nature machine intelligence

Article

https://doi.org/10.1038/s42256-023-00759-6

Multi-modal molecule structure–text model for text-based retrieval and editing

Received: 21 December 2022

Accepted: 16 October 2023

Published online: 18 December 2023

Check for updates

Shengchao Liu[1,2], Weili Nie[3], Chengpeng Wang[4], Jiarui Lu[1,2], Zhuoran Qiao[5], Ling Liu[6], Jian Tang[1,7,9], Chaowei Xiao[3,8,9] & Animashree Anandkumar[3,5,9]

There is increasing adoption of artificial intelligence in drug discovery. However, existing studies use machine learning to mainly utilize the chemical structures of molecules but ignore the vast textual knowledge available in chemistry. Incorporating textual knowledge enables us to realize new drug design objectives, adapt to text-based instructions and predict complex biological activities. Here we present a multi-modal molecule structure–text model, MoleculeSTM, by jointly learning molecules' chemical structures and textual descriptions via a contrastive learning strategy. To train MoleculeSTM, we construct a large multi-modal dataset, namely, PubChemSTM, with over 280,000 chemical structure–text pairs. To demonstrate the effectiveness and utility of MoleculeSTM, we design two challenging zero-shot tasks based on text instructions, including structure–text retrieval and molecule editing. MoleculeSTM has two main properties: open vocabulary and compositionality via natural language. In experiments, MoleculeSTM obtains the state-of-the-art generalization ability to novel biochemical concepts across various benchmarks.

Recent progress in artificial intelligence (AI) promises to be transformative for drug discovery[1]. AI methods have been used to augment and accelerate current computational pipelines[2–4], including but not limited to virtual screening[5,6], metabolic property prediction[7–9], and targeted chemical structure generation and editing[10–13].

Existing machine learning (ML) methods mainly focus on modelling the chemical structure of molecules through one-dimensional descriptions[14], two-dimensional molecular graphs[7,8,15] or three-dimensional geometric structures[16–18]. They also use supervised signals, for example, toxicity labels, quantum-mechanical properties and binding-affinity measurements. However, such a supervised setting requires expensive annotations on pre-determined label categories, impeding the application to unseen categories and tasks[19]. To overcome this issue, unsupervised pretraining on large-scale databases[20] has been proposed, with the main advantage being the ability to learn chemical structures without supervised annotation by reconstructing the masked topological[21] or geometric[22] substructures. Compared with the supervised setting, although such pretrained models[21,22] have proven to be more effective in generalizing to various downstream tasks by fine-tuning on a few labelled examples, it is still an open challenge to generalize unseen categories and tasks without such labelled examples or fine-tuning (that is, the so-called zero-shot setting[23] in ML). In addition, existing molecule pretraining methods mostly incorporate only chemical structures, leaving the multi-modal representation less explored.

We have a vast amount of textual data that is human understandable and easily accessible. This is now being harnessed in large-scale multi-modal models for images and videos[24–27]. A natural language interface is an intuitive way to enable open vocabulary and description of tasks. Pretrained multi-modal models can generalize well to new categories and tasks, even in the zero-shot setting[24–27]. They also enable agents to interactively learn to solve new tasks and explore new

[1]Mila-Québec Artificial Intelligence Institute, Montreal, Quebec, Canada. [2]Université de Montréal, Montreal, Quebec, Canada. [3]NVIDIA Research, Santa Clara, CA, USA. [4]University of Illinois Urbana-Champaign, Champaign, IL, USA. [5]California Institute of Technology, Pasadena, CA, USA. [6]Princeton University, Princeton, NJ, USA. [7]HEC Montréal, Montreal, Quebec, Canada. [8]Arizona State University, Tempe, AZ, USA. [9]These authors jointly supervised this work: Jian Tang, Chaowei Xiao, Animashree Anandkumar. e-mail: anima@caltech.edu

environments[28,29]. We believe similar capabilities can also be obtained in molecular models by incorporating the vast textual knowledge available in the literature.

Previous work[30] has attempted to leverage the textual knowledge to learn the molecule representation. However, it supports only modelling with the one-dimensional description (the simplified molecular-input line-entry system or SMILES) and learns the chemical structures and textual descriptions on a small-scale dataset (10,000 structure–text pairs). Furthermore, it unifies two modalities into a single language modelling framework and requires aligned data, that is, chemical structure and text for each sample, for training. As a result, it cannot adopt existing powerful pretrained models and the availability of aligned data is extremely limited.

Our approach

We design a multi-modal foundation model for molecular understanding that incorporates both molecular structural information and textual knowledge. We demonstrate zero-shot generalization to new drug design objectives using text-based instructions and to the prediction of new complex biological activities without the need for labelled examples or fine-tuning.

We propose MoleculeSTM, consisting of two branches, the chemical structure branch and the textual description branch, to handle the molecules' internal structures and external domain knowledge, respectively. Such a disentangled design enables MoleculeSTM to be integrated with the powerful existing models trained on each modality separately, that is, molecular structural models[11,31] and scientific language models[32]. Given these pretrained models, MoleculeSTM bridges the two branches via a contrastive learning paradigm[31,33].

To align such two branches with MoleculeSTM, we construct a structure–text dataset called PubChemSTM from PubChem[34], which is a large multi-modal dataset (28× larger than the existing dataset[30]). In PubChemSTM, each chemical structure is paired with a textual description, illustrating the chemical and physical properties or high-level bioactivities accordingly. As MoleculeSTM is trained on a large-scale structure–text pair dataset and such textual data contain open-ended chemical information, it can be generalized to diverse downstream tasks in a zero-shot manner.

To demonstrate the advantages of introducing the language modality, we design two challenging downstream tasks, the structure–text retrieval task and text-based molecule editing task, and we apply the pretrained MoleculeSTM on them in a zero-shot manner. By studying these tasks, we summarize two main attributes of MoleculeSTM: the open vocabulary and compositionality. (1) Open vocabulary means our proposed MoleculeSTM is not limited to a fixed set of pre-defined molecule-related textual descriptions and can support exploring a wide range of biochemical concepts with the unbound vocabulary depicted by the natural language. In the drug discovery pipeline, such an attribute can be used for the text-based molecule editing in the lead optimization task and the novel disease–drug relation extraction in the drug re-purposing task. (2) Compositionality implies that we can express a complex concept by decomposing it into several simple concepts. This can be applied for the text-based multi-objective lead optimization task[35] where the goal is to generate molecules satisfying multiple properties simultaneously.

Empirically, MoleculeSTM reaches the best performance on 6 zero-shot retrieval tasks (up to 50% higher accuracy) and 20 zero-shot text-based editing tasks (up to 40% higher hit ratio) compared with the state-of-the-art methods. Furthermore, for molecular editing tasks, visual inspections reveal that MoleculeSTM can successfully detect critical structures implied in text descriptions. In addition, we also explore whether MoleculeSTM can improve the performance on the standard molecular property prediction benchmark[9] via fine-tuning. Our results show that MoleculeSTM can achieve the best overall performance among nine baselines on eight property prediction tasks.

Results

Overview and preliminaries

In this section, we first provide an overview of MoleculeSTM. Then, we introduce how to pretrain MoleculeSTM and apply the pretrained MoleculeSTM to three types of downstream task (Fig. 1).

Overview. MoleculeSTM consists of two branches: the chemical structure branch and the textual description branch ($\mathbf{x}_c$ and $\mathbf{x}_t$). The chemical structure branch illustrates the arrangement of atoms in a molecule. We consider two types of encoder f_c: transformer[36] on the SMILES string and graph neural networks (GNNs)[7,8,15] on the two-dimensional molecular graph. The textual description branch provides a high-level description of the molecule's functionality, and we use the language model from a recent work[37] as the encoder f_t.

Pretraining. Within this design, MoleculeSTM aims to map the representations extracted from two branches to a joint space using two projectors (p_c and p_t) via contrastive learning[31,33]. The essential idea of contrastive learning is to reduce the representation distance between the chemical structure and textual description pairs of the same molecule and increase the representation distance between the pairs from different molecules. Specifically, we initialize these two branch encoders with the pretrained single-modal checkpoints[11,31,32] and then perform an end-to-end contrastive pretraining on collected dataset PubChemSTM. Specifically for PubChemSTM, it is constructed from PubChem[34]. We extract molecules with the textual description fields, leading to 281,000 chemical structure and text pairs. More details can be found in Supplementary Section A.1.

