



## OPEN STELLA provides a drug design framework enabling extensive fragment-level chemical space exploration and balanced multi-parameter optimization

Hokyun Jeon, Jin Gyu Lee, Wonseok Shin, Hyunjun Ji, InSuk Joung✉ & Hui Sun Lee✉

In drug discovery, identifying molecules with desired pharmacological properties remains challenging, as conventional methods often rely on exhaustive trial-and-error and limited exploration of chemical space. Here, we present STELLA, a metaheuristics-based generative molecular design framework that combines an evolutionary algorithm for fragment-based chemical space exploration with a clustering-based conformational space annealing method for efficient multi-parameter optimization. Additionally, it leverages deep learning models for accurate prediction of pharmacological properties. Our case study, which focuses on docking score and quantitative estimate of drug-likeness as primary objectives, demonstrates that STELLA generates 217% more hit candidates with 161% more unique scaffolds and achieves more advanced Pareto fronts compared to REINVENT 4. In performance evaluations optimizing 16 properties simultaneously for MolFinder, REINVENT 4, and STELLA, STELLA consistently outperforms the control methods by achieving better average objective scores and exploring a broader region of chemical space. The results highlight STELLA's superior performance in both efficient exploration of chemical space and multi-parameter optimization, indicating that STELLA is a powerful tool for de novo molecular design.

**Keywords** Multi-parameter optimization, Deep learning, Drug discovery, Chemical space exploration, Evolutionary algorithm

Small molecule drug discovery faces a fundamental challenge: the vast and nearly infinite molecular space. The number of theoretically synthesizable organic compounds is estimated to range between  $10^{30}$  and  $10^{601,2}$ . While this sheer diversity holds immense potential for discovering more effective drugs, it also presents a major bottleneck—efficiently navigating the chemical space to identify promising drug candidates. Traditional experimental screening methods cannot comprehensively explore such a vast chemical space, necessitating computational approaches to guide molecular design<sup>3</sup>. However, even computational methods, briefly described below, struggle with the trade-off between exploration and exploitation, making it difficult to efficiently generate novel, synthesizable, and biologically relevant drug candidates<sup>4</sup>. Addressing this challenge requires strategies that can systematically search chemical space while maintaining drug-like properties and synthetic feasibility.

Virtual screening is an option for computationally exploring vast chemical space. Given a 3D structure of a target protein, molecular docking screens large compound libraries by fitting molecules into the protein structure and evaluating their binding affinities<sup>5</sup>. The size of libraries for commercially available and virtually enumerated compounds has been rapidly expanding, leading to the development of ultra-large scale docking workflows<sup>6–8</sup>. However, the cost of docking increases roughly linearly with the number of compounds in the libraries, making it computationally daunting to perform docking calculations against billions of molecules. Furthermore, the compound libraries used for the large-scale virtual screening campaigns are still a tiny fraction of the full chemical space.

De novo drug design is a computational approach aimed at generating drug-like molecules from scratch. For instance, the lead optimization stage of drug discovery involves refining candidate molecules through highly iterative design-make-test cycles. An ideal de novo design method can quickly generate a relatively small

Standigm Inc, 182 Dogok-ro, Gangnam-gu, Seoul, South Korea. ✉email: insuk.joung@standigm.com; hui.sun.lee@standigm.com

number of molecules with desired pharmacological properties by focusing chemical space exploration on the most relevant areas. Unlike virtual screening, which evaluates predefined compound libraries, *de novo* design enables the creation of entirely novel molecular structures by actively exploring chemical space. The potential of *de novo* drug design to drastically reduce the time and cost of drug development has garnered considerable attention over the past decade, driving scientific communities to invest significant efforts in developing more effective molecular design methods<sup>9–12</sup>.

Drug discovery is a challenging multi-objective problem. Drugs are molecules with pharmacological profiles that encompass various properties (e.g., binding affinity, pharmacokinetic properties, toxicity, drug-drug interactions, and synthesizability), many of which conflict with each other<sup>10</sup>. This multi-parameter optimization problem in drug discovery is a process of searching for solutions within a vast chemical space, while balancing complex pharmacological properties. A key challenge in this process is to efficiently find near-optimal solutions—ideally global optima—by avoiding being trapped in low-quality local minima<sup>13</sup>. To address this, numerous computational methods have been developed to advance *de novo* drug design. The methods commonly employ two major approaches for molecular generation: metaheuristics and deep learning.

Metaheuristic methods explore chemical space by employing population-based stochastic optimization procedures<sup>14</sup>. We briefly highlight exemplar studies that adopt metaheuristic algorithms for generative molecular design. GANDI is a fragment-based method, where pre-docked fragments are encoded by the genetic algorithm, and suitable linkers are joined with a tabu search<sup>15</sup>. AutoGrow4 utilizes a genetic algorithm to generate a new population using SMARTS-reaction notation, followed by docking to select the next generation in terms of docking score<sup>16</sup>. EvoMol is a graph-based molecular generator, where molecules are represented as graphs, and the molecular graphs are sequentially mutated using seven atom-centered operators<sup>17</sup>. MolFinder uses the conformational space annealing (CSA) algorithm for global optimization of molecular properties<sup>18,19</sup>. The tool directly utilizes SMILES representation to explore chemical space. GENERA is a computational method combining a deep learning approach for drug-like analogue design with a genetic algorithm for generating molecules<sup>20</sup>. Winter et al. proposed a method called Molecule Swarm Optimization. This method integrates molecular property prediction with Particle Swarm Optimization to generate optimized molecules<sup>21</sup>.

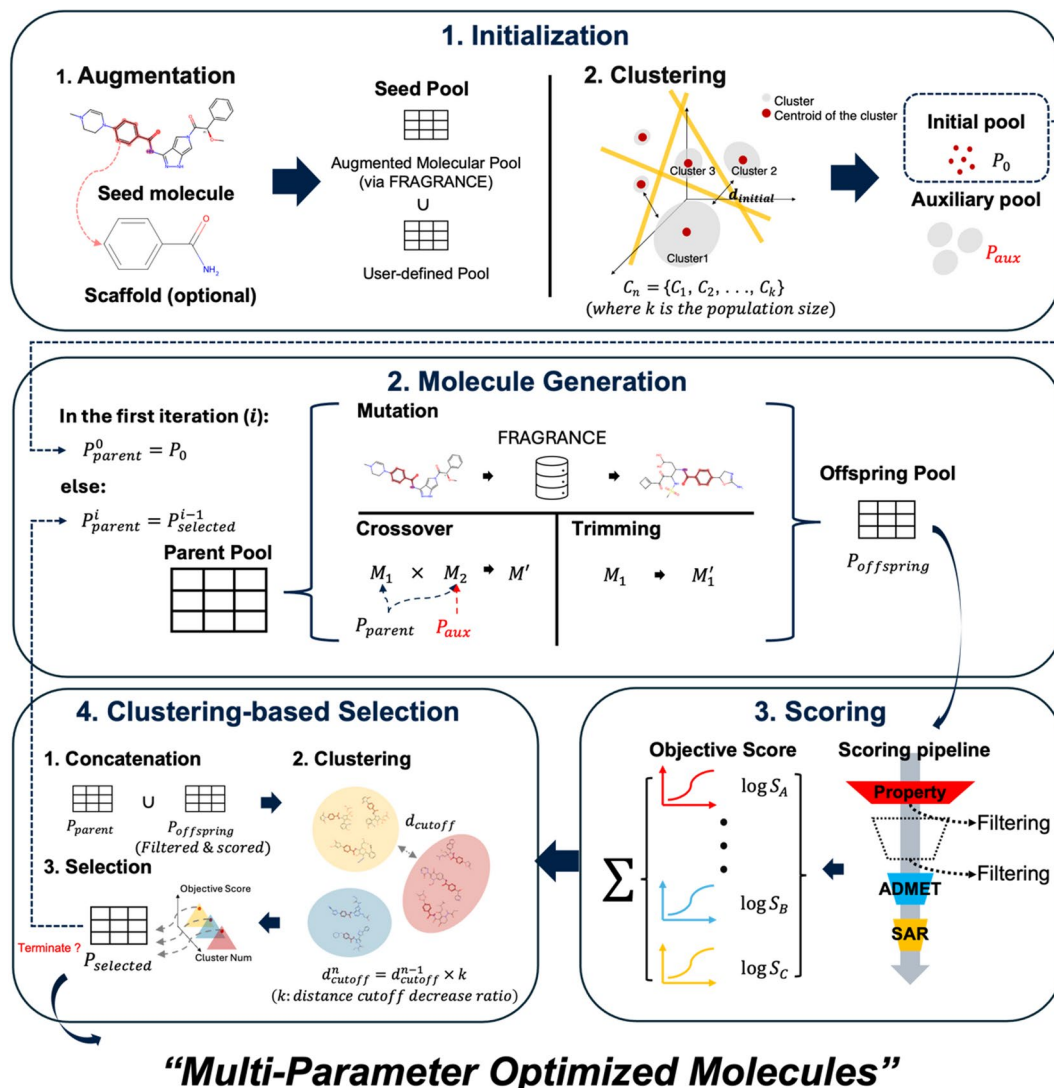
Deep learning-based algorithms can extract structural features of existing molecules, identify the patterns, and then replicate those patterns into novel molecules with notable efficacy<sup>22</sup>. A variety of deep learning architectures have been applied to molecular generation, showing the potential of generative models for chemical space exploration. REINVENT<sup>23</sup> employs reinforcement learning using recurrent neural networks and transformers as deep learning architectures to generate *de novo* compounds. This tool was expanded to REINVENT 2.0 to cover both distribution-learning and goal-directed scenarios<sup>24</sup> and to REINVENT 4 to include new functionalities such as staged learning and new transformer models for molecule optimization<sup>25</sup>. LiGAN is a computational tool for generating 3D molecular structures on a given receptor binding site. The generation model was trained using a conditional variational autoencoder on cross-docked protein-ligand structures<sup>26</sup>. QADD uses multi-objective deep reinforcement learning to generate molecules with desired properties. In this method, molecules are represented as graphs, and the molecule generator is integrated with a graph neural network-based model for accurate molecular quality assessment<sup>27</sup>. Chemistry42 is a customizable web-based platform for molecular design and optimization. It offers more than 40 multiple generative models, including generative autoencoders, generative adversarial networks, flow-based approaches, and language models<sup>28</sup>. Despite its strengths, a major challenge faced by deep learning-based molecular generation is its dependency on large, high-quality datasets for training, which limits its applicability<sup>29</sup>.

