Navigating Chemical Space with Latent Flows

Guanghao Wei[*‡]
Cornell University
gw338@cornell.edu

Yining Huang[*]
Harvard University
yininghuang@hms.harvard.edu

Chenru Duan
Deep Principle, Inc.
duanchenru@gmail.com

Yue Song[†]
California Institute of Technology
yuesong@caltech.edu

Yuanqi Du[†]
Cornell University
yd392@cornell.edu

Abstract

Recent progress of deep generative models in the vision and language domain has stimulated significant interest in more structured data generation such as molecules. However, beyond generating new random molecules, efficient exploration and a comprehensive understanding of the vast chemical space are of great importance to molecular science and applications in drug design and materials discovery. In this paper, we propose a new framework, ChemFlow, to traverse chemical space through navigating the latent space learned by molecule generative models through flows. We introduce a dynamical system perspective that formulates the problem as learning a vector field that transports the mass of the molecular distribution to the region with desired molecular properties or structure diversity. Under this framework, we unify previous approaches on molecule latent space traversal and optimization and propose alternative competing methods incorporating different physical priors. We validate the efficacy of ChemFlow on molecule manipulation and single- and multi-objective molecule optimization tasks under both supervised and unsupervised molecular discovery settings. Codes and demos are publicly available on GitHub at https://github.com/garywei944/ChemFlow.

1 Introduction

Designing new functional molecules has been a long-standing challenge in molecular discovery which concerns a wide range of applications in drug design and materials discovery [48, 57]. With the increasing interest in applying deep learning models in scientific problems [60, 70], molecular design has attracted considerable attention given its massively available data and accessible evaluations. Among the developed methods, two paradigms emerge: one paradigm searches for new molecules based on combinatorial optimization approaches respecting the discrete nature of molecules; the other paradigm builds upon the success of deep generative models in approximating the molecular distribution with a given dataset and then generating new molecules from the learned models [8]. Both of the approaches have demonstrated promising results in small molecule, protein, and materials design [61, 26, 41, 69]. Despite the promise, the chemical space is tremendously large with the number of drug-like small molecule compounds estimated to be from 10^{23} to 10^{60} [3, 39]. This necessitates either more efficient searching methods or better understanding about the structure of the chemical space. Following the progress made in studying the latent structure of deep generative models (*e.g.* generative adversarial networks (GANs) [19], variational autoencoders (VAEs) [35],

---

[*]Equal Contribution.

[†]Equal Supervision.

[‡]This work was completed while the author was at Cornell University.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

and denoising diffusion models [23]) in computer vision [28, 5, 22, 38], decent efforts have recently been made in understanding the learned latent space of molecule generative models.

Initially, disentangled representation learning becomes a popular paradigm to enforce a structured and interpretable representation [9]. Specifically, each latent dimension is expected to learn a disentangled factor of variation, and tweak the latent vector along the dimension could lead to generating new samples with changes only in one molecular property. However, even if imposing such constraints in the training of molecule generative models, the models still struggle to learn meaningful disentangled factors in the early attempts [10]. In addition to constraining the model training procedure, exploring the structure of pre-trained molecule generative models is more efficient. The main approach developed is to utilize optimization approaches to discover the region in the latent space with the desired molecular property. It often trains a proxy function to map from the latent vector to the property, providing access to gradients for gradient-based optimization [40, 20, 13]. The third line of work also builds upon pre-trained models as well. It leverages one interesting finding such that the learned latent space of molecule generative models is linearly separable [18], which is also widely studied and used as a priori in computer vision [51, 50]. ChemSpace [11] develops a highly efficient approach to use linear classifiers to identify the separation boundary and considers the normal direction to the boundary as the direction of control. Nevertheless, the linear separability assumption may be too strong. It is worth noting that the first line of work does not require labels and can be trained in an unsupervised manner (referred to as unsupervised discovery), while both the second and third lines of work require access to labels to train/identify a guidance model/direction (referred to as supervised discovery).

In this paper, we propose a new framework, ***ChemFlow***, based on flows in a dynamical system to efficiently explore the latent structure of molecule generative models. Specifically, we unify previous approaches (gradient-based optimization, linear latent traversal, and disentangled traversal) under the realm of flows that transforms data density along time via a vector field. In contrast to previous linear models, our framework is flexible to learn nonlinear transformations inspired by partial differential equations (PDEs) governing real-world physical systems such as heat and wave equations. We then analyze how different dynamics may bring special properties to solve different tasks. Our framework can also generalize both supervised and unsupervised settings under the same umbrella. Particularly in the under-studied unsupervised setting, we demonstrate a structure diversity potential can be incorporated to find trajectories that maximize the structure change of the molecules (which in turn leads to property change). We conduct extensive experiments with physicochemical, drug-related properties, and protein-ligand binding affinities on both molecule manipulation and (single- and multi-objective) molecule optimization experiments. The experiment results demonstrate the generality of the proposed framework and the effectiveness of alternative methods under this framework to achieve better or comparable results with existing approaches.

2 Background

2.1 Navigating Latent Space of Molecules

The latent space $\mathcal{Z}$ of molecule generative models is often learned through an encoder function $f_\theta(\cdot)$ and a decoder function $g_\psi(\cdot)$ such that the encoder maps the input molecular structures $\boldsymbol{x} \in \mathcal{X}$ into an (often) low-dimensional and continuous space (*i.e.* latent space) while the decoder maps the latent vectors $\boldsymbol{z} \in \mathcal{Z}$ back to molecular structures x'. Note that this encoder-decoder architecture is general and can be realized by popular generative models such as VAEs, flow-based models, GANs, and diffusion models [31, 43, 6, 58, 66]. For simplicity, we focus on VAE-based methods in this paper. To traverse the learned latent space of molecule generative models, two approaches have been proposed: gradient-based optimization and latent traversal.