Two principles for downstream task design

We want to emphasize that for these downstream tasks, the language model in the pretrained MoleculeSTM reveals certain appealing attributes for molecule modelling and drug discovery. We summarize the two key points below.

Open vocabulary. Language is by nature open vocabulary and free form[38]. The large language model has proven its generalization ability in various art-related applications[24–26], and we find that it can also provide promising and insightful observations for drug discovery tasks. In this vein, our method is not limited to a fixed set of pre-defined molecule-related annotations but can support the exploration of novel biochemical concepts with unbound vocabulary. One example is the drug re-purposing. Suppose we have a textual description for a new disease or protein target functionality. In that case, we can obtain its similarity with all the existing drugs using MoleculeSTM and retrieve the drugs with the highest rankings, which can be adopted for the later stages, such as clinical trials. Another example is text-based lead optimization. We use natural language to depict an entirely new property, which can be reflected in the generated molecules after the optimization.

Compositionality. Another attribute is compositionality. In natural language, a complex concept can be expressed by decomposing it into simple concepts. This is crucial for certain domain-specific tasks, for example, multi-objective lead optimization[35] where we need to generate molecules with multiple desired properties simultaneously. Existing solutions are either (1) learning one classifier for each desired property and doing filtering on a large candidate pool[10] or (2) optimizing a retrieval database to modify molecules to achieve the multi-objective goal[12]. The main limitation is that the success ratio highly depends on the availability of the labelled data for training the classifier or the retrieval database. With the language model in MoleculeSTM, we provide an alternative solution. We first craft a natural text, called the text prompt, as the task description. The text prompt can be multi-objective and consists of the description for each property (for example, 'molecule is soluble in water and has high permeability').

https://doi.org/10.1038/s42256-023-00759-6

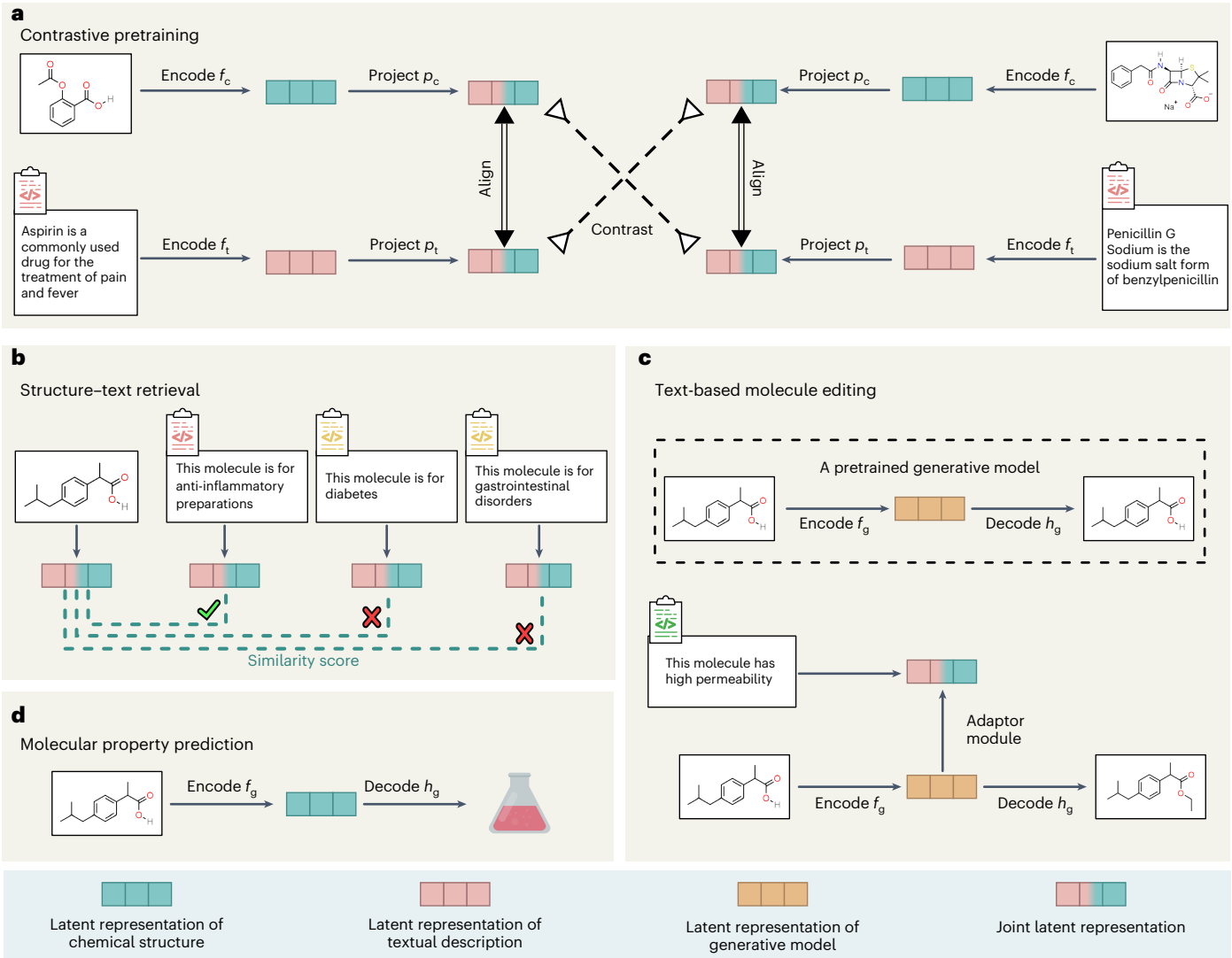

Fig. 1 | Pipeline of pretraining and downstream tasks. a, MoleculeSTM pretraining with two branches, the chemical structure (green) and textual description (pink). **b**, Structure–text retrieval downstream task. **c**, Text-based molecule editing downstream task. **d**, Molecular property prediction downstream task.

With the pretrained joint space between chemical structures and textual descriptions, MoleculeSTM can transform the molecule property compositionality problem into the language compositionality problem, which is more tractable using the language model.

Zero-shot structure–text retrieval

Experiments. For the zero-shot retrieval, we construct three datasets from DrugBank[39]. DrugBank is by far the most comprehensive database for drug-like molecules. Here we extract three fields in DrugBank: the description field, the pharmacodynamics field and the anatomical therapeutic chemical (ATC) field. These fields illustrate the chemical properties and drug effects on the target organism. Then the retrieval task can be viewed as a T-choose-one multiple-choice problem, where T is the number of choices. Specifically, we have two settings: (1) given the chemical structure to retrieve the textual description and (2) given the textual description to retrieve the chemical structure. The retrieval accuracy is used as the evaluation metric.

Baselines. We first consider two baselines with the pretrained single-modal encoders[11,31,32]. (1) 'Frozen' is that we take the pretrained encoders for the two branches and two randomly initialized projectors. (2) 'Similarity' is that we take the similarity from a single branch only. For example, in the first setting, when given a chemical structure, we retrieve the most similar chemical structure from PubChemSTM, then we take the corresponding paired text representation in PubChemSTM as the proxy representation. On this basis, we can calculate the similarity score between the proxy representation and T requested text representations. (3) We further consider the third baseline, a pretrained language model for knowledgeable and versatile machine reading (KV-PLM)[30] on SMILES–text pairs.

