OXFORD

Briefings in Bioinformatics, 2024, **25(3)**, bbae142
https://doi.org/10.1093/bib/bbae142
Problem Solving Protocol

GLDM: hit molecule generation with constrained graph latent diffusion model

Conghao Wang, Hiok Hian Ong, Shunsuke Chiba and Jagath C. Rajapakse

Corresponding author. Jagath C. Rajapakse, School of Computer Science and Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798. E-mail: ASJagath@ntu.edu.sg

Abstract

Discovering hit molecules with desired biological activity in a directed manner is a promising but profound task in computer-aided drug discovery. Inspired by recent generative AI approaches, particularly Diffusion Models (DM), we propose Graph Latent Diffusion Model (GLDM)—a latent DM that preserves both the effectiveness of autoencoders of compressing complex chemical data and the DM's capabilities of generating novel molecules. Specifically, we first develop an autoencoder to encode the molecular data into low-dimensional latent representations and then train the DM on the latent space to generate molecules inducing targeted biological activity defined by gene expression profiles. Manipulating DM in the latent space rather than the input space avoids complicated operations to map molecule decomposition and reconstruction to diffusion processes, and thus improves training efficiency. Experiments show that GLDM not only achieves outstanding performances on molecular generation benchmarks, but also generates samples with optimal chemical properties and potentials to induce desired biological activity.

Keywords: deep generative models; Diffusion Models; drug design; hit molecule discovery.

INTRODUCTION

Drug discovery with conventional computer-aided approaches are expensive and time-consuming since a large volume of candidate molecules need to be screened [1]. In the recent years, numerous deep learning-based *de novo* drug design approaches have emerged to tackle this challenge by searching for the candidate molecule with optimal properties in a directed manner [2, 3]. However, most of them focused only on chemically valid generation and optimization towards desired chemical properties, and omitted to take any biological insights into accounts [4, 5]. In this work, we study how to guide drug generation with desired biological activities defined by gene expression profiles.

Drug generation with desired biological activities

Initially, approaches targeted at designing drugs focused towards only a single target [6–8]. Then recent research have emerged to generate potential molecules that are able to induce a desired change in gene expression pattern [9, 10]. The arising of these works bridges systems biology and drug design, and remarkably expedites drug discovery process by incorporating desired genomics profiles to tailor candidate molecules. Nevertheless, these research depended on SMILES notations [11] as the representation of drug molecules. Although SMILES string is a concise model of drug molecule which are readily utilized by deep learning algorithms derived from natural language processing area

Conghao Wang received his Bachelor of Engineering degree in the School of Computer Science and Technology from Xi'an Jiaotong University, China. He is currently a PhD candidate in the School of Computer Science and Engineering at Nanyang Technological University, Singapore. His reseach interests focus on application of explainable AI on drug mechanism prediction, and *de novo* design of small molecule and biologics drugs with generative models.
Hiok Hian Ong received his Bachelor of Engineering Science (Computer Science) and Master of Science (Technology Management) from Nanyang Technological University, Singapore. He is currently working as a Machine Learning Engineer. His research interests lie in the intersection between bioinformatics, neural networks and machine learning.
Shunsuke Chiba received his PhD in 2006 under the supervision of Prof. Koichi Narasaka at the University of Tokyo. In 2007, he embarked on his independent career as the faculty of Nanyang Technological University (NTU) Singapore, where he is currently Professor of Chemistry. His research group focuses on the development of fundamentally novel and practical synthetic reactions and catalysis that are useful for efficient supply of complex organic molecules of pharmaceutical interests.
Jagath Rajapakse is Professor of Computer Science and Engineering at the Nanyang Technological University, Singapore. He has BSc degree in Electronics and Telecommunication Engineering from University of Moratuwa, Sri Lanka, and MS and PhD degrees in Electrical and Computer Engineering from University at Buffalo, USA. He was Visiting Scientist to the Max-Planck Institute of Cognitive and Brain Sciences, Germany, and the National Institute of Mental Health, USA. He was Visiting Professor to the Department of Biological Engineering at Massachusetts Institute of Technology. Professor Rajapakse's research works are in the areas of data science, machine learning, brain imaging, and computational and systems biology. His current research focus on developing techniques and tools for diagnosis and treatment of brain diseases. He is also looking into anticancer drug discovery with deep learning.
Received: November 19, 2023. **Revised:** March 8, 2024. **Accepted:** March 3, 2024

[12, 13], graphical molecular structural information is lost and chemistry validity cannot be guaranteed upon generation.

Drug representations with molecular graphs, on the contrary, are able to capture the natural structure of molecules and incorporate valence check during generation to ensure the chemistry validity. The difficulty of graph generation stems from generating discrete structures which are not differentiable. To overcome such difficulty, several probabilistic graph construction approaches have been proposed [14–16], making direct and efficient generation of molecular graphs feasible.

Deep learning-based generative approaches

Recently, Deep learning-based generative algorithms have obtained tremendous success in the domains of text [17, 18] and image generation [19]. The pictures synthesized by AI is nearly indistinguishable from human works. The blossom of generative models also facilitates the process of drug discovery by identifying the hidden patterns of chemical space and search for candidate molecules in an oriented manner. Widely used generative models include Variational Auto-encoder (VAE) [20], Generative Adversarial Networks (GAN) [21] and Diffusion Models (DM) [22].

VAE is a popular method that compresses input data into a latent distribution, then samples and reconstructs new data points from the encoded distribution. However, it would be difficult to incorporate complicated genomics profiles to induce the generation with merely VAE model. GANs are versatile approaches that have achieved great success in various image synthesis tasks [23, 24], but are also revealed to be difficult to train due to their adversarial learning process. DMs are state-of-the-art models that surpass GANs in terms of sample quality and density estimation [25, 26], but DMs are designed for grid-like data such as images and it is non-trivial to apply it for molecular generation. Nonetheless, a recent work proposing latent DM [27] has demonstrated that compressing complex input data with a VAE model and only applying DM on the encoded latent space is feasible to generate high-quality samples in an efficient way.

In this study, our main contributions include the following:

- We present GLDM, a Graph Latent Diffusion Model, to generate potential drug candidates inducing desired gene expression changes. As shown in Figure 1, GLDM incorporates the conditioning gene expression profiles during denoising of the latent representations, and generates hit-like molecules.
- GLDM is shown to be able to generate valid and versatile molecules, and outperform other state-of-the-art methods evaluated by GuacaMol benchmarking [28] in unconditional generation tasks.
- Upon generation under the guidance of gene expression profiles, GLDM significantly surpasses similarity search, which is a widely used method in traditional computer-aided drug discovery. Notably, existing works for gene expression conditioned generation are all based on SMILES and have a validity rate of lower than 10%, whereas GLDM can maintain 100% validity.

RELATED WORK

Modern AI-aided *de novo* drug discovery methods include SMILES-based and graph-based generative approaches. SMILES-based methods prefer to leverage long short-term memory (LSTM) network to encode and generate SMILES notations of drugs. A variety of such methods managed to constrain the generation

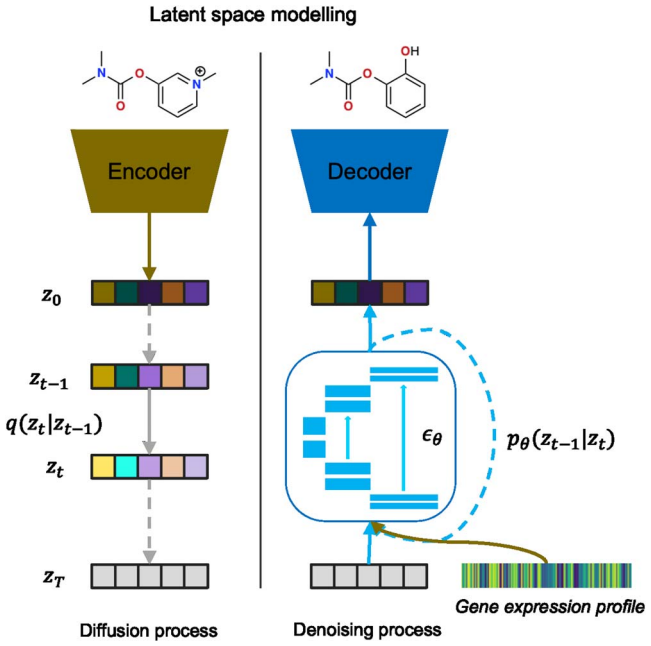

Figure 1. The schematic of GLDM architecture.

with desired pharmaceutical properties [4, 29, 30] or biological activities [9, 10, 31, 32]. Notably, a generative method proposed by [10] used stacked GAN model in the VAE-encoded latent space to generate and refine molecules conditioned by gene signatures. High similarity scores achieved by their generated molecules indicate that they are potential to become drug candidates inducing desired biological activities, which could therefore facilitate hit identification in drug discovery process [33, 34]. However, all these methods suffered from the low validity rate problem since chemical rules are broken during generation of SMILES strings.