The gradient-based optimization methods first learn a proxy function $h(\cdot)$ parameterized by a neural network that provides the direction to traverse [67]. This can be formulated as a gradient flow following the direction of steepest descent of the potential energy function $h(\cdot)$ and discretized, as follows:

$$\begin{aligned} \mathrm{d}\boldsymbol{z}_t &= -\nabla_z h(\boldsymbol{z}_t)\mathrm{d}t \\ \boldsymbol{z}_t &= \boldsymbol{z}_{t-1} - \nabla_z h(\boldsymbol{z}_{t-1})\mathrm{d}t \end{aligned} \tag{1}$$

where we take a dynamic system perspective on the evolution of latent samples. The latent traversal approaches leverage the observation of linear separability in the learned latent space of molecule gener-

2
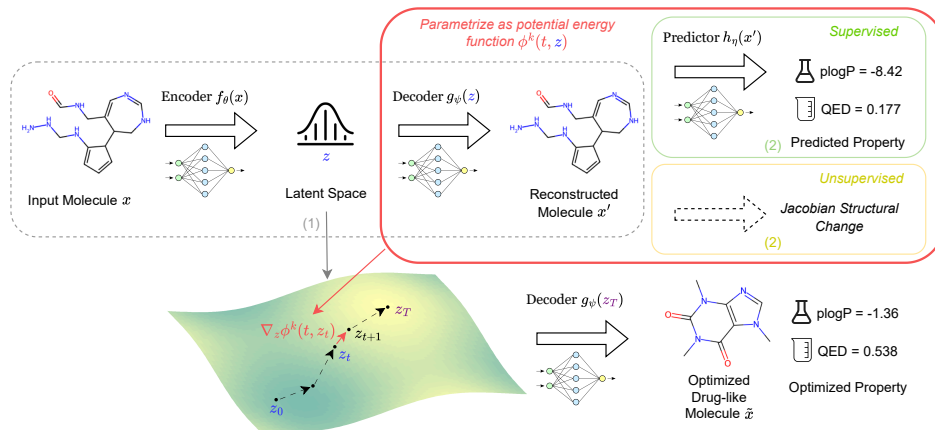
Figure 1: **ChemFlow** framework: (1) a pre-trained encoder $f_\theta(\cdot)$ and decoder $g_\psi(\cdot)$ that maps between molecules $\boldsymbol{x}$ and latent vectors $\boldsymbol{z}$, (2) we use a property predictor $h_\eta(\cdot)$ (green box) or a "Jacobian control" (yellow box) as the guidance to learn a vector field $\nabla_z \phi^k(t, \boldsymbol{z}_t)$ that maximizes the change in certain molecular properties (e.g. plogP, QED) or molecular structures, (3) during the training process, we add additional dynamical regularization on the flow. The learned flows move the latent samples to change the structures and properties of the molecules smoothly. (Better seen in color). The flow chart illustrates a case where a molecule is manipulated into a drug like caffeine.

ative models [18]. Since the direction is assumed to be linear, it can be found easily. ChemSpace [11] learns a linear classifier that defines the separation boundary of the molecular properties. Then the normal direction of the boundary provides a linear direction $\boldsymbol{n} \in \mathcal{Z}$ for traversing the latent space:

$$\boldsymbol{z}_t = \boldsymbol{z}_0 + \boldsymbol{n}t \tag{2}$$

We notice that the above gradient flow and linear traversal can be analyzed and designed from a dynamical system perspective: linear traversal can be considered as a special case of wave functions, *i.e.*, we have $\partial^2 \mathbf{z}_t / \partial^2 \mathbf{z}_{t-1} = \partial^2 \mathbf{z}_t / \partial^2 t = 0$ satisfied by wave functions. This connection inspires us to consider designing more dynamical traversal approaches in the latent space.

2.2 Wasserstein Gradient Flows

Gradient flows define the curve $\boldsymbol{x}(t) \in \mathbb{R}^n$ that evolves in the negative gradient direction of a function $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}$. The time evolution of the gradient flow is given by the ODE $\boldsymbol{x}'(t) = -\nabla \mathcal{F}(\boldsymbol{x}(t))$. Wasserstein gradient flows describe a special type of gradient flow where $\mathcal{F}$ is set to be the Wasserstein distance. For example, as introduced in Benamou and Brenier [2], the commonly used L^2 Wasserstein metric induces a dynamic formulation of optimal transport:

$$W_2(\mu, \nu)^2 = \min_{v, \rho} \left\{ \int \int \frac{1}{2} \rho(t, \boldsymbol{x}) |v(t, \boldsymbol{x})|^2 \, dt \, d\boldsymbol{x} : \partial_t \rho(t, \boldsymbol{x}) = -\nabla \cdot (v(t, \boldsymbol{x}) \rho(t, \boldsymbol{x})) \right\} \tag{3}$$

where μ, ν are two probability measures at the source and target distributions, respectively. Interestingly, if we take the gradient of a potential energy $\nabla \phi$ as the velocity field applied to a distribution, the time evolution of $\nabla \phi$ can be seen to minimize the Wasserstein distance and thus follow optimal transport. In Appendix A, we give detailed derivations of how the vector fields minimize the L^2 Wasserstein distance and discuss alternative PDEs of the density evolution recovered by Wasserstein gradient flow (*e.g.* Wasserstein gradient flow over the entropy functional recovers heat equation) following the seminal JKO scheme [34].

3 Methodology

We present **ChemFlow** as a unified framework for latent traversals in chemical latent space as latent flows. We parameterize a set of scalar-valued energy functions $\phi^k = \texttt{MLP}_{\theta^k}(t, \boldsymbol{z}) \in \mathbb{R}$ and use the

3

learned flow $\nabla_z \phi^k$ to traverse the latent samples. The traversal process can be described by the following equation in a Lagrangian way (particle trajectory):

$$z_t = z_{t-1} + \nabla_z \phi^k(t-1, z_{t-1}) \tag{4}$$

Alternatively, as an Eulerian approach, we can write the time evolution of the density through a pushforward map:

$$\rho_t = [\psi_t]_* \rho_{t-1} \tag{5}$$

where ψ_t defines the time-dependent flow that transforms the densities of latent samples through a probability path. The pushforward measure $[\psi_t]_*$ induces a change of variable formula for densities [47]:

$$[\psi_t]_* \rho_{t-1}(z) = \rho_{t-1}(\psi_t^{-1}(z)) |\det \left[\frac{\partial \psi_t^{-1}(z)}{\partial z} \right]| \tag{6}$$

In the following, we will introduce how $\nabla_z \phi^k$ is matched to some pre-defined velocities for generating different flows.

3.1 Learning Different Latent Flows

Given a pre-trained molecule generative model $g_\psi : \mathcal{Z} \to \mathcal{X}$ with prior distribution $p(z)$, we would like to model K different semantically disentangled latent trajectories that correspond to different properties of the molecules, numbered by superscript k.