In this study, we present STELLA (Systematic Tool for Evolutionary Lead optimization Leveraging Artificial intelligence), a seamless framework for generating molecules with optimized multiple pharmacological properties. In STELLA, an efficient evolutionary algorithm is used to enhance fragment-level chemical space exploration for molecular generation. In addition, STELLA utilizes a clustering-based CSA to effectively identify near-optimal solutions by balancing exploration and exploitation. A fragment replacement method and graph transformer-based deep learning models are also integrated to enhance molecule generation and improve the accuracy of pharmacological property prediction, respectively. We compare the performance of STELLA with REINVENT 4, a widely used deep learning-based tool, and MolFinder, a metaheuristics-based tool like STELLA. The evaluation results demonstrate that STELLA's reliable performance in not only exploring a wider range of chemical space but also generating molecules with better objective scores.

## Results and discussion

### Overview of STELLA workflow

Figure 1 schematically presents an overall workflow of STELLA. Each step of the workflow—*Initialization*, *Molecule Generation*, *Scoring*, and *Clustering-based Selection*—is described in more detail in the subsections of the Methods section. Given an input seed molecule, STELLA begins with the *Initialization*, where an initial pool is generated by molecular mutation using FRAGRANCE (see the Methods section for details). A user-defined pool of molecules can optionally be added to the initial pool. Variants of the molecules in the initial pool are then generated by the FRAGRANCE mutation, maximum common substructure (MCS)-based crossover, and trimming (*Molecule Generation*). Each generated molecule is scored using an objective function that incorporates user-defined molecular properties to be optimized (*Scoring*). All generated molecules, along with the initial pool are clustered with a distance cutoff, and molecules with the best objective score are selected from each cluster. If the number of the top-scoring molecules does not meet a target value, the next best molecules are iteratively selected from each cluster (*Clustering-based Selection*). The *Molecule Generation*, *Scoring*, and *Clustering-based Selection*, which are collectively referred to as multi-parameter optimization using clustering-based CSA, are repeated in a loop. The distance cutoff in the *Clustering-based Selection* is progressively reduced in each cycle, gradually transitioning the selection criteria from maintaining structural diversity to optimizing



**Fig. 1.** The overall workflow of STELLA. The initialization generates a seed pool, followed by iterative cycles of molecule generation, scoring, and clustering-based selection. The algorithm terminates when predefined conditions are met, yielding an optimized molecular pool for multi-parameter objectives.

objective function. After the selection step, the algorithm evaluates the termination conditions. If any condition is met, the iteration process terminates.

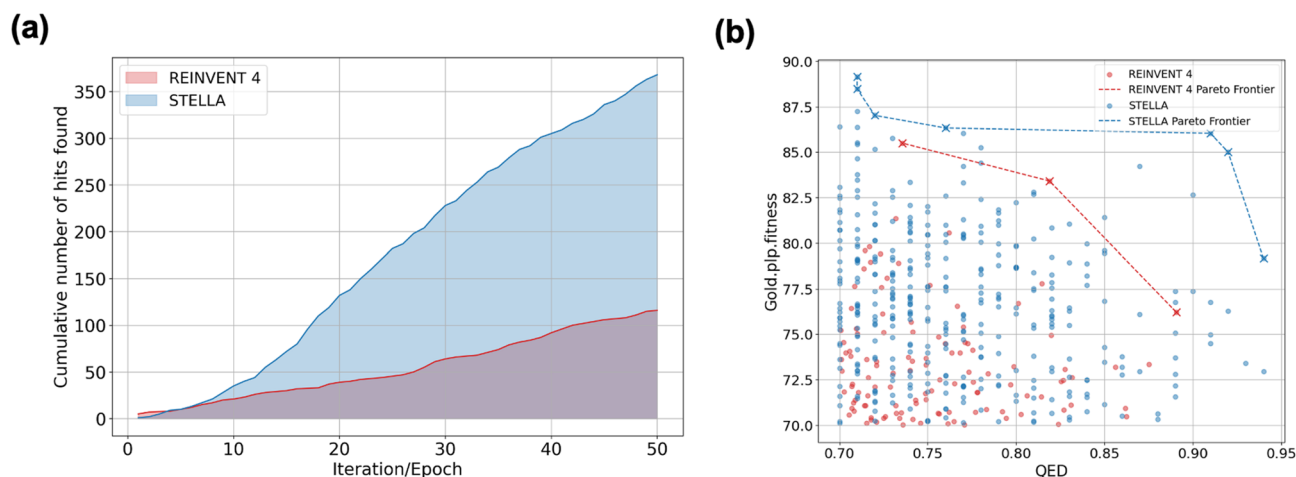
#### Comparison with the case study in REINVENT 4

REINVENT 4 is a framework for designing small molecules with optimized multiple properties using generative AI models, implemented through reinforcement learning and a curriculum learning-based optimization algorithm<sup>25</sup>. The study presented a case study demonstrating its applicability in a hypothetical virtual screening scenario to identify novel phosphoinositide-dependent kinase-1 (PDK1) inhibitors<sup>30</sup>. We reproduced this case study using REINVENT 4 and STELLA under identical computational conditions to compare the results (Table 1; Fig. 2).

Due to the limited access to the commercial tools, LigPrep and Glide were replaced with an alternative in-house ligand preparation method utilizing the OpenEye toolkit (version 2023.1.1) and CCDC's GOLD docking software (version 2024.2.0), respectively<sup>31–33</sup>. The docking score threshold for hit identification was changed from a Glide docking score of  $\leq -8$  kcal/mol to a GOLD PLP Fitness score of  $\geq 70$ , while retaining the REINVENT 4 case study's quantitative estimate of drug-likeness (QED) threshold of  $\geq 0.7$ . Both metrics were weighted equally in the objective score. We employed the same workflow as in the REINVENT 4 study, beginning with 10 epochs of transfer learning followed by 50 epochs of reinforcement learning with a batch size of 128. The batch size represents the number of molecules generated, scored, and incorporated into each epoch. To ensure comparable computational conditions in STELLA, 128 molecules were generated per iteration of the genetic algorithm, with a total of 50 iterations performed. It should be noted that the terminology differs between two methods: one

	REINVENT 4	STELLA
Cumulative number of hits	116	368
Average score of hits (Std.)		
Gold.PLP.Fitness	73.37 (3.07)	76.80 (4.81)
QED	0.75 (0.04)	0.77 (0.06)
Average hit rate per iteration/epoch (%)	1.81	5.75
Number of unique generic Murcko scaffolds in hits	115	276
Best similarity of hits to the crystal ligand	0.24	0.23

**Table 1.** A comparison of hit compounds generated by REINVENT 4 and STELLA.



**Fig. 2.** Performance comparison between REINVENT 4 and STELLA. (a) Cumulative number of hits over epochs for REINVENT 4 and iterations for STELLA. (b) Distribution of hits over the projected surface of Gold.PLP.Fitness and QED, with REINVENT 4 hits shown in red and STELLA hits in blue. The Pareto frontiers are visualized as dashed lines connecting the best-performing hits.

iteration in STELLA corresponds to one epoch in REINVENT 4. Detailed configurations of REINVENT 4 and STELLA used for this case study are provided in Supplementary Information Data S1.