To tackle the validity problem, probabilistic graph generators are brought into the area of molecule generation [14, 15], which can perform valence check to ensure chemistry validity during generation. However, early research of molecular graph decoders indicate that atom-by-atom generation is computationally intensive so that such models are only able to produce molecules of small sizes [14, 35]. MoLeR [16] and MGSSL [36] introduced motif-based generation to overcome this difficulty and produce molecules with higher quality. Moreover, NeVAE [37] and CGVAE [38] proposed to constrain the generation with desired properties. GEODIFF [39] presented a DM to generate conformations derived from existing molecules. Currently, however, there are few approaches to constrain the graph-based molecular generation with an overall transcriptomic profile that depicts the desired biological activities in detail. Therefore, motivated by these pioneering works, we propose GLDM—a latent DM to discover hit molecules with motif-based generation constrained on gene expression signatures.

METHODS

Generation of potential drug candidates with desired biological and pharmaceutical properties has remained to be unsolved due to the complexity of chemical structures and difficulty of incorporating prior biological knowledge. In this section, we introduce the proposed GLDM for molecule generation. As shown in Figure 1, our model is composed of an encoder for latent space modelling,

a DM for latent representation manufacture, and a decoder for molecule reconstruction.

Model design

In GLDM, molecules x are represented by molecular graphs, $x = (V, E)$, denoting the node set as V and edge set as E. The input graphs can be readily encoded into latent representations via the encoder, which is denoted as

$$z_0 = F(x) \tag{1}$$

where F denotes the encoder function and z_0 represents the latent code which is also the input to the diffusion process. Specifically, we use FiLMConv [40] layers to build the encoder. FiLMConv is a type of relational graph convolutional network (RGCN) that discriminates the message passing pattern of different edge (relation) types. RGCN models are gaining increasing popularity in modeling molecules over the recent years, since in molecular graphs, edges represent chemical bonds that belong to a variety of relation types.

Given the molecular graphs with N nodes, we extract an adjacency matrix $A \in \mathbb{R}^{N\times N}$, a node feature matrix $H \in \mathbb{R}^{N\times P}$, where P refers to the dimension of node features, and an edge type set R containing all chemical bond types. The FiLMConv layer is formulated by

$$h_i' = \sum_{r\in R}\sum_{j\in N_i^{(r)}} \left(\gamma_i^{(r)} \odot W^{(r)} h_j + \beta_i^{(r)}\right) \tag{2}$$

where $\beta_i^{(r)}, \gamma_i^{(r)} = g(h_i)$ and $g(h_i)$ is a linear layer.

Then we design the *diffusion process* following the denoising diffusion probabilistic model. The latent representation z_0 is destroyed by gradually adding Gaussian noise to the representation at each time step t, aiming at diffusing the data to a white noise distribution after T iterations. In this study, we choose $T = 1000$. Given the original data distribution $z_0 \sim q(z_0)$, and the variance schedule $\beta_0, \ldots, \beta_T$ of each time step, the diffusion process is formulated by

$$\begin{aligned} q(z_{1:T}|z_0) &= \prod q(z_t|z_{t-1}), \\ q(z_t|z_{t-1}) &= \mathcal{N}(z_t; \sqrt{1-\beta_t} z_{t-1}, \beta_t \mathbf{I}) \end{aligned} \tag{3}$$

By defining $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s = \prod_{s=1}^{t}(1-\beta_s)$, z_t can be sampled at an arbitrary time step t with

$$q(z_t|z_0) = \mathcal{N}(z_t; \bar{\alpha}_t z_0, (1-\bar{\alpha}_t)\mathbf{I}) \tag{4}$$

In contrast to the diffusion process, the *denoising process* targets at recovering the original data distribution from the destructed data representation z_T. The learnable denoising kernel is denoted as

$$\begin{aligned} p_\theta(z_{0:T}) &= p(z_T) \prod_{t=1}^{T} p_\theta(z_{t-1}|z_t), \\ p_\theta(z_{t-1}|z_t) &= \mathcal{N}(z_{t-1}; \mu_\theta(z_t, t), \sigma_t^2 \mathbf{I}) \end{aligned} \tag{5}$$

The distribution of z_T is set as standard normal distribution $p(z_T) = \mathcal{N}(z_T; 0, \mathbf{I})$.

Eventually, with the $\hat{z}_0$ reconstructed by the denoising process, a graph decoder is employed to ensemble the molecular graph:

$$\hat{x} = G(\hat{z}_0) \tag{6}$$

where G is the decoder function.

During the inference stage, we directly sample z_T from the standard normal distribution and apply the denoising process, and the graph decoder G generates new molecular graph samples.

Training objectives

Training of the proposed model involves two stages: (1) training the encoder and decoder to learn an expressive latent space; and (2) training the DM in the latent space.

Training the encoder and decoder

Our encoder and decoder is developed by combining the VAE and adversarial learning. The specific architecture is depicted in Figure 2. The overall loss function is given by

$$\mathcal{L}_{AE}(x) = \mathcal{L}_{rec}(x) + \lambda_{reg}\mathcal{L}_{reg}(x) \tag{7}$$

where $\mathcal{L}_{rec}$ and $\mathcal{L}_{reg}$ are the reconstruction loss and the regularization loss, respectively, and the constant λ_{reg} is set to be smaller than 1 to make the training stable.

Inspired by MoLeR [16], we formulate the reconstruction loss to be the expectation of log probability of a uniformly chosen step:

$$\mathcal{L}_{rec}(x) = \mathbb{E}_{z\sim q(z|x)} \mathbb{E}_{i\sim U} \log p(s_i|z, M_i) \tag{8}$$

where s_i is the ith generation step and M_i is the partially generated molecule. This ensures all generation steps are independent from each other and can be trained in parallel.

The target of the encoder is to compress the molecular graphs into a compact low-dimensional latent space. And according to [41], using adversarial loss helps capture a more informative latent space than the Kullback–Leibler (KL) Divergence [20] commonly used in VAE. However, the typical adversarial loss introduced by GAN [21] sometimes could not properly regularize the latent distribution to conform the prior distribution. The Wasserstein Autoencoder (WAE) [42] addresses this issue and balances the reconstruction and the regularization loss. Thus, in this work, we leverage the Wasserstein loss as the regularization loss. Following [43], we append a gradient penalty term to the original Wasserstein loss:

$$\begin{aligned} \mathcal{L}_{WAE}(x) = \mathbb{E}_{\tilde{z}\sim p(\tilde{z})}[D(\tilde{z})] - \mathbb{E}_{x\sim p_d(x)}[D(F(x))] \\ + \lambda \mathbb{E}_{\hat{z}\sim p(\hat{z})}\left[\left(\left\|\nabla_{\hat{z}} D(\hat{z})\right\|_2 - 1\right)^2\right] \end{aligned} \tag{9}$$

where $x \sim p_d(x)$ and $\tilde{z} \sim p(\tilde{z})$ denote the the input data distribution and the prior latent distribution, respectively. D is an additional discriminator attempting to discern latent representations encoded from the input data $F(x)$ and the ones sampled from the prior latent distribution $\tilde{z}$.

$\hat{z}$ are sampled randomly from the latent distribution $\hat{z} \sim p(\hat{z})$, where $p(\hat{z})$ is implicitly defined to sample uniformly along straight lines between the samples from the prior distribution $p(z)$ and data distribution $p_d(x)$. And the penalty coefficient is set to be $\lambda = 10$.