Hamilton-Jacobi Flows. One desired property for the latent traversal comes from optimal transport theory such that the transport cost is minimized (i.e. shortest path). This property can be enforced by solving Eq. (3) by Karush–Kuhn–Tucker (KKT) conditions, which will give the optimal solution — the Hamilton-Jacobi Equation (HJE):

$$\frac{\partial}{\partial t} \phi^k(t, z) + \frac{1}{2} ||\nabla_z \phi^k(t, z)||^2 = 0 \tag{7}$$

where the velocity field is defined as the flow $\nabla \phi^k$. The HJE can also be interpreted as mass transportation in fluid dynamics, *i.e.*, under the velocity field $\nabla \phi^k$, the fluid will evolve to the target distribution with an optimal transportation cost.

We achieve the HJE constraint by matching our flow fields and define the boundary condition as:

$$\mathcal{L}_r = \frac{1}{T} \sum_{t=0}^{T-1} \left(\frac{\partial}{\partial t} \phi^k(t, z) + \frac{1}{2} ||\nabla_z \phi^k(t, z)||^2 \right)^2, \ \mathcal{L}_\phi = \sum_{k=0}^{K-1} ||\nabla_z \phi^k(0, z_0)||_2^2 \tag{8}$$

where T represents the total number of traversal steps, $\mathcal{L}_r$ restricts the energy to obey our physical constraints, and $\mathcal{L}_\phi$ restricts $\phi(t, z_t)$ to match the initial condition. Our latent traversal can be thus regarded as dynamic optimal transport between distributions of molecules with different properties.

Wave Flows. Alternatively, we can pivot the optimal transport property to enforce additional physical and dynamical priors. For example, if we specify the flow to follow wave-like dynamics, we can use the second-order wave equation:

$$\frac{\partial^2}{\partial t^2} \phi^k(t, z) - c^2 \nabla_z^2 \phi^k(t, z) = 0 \tag{9}$$

The above constraint empirically produces highly diverse and realistic trajectories. Our velocity matching objective and boundary condition then become:

$$\mathcal{L}_r = \frac{1}{T} \sum_{t=0}^{T-1} ||\frac{\partial^2}{\partial t^2} \phi^k(t, z) - c^2 \nabla_z^2 \phi^k(t, z)||_2^2, \ \mathcal{L}_\phi = \sum_{k=0} K - 1 ||\nabla_z \phi^k(0, z_0)||_2^2 \tag{10}$$

where $\mathcal{L}_r$ and $\mathcal{L}_\phi$ restrict the physical constraints and the initial condition, respectively. Note that $\phi^k \equiv 0$ is a trivial optimal solution for the above two objectives regarding that both $\mathcal{L}_r$ and $\mathcal{L}_\phi$ are non-negative. To prevent the parameterized ϕ^k from converging to such a trivial solution, we introduce more guidance to the loss function in Section 3.2 separately.

Alternative Flows. Besides HJE and wave equations, our framework is also general to include other commonly used PDEs that allow for different dynamics along the flow, such as Fokker Planck equation and heat equation. In the experimental section, we will explore the effectiveness of each latent flow in different supervision settings.

4

3.2 Supervised & Unsupervised Guidance

Supervised Semantic Guidance. When an explicit semantic potential energy function or labeled data for the semantic of interest is available, we can use the provided semantic potential to guide the learning of the flow. Firstly, we train a surrogate model $h_\eta : \mathcal{X} \to \mathbb{R}$ (parameterized by a deep neural network) to predict the corresponding molecular property. Then we use the trained surrogate model as guidance to learn flows that drive the increase of the property for the trajectory of the generated molecules.

$$d = \langle -\nabla_{\boldsymbol{z}} h_\eta(g_\psi(\boldsymbol{z}_t)), \nabla_{\boldsymbol{z}} \phi^k(t, \boldsymbol{z}_t) \rangle, \quad \mathcal{L}_{\mathcal{P}} = -\operatorname{sign}(d) \|d\|_2^2 \tag{11}$$

The intuition behind this objective is to learn the vector field $\boldsymbol{z}_t$ such that it aligns with the direction of the steepest descent (negative gradient) of the objective function. Note that the sign of the dot product matters as it determines minimizing or maximizing the property.

The proposed objective function in the supervised scenario is

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_\phi + \mathcal{L}_{\mathcal{P}}$$

Unsupervised Diversity Guidance. When no explicit potential energy function is provided to learn the flow, we need to define a potential energy function that captures the change of molecular properties. As molecular properties are determined by the structures, we devise a potential energy that maximizes the continuous structure change of the generated molecules. Inspired by Song et al. [54], we couple the traversal direction with the Jacobian of the generator to maximize the traversal variations in the molecular space. The perturbation on latent samples can be approximated by the first-order Taylor approximation:

$$g(\boldsymbol{z}_t + \epsilon \nabla_{\boldsymbol{z}} \phi^k(t, \boldsymbol{z}_t)) = g(\boldsymbol{z}_t) + \epsilon \frac{\partial g(\boldsymbol{z}_t)}{\partial \boldsymbol{z}_t} \nabla_{\boldsymbol{z}} \phi^k(t, \boldsymbol{z}_t) + R_1(g(\boldsymbol{z}_t)) \tag{12}$$

where ϵ denotes perturbation strength, and $R_1(\cdot)$ is the high-order terms. In the unsupervised setting, for sufficiently small ϵ, if the Jacobian-vector product can cause large variations in the generated sample, the direction is likely to correspond to certain properties of molecules. We therefore introduce such a Jacobian-vector product guidance:

$$\mathcal{L}_{\mathcal{J}} = -\left\| \frac{\partial g(\boldsymbol{z}_t)}{\partial \boldsymbol{z}_t} \nabla_{\boldsymbol{z}} \phi^k(t, \boldsymbol{z}_t) \right\|_2^2 \tag{13}$$

Compared to the supervised setting which maximizes the change of the molecular properties, it aims to find the direction that causes the maximal change of the structures. This can in turn effectively pushes the initial data distribution to the target one concentrated on the maximum property value. The Jacobian guidance will compete with the dynamical regularization (e.g. wave-like form) on the flow to yield smooth and meaningful traversal paths.