In Table 1, we compare the number of hits, optimized properties, and scaffold diversity to evaluate the sampling efficiency of REINVENT 4 and STELLA. Over 50 training iterations/epochs, REINVENT 4 generated 116 hit compounds (average 1.81% hit rate per epoch) with mean scores of 73.37 for GOLD PLP Fitness and 0.75 for QED. In contrast, STELLA produced 368 hit compounds (average 5.75% hit rate per iteration) with higher mean scores of 76.80 for GOLD PLP Fitness and 0.77 for QED. The superior performance of STELLA was further validated by plots of cumulative number of hits over iterations/epochs and distribution of the final hits over Gold.PLP.Fitness and QED in Fig. 2. STELLA maintains a higher cumulative number of hits throughout the optimization process after the initial stages (Fig. 2a). In addition, STELLA-generated hits occupy a more favorable Pareto frontier, indicating better optimization of target properties (Fig. 2b). As shown in Table 1, STELLA also demonstrates substantially greater scaffold diversity, generating 276 unique generic Murcko scaffolds compared to 115 in REINVENT 4, while both methods yielded comparable maximum structural similarities to the native pyrazoloquinazoline inhibitor (PDB entry: 2XCH). Overall, the results indicate that STELLA not only explores a broader region of chemical space but also more effectively optimizes the target properties than REINVENT 4 in this case study.

### Benchmark performance evaluation for multi-parameter optimization

To further assess performance, we evaluated the molecular optimization capabilities of MolFinder, REINVENT 4, and STELLA for two target proteins: tyrosine-protein kinase Abl1 (Abl1) and cellular tumor antigen p53 (p53). The goal of this benchmark was to optimize a single objective score that encompassed 16 properties relevant to drug candidate evaluation, including: (1) GOLD docking score (Gold.PLP.Fitness). PDB entries 4TWP and 4AGQ, obtained from the PDBbind Core Set (v2020)<sup>34</sup> were used for Abl1 and p53, respectively. (2) 3D similarity to the reference ligand (i.e., ligands in 4TWP and 4AGQ) calculated by OpenEye's ROCS<sup>35</sup> (Tanimoto Combo score). (3) RDKit descriptors—QED, AlogP, and the synthetic accessibility score<sup>36</sup> (SAscore). (4) 11 properties from our proprietary deep learning model, QIP-ADMET (Caco-2 permeability, aqueous solubility, plasma protein binding ratio, predictions for CYP inhibition and substrate probabilities for CYP2D6, CYP3A4, and CYP2C9, hepatocyte clearance, and hERG inhibition probability)<sup>37</sup>. Designing molecules with high binding affinity is a key objective in drug discovery. Based on empirical tuning, the GOLD docking score



was assigned a weight four times higher than that of other properties to approximately balance the contribution of binding affinity with other components. To maintain drug-likeness constraints, molecules with a molecular weight exceeding 600 Da were filtered out. All scoring components were implemented identically in MolFinder, REINVENT 4, and STELLA.

In many real-world hit-to-lead or lead optimization scenarios, certain chemical moieties (i.e., scaffold) remain unaltered to reduce synthetic cost, improve SAR interpretability, and align with specific project goals. To reflect these conditions, we arbitrarily designated scaffolds in each reference ligand with restrictions of no more than two rings, a single side chain, and a molecular weight below 200 Da (Supplementary Figure S1). We used the same parameters as in the MolFinder study for molecule generation and the global optimization of molecular properties<sup>18</sup>. For molecule generation in REINVENT 4, we used LibINVENT, a generative tool specialized in reaction-based scaffold decoration<sup>38</sup>. All three methods were configured to generate 2,500 molecules per iteration/epoch with 50 iterations for MolFinder and STELLA, and 50 epochs for REINVENT 4.

The top 300 molecules were selected from the final iteration/epoch based on the objective score, where a lower value indicates more optimized properties. Table 2 summarizes the assessment results for the objective score and structural diversity of the selected molecules. STELLA achieved average objective scores that were 0.05 and 1.03 lower than MolFinder and REINVENT 4, respectively, for Abl1, and 1.39 and 0.46 lower for p53, demonstrating STELLA's effectiveness in minimizing the objective function.

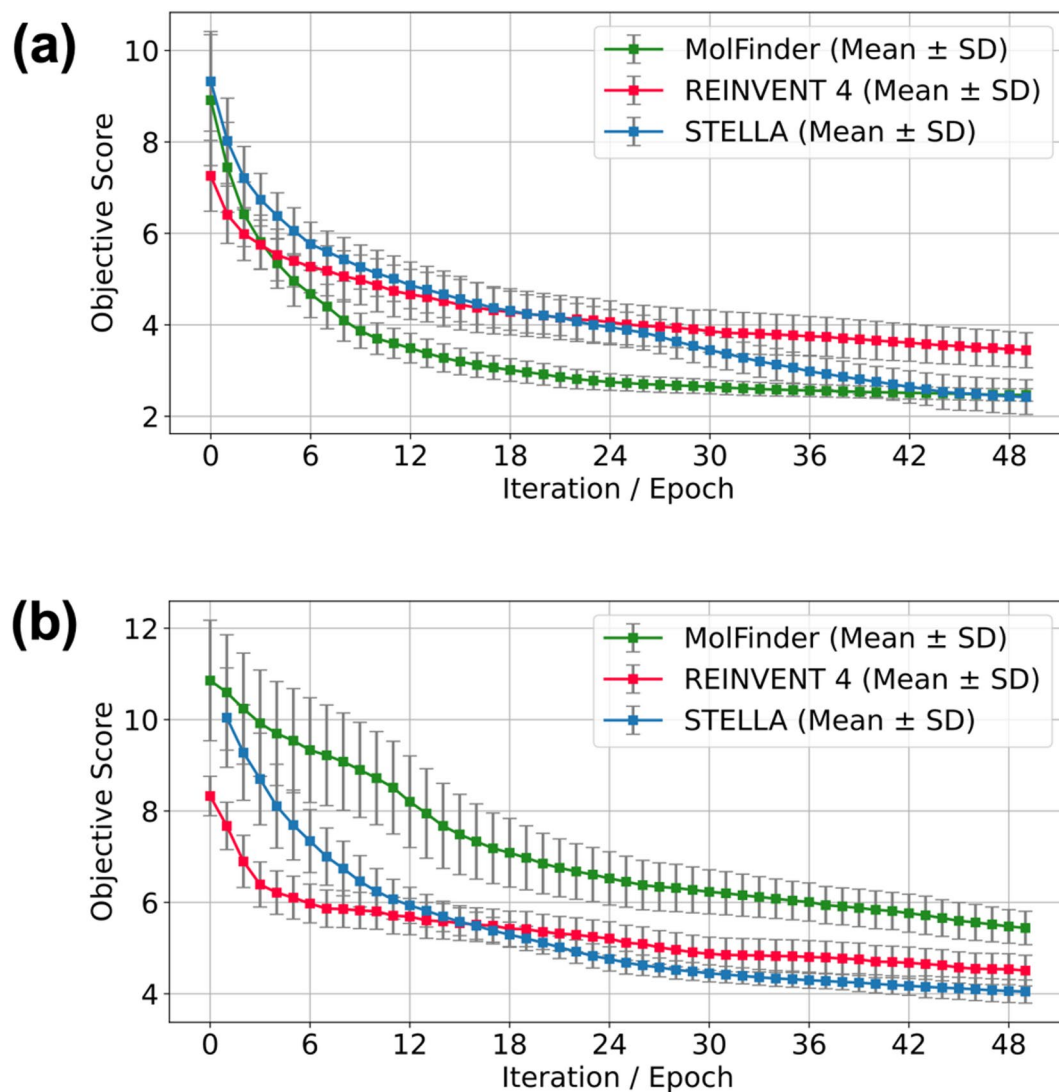
Structural diversity of the generated molecules was evaluated using an average of all-against-all pairwise Tanimoto similarity values of ECFP4 fingerprints and the number of clusters based on Sphere Exclusion algorithm with a similarity threshold of 0.3. The average pairwise Tanimoto similarity values for STELLA are 0.28 and 0.26 for Abl1 and p53, respectively, both of which are lower than those of REINVENT 4 (0.31 for both targets). Compared to MolFinder, STELLA yielded a lower similarity value for Abl1 (0.28 vs. 0.33), while MolFinder had a lower value for p53 (0.23 vs. 0.26). The Sphere Exclusion clustering analysis shows a similar trend, in which STELLA produced the largest number of clusters for Abl1, whereas MolFinder did so for p53. It is noted that STELLA maintains a balanced number of clusters across both targets (18 vs. 20 for Abl1 and p53, respectively), whereas the control tools exhibit large disparities between targets (7 vs. 27 for MolFinder and 11 vs. 21 for REINVENT 4). Although MolFinder achieved the highest number of clusters for p53 among the three methods, this result should be interpreted with caution, as its corresponding average objective score was the worst. Overall, the results demonstrate STELLA's consistently reliable performance in effectively optimizing the objective score and generating diverse molecules compared to the control tools.