Results. The zero-shot retrieval results are shown in Fig. 2a. First, we observe that all the algorithms' accuracies are quite similar between the two settings. Then, as expected, we observe that the baseline Frozen performs no better than the random guess because of the randomly initialized projectors. The Similarity baseline is better than the chance performance by a modest margin, verifying that the pretrained single-modality does learn semantic information but cannot generalize well between modalities. KV-PLM, however, learns semantically meaningful information from SMILES–text pairs, and thus it achieves much higher accuracies on three datasets. For MoleculeSTM, the graph representation from GNNs has higher accuracy on description and pharmacodynamics than the SMILES representation from the transformer model; yet, both of them outperform all the other methods

https://doi.org/10.1038/s42256-023-00759-6

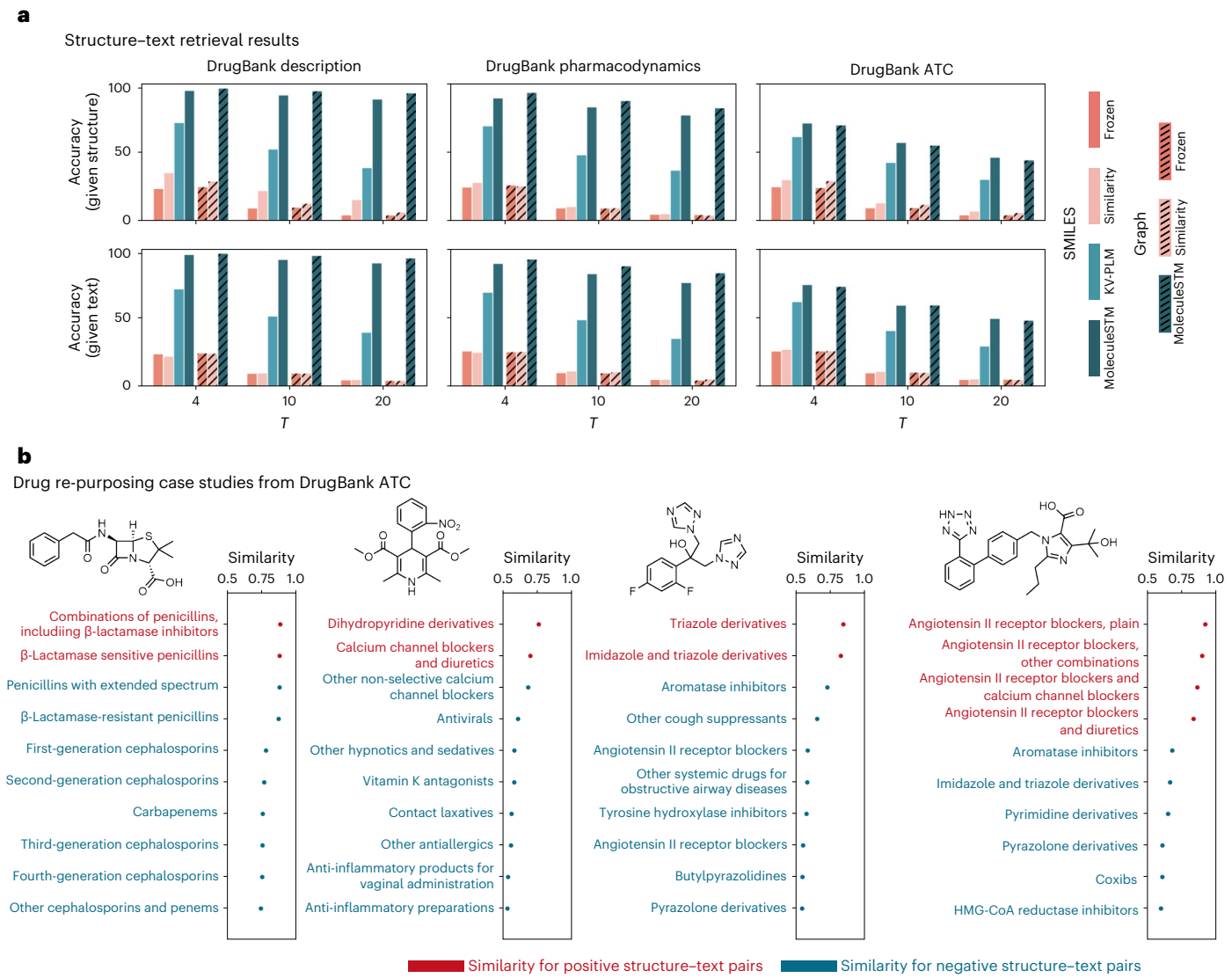

Fig. 2 | Results for zero-shot structure–text retrieval. a, Accuracy for zero-shot structure–text retrieval on three DrugBank datasets. **b**, Four case studies on DrugBank ATC retrieval. HMG-CoA denotes β-hydroxy β-methylglutaryl-CoA.

on three datasets and two settings by a large margin. For example, the accuracy improvements are around 50%, 40% and 15% compared with the best baseline with $T = 20$. Such large improvement gaps verify that MoleculeSTM can play a better role in understanding and bridging the two modalities of molecules.

Case study on drug re-purposing analysis. In Fig. 2b, we further show four case studies on the retrieval quality of ATC. Specifically, given the molecule's chemical structure, we take the 10 (out of 600) most similar ATC labels. It is observed that MoleculeSTM can retrieve the ground-truth ATC labels with high rankings.

Zero-shot text-based molecule editing

Experiments. For molecule editing, we randomly sample 200 molecules from ZINC[20] and a text prompt as the inputs. Four categories of text prompts have been covered. (1) Single-objective editing is the text prompt using the single drug-related property for editing, such as 'molecule with high solubility' and 'molecule more like a drug'. (2) Multi-objective (compositionality) editing is the text prompt applying multiple properties simultaneously, such as 'molecule with high solubility and high permeability'. (3) Binding-affinity-based editing is the text prompt for assay description, where each assay corresponds to one binding-affinity task. A concrete example is ChEMBL 1613777 (ref. 40) with prompt as 'This molecule is tested positive in an assay that are inhibitors and substrates of an enzyme protein. It uses molecular oxygen inserting one oxygen atom into a substrate, and reducing the second into a water molecule.'. The output molecules should have higher binding-affinity scores. (4) Drug-relevance editing is the text prompt to make molecules structurally similar to certain common drugs, for example, 'this molecule looks like penicillin'. We expect the output molecules to be more similar to the target drug than the input drug. For more detailed descriptions of the text prompts, see Supplementary Section D. The evaluation is the satisfactory hit ratio, and it is a hit if the metric difference between output and input is over threshold Δ. The Δ value is task specific, and we consider two typical cases: $\Delta = 0$ indicates a loose condition and $\Delta > 0$ is a strict condition with a larger positive influence. We provide the algorithm pipeline in Fig. 3, and more details can be found in Methods.

Baselines. We consider four baselines. The first three baselines[13] modify the representation of input molecules, followed by the decoding to the molecule space. 'Random' is that we take a random noise as the

https://doi.org/10.1038/s42256-023-00759-6

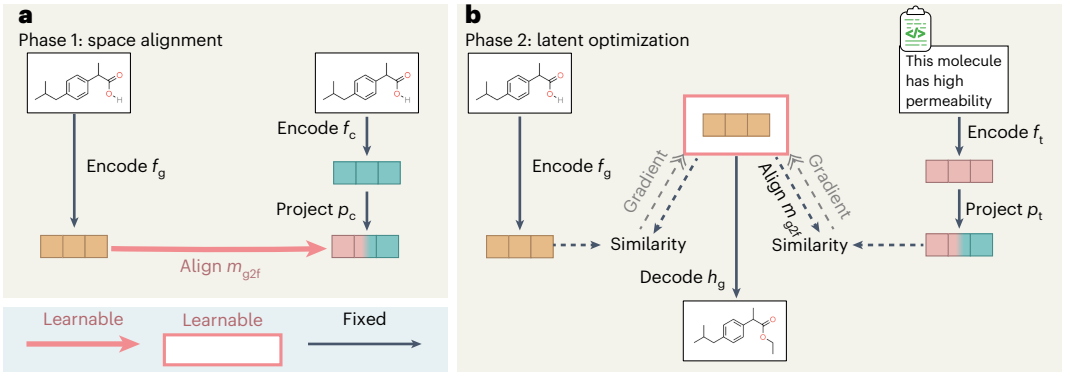

Fig. 3 | Pipelines for the zero-shot text-based molecule editing. a, The space alignment step aligns the representation space of a pretrained molecule generation model and the representation space of MoleculeSTM. **b**, The latent optimization step learns a latent representation that can be similar to both input molecules and textual descriptions.

perturbation to the representation of input molecules. 'PCA' is that we take the eigenvectors as latent directions, where the eigenvectors are obtained after decomposing the latent representation of input molecules using principal component analysis (PCA). 'High variance' is that we take the latent representation dimension with the highest variance and apply the one-hot encoding on it as a semantic direction for editing. In addition, we also consider a baseline directly modifying the molecule space, the genetic search (GS). It is a variant of graph genetic algorithm[41]; the difference is that GS does a random search instead of a guided search by a reward function as no retrieval database is available in the zero-shot setting.