Disentanglement Regularization. While the above formulation can encourage smooth dynamics and meaningful output variations, the flows are likely to mine identical directions which all correspond to the maximum Jacobian change. To avoid such a trivial solution, we adopt an auxiliary classifier l_γ following Song et al. [54] to predict the flow index and use the cross-entropy loss to optimize it:

$$\mathcal{L}_k = \mathcal{L}_{CE}(l_\gamma(g_\psi(\boldsymbol{z}_t); g_\psi(\boldsymbol{z}_{t+1})), k) \tag{14}$$

Where $\boldsymbol{x}_t = g(\boldsymbol{z}_t)$ is the generated sample from timestep t. We see the extra classifier guidance would encourage each flow to be independent and find distinct properties. For each target property, we compute the Pearson correlation coefficient using a randomly generated test set. This coefficient measures the correlation between the property and a natural sequence (from 1 to time step t) along the optimization trajectory. We then select the energy network that achieves the highest correlation score for optimizing molecules with that specific property.

The proposed objective function in the unsupervised scenario is

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_\phi + \mathcal{L}_{\mathcal{J}} + \mathcal{L}_k$$

3.3 Connection with Langevin Dynamics for Global Optimization

In scenarios where our flow adheres to the dynamics of the Fokker-Planck equation, our approach may also be interpreted as employing a learned potential energy function to simulate Langevin Dynamics

5

for global optimization [15]. Notably, the convergence of Langevin dynamics, particularly at low temperatures, tends to occur around the global minima of the potential energy function [7]. The continuous and discretized Langevin dynamics are as follows:

$$\mathrm{d}z_t = -\nabla_z h_\eta(z_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{w}_t$$
$$z_t = z_{t-1} - \nabla_z h_\eta(z_{t-1})\mathrm{d}t + \sqrt{2\mathrm{d}t}\mathcal{N}(0, I)$$

(15)

Proposition 3.1. *(Global Convergence of Langevin Dynamics, adapted from Gelfand and Mitter [16]). Given a Langevin dynamics in the form of*

$$z_t = z_{t-1} - a_t(\nabla_z h_\eta(z_{t-1}) + u_t) + b_t \mathbf{w}_t$$

where $\mathbf{w}_t$ is a d-dimensional Brownian motion, a_t and b_t are a set of positive numbers with $a_T, b_T \to 0$, and u_t is a set of random variables in $\mathbb{R}^n$ denoting noisy measurements of the energy function $h_\eta(\cdot)$. Under mild assumptions, z_t converges to the set of global minima of $h_\eta(\cdot)$ in probability.

Following Proposition 3.1, the learned latent flow can be used to search for molecules with optimal properties and it converges to the global minimizers of the learned latent potential energy function.

4 Experiments

4.1 Experiment Set-up

Datasets & Molecular properties.   We extract 4,253,577 molecules from the three commonly used datasets for drug discovery including **MOSES** [46], **ZINC250K** [27], and **ChEMBL** [68]. Molecules are represented by SELFIES strings [37]. All input molecules are padded to the maximum length in the dataset before fitting into the generative model. We consider a total of 8 molecular properties which include *3 general drug-related properties* — Quantitative Estimate of Drug-likeness (QED), Synthesis Accessibility (SA), and penalized Octanol-water Partition Coefficient (plogP) and *3 machine learning-based target activities* — DRD2, JNK3 and GSK3B [25] , *2 simulation-based target activities* — docking scores for two human proteins ESR1 and ACAA1. See Appendix D.3 for details.

Implementations.   We establish our framework by pre-training a VAE model that learns a latent space of molecules that can generate new molecules by decoding latent vectors from the latent space. We adapt the framework in Eckmann et al. [13] which is a basic VAE architecture with molecular SELFIES string representations and an additional MLP model as the surrogate property predictor. See Appendix D.5 for all implementation and hyper-parameter details.

Model variants.   As discussed in Section 3.1, our proposed framework is general to incorporate different dynamical priors to learn the flow. For the experiments, we consider four types of dynamics including *gradient flow (GF)*, *Wave flow (Wave, eq. (9))*, *Hamilton Jacobi flow (HJ, eq. (7))* and *Langevin Dynamics or equivalently Fokker Planck flow (LD, eq. (15))*.

For the specific molecular properties and evaluations, the readers are kindly referred to Appendix D for details. We also move qualitative evaluations to Appendix F due to space limit.

4.2 Molecule Optimization

Molecule optimization is key in drug design and materials discovery, aiming to identify molecules with optimal properties [4]. Various machine learning methods have accelerated this process [8]. Our discussion focuses on optimization within the latent space of generative models, primarily using gradient-based optimization as outlined in Section 2.1. We categorize molecule optimization into three scenarios: (1) unconstrained optimization to identify molecules with the best properties, (2) constrained optimization to find molecules with the best-expected property and similar to specific structures—a common step in the lead optimization process, and (3) multi-objective optimization to simultaneously enhance multiple properties of a molecule.

6

Table 1: **Unconstrained plogP, QED maximization, and docking score minimization.** (SPV denotes supervised scenarios, UNSUP denotes unsupervised scenarios). Boldface highlights the highest-performing generation for each property within each rank.

METHOD	PLOGP ↑			QED ↑			ESR1 DOCKING ↓			ACAA1 DOCKING ↓		
	1ST	2ND	3RD	1ST	2ND	3RD	1ST	2ND	3RD	1ST	2ND	3RD
RANDOM	3.52	3.43	3.37	0.940	0.933	0.932	-10.32	-10.18	-10.03	-9.86	-9.50	-9.34
CHEMSPACE	3.74	3.69	3.64	0.941	0.936	0.933	**-11.66**	-10.52	-10.43	-9.81	-9.72	-9.63
GRADIENT FLOW	4.06	3.69	3.54	0.944	0.941	0.941	-11.00	-10.67	-10.46	-9.90	-9.64	-9.61
WAVE (SPV)	4.76	3.78	3.71	**0.947**	0.934	0.932	-11.05	**-10.71**	**-10.68**	**-10.48**	**-10.04**	**-9.88**
WAVE (UNSUP)	**5.30**	**5.22**	**5.14**	0.905	0.902	0.978	-10.22	-10.06	-9.97	-9.69	-9.64	-9.57
HJ (SPV)	4.39	3.70	3.48	0.946	0.941	0.940	-10.68	-10.56	-10.52	-9.89	-9.61	-9.60
HJ (UNSUP)	4.26	4.10	4.07	0.930	0.928	0.927	-10.24	-9.96	-9.92	-9.73	-9.31	-9.24
LD	4.74	3.61	3.55	**0.947**	**0.947**	**0.942**	-10.68	-10.29	-10.28	-10.34	-9.74	-9.64

Baselines. For molecule optimization, we follow the same experiment procedure as in Eckmann et al. [13][1]. To ensure a fair comparison, we use the same pre-trained VAE model for all the methods. The details about the baselines are deferred to Appendix D.1. We also propose evolutionary algorithm (EA)-based latent optimization approaches in our comparison (Appendix D.7).