Figure 3 are plots of average objective scores over iterations/epochs. The top 300 molecules were selected in the first iteration/epoch. In the subsequent iterations/epochs, newly generated molecules were combined with the 300 molecules selected in the previous iteration/epoch, and the top 300 molecules were then selected from the combined pool. The average objective scores were calculated at each iteration/epoch. All three methods exhibit a similar trend, where initial average objective scores gradually decrease as iterations/epochs progress. While STELLA initially generates molecules with higher average objective scores compared to REINVENT 4, STELLA outperforms REINVENT 4 after 23 and 17 iterations/epochs for Abl1 and p53, respectively. This suggests that REINVENT 4, which is based on a pre-trained generative model, has already learned general patterns of drug-like molecules and can generate promising candidates early in the optimizing process. However, its sampling is less robust than that of STELLA, resulting in reduced effectiveness in optimizing the objective score as the optimizing progresses. MolFinder exhibited a trend similar to STELLA for Abl1, achieving a comparable average objective score at the final iteration (2.47 for MolFinder and 2.42 for STELLA). However, MolFinder's ability to optimize the objective score for p53 was significantly lower compared to REINVENT 4 and STELLA.

We plotted the distribution of each property for the top 300 molecules generated in the final iteration/epoch by MolFinder, REINVENT 4, and STELLA (Supplementary Figure S2). To easily assess how many compounds fall within a pharmacologically reliable property range, we defined hit criteria for each property (Supplementary Table S1). Predefined hit thresholds were applied to QED, SAScore, AlogP, Gold docking score, and ROCS Tanimoto combo. ADMET prediction values were evaluated using a threshold of 0.5 for classification models, while for regression models, we adopted criteria reported in a previous study<sup>39</sup> on ADMET predictions. MolFinder, REINVENT 4, and STELLA achieved balanced optimization across multiple properties, avoiding over-reliance on a single score. MolFinder and STELLA outperformed in optimizing GOLD docking score for both Abl1 and p53 (see Supplementary Figure S3 for the progression of docking scores over the optimization steps), whereas REINVENT 4 focused more on SAScore and QED. Notably, most molecules generated by MolFinder exhibited unfavorable QED values for both targets, often featuring long branched structures that appear non-druggable (Supplementary Figure S4). For CYP2C9 inhibition in the p53 case, STELLA produced a significantly higher

	MolFinder		REINVENT 4		STELLA	
	Abl1	p53	Abl1	p53	Abl1	p53
Average objective score (Std.) ↓	2.47 (0.14)	5.44 (0.37)	3.45 (0.38)	4.51 (0.34)	2.42 (0.38)	4.05 (0.25)
Average pairwise similarity (Std.) ↓	0.33 (0.1)	0.23 (0.1)	0.31 (0.07)	0.31 (0.07)	0.28 (0.11)	0.26 (0.1)
Number of clusters ↑	7	27	11	21	18	20

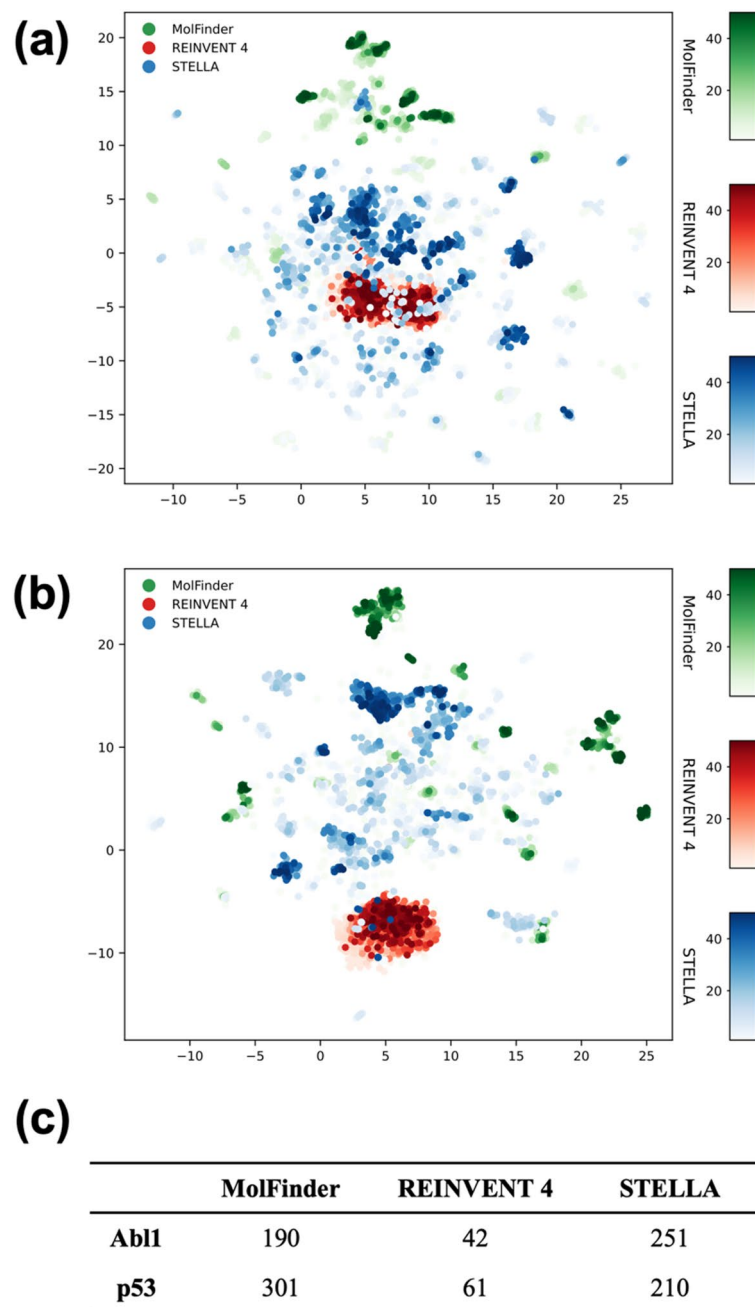
**Table 2.** Comparison of molecular optimization performance among MolFinder, REINVENT 4, and STELLA for the top 300 molecules in the final iteration/epoch. Performance metrics include average objective score, average pairwise Tanimoto similarity, and the number of clusters based on Sphere Exclusion algorithm with a similarity threshold of 0.3.



**Fig. 3.** The progression of average objective score over 50 iterations/epochs for MolFinder, REINVENT 4, and STELLA on (a) Abl1 and (b) p53. Lower objective scores indicate better performance.

proportion of molecules within the hit area, whereas most molecules generated by MolFinder and REINVENT 4 did not achieve favorable values. The representative 12 molecules generated by MolFinder, REINVENT 4, and STELLA for Abl1 and p53 are presented in Supplementary Figures S4–S6.

We compared the ability of MolFinder, REINVENT 4, and STELLA in exploring the chemical space (Fig. 4). The top 300 molecules selected at each iteration/epoch were transformed into ECFP4 fingerprints and projected onto a 2D space using DensMap<sup>40</sup> for dimensionality reduction. To illustrate the trajectory over exploration, later iterations/epochs are represented by darker points (Fig. 4a and b). Furthermore, Sphere Exclusion clustering was applied to all 15,000 molecules generated over 50 iterations/epochs to determine the number of distinct clusters (Fig. 4c). As iterations/epochs progress, STELLA demonstrates a stronger tendency to expand its search space, whereas REINVENT 4 maintains a more focused search within its initial generation space. The number of clusters for REINVENT 4 and STELLA was 42 and 251 for Abl1, and 61 and 210 for p53, respectively. These results indicate that STELLA conducts a broader exploration of chemical space compared to REINVENT 4. While REINVENT 4 has mechanisms to balance exploration and exploitation, its optimization likely remains confined within the generation space, possibly due to constraints from its prior model. A well-known limitation of genetic algorithm-based approaches is their tendency to become trapped in suboptimal regions of the chemical space. However, STELLA effectively addresses this issue by incorporating clustering-based CSA, which preserves structural diversity while dynamically adjusting selection pressure. This approach progressively enhances optimization efficiency, steering molecular design towards high-quality solutions. MolFinder, which employs metaheuristic approaches similar to STELLA, also demonstrates superior capability in exploring broader chemical space than REINVENT 4. However, unlike STELLA, its capability is inconsistent across the two targets, suggesting that the efficiency of chemical space exploration may be highly dependent on specific protein targets or parameter settings.



**Fig. 4.** Visualization of the 300 top molecules generated at each iteration/epoch for (a) Abl1 and (b) p53 case studies. Each point represents a single molecule, with color intensity increasing according to iteration/epoch progression. Molecules generated by MolFinder, REINVENT 4, and STELLA are shown in green, red, and blue gradients, respectively. (c) The number of clusters identified using the Sphere Exclusion algorithm with a similarity threshold of 0.3, based on the 15,000 molecules generated over 50 iterations/epochs.

One notable advantage of deep learning-based methods over metaheuristic approaches is their ability to incorporate an implicit scoring function within the generative model. As generation and scoring are performed simultaneously within a unified framework, the overall computational burden is typically lower. However, in this study, we adopted an objective function-based generation strategy, where the scoring process is delegated to an external scoring function. In this setting, scoring all generated molecules—particularly molecular docking—is significantly more expensive than the molecule generation process itself throughout the de novo design workflow. Although the molecule generation process in REINVENT 4 may be faster in isolation than STELLA, its computational cost is approximately orders of magnitude lower than that of the scoring process. When generation and scoring are executed using parallel computing, as in our implementation, the negligible cost of generation has little impact on overall wall-clock time of iterative generation-scoring loop. In this study, since

the total number of calls to the evaluation functions was fixed across the methods, the overall computational workload was roughly comparable regardless of the molecule generation strategy.