Results. First, we provide the quantitative results for 20 editing tasks across four editing task types in Fig. 4. The empirical results illustrate that the satisfactory hit ratios of MoleculeSTM are the best among all 20 tasks. It verifies that for both SMILES and molecular graph encoders, MoleculeSTM enables a better semantic understanding of the natural language to explore output molecules with the desired properties. Next, we scrutinize the quality of output molecules in Fig. 5 with detailed analysis.

Visual analysis on single-objective molecule editing. We visually analyse the difference between input and output molecules using the single-objective property. Typical modifications are the addition, removal and replacement of functional groups or cores of the molecules. For example, Fig. 5a,b shows two different edits on the same molecule leading to opposite directions in solubility change depending on the text prompt. Replacement of pyridine to a pyrazine core improves the solubility, while insertion of a benzene linkage yields an insoluble molecule. In Fig. 5c,d, changing an amide linkage to an alkyl amine and a urea results in higher and lower permeability of the edited molecules, respectively. Finally, in Fig. 5e,f, a butyl adding-ether and a primary amine to the exact position of the molecule brings more hydrogen-bond acceptors (HBA) and hydrogen-bond donors (HBD), respectively.

Visual analysis on multi-objective molecule editing. We further analyse the multi-objective (compositional) property editing. Water solubility improvement and permeability reduction are consistent when introducing polar groups to the molecule and removing lipophilic hydrocarbons, such as an amide or primary amine replacing a methyl or phenyl in Fig. 5g. However, higher solubility and permeability are achievable if polar functionalities are removed or reduced in number together with hydrophobic components. For example, in Fig. 5h, an amide and a benzene linkage are both removed in the left case, and a [1,2]oxazolo[5,4-b]pyridine substituent is replaced by a water-soluble imidazole with a smaller polar surface in the right case.

Case studies on neighbourhood searching for patent drug molecules. In drug discovery, improvement of drug-like properties of lead molecules is crucial for finding drug candidates[35]. Herein we demonstrate two examples of generating approved drugs from their patented analogues by addressing their property deficiencies based on text prompts. Figure 5i generates celecoxib from its amino-substituted derivative[42], where the removal of the amino group yields a greater intestinal permeability of the molecule leading to higher bioavailability[43]. In Fig. 5j, the trimethoxy benzene moiety, an electron-rich arene known to undergo oxidative phase I metabolisms[44], is replaced by a dimethoxy arene in donepezil by calling for a metabolically stable molecule.

In summary, we conduct rich experiments on 4 types and 20 text-based molecule editing tasks, where the satisfactory hit ratios of MoleculeSTM are superior to baseline methods. Moreover, our editing results can match the expected outcomes based on chemistry domain knowledge. Both quantitative and qualitative results illustrate that MoleculeSTM can learn semantically meaningful information useful for domain applications, which encourages us to explore more challenging tasks with MoleculeSTM in the future.

Molecular property prediction

Experiments. One advantage of MoleculeSTM is that the pretrained chemical structure representation shares information with the external domain knowledge, and such implicit bias can be beneficial for the property prediction tasks. Similar to previous studies on molecule pretraining[21,31], we adopt the MoleculeNet benchmark[9]. It contains eight single-modal binary classification datasets to evaluate the expressiveness of the pretrained molecule representation methods. The evaluation metric is the area under the receiver operating characteristic curve[45].

Baselines. We consider two types of chemical structure, the SMILES string and the molecular graph. For the SMILES string, we take three baselines: the randomly initialized models and two pretrained language models (MegaMolBART[11] and KV-PLM[30]). For the molecular graph, in addition to the random initialization, we consider five pretraining-based methods as baselines: AttrMasking[21], ContextPred[21], InfoGraph[46], MolCLR[47] and GraphMVP[8].

Results. As shown in Table 1, we first observe that pretraining-based methods improve the overall classification accuracy compared with the randomly initialized ones. MoleculeSTM on the SMILES string has consistent improvements on six out of eight tasks compared with the

https://doi.org/10.1038/s42256-023-00759-6

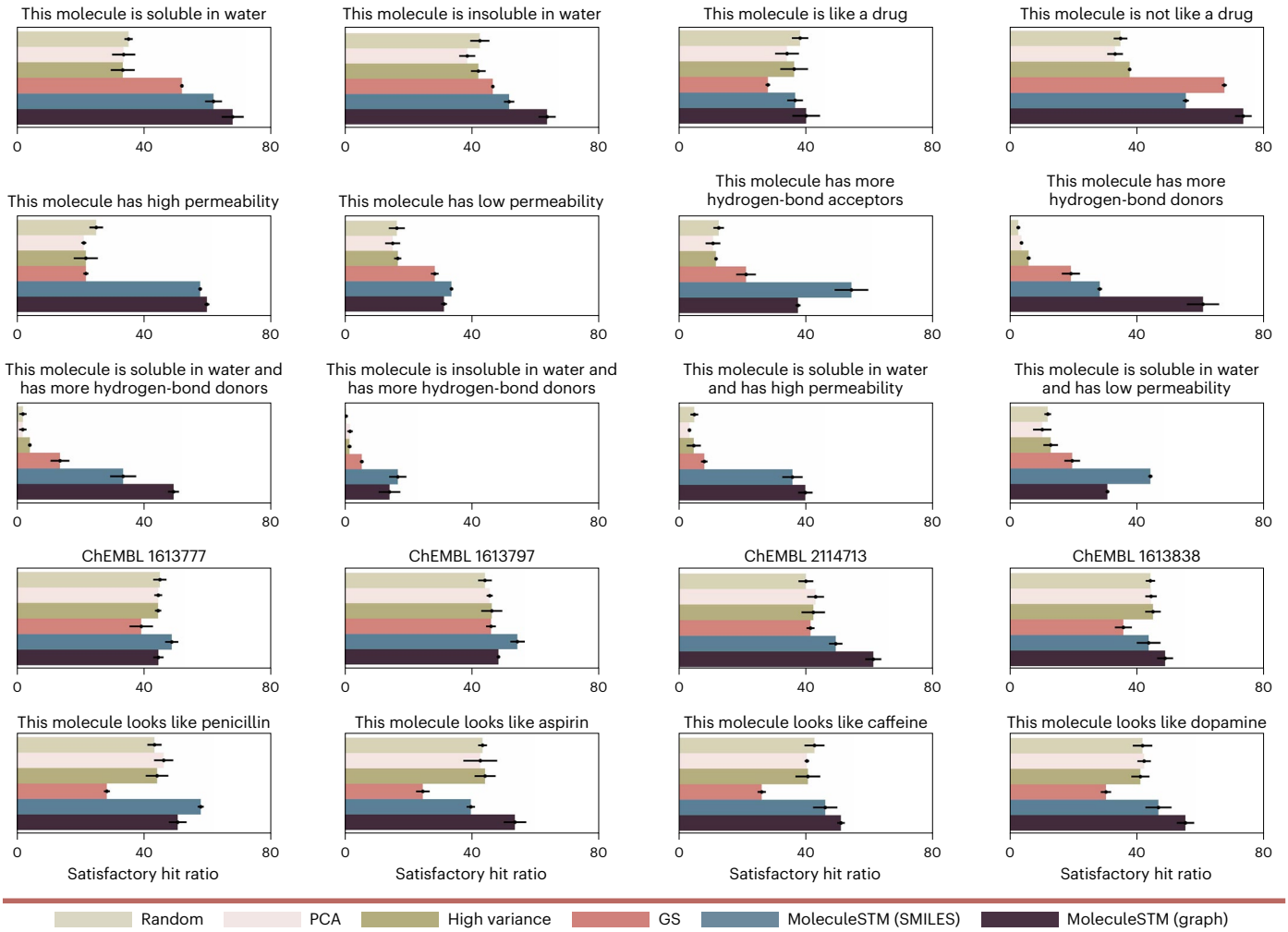

Fig. 4 | Visualization results for the zero-shot text-based molecule editing. Satisfactory hit ratios (%) of four types of text-based editing task: eight single-objective, four multi-objective, four ChEMBL binding-affinity-based editing tasks (pretrained random forest as an evaluator, and detailed text prompts are in Supplementary Section D), and four drug-relevance editing tasks. The satisfactory threshold (Δ) is 0 for all visualized results. Each task runs for three random seeds, and the length of each error bar represents the standard deviation.

three baselines. MoleculeSTM on the molecular graph performs the best on four out of eight tasks, while it performs comparably to the best baselines in other four tasks. In both cases, the overall performances (that is, taking an average across all eight tasks) of MoleculeSTM are the best among all the methods.