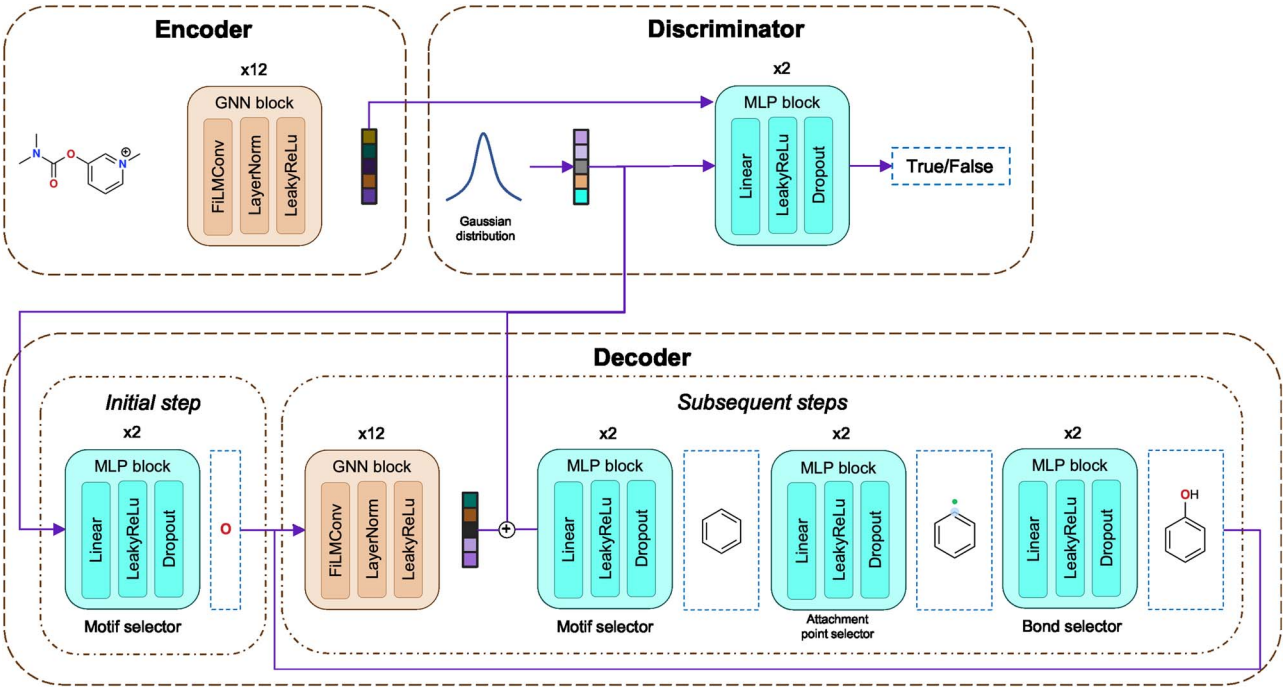

Figure 2. Autoencoder architechture. In the encoder, the input molecular graph is compiled into the latent representation conforming Gaussian distributions by the GNN layers. Then, a sampled latent representation is passed to the decoder to reconstruct the molecule structures. Following MoLeR, our decoder is composed of three MLPs, i.e. motif selector, attachment point selector and bond selector. In the initial step, only the motif selector is applied to predict the first fragment or single atom. In subsequent steps, another GNN block is used to convert the partially constructed graph into the latent representation, which is combined with the sampled latent representation as the input to the three selectors. Attachment point selector is responsible for predicting the node on the new fragment to be attached to the partially constructed graph. And the bond selector predicts the chemical bond to connect them. In the end, the discriminator is adopted to distinguish sampled and original latent representations.

Training the DM

Following [22], training of the DM is performed to optimize the evidence lower bound:

$$\mathcal{L}_{ELBO} = \mathcal{D}_{KL}(q(z_T|z_0)\|p(z_T)) + \sum_{t=1}^{T} \mathbb{E}_{q(z_t|z_0)}\mathcal{L}_{t-1}, \quad \text{where} \quad \mathcal{L}_{t-1} = \mathcal{D}_{KL}(q(z_{t-1}|z_t, z_0)\|p(z_{t-1}|z_t)) \tag{10}$$

which consists of the KL Divergence at the final time step and the sum of KL Divergence over the previous time steps $\mathcal{L}_{t-1}$.

The variance schedule β_t of the diffusion process (3) are set to constants so that the diffusion process is not learnable and the loss term (the first term in (10)) is neglected. Since the true diffusion posterior is tractable as a Gaussian distribution according to (4), we parameterize the denoising process to be a Gaussian distribution as well with $p_\theta(z_{t-1}|z_t) = \mathcal{N}(z_{t-1}; \mu_\theta(z_t, t), \Sigma_\theta(z_t, t))$, so that the loss term $\mathcal{L}_{t-1}$ is readily computed with the KL Divergence of two Gaussian distributions. Thereby, after simplifications, the ultimate loss function used for optimizing our latent DM is defined as follows:

$$\mathcal{L}_{DM}(\theta) = \mathbb{E}_{z_0,t,\epsilon}[\|\epsilon - \epsilon_\theta(z_t, t)\|^2] \tag{11}$$

where $z_0 \sim q(z_0)$, $t \sim \mathcal{U}(1, T)$ and $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. ϵ_θ is the trainable neural network implemented as a U-Net backbone, and the representation at time step t is sampled by $z_t = \sqrt{\bar{\alpha}_t}z_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$. Detailed architecture of the U-Net backbone is shown in Figure 3.

Conditional generation

When gene expression profiles are used as the condition y to guide the generation, our training objective is to optimize the loss

$$\mathcal{L}_{cond\_DM}(\theta) = \mathbb{E}_{z_0,t,\epsilon,y}\left[\|\epsilon - \epsilon_\theta(z_t, y, t)\|^2\right] \tag{12}$$

Cross-attention conditioning

DMs are not only capable of generating high-quality samples, but also able to be extended with conditional generation [27]. We incorporate multi-head cross-attention mechanism [44] into the intermediate layers of U-Net backbone to condition the molecular latent code z_t at time step t with gene expression differences $y \in \mathbb{R}^{N\times d_c}$. Each head H_i can be formulated as

$$H_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{13}$$

where we can use the flattened intermediate representation of $\varphi_i(z_i) \in \mathbb{R}^{N\times d_\epsilon^i}$ as Q, and use y as K and V. $W_i^V \in \mathbb{R}^{d\times d_\epsilon^i}$ and $W_i^Q, W_i^K \in \mathbb{R}^{d\times d_c}$ are learned weights during optimizing the conditional loss function (12).

Training and sampling procedure

Steps to train GLDM are illustrated in Algorithm 1. Specifically, when training GLDM on an unconstrained generation task, gene expression input y will be neglected and the cross-attention layers in the transformer blocks in the U-Net backbone ϵ_θ will become self-attention layers.

During the inference stage, the developed latent DM and decoder are combined for generation of new samples. Details are given in Algorithm 2. Starting from a standard Gaussian noise

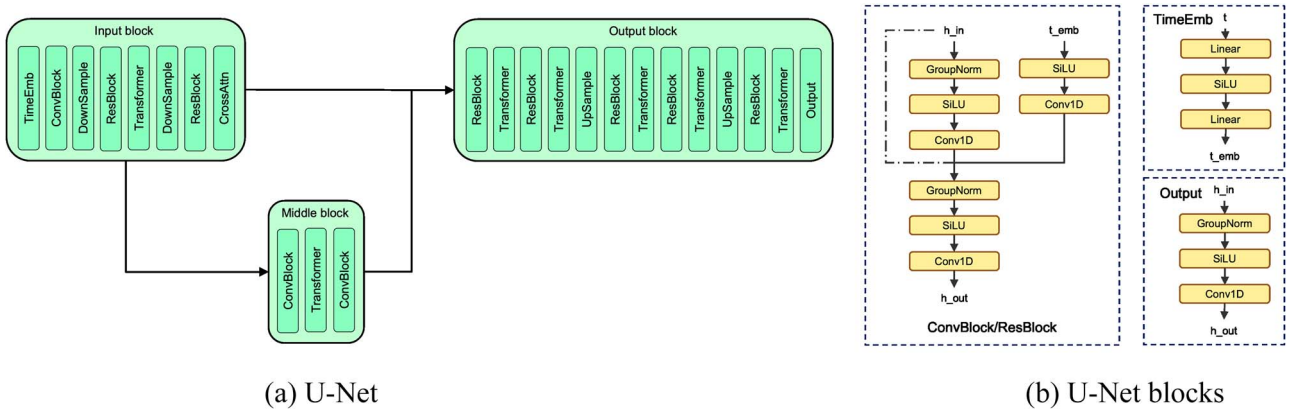

Figure 3. Architecture of the U-Net backbone. (**A**) The backbone is composed of a single-level U-Net. The ConvBlock and ResBlock are used for conditioning the latent codes with the time embedding. And Transformer blocks are used for conditioning the latent codes with the gene expression signatures. (**B**) Specific layers for the ConvBlock, ResBlock, TimeEmb and Output block. Transformer block is inherited from the typical transformer architecture with cross-attention for constrained generation and self-attention for unconstrained generation [44]. ResBlock has identical layers with ConvBlock, but have an additional connection to add the initial input before the second GroupNorm layer. DownSample and UpSample blocks consist of one 1D convolutional layer to decrease or increase the channels, respectively.