In STELLA, the specification of a scaffold is designed as a user-driven process to reflect practical needs. The platform provides an intuitive web-based graphic interface that enables users to easily define the scaffold of a user-uploaded seed molecule by directly selecting atoms on the interface. While scaffold specification is a manual step in the current implementation, automating this step is feasible and potentially useful for broader exploratory tasks. We plan to integrate automated scaffold selection options (e.g., Murcko scaffolds or frequency-based core selection) in future versions.

## Conclusions

In this study, we presented STELLA, a novel de novo molecular design framework. STELLA employs both an evolutionary algorithm and clustering-based CSA for efficient exploration of chemical space and global optimization of objective function. It also leverages an advanced fragment replacement method and graph transformer-based deep learning models to enhance the efficiency of molecule generation and the accuracy of pharmacological property prediction, respectively, aiming to facilitate the evolutionary process toward more optimized drug-like candidates with unique structures.

Our case study, which focuses on docking score and QED, demonstrates that STELLA generates more hit candidates with a great diversity of scaffolds and achieves superior multi-objective optimization compared to REINVENT 4. In performance evaluations involving the simultaneous optimization of 16 properties to more thoroughly assess multi-parameter optimization capabilities, STELLA consistently outperforms the control methods by achieving better average objective scores and exploring a broader region of chemical space. Overall, these results highlight that STELLA's robust capability for de novo design of molecules with desirable pharmaceutical properties.

Looking ahead, future research could enhance STELLA by integrating deep learning models for molecule generation and for predicting additional pharmacological properties and synthetic feasibility. We believe that STELLA lays a strong foundation for hybrid metaheuristic-deep learning approaches, offering a promising tool for efficiently generating novel drug candidates across vast regions of chemical space.

## Methods

### Multi-parameter optimization using clustering-based CSA

In STELLA, we utilized the principles of conformational space annealing (CSA), a highly effective global optimization algorithm, to enhance search efficiency while maintaining structural diversity during chemical space exploration. CSA is an algorithm integrating the strengths of genetic algorithms and simulated annealing<sup>41</sup> to balance exploration and exploitation during the optimization process and has been successfully applied to various global optimization problems<sup>19,42,43</sup>.

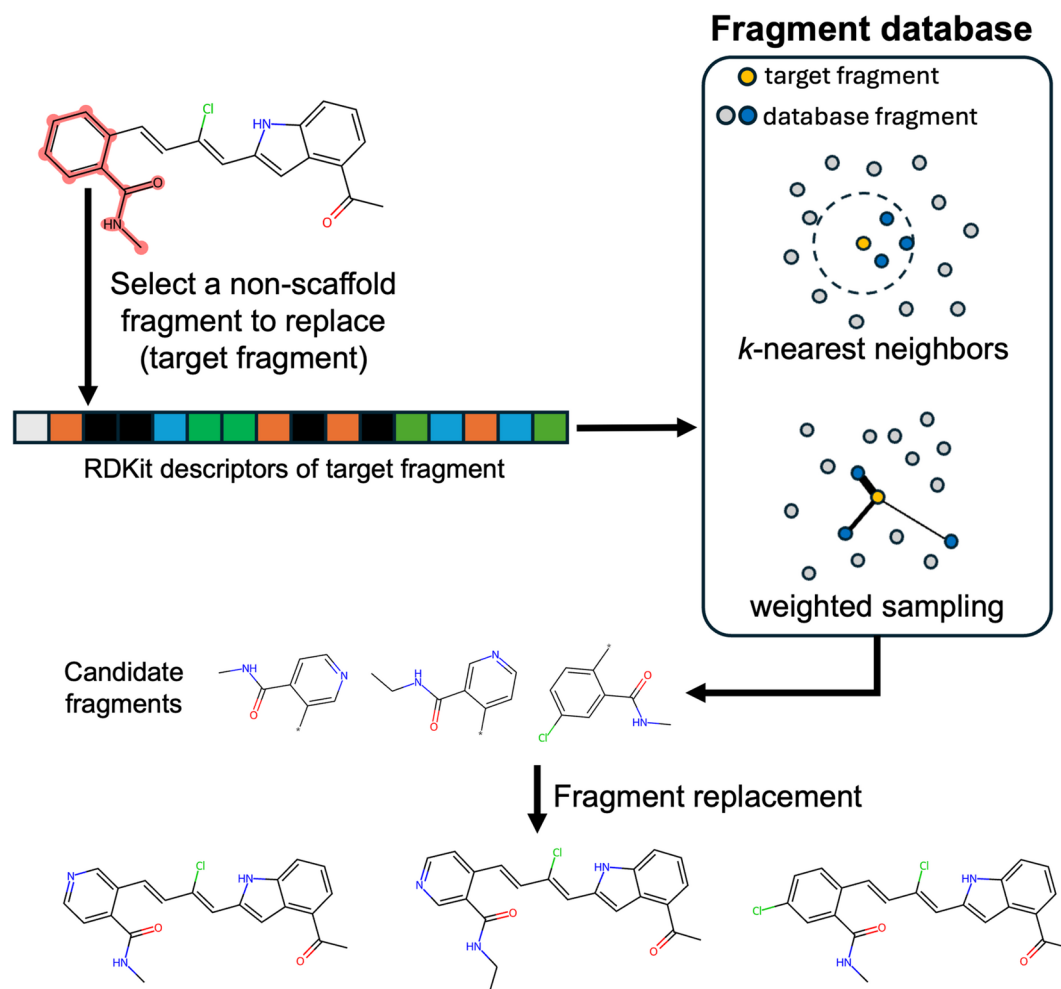
Conventional CSA employs an individual molecule-based approach, where a generated molecule replaces either the closest molecule with worse objective score (if the distance  $\leq$  cutoff) or the molecule with the worst objective score (if the distance  $>$  cutoff). In this process, newly generated molecules with better objective scores relative to the current population are discarded when the closest molecules within the cutoff have better scores. On the other hand, our approach, clustering-based CSA, groups newly generated molecules together with the parent pool into clusters and then selects the target number of top-scoring candidates from these clusters, allowing the molecules with good objective scores to be included in the new parent pool (see the Clustering-based Selection subsection below). This approach enables efficient global optimization of the objective function, while maintaining the overall diversity of the population. As a result, when the algorithm struggles to identify promising candidates in a specific local region, it can efficiently redirect exploration to other areas, accelerating the discovery of diverse, high-quality solutions.

## FRAGRANCE

We introduce FRAGRANCE (Fragment Retrieval and Combination-based Enumerator), a computational method for modifying a given molecule by substituting its fragments with alternatives from a fragment database, ensuring similar chemical properties. The workflow of FRAGRANCE is illustrated in Fig. 5. To construct the fragment database, we collected compounds from ChEMBL24.1 (<https://ftp.ebi.ac.uk/pub/databases/chembl>) and SureChEMBL\_20210401 (about 23 million unique molecules) and fragmented each molecule as follows: (1) A molecule is separated into rings and linkers. Fused ring systems are considered single components. Side chains remain attached to their respective rings. Linkers are defined as fragments connecting one ring system to another. (2) BRICS<sup>44</sup> bonds are enumerated within each ring and linker, and a set of breakable bonds that produce fragments with reasonable size and synthetic accessibility are identified using beam search guided by a penalty metric. An empirical size-based penalty is applied to fragmentation, based on the largest fragment produced when breaking bonds. Specifically, the penalty is given as  $\max(\text{num. heavy atoms in the largest fragment} - 15, 0)$  to discourage the generation of excessively large fragments, thereby resulting in more balanced fragment sizes. Another constraint is that all fragments produced by breakable bonds should have at least three heavy atoms. In cases where the size-based penalties are equal, the maximum SAScore is evaluated for each set of fragments to determine the optimum fragmentation.

Fragments were removed from the database if they were redundant, failed sanitization using RDKit (version 2023.9.5), had a molecular weight of  $> 600$  Da, contained PAINS<sup>45</sup> substructures, or failed to pass the Eli Lilly medicinal chemistry rules<sup>46</sup>. Among the Eli Lilly rules, those rejecting molecules with fewer than seven atoms or molecules lacking specific atoms were not applied, as such rules are not suitable for evaluating fragments. While the fragment library included large fragments to ensure broad coverage, only smaller fragments (typically  $< 300$





**Fig. 5.** Schematic illustration of FRAGRANCE workflow.

Da) were effectively utilized during molecule generation due to the overall molecular weight constraint. The resulting database consists of about 2.6 million unique fragments. For each fragment, we computed RDKit-derived physicochemical and topological properties, resulting in a 115-dimensional feature vector for each fragment (see Supplementary Table S2 for the full list of properties). To address inconsistencies in the scales of the properties and mitigate the impact of outliers, we scaled them using RobustScaler in Scikit-learn (version 1.6.0).