Discussion

In this work, we have presented a multi-modal model, MoleculeSTM, to illustrate the effectiveness of incorporating textual descriptions for molecule representation learning. On two newly proposed zero-shot tasks and one standard property prediction benchmark, we confirmed consistently improved performance of MoleculeSTM compared with the existing methods. In addition, we observed that MoleculeSTM can retrieve novel drug–target relations and successfully modify molecule substructures to gain the desired properties. These functionalities may accelerate various downstream drug discovery practices, such as re-purposing and multi-objective lead optimization. Furthermore, the outcomes of such downstream tasks have been found to be consistent with the feedback from chemistry experts, reflecting the domain knowledge exploration ability of MoleculeSTM.

One limitation of this work is data insufficiency. Although PubChemSTM is 28× larger than the dataset used in existing studies, it can be further improved and may require support from the entire community in the future. The second bottleneck of this work is the expressiveness of chemical structure models, including the SMILES encoder, the GNN encoder and the SMILES-based molecule generative model. The development of more expressive architectures is perpendicular to this work and can be feasibly adapted to our multi-modal pretraining framework.

For future directions, we would like to extend MoleculeSTM from cheminformatics to bioinformatics tasks with richer textual information. This enables us to consider structure-based drug design problems such as protein–ligand binding and fragment design. Besides, the three-dimensional geometric information has become more important for small molecules and polymers and can thus be merged into our foundation model. Last but not least, the tokenization of the textual description may require extra effort. Certain tasks possess rich terminologies (for example, the ATC codes in DrugBank ATC), and the overall performance is affected accordingly. Such fundamental problems should be handled carefully.

Methods

This section briefly describes certain modules in both pretraining and downstream tasks. Detailed specifications, such as dataset construction, model architectures and hyperparameters, can be found in Supplementary Section A.

https://doi.org/10.1038/s42256-023-00759-6

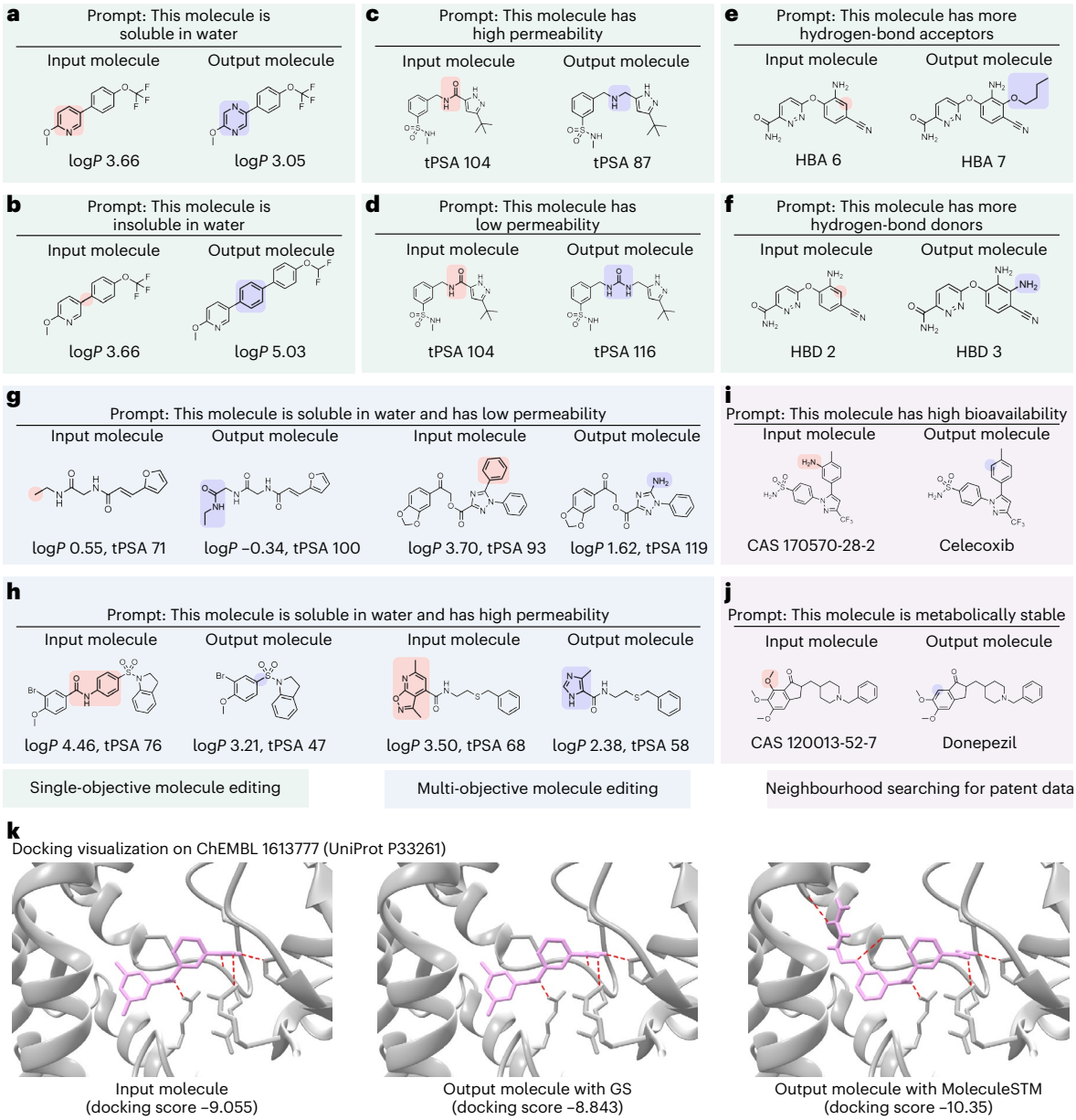

Fig. 5 | Visual analysis on text-based molecule editing. a–j, Case studies for solubility editing (**a**,**b**), permeability editing (**c**,**d**), acceptor and donor editing (**e**,**f**), solubility and permeability editing (**g**,**h**) and neighbourhood searching for patent data (**i**,**j**). The pink and blue regions mark the functional groups before and after the editing, and we list the chemical abstracts service (CAS) registry number. **k**, Binding-affinity-based editing. The dashed red lines mark the potential bindings.

MoleculeSTM pretraining

Dataset construction. For the structure–text pretraining, we consider the PubChem database[34] as the data source. PubChem includes 112 million molecules, which is one of the largest public databases for molecules. The PubChem database has many fields, and previous work[30] uses the synonym field to match with an academic paper corpus[48], resulting in a dataset with 10,000 structure–text pairs. Meanwhile, the PubChem database has another field called 'string' with more comprehensive and versatile molecule annotations. We utilize this field to construct a large-scale dataset called PubChemSTM, consisting of 250,000 molecules and 281,000 structure–text pairs.

In addition, even though PubChemSTM is the largest dataset with textual descriptions, its dataset size is comparatively small compared with the peers from other domains (for example, 400 million in the vision-language domain[24]). To mitigate such a data insufficiency issue, we adopt the pretrained models from existing checkpoints and then conduct the end-to-end pretraining, as discussed next.

Chemical structure branch f_c. This work considers two types of chemical structure: the SMILES string views the molecule as a sequence, and the two-dimensional molecular graph takes the atoms and bonds as the nodes and edges, respectively. Then, based on the chemical structures, we apply a deep learning encoder f_c to get a latent vector as molecule representation. Specifically, for the SMILES string, we take the encoder from MegaMolBART[11], which is pretrained on 500 million molecules from the ZINC database[49]. For the molecular graph, we take a pretrained graph isomorphism network[15] using GraphMVP pretraining[31]. GraphMVP is doing a multi-view pretraining between the two-dimensional topologies and three-dimensional geometries on 250,000 conformations from the Geometric Ensemble of Molecules