Algorithm 1 Training algorithm of GLDM

Input: Molecular graph data $x = (V, E)$, gene expression signature y
Stage 1: Train the encoder and decoder
Initial: Encoder F_ϕ, decoder G_ξ and discriminator D_ω
while ϕ, ξ, ω have not converged **do**
$z = F_\phi(x)$
$\mu, \sigma = MLP(z)$
$\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$\tilde{z} = \epsilon \odot \sigma + \mu$ ▷ Reparameterization
$\hat{x} = G_\xi(\tilde{z})$ ▷ Reconstruction
$\mathcal{L}_{AE} = \mathcal{L}_{rec}(x, \hat{x}) + \lambda_{reg}\mathcal{L}_{reg}(x, \tilde{z})$
$\phi, \xi, \omega \leftarrow \text{optimizer}(\mathcal{L}_{AE}; \phi, \xi, \omega)$
end while
Stage 2: Train the latent diffusion model
Initial: Diffusion backbone ϵ_θ
Fix encoder parameters ϕ
while θ has not converged **do**
$z_0 = F_\phi(x)$
$t \sim \mathcal{U}(1, T)$
$\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$ ▷ Diffusion process
$\mathcal{L}_{DM}(\theta) = \mathbb{E}\left[\|\epsilon - \epsilon_\theta(z_t, y, t)\|^2\right]$
$\theta \leftarrow \text{optimizer}(\mathcal{L}_{DM}; \theta)$
end while

Algorithm 2 Sampling algorithm of GLDM

Input: Standard Gaussian noise $z_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, gene expression signature y
for $t = T : 1$ **do** ▷ Denoising loop
$\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$z_{t-1} = \frac{1}{\sqrt{1-\beta_t}}(z_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(z_t, y, t)) + \sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}\epsilon$
end for
$\hat{x} = G_\xi(z_0)$ ▷ Reconstruction
return $\hat{x}$

signal, the DM is first applied to remove noise gradually after $T = 200$ steps. And the decoder is then used for reconstructing the molecular graph. For constrained generation, gene expression signature y is also fed into the diffusion backbone ϵ_θ to control the denoising procedure. For unconstrained generation, y is neglected and the generated molecules are not expected to induce potential gene expression changes.

Implementation details

We set the dimension of the latent representations to be 1×512, and the dimension of the conditioning gene signature to be 1×979, which equals to 978 genes plus 1 dosage value. We use a linear learning rate scheduler to gradually increase the learning rate from $1e-6$ to $1e-3$. The batch size used for training the LDM is 50 or 100. The other hyperparameter configurations are shown in Table 1. Eventually, we use the GLDM-1 configuration to report all the results. Channels refer to the number of channel in the first ConvBlock. Channel multipliers control how the number of channel multiplies or divides in the subsequent ConvBlock or ResBlock. Attention downsampling factor affects when a Transformer block is appended after the ResBlock. Head channels reflect the number of channels assigned to each head in the multi-head attention layer. The number of attention heads equals to current channel number divided by head channel number.

RESULTS

Datasets

Two generative chemistry databases were used to sustain this study: ChEMBL [45] and LINCS [46]. ChEMBL database collects bioactive molecules with drug-like properties together with bioactivity and genomics data. LINCS database is a program facilitating network-based apprehension of biology by cataloging changes in various cellular processes when cells are exposed to certain perturbagens. LINCS L1000 project collects gene expression profiles of cell lines for thousands of perturbagens such as drugs at different time points. Both databases aim at bridging the relationship between chemistry and biology and boosting application of genomics profiles in novel drug discovery.

Table 1: Hyperparameters of GLDM

	GLDM-1	GLDM-2	GLDM-3	GLDM-4	GLDM-5	GLDM-6
Channels	64	64	256	256	64	64
Attention downsampling factors	4, 2	8, 4, 2	4, 2	8, 4, 2	4, 2	8, 4, 2
Channel multipliers	1, 2, 3	1, 2, 3, 4	1, 2, 3	1, 2, 3, 4	1, 2, 3	1, 2, 3, 4
Number of ResNet blocks	1	1	1	1	2	2
Head channels	8	8	32	32	8	8

In this study, we first pretrain our encoder and decoder on a dataset curated by GuacaMol [28] from ChEMBL database. This dataset contains ˜1.5M drug-like molecules and is large enough for our AE models to learn a general latent distribution of drug-like structures. Then we develop LDMs on the latent space encoded by AE models, and compare their performances with plain AEs and baseline models by evaluating on the GuacaMol benchmarks. Then we perform gene expression conditioned generation on a dataset curated by [9] from the L1000 project. This dataset consists of only ˜5.2k molecules, which we believe is insufficient to cover general drug distributions. Therefore, we apply transfer learning to fine-tune our pretrained AE models on L1000 dataset, and develop the LDM models to generate hit-like drug candidates inducing target gene expression signatures.

Unconstrained generation

In this section, the developed and baseline models were evaluated on the GuacaMol distribution learning benchmarks [28]. In particular, five benchmarks were evaluated, i.e., validity, uniqueness, novelty, KL Divergence and Frechet Chemnet Distance (FCD) [47]. Validity assesses if the generated molecules are valid. Uniqueness evaluates if the products of generation correspond to different molecules. Novelty assesses if the model is able to generate novel molecules that are never present in the training set. KL Divergence and FCD measure if the generated molecule set has captured the probability distribution of the training set in terms of a variety of physicochemical properties or chemical and biological features. All the five measurements are converted by GuacaMol benchmark into scores ranging from 0 to 1, which is the higher the better. 10k molecules are generated to evaluate these benchmarks.

Comparison with existing methods

To demonstrate our proposed approach, we consider 4 generative models as baselines: JT-VAE [48], MoLeR [16], GDSS [49] and DiGress [50]. Notably, JT-VAE and MoLeR are VAE models, and GDSS and DiGress are DMs. We directly use the JT-VAE model pretrained on the ChEMBL dataset to evaluate the GuacaMol distribution learning benchmarks without retraining. We transform MoLeR implementation from Tensorflow to PyTorch to keep the framework consistent with other models. GDSS is not initially developed on the GuacaMol dataset, so we train the model with default hyperparameters on this dataset for a fair comparison. For DiGress, we directly use the scores reported in their work.

In Table 2, the first four rows correspond to the four baseline models, and the last row reports the performance of our GLDM. As shown, JT-VAE fails to generate unique and novel molecules. Despite that MoLeR and GDSS are able to achieve high uniqueness and novelty scores, they perform poorly with regard to KL Divergence and FCD metrics. Only GLDM and DiGress perform decently in KL Divergence and FCD evaluations, whereas DiGress fails to generate 100% valid molecules. This demonstrates that by applying diffusion processes in the latent space, our model is able to capture similar diversity and property distributions with the original training molecule set, thus generate versatile molecules with higher potential to become drugs maintaining chemical validity.

Effect of loss functions on encoder

As discussed in Section 3.2.1, the regularization loss $\mathcal{L}_{reg}$ is implemented as the Wasserstein loss $\mathcal{L}_{WAE}$ (9). To demonstrate the rationale of using this loss function, we designed an ablation experiment to compare the performance of GLDMs using various regularization loss functions. Besides the Wasserstein loss, we considered the KL Divergence [20] and the adversarial loss [21].

KL Divergence, which is commonly used in VAE model training, is formulated as

$$\mathcal{L}_{\text{VAE}}(\mathrm{x}) = -\mathcal{D}_{KL}(q_\theta(\mathrm{z}|\mathrm{x})\|p(\mathrm{z})) \tag{14}$$

where q_θ represents the encoder posterior distribution and $p(\mathrm{z})$ is the prior Gaussian distribution.