Given a molecule with a user-defined scaffold, FRAGRANCE first identifies the fragments that do not intersect with the provided scaffold. For these non-scaffold fragments, the scaled feature vectors are calculated. Candidate fragments for replacement are obtained from the database using the following two methods: (1) Approximate *k*-nearest neighbors are retrieved based on the feature vector to identify database fragments structurally similar to the target fragment. We used a HNSW (Hierarchical Navigable Small Worlds)<sup>47</sup> index for efficient nearest neighbor querying (default: 10). (2) Fragments are selected using weighted sampling (default: 10), where the weights are determined by  $e^{-\|x-x_i\|^2/T}$ , where  $x$  and  $x_i$  are feature vectors for a target fragment and a database fragment, respectively.  $T$  is a temperature parameter that controls the preference for fragments similar to the target (default: 1.0). This method occasionally introduces fragments distant from the target fragment, enhancing the structural diversity. Among the candidate database fragments obtained from both methods, those with the same number of attachment points (i.e., broken inter-fragment bonds during fragmentation) are selected to ensure that the replacement yields a chemically valid molecule.

When multiple attachment points exist in the target fragment, the reconstruction process can become ambiguous due to potential permutations in the mapping between them. To minimize structural deviations, we reconnect the bonds in a way that preserves the original topological distances between the attachment points as closely as possible. Let  $\{i_1, i_2, \dots, i_n\}$  and  $\{j_1, j_2, \dots, j_n\}$  denote the indices of the attachment points in the original fragment and the replacement fragment, respectively. We find the permutation  $\pi$  of the atom indices that minimizes  $Penalty(\pi)$ .

$$Penalty(\pi) = \sum_{a=1}^n \sum_{b=a+1}^n \left| \frac{1}{d_S(i_a, i_b)} - \frac{1}{d_T(j_{\pi(a)}, j_{\pi(b)})} \right|$$

where  $d_S$  and  $d_T$  denote the topological distances between atoms in the original and the replacement fragment, respectively.

### Initialization

This step is an augmentation process for a single seed molecule to generate initial population for the genetic algorithm. The scaffold atoms, defined as atoms that should not be modified, can be specified by the user. The FRAGRANCE algorithm modifies the non-scaffold fragments of the seed molecule to generate diverse molecule structures. The resulting augmented molecular population is referred to as the “seed pool.” By default, FRAGRANCE generates 10,000 molecules. This pool can be further expanded by incorporating a custom molecule pool provided by the user.

Each molecule in the seed pool is converted into ECFP4 fingerprints, and pairwise Tanimoto similarity values are calculated to construct a distance map. Agglomerative Hierarchical Clustering (AHC) is then applied to this distance map to group molecules into clusters. Clustering is performed to generate the number of clusters that matches the target population size ( $N_t$ , default: 300). The centroid molecules from each cluster are selected as the initial pool ( $P_0$ ). Remaining molecules are assigned to an auxiliary pool ( $P_{aux}$ ).

### Molecule generation

In STELLA, new molecule structures are augmented by generating an offspring pool from a parent pool using the genetic algorithm. In the first iteration after the *Initialization*, the initial pool serves as the parent pool. STELLA employs three distinct methods, FRAGRANCE mutation, maximum common substructure (MCS)-based crossover, and trimming, to generate offspring molecules.

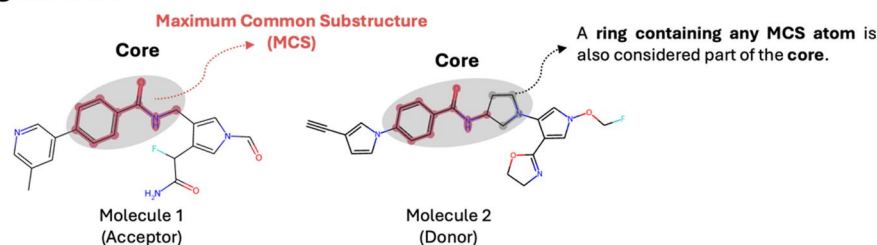
FRAGRANCE mutation randomly selects a molecule from the parent pool and modifies one non-scaffold fragment. This mutation is repeated until the target number of unique molecules is reached (default: 3,000 molecules).

In the MCS-based crossover method, two molecules are randomly selected from the parent/auxiliary pool, where one is designated as the acceptor molecule and the other as the donor molecule. The donor molecules are selected from either the parent or auxiliary pool, depending on a user-defined auxiliary ratio, whereas the acceptor molecules are always chosen only from the parent pool. For example, if the total number of molecules for MCS crossover and the auxiliary ratio are set to 1,000 and 0.3, respectively, 300 donor molecules will be sampled from the auxiliary pool, while the remaining 500 acceptor molecules and 200 donor molecules will both be sampled from the parent pool. The MCS between an acceptor and a donor molecule is identified, and the indices of MCS atoms in both the acceptor and donor molecules are set to be the same based on those of the acceptor molecule. The identified MCS is designated as core fragment, and fragments with MCS atoms are unified as the core fragment. The molecules are fragmented at cleavable bonds defined by RDKit's BRICS rules, while preserving the core fragment intact. The molecules are represented as fragment-level graphs, with each fragment treated as a node and bonds cleaved during fragmentation as edges (Fig. 6). Each node in the fragment-level graphs is assigned two key properties: Nearest MCS Atom Index and Distance from Core. The Nearest MCS Atom Index is the index of the MCS atom with the shortest path length from any fragment atom. The Distance from Core represents the relative distance of each fragment from the core fragment (The core fragment is assigned 0, and fragments directly connected to the core fragment are assigned 1 with the value increasing based on their distance from the core). Fragments in the donor molecule are sorted in ascending order in terms of their Distance from Core values, prioritizing fragments closest to the core. Crossover occurs when the acceptor molecule has a fragment with the same node property values as ones of a fragment in the donor molecule, replacing the fragment in the acceptor molecule with its counterpart in the donor molecule. All fragments contain dummy atoms that represent the originally connected atoms. Bonds between fragments are formed randomly among the atoms associated with these dummy atoms, diversifying the products of the recombination while preserving their original connectivity. An addition is attempted when the acceptor molecule has a fragment with the same Nearest MCS Atom Index but a Distance from Core that is one less than that of a donor fragment. In the addition, the donor fragment is connected to the acceptor atom replacing one of the hydrogen atoms.

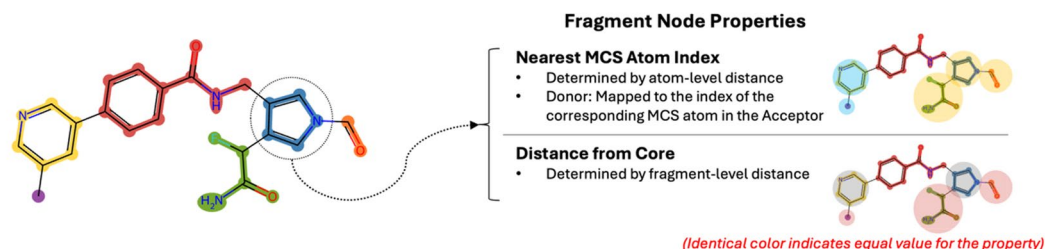
Trimming is designed to randomly remove a single atom and reconnect its neighboring atoms, provided that the removed atom has more than one neighbor. It removes an atom when the atom is not part of a ring, not part of a conjugated system, and not included in the atom indices matched to the scaffold structure. If trimming results in an atom violation of valence, it is excluded from the trimming candidates.

Evolutionary operations for molecule generation can be implemented using either atom-based or fragment-based approaches<sup>48</sup>. STELLA employs fragment-based approaches to mutation and crossover, while MolFinder utilizes them only for crossover<sup>18</sup>. In MolFinder, fragments are defined based on ring structures extracted from SMILES strings. This contrasts with our approach, which leverages MCS matching in combination with BRICS-based fragmentation to define chemically meaningful fragments, enabling efficient and synthesizable molecule generation. EvoMol<sup>17</sup> randomly removes an atom and reconnects its bonded neighbors (i.e., the Cut atom action), a strategy conceptually similar to the Trimming operator in STELLA. While EvoMol also supports the freezing of scaffold atoms, its removal operation is primarily applied to an atom bonded to two other atoms that are not connected to each other. This differs from STELLA, where removal targets atoms that are not part of a ring or conjugated system, irrespective of their number of bonded neighbors, thereby enabling both broader and chemically more stable molecule modifications.

## Assigning the Core



## Fragment-Level Graph



**Fig. 6.** Schematic illustration of the core assignment process in the MCS-based crossover and the conversion of a molecule to a fragment-level graph. At the atom level, atoms identified as part of the MCS in two sampled parent molecules are designated as core atoms and are protected from modifications during the crossover operation. In the fragment-level representation, each node (fragment) is annotated with two properties—Nearest MCS Atom Index and Distance from Core—which guide the crossover operation.