Unconstrained Molecule Optimization. In this study, we randomly sample 100,000 molecules from the latent space and assess the top three scores after 10 steps of optimization for each method (details in Table 1). For two specific docking scores tasks, however, only 10,000 molecules are sampled due to computational resource constraints. All methods employ a step size of 0.1 to ensure a fair comparison. Our findings reveal that the efficacy of optimization methods varies with target properties, highlighting the necessity of employing a diverse set of approaches within the optimization framework, rather than depending on a single dominant method. We also visualize some generated ligands docked into protein pockets in Figure 2.
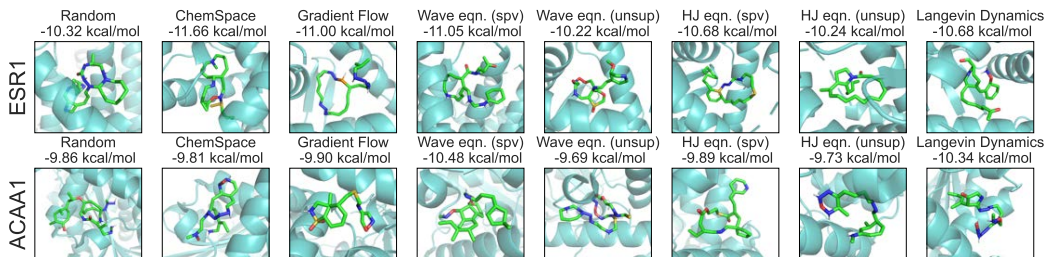
Figure 2: **Visualization of generated ligands docked against target ESR1 and ACAA1.**

Furthermore, when extending the optimization procedure to 1,000 steps, illustrated in Figure 3, Langevin dynamics significantly pushes the entire distribution to molecules with better properties, surpassing other methods in performance. Although the random direction method is effective in optimizing molecules, it does not consistently produce significant shifts in the distribution. Moreover, the outputs from ChemSpace often converge to just a few molecules, which is indicative of the challenge posed by Out-of-Distribution (OoD) generation.

Similarity-constrained Molecule Optimization. Adopting methodologies from JT-VAE [31] and LIMO [13], we select 800 molecules with the lowest partition coefficient (plogP) scores from the ZINC250k dataset. These molecules undergo 1,000 steps of optimization until convergence is achieved for all methods. The optimal results for each molecule, adhering to a predefined similarity constraint (δ), is reported as in Table 2. The structural similarity is measured using Tanimoto similarity between Morgan fingerprints with a radius of 2.

Without any similarity constraints, the unsupervised approaches significantly improve molecular properties with a high success rate. However, as the similarity constraints increase, both the magnitude

---

[1]Note that we notice there was a misalignment of normalization schemes for the plogP property in the previous literature, so we only rerun and compare with related methods that align with our normalization scheme. Details can be found in Appendix D.3.

7
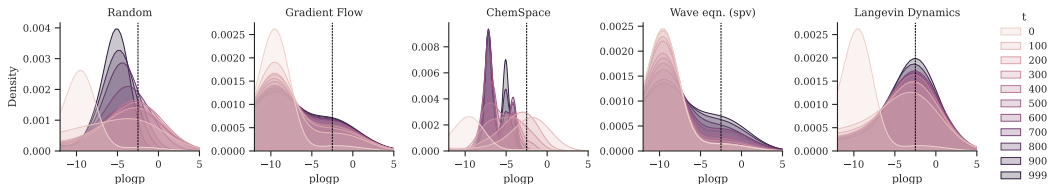
Figure 3: **Molecular property plogP distribution shifts following the latent flow path.**

Table 2: **Similarity-constrained plogP maximization.** For each method with minimum similarity constraint δ, the results in reported in format mean ± standard derivation (success rate %) of absolute improvement, where the mean and standard derivation are calculated among molecules that satisfy the similarity constraint.

Method	$\delta = 0$	$\delta = 0.2$	$\delta = 0.4$	$\delta = 0.6$
Random	11.76 ± 6.18 (99.0)	7.64 ± 6.38 (80.0)	5.03 ± 5.70 (52.1)	2.37 ± 3.71 (21.1)
ChemSpace	12.13 ± 6.41 (99.8)	9.07 ± 6.80 (90.2)	$\mathbf{7.52 \pm 6.29}$ **(59.4)**	$\mathbf{5.70 \pm 5.84}$ **(20.2)**
Gradient Flow	7.88 ± 7.28 (60.4)	7.20 ± 6.98 (56.5)	5.45 ± 6.45 (41.9)	3.60 ± 5.50 (18.4)
Wave (spv)	6.83 ± 7.15 (59.6)	5.62 ± 6.42 (54.9)	4.31 ± 5.55 (41.9)	2.47 ± 4.21 (20.6)
Wave (unsup)	19.76 ± 13.62 (99.6)	7.47 ± 9.62 (50.2)	2.06 ± 4.37 (27.3)	0.77 ± 2.21 (16.8)
HJ (spv)	8.58 ± 8.08 (68.0)	6.62 ± 7.44 (60.0)	4.27 ± 5.40 (40.6)	2.39 ± 4.10 (18.5)
HJ (unsup)	$\mathbf{20.64 \pm 12.93}$ **(98.0)**	8.57 ± 9.69 (50.1)	2.12 ± 3.55 (19.5)	0.67 ± 0.86 (8.6)
LD	12.98 ± 6.23 (99.6)	$\mathbf{9.70 \pm 6.21}$ **(94.4)**	6.14 ± 5.99 (70.9)	2.94 ± 4.34 (35.4)

and success rate of unsupervised methods decrease notably. The most considerable improvements of ChemSpace are observed when similarity constraints are set at 0.4 and 0.6. Despite this, as shown in Figure 3 and previously discussed, the generation within ChemSpace encounters significant OoD issues after extensive iterations. Among all techniques evaluated, Langevin dynamics stands out for its overall high improvements and high success rate. As further illustrated in Figure 5, Langevin dynamics also demonstrates a notably faster empirical convergence rate. We also report the performance in optimizing the QED property in Appendix D.8.