Table 1 | Results on eight MoleculeNet[9] binary classification tasks

	Method	BBBP ↑	Tox21 ↑	ToxCast ↑	Sider ↑	ClinTox ↑	MUV ↑	HIV ↑	Bace ↑	Average ↑
SMILES	– (random initialized)	66.54±0.95	71.18±0.67	61.16±1.15	58.31±0.78	88.11±0.70	62.74±1.57	70.32±1.51	80.02±1.66	69.80
	MegaMolBART	68.89±0.17	73.89±0.67	63.32±0.79	59.52±1.79	78.12±4.62	61.51±2.75	71.04±1.70	**82.46±0.84**	69.84
	KV-PLM	70.50±0.54	72.12±1.02	55.03±1.65	59.83±0.56	**89.17±2.73**	54.63±4.81	65.40±1.69	78.50±2.73	68.15
	MoleculeSTM	**70.75±1.90**	**75.71±0.89**	**65.17±0.37**	**63.70±0.81**	86.60±2.28	**65.69±1.46**	**77.02±0.44**	81.99±0.41	**73.33**
Graph	– (random initialized)	63.90±2.25	75.06±0.24	64.64±0.76	56.63±2.26	79.86±7.23	70.43±1.83	76.23±0.80	73.14±5.28	69.99
	AttrMasking	67.79±2.60	75.00±0.20	63.57±0.81	58.05±1.17	75.44±8.75	73.76±1.22	75.44±0.45	80.28±0.04	71.17
	ContextPred	63.13±3.48	74.29±0.23	61.58±0.50	60.26±0.77	80.34±3.79	71.36±1.44	70.67±3.56	78.75±0.35	70.05
	InfoGraph	64.84±0.55	76.24±0.37	62.68±0.65	59.15±0.63	76.51±7.83	72.97±3.61	70.20±2.41	77.64±2.04	70.03
	MolCLR	67.79±0.52	75.55±0.43	64.58±0.07	58.66±0.12	84.22±1.47	72.76±0.73	75.88±0.24	71.14±1.21	71.32
	GraphMVP	68.11±1.36	**77.06±0.35**	**65.11±0.27**	60.64±0.13	84.46±3.10	**74.38±2.00**	**77.74±2.51**	80.48±2.68	73.50
	MoleculeSTM	**69.98±0.52**	76.91±0.51	65.05±0.39	**60.96±1.05**	**92.53±1.07**	73.40±2.90	76.93±1.84	**80.77±1.34**	**74.57**

The mean and standard deviation of test area under the receiver operating characteristic curve on three random seeds are reported. The optimal results of using SMILES and Graph are indicated with bold.

(GEOM) dataset[50]. Thus, although we are not explicitly utilizing the three-dimensional geometries, the state-of-the-art pretrained graph isomorphism network models can implicitly encode such information.

Textual description branch f_t. The textual description branch provides a high-level description of the molecule's functionality. We can view this branch as domain knowledge to strengthen the molecule representation. Such domain knowledge is in the form of natural language, and we use the BERT model[37] as the text encoder f_t. We further adapt the pretrained SciBERT[32], which was pretrained on the textual data from the chemical and biological domain.

Contrastive pretraining. For the MoleculeSTM pretraining, we adopt the contrastive learning strategy, for example, EBM-NCE[31] and InfoNCE[33]. EBM-NCE and InfoNCE align the structure–text pairs for the same molecule and contrast the pairs for different molecules simultaneously. We consider the selection of contrastive pretraining methods as one important hyperparameter. The objectives for EBM-NCE and InfoNCE are

$$\begin{aligned}&\mathcal{L}_{\text{EBM-NCE}}\\&=-\frac{1}{2}\left(\mathbb{E}_{\mathbf{x}_c,\mathbf{x}_t}\left[\log\sigma\left(E\left(\mathbf{x}_c,\mathbf{x}_t\right)\right)\right]+\mathbb{E}_{\mathbf{x}_c,\mathbf{x}_t'}\left[\log\left(1-\sigma(E(\mathbf{x}_c,\mathbf{x}_t'))\right)\right]\right)\\&\qquad+\mathbb{E}_{\mathbf{x}_c,\mathbf{x}_t}\left[\log\sigma\left(E\left(\mathbf{x}_c,\mathbf{x}_t\right)\right)\right]+\mathbb{E}_{\mathbf{x}_c',\mathbf{x}_t}\left[\log\left(1-\sigma(E(\mathbf{x}_c',\mathbf{x}_t))\right)\right],\\&\mathcal{L}_{\text{InfoNCE}}\\&=-\frac{1}{2}\mathbb{E}_{\mathbf{x}_c,\mathbf{x}_t}\left[\log\frac{\exp(E(\mathbf{x}_c,\mathbf{x}_t))}{\exp(E(\mathbf{x}_c,\mathbf{x}_t))+\sum_{\mathbf{x}_t'}\exp(E(\mathbf{x}_c,\mathbf{x}_t'))}+\log\frac{\exp(E(\mathbf{x}_c,\mathbf{x}_t))}{\exp(E(\mathbf{x}_c,\mathbf{x}_t))+\sum_{\mathbf{x}_c'}\exp(E(\mathbf{x}_c',\mathbf{x}_t))}\right],\end{aligned}\tag{1}$$

where σ is the sigmoid activation function, $\mathbf{x}_c$ and $\mathbf{x}_t$ form the structure–text pair for each molecule, and $\mathbf{x}_{c'}$ and $\mathbf{x}_{t'}$ are the negative samples randomly sampled from the noise distribution, which we use the empirical data distribution. $E(\cdot)$ is the energy function with a flexible formulation, and we use the dot product on the jointly learned space, that is, $E(\mathbf{x}_c, \mathbf{x}_t) = \langle p_c \circ f_c(\mathbf{x}_c), p_t \circ f_t(\mathbf{x}_t)\rangle$, where $\circ$ is the function composition.

Zero-shot structure–text retrieval

Given a chemical structure and T textual descriptions, the retrieval task is to select the textual description with the highest similarity to the chemical structure (or vice versa) based on a score calculated on the joint representation space. This is appealing for specific drug discovery tasks, such as drug re-purposing or indication expansion[30,51]. We highlight that pretrained models are used for retrieval in the zero-shot setting, that is, without model optimization for this retrieval task. Existing studies[52] have witnessed the potential issue that utilizing the chemical structure alone is not sufficient, while MoleculeSTM enables a novel perspective by adopting the textual description with the utilization of the high-level functionality of molecules.

In such a zero-shot task setting, all the encoders (f_c, f_t) and projectors (p_c, p_t) are pretrained from MoleculeSTM, and stay frozen in this downstream task. An example of the retrieval task of setting 1 is

$$\text{Retrieval}(\mathbf{x}_c)=\arg\max_{\tilde{\mathbf{x}}_t}\left\{\langle p_c\circ f_c(\mathbf{x}_c), p_t\circ f_t(\tilde{\mathbf{x}}_t)\rangle \mid \tilde{\mathbf{x}}_t\in T\text{ textual descriptions}\right\},\tag{2}$$

Zero-shot text-based molecule editing

The objective of the molecule editing task is to modify the chemical structure of molecules such as functional group change[53] and scaffold hopping[54,55]. Traditional methods for molecule editing highly rely on domain experts and could be subjective or biased[56,57]. ML methods have provided an alternative strategy to solve this issue. Given a fixed pretrained molecule generative model (encoder f_g and decoder h_g), the ML editing methods learn a semantically meaningful direction on the latent representation (or latent code) space. The decoder h_g then generates output molecules with the desired properties by moving along the direction. In MoleculeSTM, with the pretrained joint representation space, we can accomplish this task by injecting the textual description in a zero-shot manner. As shown in Fig. 3, we need two phases. The first phase is space alignment, where we train an adaptor module to align the representation space of the generative model to the joint representation space of MoleculeSTM. The second phase is latent optimization, where we directly learn the latent code using two similarity scores as the objective function. Finally, decoding the optimized latent code can lead to the output molecules. Notice that during this editing process, both the MoleculeSTM (f_c, p_c, f_t, p_t) and a pretrained molecule generative model (f_g, h_g) are frozen.

Phase 1 space alignment. In this phase, the goal is to learn an adaptor module to align the representation space of the generative model to the joint representation space of MoleculeSTM. Following the Gaussian distribution, the objective function is

$$\mathcal{L}=\| m_{g2f}\circ f_g(\mathbf{x}_c)-p_c\circ f_c(\mathbf{x}_c)\|^2,\tag{3}$$

where m_{g2f} is the adaptor module optimized to align the two latent spaces.