## Scoring

The offspring molecules are evaluated using an objective function for optimization. STELLA's objective function encompasses various pharmacological properties, including molecular weight, AlogP, polar surface area, the number of rotatable bonds, hydrogen bond donors, and acceptors, SAScore, maximum fused ring size, maximum ring size, the number of chiral centers, ring size penalty, undesirable SMARTS pattern-based penalty, and scores from docking tools such as GOLD. Moreover, it incorporates scores from QIP-ADMET, our proprietary ADMET property prediction model, which leverages a graph transformer-based deep learning approach pretrained on quantum data and fine-tuned using transfer learning<sup>37</sup>. STELLA also provides diverse filtering options, including substructure filters, interaction fingerprint filters, PAINS filters (OpenEye OMEGA version 5.0.1.3, <https://www.eyesopen.com/filter>), Eli Lilly's medicinal chemistry rules-based filters (version 1.0, <https://github.com/IanAWatson/Lilly-Medchem-Rules/tree/v1.0>), and 3D shape similarity filter using ROCS (version 3.6.2.3, <https://www.eyesopen.com/rocs>). For computational efficiency, all scoring methods are executed in a predefined order, skipping computationally intensive methods if a molecule is filtered out by user-defined thresholds for less resource-demanding methods.

The calculated scores for all properties are integrated into a single objective function. Each property score is transformed using its own sigmoid function, mapping the value to a range between 0 and 1 based on the upper ( $M$ ) and lower ( $m$ ) thresholds of the property. For properties where the target is the upper threshold, the sigmoid function is defined as:

$$S(x) = \frac{1}{1 + e^{-\alpha(x-c)}}$$

where  $c$  represents the midpoint between the thresholds ( $c = (m + M)/2$ ), and  $\alpha$  is calibrated such that the upper threshold corresponds to a value of 0.05, and the lower threshold to 0.95:

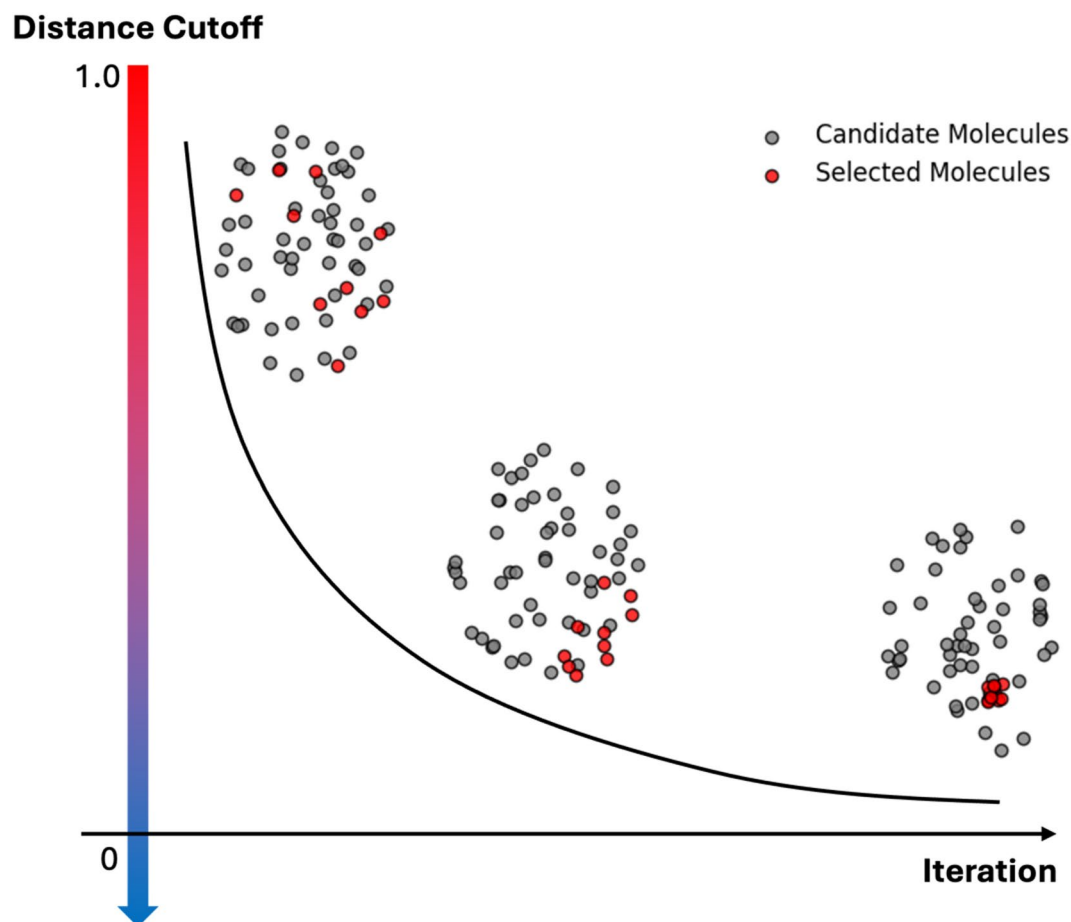
$$\alpha = \frac{\log(0.05/0.95)}{c - M}$$

For properties targeting the lower threshold, the sigmoid function is adjusted as follows:

$$S(x) = 1 - \frac{1}{1 + e^{-\alpha(x-c)}}$$

The transformed scores can be adjusted using user-specified weights ( $w_i$ ), with default values of 1 for all properties), and the final objective score ( $F$ ) is the sum of the weighted scores:

$$F = \sum_i w_i \cdot S_i$$



**Fig. 7.** Schematic visualization of molecule selection over iterations with progressively decreasing distance cutoff. Gray dots represent candidate molecules, while red dots indicate selected molecules. The curved boundary illustrates the reduction of the distance cutoff as iterations progress, shifting the selection strategy from promoting diversity to focusing on local exploration.

where  $S_i$  represents the transformed score for property  $i$ .

#### Clustering-based selection

Clustering is performed on the scored offspring pool together with the parent pool using AHC, as in the initialization step. In the first iteration,  $d_{cutoff}$  is optimized to align the number of clusters with the target number of molecules to be selected. This optimization minimizes the difference between the desired number of clusters ( $N_{desired}$ , equal to the target number of molecules) and the actual number of clusters ( $N_{clusters}$ ):

$$d_{cutoff} = \operatorname{argmin} |N_{clusters}(d) - N_{desired}|$$

In subsequent iterations,  $d_{cutoff}$  is scaled by a predefined decrease ratio  $k$  (default: 0.98):

$$d_{cutoff}^{(i+1)} = d_{cutoff}^{(i)} \times k$$

The minimum  $d_{cutoff}$ , determined by its lower bound, represents the smallest value achievable through iterative scaling. This lower bound is typically set between 0.2 and 0.5 depending on the configuration. Once this minimum is reached,  $d_{cutoff}$  remains constant until the end of the run. The gradual decrease of  $d_{cutoff}$  during the iterations shifts the optimization focus from global exploration to local exploitation, refining the search within more specific regions of chemical space (Fig. 7).

A molecule with the minimum objective score is selected for each cluster. If the target number of molecules ( $N_t$ ) is not reached, the next lowest-scoring molecule in each cluster is iteratively selected until the target count is achieved. If the number of clusters exceeds  $N_t$ , the top molecules in each cluster are collected, sorted by objective score, and the top  $N_t$  among them are selected.



## Termination

After the selection step, the algorithm checks whether the termination conditions are met. The primary condition is reaching the maximum allowed iterations (default: 50). In addition, two optional conditions can also terminate the selection process. The first option, referred to as the evolution efficiency of the pool, checks whether the current pool includes a sufficient proportion of newly added molecules compared to the previous iteration (default: 0.05). To account for cases where extensive filtering makes it difficult to find new molecules, this condition is only applied after 10 iterations by default, but this value can be adjusted in the configuration. The second condition, early stopping, is triggered when consecutive failures to improve the objective score exceed a predefined number (default: 10). The improvement (default threshold: 0.01) is evaluated based on either the average or median objective score. The process of molecule *generation*, *scoring*, and *selection* is then repeated in a loop until one of the termination criteria is met.

## Data availability

All data generated or analyzed during this study are available as supplementary data (Supplementary\_Information\_Raw\_Data.zip). STELLA presented in this study is a commercial tool but is available upon request by completing a form on <https://www.standigm.com/ai-saas/stella>.