The adversarial loss is defined as

$$\mathcal{L}_{\text{GAN}}(\mathrm{x}) = \mathbb{E}_{z\sim p(\mathrm{z})} \log D(\mathrm{z}) + \mathbb{E}_{x\sim p_d(\mathrm{x})} \log(1 - D(\mathrm{E}(\mathrm{x}))) \tag{15}$$

where D is the discriminator function, and $\mathrm{x} \sim p_d(\mathrm{x})$ and $\mathrm{z} \sim p(\mathrm{z})$ are identical to (9).

As shown in Table 3, the three variants of our model did not differ much in terms of validity, uniqueness and novelty. However, GLDM with the GAN loss performed slightly better than VAE and WAE losses on KL Divergence and FCD.

Constrained generation of hit molecules

In this section, we present the generation results of GLDM conditioned on anticipated gene expression profiles. In Section 4.2, we developed our DM on the top of the autoencoder model pretrained on the GuacaMol dataset containing millions of drug-like molecules. Now we apply transfer learning to fine-tune the pretrained autoencoder model on the L1000 dataset to maintain the model's capability of encoding general drug-like molecules and adjust it to fit the specific distributions of target dataset. Then we train conditional LDM by optimizing (12) and evaluate the generation conditioned on gene signatures.

In the L1000 dataset, each gene signature or gene expression profile is composed of expression values of 978 Landmark genes. Landmark gene expression can be used for inferring expression of another ˜12k genes and thus depicting the overall transcriptome [46]. Then we calculate the differences between the gene signatures of treated cell lines and untreated control ones, and combine each of the gene expression change with the dosage value of the treatment to form the condition vector $y \in \mathbb{R}^{1\times 979}$ to train GLDM. For evaluation, we first filtered the L1000 dataset with

Table 2: GuacaMol distribution learning scores of baselines, i.e., JT-VAE, MoLeR, GDSS and DiGress, and our model, GLDM

Model	Validity (↑)	Uniqueness (↑)	Novelty (↑)	KL Divergence (↑)	FCD (↑)
JT-VAE	1.0	0.376	0.626	0.320	0.005
MoLeR	1.0	0.955	0.968	0.471	0.087
GDSS	1.0	0.996	0.991	0.673	0.020
DiGress	0.852	1.0	0.999	0.929	0.680
GLDM (Ours)	1.0	0.999	0.997	0.926	0.424

Table 3: Ablation study on GuacaMol distribution learning benchmarks

GLDM Model	Validity (↑)	Uniqueness (↑)	Novelty (↑)	KL Divergence (↑)	FCD (↑)
$\mathcal{L}_{reg} = \mathcal{L}_{VAE}$	1.0	0.999	0.993	0.798	0.248
$\mathcal{L}_{reg} = \mathcal{L}_{GAN}$	1.0	0.999	0.997	0.926	0.424
$\mathcal{L}_{reg} = \mathcal{L}_{WAE}$	1.0	0.999	0.995	0.870	0.295

gene signatures of MCF7 or VCAP cell lines treated by drug-like compounds at 5 or 10 μM after exposure of 24 hours. Then only the gene signatures perturbed by active molecules, whose target is among nine famous protein targets (i.e., AKT1, AKT2, EGFR, AURKB, HDAC1, SMAD, MTOR, PIK3CA, TP53), were kept, following [10]. ExCAPE database [51] was used for protein target retrieval. During inference, 100 molecules are generated for each of the 332 gene expression differences, which are caused by treatments of drug-like compounds serving as inhibitors of pharmaceutically interested genes.

Among existing works [9, 10, 31, 32] that constrain molecule generation with gene expression signatures, only BiAAE [9] and TRIOMPHE [31] provided their implementations and were developed on the L1000 dataset. However, after scrutinizing TRIOMPHE, we realized their workflow is to find an existing drug-like compound that induces the most similar gene expression signature with the desired signature, then build novel molecules on the top of the existing compound. Unlike our method, the input to TRIOMPHE model is in fact the structure of the existing compound, but not the desired gene expression profile. Therefore, we only considered BiAAE as the baseline method for comparison in the following experiments.

Structural similarity evaluation

Similarity search is a well-known approach in drug repurposing and drug discovery by comparing the similarity between gene expression signatures of drugs or between gene expression signatures induced by potentially active molecules and a knocked-out target protein. In this research, we use similarity search as the baseline to demonstrate our proposed approach. Comparisons are made by calculating structural similarity scores. Specifically, the similarity scores between the generated molecules conditioned on the target gene expression changes and the active drugs for the target extracted from the ExCAPE database [51] are calculated as the metrics of our model. Then we compute the euclidean and cosine distances between the target gene expression changes and the compound-induced gene expression signatures in the dataset, and select the nearest neighbour molecules, whose structural similarity scores with the active drugs are used as baselines to evaluate our models.

In particular, we measured Fraggle similarity, and Tanimoto similarity with MACCS Fingerprints and Morgan Fingerprints as the metrics for evaluation. The metrics are implemented using RDKit [52], and shown in Figure 4. We use a one-sided Mann–Whitney U-test to examine if the similarity scores obtained by our models are significantly greater than the baselines. Compared with similarity search, GLDMs using all kinds of loss functions are always able to generate molecules with significantly higher similarity (P-value < 0.001). However, our method's improvement over BiAAE is not statistically significant (P-value ≥ 0.001).

Synthetic accessibility and quantitative estimate of drug-likeness

We computed the Synthetic accessibility (SA) score [53] for molecules produced by GLDM and the baseline model, and compare the results with the original dataset. There are ˜5k molecules in the L1000 dataset, and we generated 100 molecules for each of the 332 target gene expression changes. Since the number of samples varies to a great extent, we calculated the percentage of easily synthesized molecules to perform a fair comparison. Notably, the SA score ranges from 1 (easy to synthesize) to 10 (difficult to synthesize). Thus, we consider a SA score ≤ 4.5 as the criterion of easily synthesized.

As shown in Figure 5, molecules from the L1000 dataset exhibit high rates of easily synthesized. We observe that with VAE or WAE loss, percentage of easily synthesized molecules generated by GLDM approximates the original dataset and outperforms BiAAE. BiAAE generates SMILES rather than the molecule structures, and around 34.7% of the generated SMILES strings violated the chemical rules. We considered such samples as unsynthesizable (SA score = 10). However, GLDM with GAN loss performs much worse. By scrutinizing the generated samples, we find GLDM with GAN loss produces complicated structures such as large rings and long chains, which make the molecules difficult to synthesize.

Quantitative estimate of drug-likeness (QED) score is a criterion to evaluate the molecular properties of drug candidates. It integrates the assessment of 8 commonly used properties, including number of structural alerts, octanol–water partition coefficient, number of aromatic rings, etc. All these properties are closely relevant to drug attrition. In order to identify possible attrition in advance and prevent failures after devoting huge resources to clinical trials, we intend to evaluate our approach with QED score. Specifically, average QED scores are computed for all tasks and baselines.

In Figure 6, we find that, consistent with our observations on SA scores, GLDM with the VAE loss achieves the QED score higher

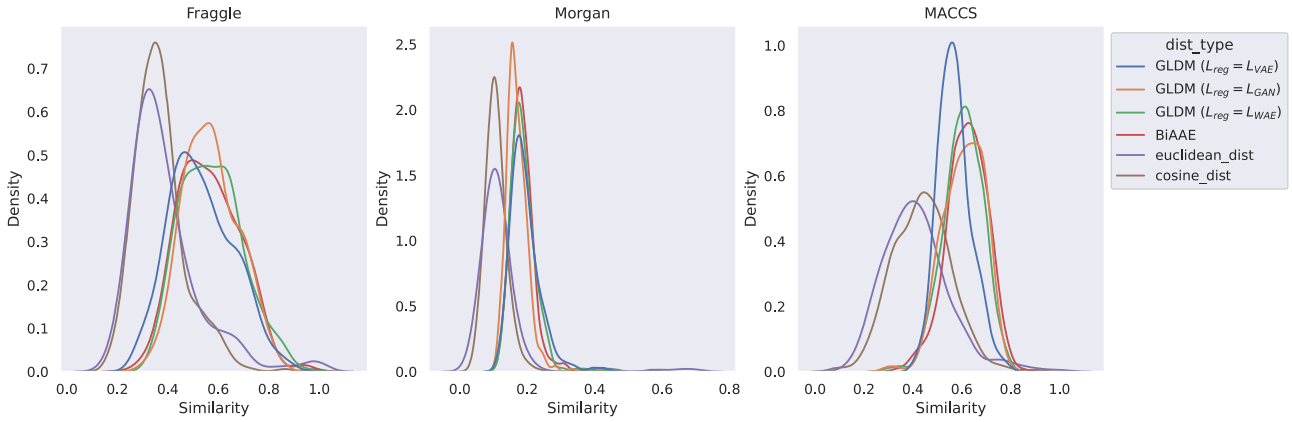

Figure 4. Distribution of structural similarity scores of Fraggle similarity, and Tanimoto similarity with MACCS Fingerprints and Morgan Fingerprints.