Surprisingly, we observe that the random direction performs well on molecule optimization tasks. This observation motivates us to study the structure of the latent space. We show that the molecular structure distribution on the latent space follows a high-dimensional Gaussian distribution and the random direction increases the norm of the latent vectors that have strong correlations with molecular properties. We analyze this systematically in Appendix E. It is also notable that although random directions could be effective in optimizing molecules, the distribution of the entire molecule sets being optimized does not change accordingly as shown in Figure 3.

Multi-objective Molecule Optimization. As we are learning distinct vector fields and potential energy functions for each property, they can be readily added together for multi-objective optimization [13, 11]. To generate molecules that are optimized on multiple properties, we use a similar setting as similarity-constrained molecule optimization to select 800 molecules from the ZINC250k dataset with the lowest QED and aim to generate molecules with high QED as well as low SA simultaneously. At each time step, the latent vector is optimized following the averaged direction of two corresponding flow directions. This scheme could be seamlessly generalized to k-objectives optimization. Table 3 shows that Langevin dynamics and ChemSpace achieve the best or competitive performance at all similarity cutoff levels.

4.3   Molecule Manipulation

Molecule manipulation is a relatively new task proposed in Du et al. [11] to study the performance of latent traversal methods. Specifically, the main idea of molecule manipulation is to find smooth local changes of molecular structures that simultaneously improve molecular properties which is essential to help chemists systematically understand the chemical space.

Supervised Molecule Manipulation Table 4 shows the success rate results of manipulating 1,000 randomly sampled molecules to optimize each desired property. Following Du et al. [11], we traverse

8

Table 3: **Similarity-constrained Multi-objective (QED-SA) maximization.** The value of QED and SA is scaled to both have a range from 0 to 100 for an equal-weighted sum. The method with the highest equal-weighted sum improvement of QED and SA of each structure similarity level is bolded.

	$\delta = 0$	$\delta = 0.2$	$\delta = 0.4$	$\delta = 0.6$
	QED			
Random	45.5 ± 13.3 (99.5)	20.7 ± 14.4 (81.8)	12.8 ± 10.7 (57.0)	8.0 ± 8.3 (29.5)
ChemSpace	47.0 ± 12.9 (99.6)	25.9 ± 16.7 (88.2)	**15.5 ± 13.2 (63.4)**	**9.7 ± 10.1 (31.6)**
Gradient Flow	31.9 ± 17.9 (89.9)	23.0 ± 16.4 (80.0)	14.4 ± 12.6 (59.4)	9.4 ± 9.1 (32.2)
Wave (SPV)	14.6 ± 14.4 (26.8)	12.3 ± 11.7 (24.2)	9.8 ± 10.0 (19.8)	6.8 ± 6.5 (12.4)
Wave (UNSUP)	39.0 ± 18.5 (96.1)	18.3 ± 14.0 (69.4)	10.0 ± 9.6 (43.2)	6.4 ± 7.0 (24.9)
HJ (SPV)	45.2 ± 13.7 (98.9)	22.7 ± 15.6 (84.9)	13.9 ± 12.0 (57.5)	9.3 ± 9.7 (30.2)
HJ (UNSUP)	40.6 ± 19.4 (96.8)	15.5 ± 12.8 (71.4)	9.2 ± 9.4 (46.2)	6.4 ± 7.3 (26.9)
LD	**47.0 ± 13.1 (99.6)**	**27.6 ± 16.3 (92.1)**	15.4 ± 12.5 (71.4)	9.6 ± 9.4 (40.1)
	SA			
Random	8.34 ± 8.06 (37.2)	6.70 ± 7.11 (27.0)	4.80 ± 5.73 (19.1)	2.61 ± 3.21 (10.9)
ChemSpace	7.92 ± 7.71 (42.8)	6.63 ± 6.71 (36.1)	**4.75 ± 5.45 (25.2)**	**2.89 ± 3.22 (15.4)**
Gradient Flow	9.56 ± 8.49 (44.0)	7.19 ± 6.66 (35.8)	5.27 ± 5.30 (26.6)	3.04 ± 3.62 (17.1)
Wave (SPV)	6.37 ± 6.30 (12.9)	5.77 ± 5.81 (12.4)	4.54 ± 4.51 (10.5)	3.44 ± 3.59 (7.1)
Wave (UNSUP)	15.21 ± 10.17 (89.2)	7.69 ± 6.51 (70.8)	3.92 ± 3.86 (45.9)	2.22 ± 1.97 (25.9)
HJ (SPV)	8.93 ± 8.39 (52.1)	6.69 ± 6.61 (38.8)	5.21 ± 5.72 (29.2)	3.23 ± 3.40 (18.6)
HJ (UNSUP)	16.03 ± 10.31 (91.0)	7.35 ± 6.34 (68.5)	4.22 ± 4.09 (46.8)	2.73 ± 2.85 (27.8)
LD	**11.51 ± 10.44 (69.2)**	**7.51 ± 7.41 (45.4)**	4.50 ± 4.95 (33.6)	2.75 ± 3.09 (20.1)

Table 4: **Success Rate of traversing latent molecule space to manipulate over a variety of molecular properties.** Numbers reported are strict success rate/relaxed success rate in %. (SPV denotes supervised scenarios, UNSUP denotes unsupervised scenarios). The ranking is the average between the ranking of average strict success rate and ranking of the average relaxed success rate.

