566**, 224–229.
122 B. W. Hall, T. A. Tummino, K. Tang, O. Mailhot, M. Castanon, J. J. Irwin and B. K. Shoichet, *J. Chem. Inf. Model.*, 2025, **65**, 4458–4467.
123 O. Trott and A. J. Olson, *J. Comput. Chem.*, 2009, **31**, 455–461.