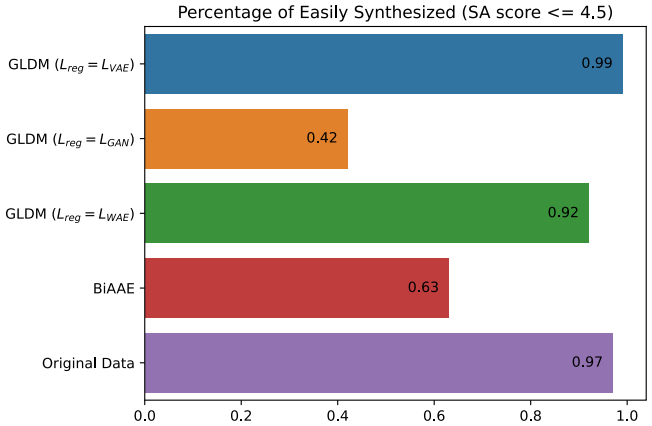

Figure 5. Percentage of easily synthesized molecules.

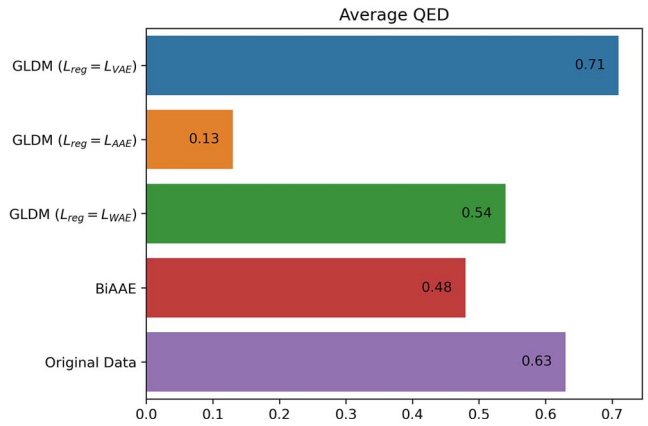

Figure 6. Average QED scores.

than the original dataset. GLDM with the WAE loss is the second best model, which is lower than the original data but higher than the BiAAE baseline. Similar to SA score computation, we consider invalid samples generated by BiAAE as zero drug-likeness (QED score = 0). GLDM with the GAN loss still performs the worst. Both SA scores and QED scores indicate that using the GAN loss causes the model to generate molecules that are less potential to become drug candidates.

Binding affinity

Previous studies [9, 10, 31, 32] only evaluated structural similarity, which is a commonly used metric to estimate efficacy of potential drug candidates. However, similar structures do not necessarily lead to similar biological functions. Moreover, if we only assess models using structural similarity, the best performance will be achieved by generating existing structures rather than new ones, which is contradictory to our aim to discovering novel drugs. Since the molecules are generated using gene expression changes caused by the reference active molecules, we hypothesize that the generated and the reference molecules should induce similar biological activities, which means they should have the same protein targets. Thus, we believe it is more sensible to test this hypothesis by assessing the binding affinity between the generated molecules and their potential protein targets.

Table 4: Information about the evaluated binding pose structures

PDB id	Target protein	Ligand name
4HJO	EGFR	Erlotinib
1M17	EGFR	Erlotinib
2ITO	EGFR	Gefitinib
2ITY	EGFR	Gefitinib
2ITZ	EGFR	Gefitinib
7ZYN	EGFR	–
5X2K	EGFR	–
6HSK	HDAC8	Quisinostat
4LXZ	HDAC2	Vorinostat
7ZZS	HDAC2	Vorinostat

In BindingDB [54], we searched for active molecules whose binding structures are clearly recorded in Protein Data Bank (PDB) [55], and 5 active molecules with 10 binding poses were finally selected for evaluation. The PDB id, target protein and ligand name of the selected binding poses are shown in Table 4. We generated 100 molecules using the gene expression changes caused by the 5 reference molecules. Gnina [56] was leveraged to perform the molecular docking experiments and predict the binding affinity scores of the generated molecules. Figure 7 displays the distribution of Vina scores when docking generated molecules with known protein pockets. It is evident that GLDMs with the GAN loss and the WAE loss achieve lower Vina scores in all the ten docking experiments, indicating better binding affinity, whereas GLDM with the VAE loss performs much worse. However, as observed in Section 4.3.2, the GAN loss induces our model to generate large molecules with complex structures. Since large molecules are inclined to achieve high binding affinity in the molecular docking experiments, we cannot fairly demonstrate GLDM with the GAN loss generates potential drug candidates. On

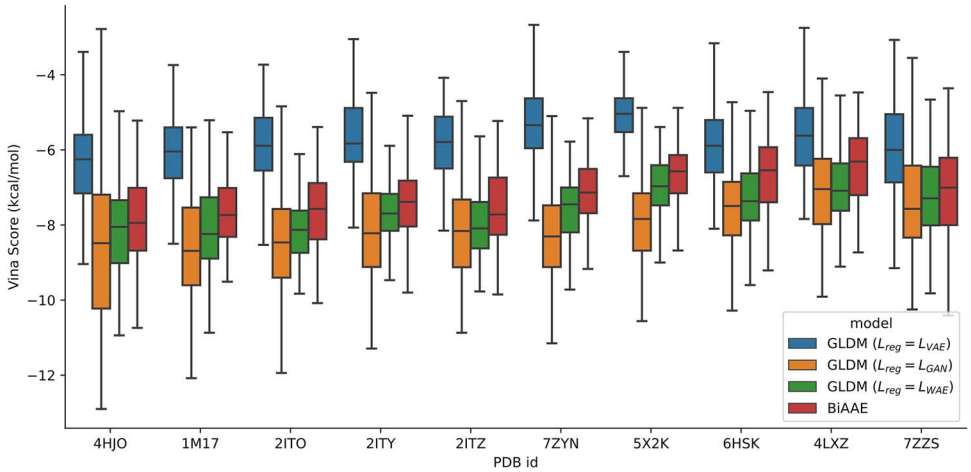

Figure 7. Vina score distribution of our model (GLDM) and baseline model (BiAAE). Outliers are removed.

Table 5: Ratio of generated molecules with better binding affinity and Vina scores of our model and baselines

PDB id	High affinity ratio (%) (↑)		Binding score (kcal/mol) (↓)		
	BiAAE	GLDM	Reference	BiAAE	GLDM
4HJO	64.0	73.8	−7.36	−10.74	−10.94
1M17	73.0	76.4	−7.22	−9.51	−10.82
2ITO	27.0	40.8	−8.33	−10.08	−11.07
2ITY	48.0	60.2	−7.48	−9.80	−10.02
2ITZ	16.0	31.6	−8.48	−9.85	−9.77
7ZYN	25.0	39.8	−7.74	−9.17	−10.30
5X2K	11.0	16.3	−7.80	−9.13	−9.76
6HSK	44.0	74.0	−6.72	−9.21	−9.60
4LXZ	52.0	77.3	−6.27	−8.73	−9.68
7ZZS	41.0	48.5	−7.31	−11.18	−9.82

the contrary, GLDM with the WAE loss not only achieved high binding affinity with even smaller range of Vina scores, but also retain decent SA and QED scores. We therefore conclude that GLDM with the WAE loss is potential for hit candidate discovery.

Furthermore, we evaluated the ratio of molecules that exhibit lower binding scores than the original reference molecules. For brevity, we only consider GLDM with the WAE loss. In Table 5, GLDM always generated a significant amount of molecules that have better binding affinities than reference compounds, and performed better than BiAAE in all assessed docking experiments except 7ZZS. In addition, the Vina score achieved by reference molecules and the lowest Vina score achieved by BiAAE and our model are also shown in Table 5. Figure 8 presents generated molecules including the only case (7ZZS) that BiAAE generates a better molecule.