Phase 2 latent optimization. In this phase, given an input molecule $\mathbf{x}_{c,in}$ and a text prompt $\mathbf{x}_t$, the goal is to optimize a latent code w directly. The optimal w should be close to the representations of $\mathbf{x}_{c,in}$ and $\mathbf{x}_t$ simultaneously, as:

$$w = \underset{w\in\mathcal{W}}{\operatorname{argmin}}\left(-\mathscr{L}_{\text{cosine-sim}}\left(m_{g2f}(w), p_t \circ f_t(\mathbf{x}_t)\right) + \lambda \cdot \mathscr{L}_{l_2}\left(w, f_g(\mathbf{x}_{c,in})\right)\right), \quad (4)$$

where $\mathcal{W}$ is the latent code space, $\mathscr{L}_{\text{cosine-sim}}$ is the cosine-similarity, $\mathscr{L}_{l_2}$ is the l_2 distance, and λ is a coefficient to balance these two similarity terms. Finally, after we optimize the latent code w, we will do decoding using the decoder from the pretrained generative model to obtain the output molecule: $\mathbf{x}_{c,out} = h_g(w)$.

Evaluation. The evaluation metric is the satisfactory hit ratio. Suppose we have an input molecule $\mathbf{x}_{c,in}$ and a text prompt $\mathbf{x}_t$, the editing algorithm will generate an output molecule $\mathbf{x}_{c,out}$. Then we use the hit ratio to measure if the output molecule can satisfy the conditions as indicated in the text prompt.

$$\text{hit}(\mathbf{x}_{c,in}, \mathbf{x}_t) = \begin{cases} 1, & \exists\lambda,\ \text{s.t.}\ \mathbf{x}_{c,out} = h_g(w;\lambda) \wedge \text{satisfy}\left(\mathbf{x}_{c,in}, \mathbf{x}_{c,out}, \mathbf{x}_t\right) \\ 0, & \text{otherwise} \end{cases},$$

$$\text{hit}\,(t) = \frac{\sum_{i=1}^{N} \text{hit}(\mathbf{x}^i_{c,in}, \mathbf{x}_t)}{N}, \quad (5)$$

where N is the total number of editing outputs, and satisfy$(\cdot)$ is the satisfaction condition. It is task specific, and we list the five key points below. (1) For single-objective property-based editing, we use the logarithm of partition coefficient (log$P$), quantitative estimate of drug-likeness (QED) and topological polar surface area (tPSA) as the proxies to measure the molecule solubility[58], drug likeness[59] and permeability[60], respectively. The count of HBA and HBD are calculated explicitly. It will be a successful hit once the measurement difference between the input molecule and output molecule is above a certain threshold Δ. (2) For multiple-objective property-based editing, we feed in a text prompt describing multiple properties' composition. The Δ is composed of the threshold on each individual property, and a successful hit needs to satisfy all the properties simultaneously. (3) For binding-affinity-based editing, we take the ground-truth data from ChEMBL to train a binary classifier and test if the output molecules have higher confidence than the input molecules, and Δ is fixed to 0. (4) For drug-relevance editing, we use Tanimoto similarity to quantify the structural similarity[61]. It will be a hit if the similarity score between the output molecule and target drug is higher than the similarity between the input molecule and target drug by a threshold Δ. (5) Besides, the choice of satisfactory threshold Δ is also task specific, and the higher the values are, the stricter the satisfaction condition is. The details of the threshold values can be found in Supplementary Section D.

Molecular property prediction

For modelling, we take the pretrained encoder f_c and add a prediction head h_c to predict a categorical-valued or scalar-valued molecular property such as binding affinity or toxicity. Both f_c and h_c are optimized to fit the target property, that is, in a fine-tuning manner[21,31].

Data availability

All the datasets are provided on Hugging Face at https://huggingface.co/datasets/chao1224/MoleculeSTM/tree/main. Specifically for the release of PubChemSTM, we encountered a big challenge regarding the textual data license. As confirmed with the PubChem group, performing research on these data does not violate their license; however, PubChem does not possess the license for the textual data, which necessitates an extensive evaluation of the license for each of the 280 structure–text pairs in PubChemSTM. This has hindered the release of PubChemSTM. Nevertheless, we have (1) described the detailed preprocessing steps in Supplementary Section A.1, (2) provided the molecules with CID file (https://huggingface.co/datasets/chao1224/MoleculeSTM/blob/main/PubChemSTM_data/raw/CID2SMILES.csv) in PubChemSTM and (3) have also provided the detailed preprocessing scripts (https://github.com/chao1224/MoleculeSTM/tree/main/preprocessing/PubChemSTM). By utilizing these scripts, users can easily reconstruct the PubChemSTM dataset.

Code availability

The source code can be found on GitHub (https://github.com/chao1224/MoleculeSTM/tree/main) and Zenodo[62]. The scripts for pretraining and three downstream tasks are provided at https://github.com/chao1224/MoleculeSTM/tree/main/scripts. The checkpoints of the pretrained models are provided on Hugging Face at https://huggingface.co/chao1224/MoleculeSTM/tree/main. Beyond the methods described so far, to help users try our MoleculeSTM model, this release includes demos in notebooks (https://github.com/chao1224/MoleculeSTM). Furthermore, users can customize their own datasets by checking the datasets folder (https://github.com/chao1224/MoleculeSTM/tree/main/MoleculeSTM/datasets).

References

1. Sullivan, T. A tough road: cost to develop one new drug is $2.6 billion; approval rate for drugs entering clinical development is less than 12%. *Policy Medicine* https://www.policymed.com/2014/12/a-tough-road-cost-to-develop-one-new-drug-is-26-billion-approval-rate-for-drugs-entering-clinical-de.html (2019).
2. Patronov, A., Papadopoulos, K. & Engkvist, O. in *Artificial Intelligence in Drug Design* (ed. Heietz, A.) 153–176 (Springer, 2022).
3. Jayatunga, M. K., Xie, W., Ruder, L., Schulze, U. & Meier, C. AI in small-molecule drug discovery: a coming wave. *Nat. Rev. Drug Discov.* **21**, 175–176 (2022).
4. Jumper, J. et al. Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).
5. Rohrer, S. G. & Baumann, K. Maximum unbiased validation (MUV) data sets for virtual screening based on PubChem bioactivity data. *J. Chem. Inf. Model.* **49**, 169–184 (2009).
6. Liu, S. et al. Practical model selection for prospective virtual screening. *J. Chem. Inf. Model.* **59**, 282–293 (2018).
7. Duvenaud, D. K. et al. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems* Vol. 2 (eds Cortes, C. et al.) 2224–2232 (Curran Associates, 2015).
8. Liu, S., Demirel, M. F. & Liang, Y. *N*-gram graph: simple unsupervised representation for graphs, with applications to molecules. In *Advances in Neural Information Processing Systems* Vol. 32 (eds Wallach, H. et al.) 8464–8476 (Curran Associates, 2019).
9. Wu, Z. et al. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.* **9**, 513–530 (2018).
10. Jin, W., Barzilay, R. & Jaakkola, T. Hierarchical generation of molecular graphs using structural motifs. In *International Conference on Machine Learning* Vol. 119, 4839–4848 (PMLR, 2020).
11. Irwin, R., Dimitriadis, S., He, J. & Bjerrum, E. J. Chemformer: a pre-trained transformer for computational chemistry. *Mach. Learn. Sci. Technol.* **3**, 015022 (2022).
12. Wang, Z. et al. Retrieval-based controllable molecule generation. In *International Conference on Learning Representations* (PMLR, 2023).