	RANKING	AVERAGE	PLOGP (↑)	QED (↑)	SA (↓)	DRD2 (↑)	JNK3 (↑)	GSK3B (↑)
RANDOM-1D	9	1.42 / 6.85	6.00 / 31.60	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	2.50 / 9.50	0.00 / 0.00
RANDOM	6	0.57 / 42.3	0.00 / 32.60	0.10 / 3.20	0.40 / 8.60	0.50 / 87.10	1.50 / **81.40**	0.90 / 40.90
CHEMSPACE	3	6.17 / 22.83	5.20 / 25.00	6.00 / **18.10**	**6.80** / 26.50	3.20 / 18.00	8.60 / 25.80	**7.20** / 23.60
WAVE (UNSUP)	3	1.18 / **45.28**	0.60 / **40.30**	0.60 / 6.20	1.90 / 16.50	0.40 / 86.40	1.80 / 78.20	1.80 / **44.10**
WAVE (SPV)	7	1.85 / 8.08	0.00 / 0.20	3.40 / 12.10	4.00 / 18.60	3.50 / 17.10	0.00 / 0.20	0.20 / 0.30
HJ (UNSUP)	3	2.3 / 25.28	3.00 / 15.60	0.70 / 3.20	1.70 / 13.00	0.20 / **87.20**	4.80 / 18.70	3.40 / 14.00
HJ (SPV)	7	1.97 / 7.4	3.00 / 13.20	3.00 / 7.20	3.70 / 15.00	1.90 / 8.50	0.20 / 0.50	0.00 / 0.00
GF (SPV)	1	**7.62** / 28.78	**6.90** / 28.30	**6.60** / 16.70	6.30 / 25.10	7.10 / 36.10	**11.70** / 35.50	7.10 / 31.00
LD (SPV)	2	6.23 / 26.68	5.90 / 26.00	6.20 / 15.50	5.20 / 22.90	**6.00** / 33.30	8.40 / 33.80	5.70 / 28.60

the latent space for 10 steps in the traversal direction of each method and report the strict and relaxed success rate. Details of the definition of these metrics can be found in Appendix D.6. Among all approaches, the gradient flow achieves the highest success rates in multiple properties such that it takes the steepest descent of the surrogate model. When the step size is small enough, it is reasonable to learn a smooth path. However, the results still vary across properties.

Unsupervised Molecule Manipulation As the correspondence between specific molecular properties and learned latent flows is not explicitly given in the unsupervised scenario, we use an artificial process to mimic the use case in reality. Specifically, we learn 10 different potential energy functions representing 10 disentangled flows following Algorithm 1 using Wave equation and Hamilton-Jacobi equation and validated them on 1,000 unseen molecules. For each flow, we evaluate the properties of molecules generated along the 10-step manipulation trajectory. The learned potential energy function with the highest correlation score is selected for each property representing the learned Jacobian structural change that would most effectively optimize the corresponding property.

In Table 4, we can observe that even though it is without supervised training of traversal directions, the flow still learns meaningful directions from molecular structure to property changes. Surprisingly, the relaxed success rate of manipulating molecules for JNK3, and GSK3B in unsupervised settings are better than in supervised settings. We hypothesize that this is partially because of the training and generalization errors of the surrogate model. On the contrary, the structure change measurement does not provide supervision but is correlated with key molecular properties. We would like to point out that this is an open question in chemistry, often referred as to the structure-activity relationship [12], such that it is important to know the correspondence between structure and activity. We believe this is a promising result to demonstrate that generative models "realize" molecular property by learning from structures.

9

Among the quantitative results, it is interesting that the random direction achieves a good relaxed success rate for some properties, we argue this is because of the specific property of the learned latent space. The latent space learned by generative models tend to be smooth such that similar molecular structures are often mapped to close areas in the latent space. In Appendix E, we find that some molecular properties are highly correlated with their latent vector norms, in which a random direction always increases the norm and thus successfully manipulates a portion of molecules by chance.

5 Conclusion, Limitation and Future Work

In this paper, we propose a unified framework for navigating chemical space through the learned latent space of molecule generative models. Specifically, we formulate the traversal process as a flow that defines a vector to transport the mass of molecular distribution through time to desired concentrations (e.g. high properties). Two forces (supervised potential guidance and unsupervised structure diversity guidance) are derived to drive the dynamics. We also propose a variety of new physical PDEs on the dynamics which exhibit different properties. We hope this general framework can open up a new research avenue to study the structure and dynamics of the latent space of molecule generative models.

Limitation and future work. This work is a preliminary study on small molecules and it may be interesting to see it transfer to larger molecular systems or more specialized systems and properties. Beyond molecules, this approach has the potential to be extended to languages and other data modalities.

6 Acknowledgement

Y.D. would like to thank Ziming Liu and Kirill Neklyudov for helpful discussions.

References

[1] L. Ambrosio, N. Gigli, and G. Savaré. Gradient flows: in metric spaces and in the space of probability measures. Springer Science & Business Media, 2005.

[2] J.-D. Benamou and Y. Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. Numerische Mathematik, 84(3):375–393, 2000.

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

[4] N. Brown, M. Fiscato, M. H. Segler, and A. C. Vaucher. Guacamol: benchmarking models for de novo molecular design. Journal of chemical information and modeling, 59(3):1096–1108, 2019.

[5] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in β-vae. ArXiv, abs/1804.03599, 2018.

[6] N. D. Cao and T. Kipf. MolGAN: An implicit generative model for small molecular graphs, 2018.

[7] T.-S. Chiang, C.-R. Hwang, and S. J. Sheu. Diffusion for global optimization in rˆn. SIAM Journal on Control and Optimization, 25(3):737–753, 1987.

[8] Y. Du, T. Fu, J. Sun, and S. Liu. Molgensurvey: A systematic survey in machine learning models for molecule design. arXiv preprint arXiv:2203.14500, 2022.

[9] Y. Du, X. Guo, A. Shehu, and L. Zhao. Interpretable molecular graph generation via monotonic constraints. In Proceedings of the 2022 SIAM International Conference on Data Mining (SDM), pages 73–81. SIAM, 2022.

[10] Y. Du, X. Guo, Y. Wang, A. Shehu, and L. Zhao. Small molecule generation via disentangled representation learning. Bioinformatics, 38(12):3200–3208, 2022.

[11] Y. Du, X. Liu, N. M. Shah, S. Liu, J. Zhang, and B. Zhou. Chemspace: Interpretable and interactive chemical space exploration. Trans. Mach. Learn. Res., 2023, 2023.

10

[12] A. Z. Dudek, T. Arodz, and J. Gálvez. Computational methods in developing quantitative structure-activity relationships (qsar): a review. Combinatorial chemistry & high throughput screening, 9(3):213–228, 2006.

[13] P. Eckmann, K. Sun, B. Zhao, M. Feng, M. Gilson, and R. Yu. Limo: Latent inceptionism for targeted molecule generation. In International Conference on Machine Learning, pages 5777–5792. PMLR, 2022.