CONCLUSION AND DISCUSSION

In this study, we presented GLDM, a latent DM to generate hit-like drug molecules that are able to induce desired biological activities. We used an autoencoder model made out of graph neural networks to obtain latent codes of molecular graphs, and developed a DM in the latent space to produce molecular representations. Gene expressions were added as constraints to GLDM to generate molecules with desired biological activity. Our experiments demonstrate that GLDM achieves state-of-the-art results on both unconstrained and constrained generation tasks.

Previous studies directly developed their models on a small dataset targeting at the constrained generation task. However, the widely used LINCS L1000 dataset does not contain sufficient drug-like molecules so that their models are not able to learn a comprehensive distribution of potential drug structures. This means the generated molecules are limited to a small portion of possible structures which are similar with existing drugs. Although similar structures sometimes lead to similar biological activities, we might lose the chance to discover brand new drugs. Therefore, we start our model development with the much larger GuacaMol dataset. GLDM exhibits superior GuacaMol distribution learning scores to baseline models, meaning it is capable of generating valid, novel and versatile drug-like molecules.

Then we apply transfer learning to finetune the model on L1000 dataset. Constrained on desired gene expression changes, the high structural similarity between generated molecules and known active compounds further indicates that GLDM can discover hit candidates. Moreover, compared with the baseline model, our model produced molecules with higher binding affinity while retaining reasonable SA and QED scores. We believe GLDM will be a promising approach to expedite drug discovery process.

Key Points

- GLDM adapts diffusion operations in the latent space modelled by a graph encoder, which generates high-quality molecules in an efficient way.
- GLDM leverages cross attention mechanism during the denoising process of the Diffusion Model to constrain the

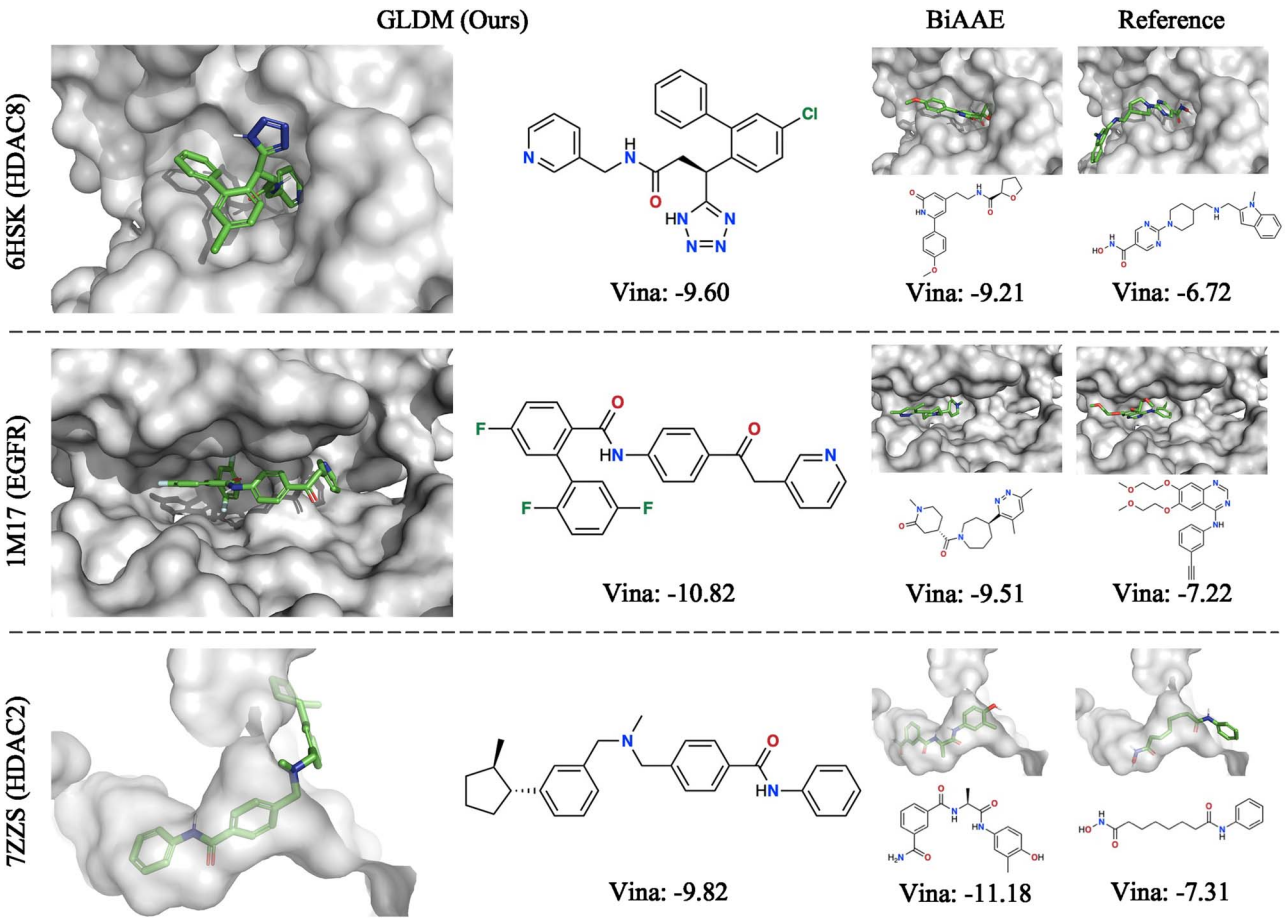

Figure 8. Example molecules with high predicted binding affinity generated by GLDM and BiAAE, together with the reference molecule. Vina score is reported under each subfigure.

generation with gene expression profiles, thus producing hit molecules with desired biological activities.
- GLDM achieved state-of-the-art performance in GuacaMol distribution learning tasks, indicating our model is able to generate valid and versatile molecule samples.
- Constrained by desired transcriptomic profiles, GLDM generated hit candidates that exhibited high structural similarity with known active drugs and high binding affinity with potential protein targets.

FUNDING

This research was supported by by AcRF Tier-1 grant RG14/23 of Ministry of Education, Singapore.

AUTHOR CONTRIBUTIONS

C.W., O.H.H. and J.C.R. conceived the experiment(s); C.W. and O.H.H. conducted the experiment(s); C.W., J.C.R. and S.C. analysed the results; and C.W. and J.C.R. wrote and reviewed the manuscript.

DATA AND CODE AVAILABILITY

The data underlying unconstrained and constrained generation are available in the GuacaMol repository and the L1000 dataset, respectively. The protein structure data for binding affinity validation is downloaded from the PDB database using the identifier provided in Table 4. Codes of our approach are available in our GitHub repository

REFERENCES

1. Wouters OJ, McKee M, Luyten J. Estimated research and development investment needed to bring a new medicine to market, 2009-2018. *JAMA* 2020;**323**(9):844–53.
2. Gómez-Bombarelli R, Wei JN, Duvenaud D, *et al.* Automatic chemical design using a data-driven continuous representation of molecules. *ACS Cent Sci* 2018;**4**(2):268–76.
3. Blaschke T, Olivecrona M, Engkvist O, *et al.* Application of generative autoencoder in de novo molecular design. *Mol Inform* 2018;**37**(1–2):1700123.
4. Guimaraes GL, Sanchez-Lengeling B, Outeiral C, *et al.* Objective-Reinforced Generative Adversarial Networks (ORGAN) for sequence generation models. arXiv preprint arXiv:1705.10843. 2017. https://arxiv.org/abs/1705.10843.
5. Kadurin A, Nikolenko S, Khrabrov K, *et al.* DruGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Mol Pharm* 2017;**14**(9):3098–104.
6. Sattarov B, Baskin II, Horvath D, *et al.* De novo molecular design by combining deep autoencoder recurrent neural networks with generative topographic mapping. *J Chem Inf Model* 2019;**59**(3):1182–96.