13. Liu, S. et al. GraphCG: unsupervised discovery of steerable factors in graphs. In *NeurIPS 2022 Workshop: New Frontiers in Graph Learning* (NeurIPS, 2022).
14. Krenn, M., Häse, F., Nigam, A., Friederich, P. & Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): a 100% robust molecular string representation. *Mach. Learn. Sci. Technol.* **1**, 045024 (2020).
15. Xu, K., Hu, W., Leskovec, J. & Jegelka, S. How powerful are graph neural networks? In *International Conference on Learning Representations* (PMLR, 2019).
16. Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A. & Müller, K.-R. SchNet—a deep learning architecture for molecules and materials. *J. Chem. Phys.* **148**, 241722 (2018).
17. Satorras, V. G., Hoogeboom, E. & Welling, M. E(*n*) equivariant graph neural networks. In *International Conference on Machine Learning* Vol. 139, 9323–9332 (2021).
18. Atz, K., Grisoni, F. & Schneider, G. Geometric deep learning on molecular representations. *Nat. Mach. Intell.* **3**, 1023–1032 (2021).
19. Ji, Y. et al. DrugOOD: out-of-distribution dataset curator and benchmark for AI-aided drug discovery—a focus on affinity prediction problems with noise annotations. In *Proc. AAAI Conference on Artificial Intelligence* Vol. 37, 8023–8031 (2023).
20. Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S. & Coleman, R. G. ZINC: a free tool to discover chemistry for biology. *J. Chem. Inf. Model.* **52**, 1757–1768 (2012).
21. Hu, W. et al. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations* (PMLR, 2020).
22. Liu, S., Guo, H. & Tang, J. Molecular geometry pretraining with SE(3)-invariant denoising distance matching. In *International Conference on Learning Representations* (PMLR, 2022).
23. Larochelle, H., Erhan, D. & Bengio, Y. Zero-data learning of new tasks. In *Proc. AAAI Conference on Artificial Intelligence* Vol. 2, 646–651 (AAAI, 2008).
24. Radford, A. et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning* Vol. 139, 8748–8763 (PMLR, 2021).
25. Nichol, A. et al. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning* Vol. 162, 16784–16804 (PMLR, 2022).
26. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. & Chen, M. Hierarchical text-conditional image generation with clip latents. Preprint at https://arxiv.org/abs/2208.11126 (2022).
27. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D. & Lischinski, D. StyleCLIP: text-driven manipulation of StyleGAN imagery. In *Proc. IEEE/CVF International Conference on Computer Vision* 2085–2094 (IEEE, 2021).
28. Li, S. et al. Pre-trained language models for interactive decision-making. In *Advances in Neural Information Processing Systems* Vol. 35 (eds Koyejo, S. et al.) 31199–31212 (Curran Associates, 2022).
29. Fan, L. et al. MineDojo: building open-ended embodied agents with internet-scale knowledge. In *Advances in Neural Information Processing Systems* Vol. 35 (eds Koyejo, S. et al.) 18343–18362 (Curran Associates, 2022).
30. Zeng, Z., Yao, Y., Liu, Z. & Sun, M. A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. *Nat. Commun.* **13**, 862 (2022).
31. Liu, S. et al. Pre-training molecular graph representation with 3D geometry. In *International Conference on Learning Representations* (PMLR, 2022).
32. Beltagy, I., Lo, K. & Cohan, A. SciBERT: pretrained language model for scientific text. In *Proc. 2019 Conference on Empirical Methods in Natural Language Processing* (eds Inui, K. et al.) 3615–3620 (ACL, 2019).
33. Oord, A.V., Li, Y. & Vinyals, O. Representation learning with contrastive predictive coding. Preprint at https://arxiv.org/abs/1807.03748 (2018).
34. Kim, S. et al. PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Res.* **49**, D1388–D1395 (2021).
35. Hughes, J. P., Rees, S., Kalindjian, S. B. & Philpott, K. L. Principles of early drug discovery. *Br. J. Pharmacol.* **162**, 1239–1249 (2011).
36. Vaswani, A. et al. Attention is all you need. In *Advances in Neural Information Processing System* Vol. 30 (eds von Luxburg, U. et al.) 6000–6010 (Curran Associates, 2017).
37. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proc. 2019 Association for Computational Linguistics* (eds Burstein, J. et al.) 4171–4186 (ACL, 2019).
38. Gu, X., Lin, T.-Y., Kuo, W. & Cui, Y. Open-vocabulary object detection via vision and language knowledge distillation. In *International Conference on Learning Representations* (PMLR, 2022).
39. Wishart, D. S. et al. DrugBank 5.0: a major update to the drugbank database for 2018. *Nucleic Acids Res.* **46**, D1074–D1082 (2018).
40. Mendez, D. et al. ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Res.* **47**, D930–D940 (2018).
41. Jensen, J. H. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chem. Sci.* **10**, 3567–3572 (2019).
42. Talley, J. J. et al. Substituted pyrazolyl benzenesulfonamides for the treatment of inflammation. US patent 5,760,068 (1998).
43. Dahlgren, D. & Lennernäs, H. Intestinal permeability and drug absorption: predictive experimental, computational and in vivo approaches. *Pharmaceutics* **11**, 411 (2019).
44. Guroff, G. et al. Hydroxylation-induced migration: the NIH shift. Recent experiments reveal an unexpected and general result of enzymatic hydroxylation of aromatic compounds. *Science* **157**, 1524–1530 (1967).
45. Bradley, A. P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.* **30**, 1145–1159 (1997).
46. Sun, F.-Y., Hoffmann, J., Verma, V. & Tang, J. InfoGraph: unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *International Conference on Learning Representations* (PMLR, 2020).
47. Wang, Y., Wang, J., Cao, Z. & Farimani, A. B. Molecular contrastive learning of representations via graph neural networks. *Nat. Mach. Intell.* **4**, 279–287 (2022).
48. Lo, K., Wang, L. L., Neumann, M., Kinney, R. & Weld, D. S. S2ORC: the semantic scholar open research corpus. In *Proc. Association for Computational Linguistics* (eds Jurafsky, D. et al.) 4969–4983 (ACL, 2020).
49. Sterling, T. & Irwin, J. J. ZINC 15—ligand discovery for everyone. *J. Chem. Inf. Model.* **55**, 2324–2337 (2015).
50. Axelrod, S. & Gomez-Bombarelli, R. GEOM, energy-annotated molecular conformations for property prediction and molecular generation. *Sci. Data* **9**, 185 (2022).
51. Aggarwal, S. Targeted cancer therapies. *Nat. Rev. Drug Discov.* **9**, 427–428 (2010).
52. Guney, E. Reproducible drug repurposing: when similarity does not suffice. In *Pacific Symposium on Biocomputing* (eds Altaman, R. B. et al.) 132–143 (World Scientific, 2017).
53. Ertl, P., Altmann, E. & McKenna, J. M. The most common functional groups in bioactive molecules and how their popularity has evolved over time. *J. Med. Chem.* **63**, 8408–8418 (2020).
54. Böhm, H.-J., Flohr, A. & Stahl, M. Scaffold hopping. *Drug Discov. Today Technol.* **1**, 217–224 (2004).
55. Hu, Y., Stumpfe, D. & Bajorath, J. Recent advances in scaffold hopping: miniperspective. *J. Med. Chem.* **60**, 1238–1246 (2017).

56. Drews, J. Drug discovery: a historical perspective. *Science* **287**, 1960–1964 (2000).
57. Gomez, L. Decision making in medicinal chemistry: the power of our intuition. *ACS Med. Chem. Lett.* **9**, 956–958 (2018).
58. Leo, A., Hansch, C. & Elkins, D. Partition coefficients and their uses. *Chem. Rev.* **71**, 525–616 (1971).
59. Bickerton, G. R., Paolini, G. V., Besnard, J., Muresan, S. & Hopkins, A. L. Quantifying the chemical beauty of drugs. *Nat. Chem.* **4**, 90–98 (2012).
60. Ertl, P., Rohde, B. & Selzer, P. Fast calculation of molecular polar surface area as a sum of fragment-based contributions and its application to the prediction of drug transport properties. *J. Med. Chem.* **43**, 3714–3717 (2000).
61. Butina, D. Unsupervised data base clustering based on daylight's fingerprint and Tanimoto similarity: a fast and automated way to cluster small and large data sets. *J. Chem. Inf. Comput. Sci.* **39**, 747–750 (1999).
62. Liu, S. et al. Multi-modal molecule structure-text model for text-based editing and retrieval. *Zenodo* https://doi.org/10.5281/zenodo.8303265 (2023).

Acknowledgements

This work was done during S.L.'s internship at NVIDIA Research. We thank the insightful comments from M. L. Gill, A. Stern and other team members from AIAlgo and Clara team at NVIDIA. We also thank the kind help from T. Dierks, E. Bolton, P. Thiessen and others from PubChem for confirming the PubChem license.

Author contributions

S.L., W.N., C.W., Z.Q., C.X. and A.A. conceived and designed the experiments. S.L. performed the experiments. S.L. and C.W. analysed the data. S.L., C.W. and J.L. contributed analysis tools. S.L., W.N., C.W., J.L., Z.Q., L.L., J.T., C.X. and A.A. wrote the paper. J.T., C.X. and A.A. contributed equally to advising this project.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-023-00759-6.

Correspondence and requests for materials should be addressed to Animashree Anandkumar.

Peer review information *Nature Machine Intelligence* thanks Rocío Mercado and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editor: Jacob Huth, in collaboration with the *Nature Machine Intelligence* team.