Received: 17 March 2025; Accepted: 18 July 2025

Published online: 01 August 2025

## References

- Aherne, G. W., McDonald, E. & Workman, P. Finding the needle in the haystack: why high-throughput screening is good for your health. *Breast Cancer Res.* **4**, 148–154 (2002).
- Bohacek, R. S., McMartin, C. & Guida, W. C. The Art and practice of structure-based drug design: a molecular modeling perspective. *Med. Res. Rev.* **16**, 3–50 (1996).
- Kumar, A. & Zhang, K. Y. Hierarchical virtual screening approaches in small molecule drug discovery. *Methods* **71**, 26–37 (2015).
- Langevin, M., Bianciotto, M. & Vuilleumier, R. Balancing exploration and exploitation in de Novo drug design. *Digit. Discovery*. **3**, 2572–2588 (2024).
- Kitchen, D. B., Decornez, H., Furr, J. R. & Bajorath, J. Docking and scoring in virtual screening for drug discovery: methods and applications. *Nat. Rev. Drug Discov.* **3** (2004).
- Lyu, J. et al. Ultra-large library Docking for discovering new chemotypes. *Nature* **566**, 224–229 (2019).
- Rogers, D. M. et al. SARS-CoV2 billion-compound Docking. *Sci. Data*. **10**, 173. <https://doi.org/10.1038/s41597-023-01984-9> (2023).
- Tingle, B. I. et al. ZINC-22—a free multi-billion-scale database of tangible compounds for ligand discovery. *J. Chem. Inf. Model.* **63**, 1166–1176 (2023).
- Mouchlis, V. D. et al. Advances in de Novo drug design: from conventional to machine learning methods. *Int. J. Mol. Sci.* **22** <https://doi.org/10.3390/ijms22041676> (2021).
- Nicolaou, C. A. & Brown, N. Multi-objective optimization methods in drug design. *Drug Discov. Today Technol.* **10**, e427–435. <https://doi.org/10.1016/j.ddtec.2013.02.001> (2013).
- Tang, X. et al. A survey of generative AI for de Novo drug design: new frontiers in molecule and protein generation. *Brief. Bioinform.* **25** <https://doi.org/10.1093/bib/bbae338> (2024).
- Xie, W., Wang, F., Li, Y., Lai, L. & Pei, J. Advances and challenges in de Novo drug design using three-dimensional deep generative models. *J. Chem. Inf. Model.* **62**, 2269–2279 (2022).
- Nicolaou, C. A., Apostolakis, J. & Pattichis, C. S. De Novo drug design using multiobjective evolutionary graphs. *J. Chem. Inf. Model.* **49**, 295–307 (2009).
- Meyers, J., Fabian, B. & Brown, N. De Novo molecular design and generative models. *Drug Discov. Today*. **26**, 2707–2715 (2021).
- Dey, F. & Caffisch, A. Fragment-based de Novo ligand design by multiobjective evolutionary optimization. *J. Chem. Inf. Model.* **48**, 679–690 (2008).
- Spiegel, J. O. & Durrant, J. D. AutoGrow4: an open-source genetic algorithm for de Novo drug design and lead optimization. *J. Cheminform.* **12**, 25. <https://doi.org/10.1186/s13321-020-00429-4> (2020).
- Leguy, J., Cauchy, T., Glavatskikh, M. & Duval, B. Da mota, B. EvoMol: a flexible and interpretable evolutionary algorithm for unbiased de Novo molecular generation. *J. Cheminform.* **12**, 55. <https://doi.org/10.1186/s13321-020-00458-z> (2020).
- Kwon, Y. & Lee, J. MolFinder: an evolutionary algorithm for the global optimization of molecular properties and the extensive exploration of chemical space using SMILES. *J. Cheminform.* **13** <https://doi.org/10.1186/s13321-021-00501-7> (2021).
- Lee, J., Scherage, H. A. & Rackovsky, S. New optimization method for conformational energy calculations on polypeptides: conformational space annealing. *J. Comput. Chem.* **18**, 1222–1232 (1997).
- Lamanna, G. et al. GENERA: a combined genetic/deep-learning algorithm for multiobjective target-oriented de Novo design. *J. Chem. Inf. Model.* **63**, 5107–5119 (2023).
- Winter, R. et al. Efficient multi-objective molecular optimization in a continuous latent space. *Chem. Sci.* **10**, 8016–8024 (2019).
- Wang, M. et al. Deep learning approaches for de Novo drug design: an overview. *Curr. Opin. Struct. Biol.* **72**, 135–144 (2022).
- Olivecrona, M., Blaschke, T., Engkvist, O. & Chen, H. Molecular de-novo design through deep reinforcement learning. *J. Cheminform.* **9** <https://doi.org/10.1186/s13321-017-0235-x> (2017).
- Blaschke, T. et al. REINVENT 2.0: an AI tool for de Novo drug design. *J. Chem. Inf. Model.* **60** (2020).
- Loeffler, H. H. et al. Reinvent 4: modern AI-driven generative molecule design. *J. Cheminform.* **16** <https://doi.org/10.1186/s13321-024-00812-5> (2024).
- Ragoza, M., Masuda, T. & Koes, D. R. Generating 3D molecules conditional on receptor binding sites with deep generative models. *Chem. Sci.* **13**, 2701–2713 (2022).
- Fang, Y., Pan, X. & Shen, H. B. De Novo drug design by iterative multiobjective deep reinforcement learning with graph-based molecular quality assessment. *Bioinformatics* **39** <https://doi.org/10.1093/bioinformatics/btad157> (2023).
- Ivanenkov, Y. A. et al. Chemistry42: an AI-driven platform for molecular design and optimization. *J. Chem. Inf. Model.* **63**, 695–701 (2023).
- Elton, D. C., Boukouvalas, Z., Fugea, M. D. & Chung, P. W. Deep learning for molecular design—a review of the state of the Art. *Mol. Syst. Des. Eng.* **4**, 828–849 (2019).
- Angiolini, M. et al. Structure-based optimization of potent PDK1 inhibitors. *Bioorg. Med. Chem. Lett.* **20**, 4095–4099 (2010).
- Friesner, R. A. et al. Extra precision glide: Docking and scoring incorporating a model of hydrophobic enclosure for protein-ligand complexes. *J. Med. Chem.* **49**, 6177–6196 (2006).

32. Guo, J. et al. DockStream: a Docking wrapper to enhance de Novo molecular design. *J. Cheminform.* **13**, 89. <https://doi.org/10.1186/s13321-021-00563-7> (2021).
33. Jones, G., Willett, P., Glen, R. C., Leach, A. R. & Taylor, R. Development and validation of a genetic algorithm for flexible Docking. *J. Mol. Biol.* **267**, 727–748 (1997).
34. Su, M. et al. Comparative assessment of scoring functions: the CASF-2016 update. *J. Chem. Inf. Model.* **59**, 895–913 (2019).
35. Hawkins, P. C., Skillman, A. G. & Nicholls, A. Comparison of shape-matching and Docking as virtual screening tools. *J. Med. Chem.* **50**, 74–82 (2007).
36. Ertl, P. & Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminform.* **1**(8). <https://doi.org/10.1186/1758-2946-1-8> (2009).
37. Kim, J., Chang, W., Ji, H. & Joung, I. Quantum-informed molecular representation learning enhancing ADMET property prediction. *J. Chem. Inf. Model.* **64**, 5028–5040 (2024).
38. Fialkova, V. et al. LibINVENT: reaction-based generative scaffold decoration for in Silico library design. *J. Chem. Inf. Model.* **62**, 2046–2063 (2022).
39. Dong, J. et al. ADMETlab: a platform for systematic ADMET evaluation based on a comprehensively collected ADMET database. *J. Cheminform.* **10** <https://doi.org/10.1186/s13321-018-0283-x> (2018).
40. Narayan, A., Berger, B. & Cho, H. Assessing single-cell transcriptomic variability through density-preserving data visualization. *Nat. Biotechnol.* **39**, 765–774 (2021).
41. Kirkpatrick, S., Gelatt, C. D. Jr. & Vecchi, M. P. Optimization by simulated annealing. *Science* **220**, 671–680 (1983).
42. Lee, J. et al. De Novo protein structure prediction by dynamic fragment assembly and conformational space annealing. *Proteins* **79**, 2403–2417 (2011).
43. Joung, I. S., Kim, J. Y., Gross, S. P., Joo, K. & Lee, J. Conformational space annealing explained: a general optimization algorithm, with diverse applications. *Comput. Phys. Commun.* **223**, 28–33 (2018).
44. Degen, J., Wegscheid-Gerlach, C., Zaliani, A. & Rarey, M. On the Art of compiling and using ‘drug-like’ chemical fragment spaces. *ChemMedChem* **3**, 1503–1507 (2008).
45. Baell, J. B. & Holloway, G. A. New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. *J. Med. Chem.* **53**, 2719–2740 (2010).
46. Bruns, R. F. & Watson, I. A. Rules for identifying potentially reactive or promiscuous compounds. *J. Med. Chem.* **55**, 9763–9772 (2012).
47. Malkov, Y. A. & Yashunin, D. A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **42**, 824–836 (2020).
48. Devi, R. V., Sathya, S. S. & Coumar, M. S. Evolutionary algorithms for de Novo drug design – A survey. *Appl. Soft Comput.* **27**, 543–552 (2015).

## Acknowledgements

This research was supported by Standigm Inc. We would like to thank Standigm Inc. for their financial support.

## Author contributions

Hui Sun Lee conceived and designed the experiments. Hokyun Jeon, Jin Gyu Lee, Wonseok Shin, and Hyunjun Ji developed the code required for the experiments. Hokyun Jeon and Jin Gyu Lee conducted the experiments. Insuk Joung supervised the project. Hui Sun Lee, Hokyun Jeon, Jin Gyu Lee, Wonseok Shin, and Insuk Joung contributed to the manuscript preparation. All authors reviewed and approved the final manuscript.

## Declarations

## Competing interests

The authors declare no competing interests.  
Tables.

## Additional information

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1038/s41598-025-12685-1>.

**Correspondence** and requests for materials should be addressed to I.J. or H.S.L.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025