7. Polykovskiy D, Zhebrak A, Vetrov D, *et al*. Entangled conditional adversarial autoencoder for de novo drug discovery. *Mol Pharm* 2018;**15**(10):4398–405.
8. Mohammadi S, O'Dowd B, Paulitz-Erdmann C, Goerlitz L. Penalized variational autoencoder for molecular design. *ChemRxiv* 2019. https://doi.org/10.26434/chemrxiv.7977131.v1.
9. Shayakhmetov R, Kuznetsov M, Zhebrak A, *et al*. Molecular generation for desired transcriptome changes with adversarial autoencoders. *Front Pharmacol* 2020;**11**:269.
10. Méndez-Lucio O, Baillif B, Clevert D-A, *et al*. De novo generation of hit-like molecules from gene expression signatures using artificial intelligence. *Nat Commun* 2020;**11**(1):10.
11. Weininger D. Smiles, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J Chem Inf Comput Sci* 1988;**28**(1):31–6.
12. Segler MHS, Kogej T, Tyrchan C, Waller MP. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent Sci* 2018;**4**(1):120–31.
13. Winter R, Montanari F, Noé F, Clevert D-A. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem Sci* 2019;**10**(6):1692–701.
14. Simonovsky M, Komodakis N. GraphVAE: towards generation of small graphs using variational autoencoders. In: *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4–7 2018, Proceedings, Part I 27*, Springer, Cham, 2018, p. 412–22.
15. Kipf TN, Welling M. Variational graph auto-encoders. arXiv preprint arXiv:1611.07308. 2016. https://arxiv.org/abs/1611.&break;07308.
16. Maziarz K, Jackson-Flux H, Cameron P, *et al*. Learning to extend molecular scaffolds with structural motifs. arXiv preprint arXiv:2103.03864. 2021. https://arxiv.org/abs/2103.03864.
17. Celikyilmaz A, Clark E, Gao J. Evaluation of text generation: a survey. arXiv preprint arXiv:2006.14799. 2020. https://arxiv.org/abs/2006.14799.
18. Fedus W, Goodfellow I, Dai AM. MaskGAN: better text generation via filling in the. arXiv preprint arXiv:1801.07736. 2018. https://arxiv.org/abs/1801.07736.
19. Li Z, Wu X, Xia B, *et al*. A comprehensive survey on data-efficient GANs in image generation. arXiv preprint arXiv:2204.08329, 2022. https://arxiv.org/abs/2204.08329.
20. Kingma DP, Welling M. Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114. 2013. https://arxiv.org/abs/1312.6114.
21. Goodfellow I, Pouget-Abadie J, Mirza M, *et al*. Generative Adversarial Networks. *Commun ACM* 2020;**63**(11):139–44.
22. Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. *Adv Neural Inf Process Syst* 2020;**33**:6840–51.
23. Brock A, Donahue J, Simonyan K. Large scale GAN training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096. 2018. https://arxiv.org/abs/1809.11096.
24. Karras T, Laine S, Aila T. A style-based generator architecture for Generative Adversarial Networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, p. 4401–10.
25. Kingma D, Salimans T, Poole B, Ho J. Variational Diffusion Models. *Adv Neural Inf Process Syst* 2021;**34**:21696–707.
26. Dhariwal P, Nichol A. Diffusion Models beat GANs on image synthesis. *Adv Neural Inf Process Syst* 2021;**34**:8780–94.
27. Rombach R, Blattmann A, Lorenz D, *et al*.. High-resolution image synthesis with latent Diffusion Models. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, LA, 10684–95. 2022. IEEE, New York, NY, USA.
28. Brown N, Fiscato M, Segler MHS, Vaucher AC. GuacaMol: benchmarking models for de novo molecular design. *J Chem Inf Model* 2019;**59**(3):1096–108.
29. Bian Y, Xie X-Q. Artificial intelligent deep learning molecular generative modeling of scaffold-focused and cannabinoid CB2 target-specific small-molecule sublibraries. *Cells* 2022;**11**(5):915.
30. Prykhodko O, Johansson SV, Kotsias P-C, *et al*. A de novo molecular generation method using latent vector based Generative Adversarial Network. *J Chem* 2019;**11**(1):1–13.
31. Kaitoh K, Yamanishi Y. TRIOMPHE: transcriptome-based inference and generation of molecules with desired phenotypes by machine learning. *J Chem Inf Model* 2021;**61**(9):4303–20.
32. Born J, Manica M, Oskooei A, *et al*. PaccMannRL: De novo generation of hit-like anticancer molecules from transcriptomic data via reinforcement learning. *Iscience* 2021;**24**(4):102269.
33. Phatak SS, Stephan CC, Cavasotto CN. High-throughput and in silico screenings in drug discovery. *Expert Opin Drug Discovery* 2009;**4**(9):947–59.
34. Paricharak S, Méndez-Lucio O, Ravindranath AC, *et al*. Data-driven approaches used for compound library design, hit triage and bioactivity modeling in high-throughput screening. *Brief Bioinform* 2018;**19**(2):277–85.
35. De Cao N, Kipf T. MolGAN: an implicit generative model for small molecular graphs. arXiv preprint arXiv:1805.11973. 2018. https://arxiv.org/abs/1805.11973.
36. Zhang Z, Liu Q, Wang H, *et al*. Motif-based graph self-supervised learning for molecular property prediction. *Adv Neural Inf Process Syst* 2021;**34**:15870–82.
37. Samanta B, De A, Jana G, *et al*. NeVAE: a deep generative model for molecular graphs. *J Mach Learn Res* 2020;**21**(1):4556–88.
38. Liu Q, Allamanis M, Brockschmidt M, Gaunt A. Constrained graph variational autoencoders for molecule design. *Adv Neural Inf Process Syst* 2018;**31**.
39. Xu M, Yu L, Song Y, *et al*. GeoDiff: a geometric Diffusion Model for molecular conformation generation. arXiv preprint arXiv:2203.02923. 2022. https://arxiv.org/abs/2203.02923.
40. Brockschmidt M. GNN-FiLM: graph neural networks with feature-wise linear modulation. In: *International Conference on Machine Learning*, Vienna, Austria: PMLR, 2020, p. 1144–52.
41. Makhzani A, Shlens J, Jaitly N, *et al*. Adversarial autoencoders. arXiv preprint arXiv:1511.05644. 2015.
42. Arjovsky M, Chintala S, Bottou L. Wasserstein Generative Adversarial Networks. In: *International Conference on Machine Learning*. Sydney, NSW, Australia: PMLR, 2017, p. 214–23.
43. Gulrajani I, Ahmed F, Arjovsky M, *et al*. Improved training of Wasserstein GANs. *Adv Neural Inf Process Syst* 2017;**30**.
44. Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. *Adv Neural Inf Process Syst* 2017;**30**.
45. Gaulton A, Hersey A, Michał Nowotka A, *et al*. The ChembI database in 2017. *Nucleic Acids Res* 2017;**45**(D1):D945–54.
46. Duan Q, Reid SP, Clark NR, *et al*. L1000CDS2: LINCS L1000 characteristic direction signatures search engine. *NPJ Syst Biol Appl* 2016;**2**(1):1–12.
47. Preuer K, Renz P, Unterthiner T, *et al*. Fréchet ChemBINet distance: a metric for generative models for molecules in drug discovery. *J Chem Inf Model* 2018;**58**(9):1736–41.
48. Jin W, Barzilay R, Jaakkola T. Junction tree variational autoencoder for molecular graph generation. In: Dy J, Krause A (eds). *Proceedings of the 35th International Conference on Machine Learning*, vol. **80** of *Proceedings of Machine Learning Research*, Stockholmsmässan, Stockholm, Sweden: PMLR, 10–15 Jul 2018, p. 2323–2332.

49. Jo J, Lee S, Hwang SJ. Score-based generative modeling of graphs via the system of stochastic differential equations. In: *International Conference on Machine Learning*, Baltimore, MD: PMLR, 2022, p. 10362–83.
50. Vignac C, Krawczuk I, Siraudin A, *et al.* DiGress: discrete denoising diffusion for graph generation. arXiv preprint arXiv: 2209.14734. 2022. https://arxiv.org/abs/2209.14734.
51. Sun J, Jeliazkova N, Chupakhin V, *et al.* ExCAPE-DB: an integrated large scale dataset facilitating big data analysis in chemogenomics. *J Chem* 2017;**9**:1–9.
52. Landrum G. *RDKit: Open-Source Cheminformatics Software*. 2016. http://www.rdkit.org/, https://github.com/rdkit/rdkit.
53. Ertl P, Schuffenhauer A. Estimation of Synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J Chem* 2009;**1**:8.
54. Gilson MK, Liu T, Baitaluk M, *et al.* BindingDB in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Res* 2016;**44**(D1): D1045–53.
55. Berman HM, Westbrook J, Feng Z, *et al.* The protein data bank. *Nucleic Acids Res* 2000;**28**(1):235–42.
56. McNutt AT, Francoeur P, Aggarwal R, *et al.* GNINA 1.0: molecular docking with deep learning. *J Chem* 2021;**13**(1): 1–20.