[14] T. Fu, W. Gao, C. Coley, and J. Sun. Reinforced genetic algorithm for structure-based drug design. Advances in Neural Information Processing Systems, 35:12325–12338, 2022.

[15] C. W. Gardiner et al. Handbook of stochastic methods, volume 3. springer Berlin, 1985.

[16] S. B. Gelfand and S. K. Mitter. Recursive stochastic algorithms for global optimization in r^d. SIAM Journal on Control and Optimization, 29(5):999–1018, 1991.

[17] L. Goetschalckx, A. Andonian, A. Oliva, and P. Isola. Ganalyze: Toward visual definitions of cognitive image properties. In ICCV, 2019.

[18] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. ACS central science, 4(2):268–276, 2018.

[19] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. Deep learning, volume 1. MIT Press, 2016.

[20] R.-R. Griffiths and J. M. Hernández-Lobato. Constrained bayesian optimization for automatic chemical design using variational autoencoders. Chemical science, 11(2):577–586, 2020.

[21] G. L. Guimaraes, B. Sanchez-Lengeling, C. Outeiral, P. L. C. Farias, and A. Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models. arXiv preprint arXiv:1705.10843, 2017.

[22] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris. Ganspace: Discovering interpretable gan controls. Advances in neural information processing systems, 33:9841–9850, 2020.

[23] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020.

[24] E. Hoogeboom, V. G. Satorras, C. Vignac, and M. Welling. Equivariant diffusion for molecule generation in 3d. In International conference on machine learning, pages 8867–8887. PMLR, 2022.

[25] K. Huang, T. Fu, W. Gao, Y. Zhao, Y. Roohani, J. Leskovec, C. W. Coley, C. Xiao, J. Sun, and M. Zitnik. Artificial intelligence foundation for therapeutic science. Nature chemical biology, 18(10):1033–1036, 2022.

[26] J. B. Ingraham, M. Baranov, Z. Costello, K. W. Barber, W. Wang, A. Ismail, V. Frappier, D. M. Lord, C. Ng-Thow-Hing, E. R. Van Vlack, et al. Illuminating protein space with a programmable generative model. Nature, pages 1–9, 2023.

[27] J. J. Irwin and B. K. Shoichet. Zinc- a free database of commercially available compounds for virtual screening. Journal of chemical information and modeling, 45(1):177–182, 2005.

[28] A. Jahanian, L. Chai, and P. Isola. On the" steerability" of generative adversarial networks. In International Conference on Learning Representations, 2019.

[29] A. Jahanian, L. Chai, and P. Isola. On the" steerability" of generative adversarial networks. ICLR, 2020.

[30] J. H. Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. Chemical science, 10(12):3567–3572, 2019.

[31] W. Jin, R. Barzilay, and T. Jaakkola. Junction tree variational autoencoder for molecular graph generation. ICML, 2018.

[32] W. Jin, R. Barzilay, and T. Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In International Conference on Machine Learning, 2020.

[33] J. Jo, S. Lee, and S. J. Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In International Conference on Machine Learning, pages 10362–10383. PMLR, 2022.

11

[34] R. Jordan, D. Kinderlehrer, and F. Otto. The variational formulation of the fokker-planck equation. Siam Journal on Applied Mathematics, 1996.

[35] D. P. Kingma and M. Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[36] M. Krenn, F. Hase, A. Nigam, P. Friederich, and A. Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. Machine Learning: Science and Technology, 1, 2019.

[37] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100% robust molecular string representation. Machine Learning: Science and Technology, 1 (4):045024, 2020.

[38] M. Kwon, J. Jeong, and Y. Uh. Diffusion models already have a semantic latent space. In The Eleventh International Conference on Learning Representations, 2022.

[39] C. A. Lipinski, F. Lombardo, B. W. Dominy, and P. J. Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. Advanced drug delivery reviews, 64:4–17, 2012.

[40] Q. Liu, M. Allamanis, M. Brockschmidt, and A. Gaunt. Constrained graph variational autoencoders for molecule design. Advances in neural information processing systems, 31, 2018.

[41] H. Loeffler, J. He, A. Tibo, J. P. Janet, A. Voronov, L. Mervin, and O. Engkvist. Reinvent4: Modern ai–driven generative molecule design. 2023.

[42] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In International Conference on Learning Representations, 2017.

[43] K. Madhawa, K. Ishiguro, K. Nakago, and M. Abe. Graphnvp: An invertible flow model for generating molecular graphs. arXiv preprint arXiv:1905.11600, 2019.

[44] D. Misra. Mish: A self regularized non-monotonic activation function. In British Machine Vision Conference, 2020.

[45] W. Peebles, J. Peebles, J.-Y. Zhu, A. Efros, and A. Torralba. The hessian penalty: A weak prior for unsupervised disentanglement. In ECCV, 2020.

[46] D. Polykovskiy, A. Zhebrak, B. Sánchez-Lengeling, S. Golovanov, O. Tatanov, S. Belyaev, R. Kurbanov, A. A. Artamonov, V. Aladinskiy, M. Veselov, A. Kadurin, S. I. Nikolenko, A. Aspuru-Guzik, and A. Zhavoronkov. Molecular sets (moses): A benchmarking platform for molecular generation models. Frontiers in Pharmacology, 11, 2018.

[47] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In International conference on machine learning, pages 1530–1538. PMLR, 2015.

[48] B. Sanchez-Lengeling and A. Aspuru-Guzik. Inverse molecular design using machine learning: Generative models for matter engineering. Science, 361(6400):360–365, 2018.

[49] A. Schneuing, Y. Du, C. Harris, A. Jamasb, I. Igashov, W. Du, T. Blundell, P. Lió, C. Gomes, M. Welling, et al. Structure-based drug design with equivariant diffusion models. arXiv preprint arXiv:2210.13695, 2022.

[50] Y. Shen and B. Zhou. Closed-form factorization of latent semantics in gans. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 1532–1540, 2021.

[51] Y. Shen, C. Yang, X. Tang, and B. Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. IEEE transactions on pattern analysis and machine intelligence, 44(4):2004–2018, 2020.

[52] Y. Song, N. Sebe, and W. Wang. Orthogonal svd covariance conditioning and latent disentanglement. IEEE T-PAMI, 2022.

[53] Y. Song, A. Keller, N. Sebe, and M. Welling. Flow factorzied representation learning. In NeurIPS, 2023.

[54] Y. Song, A. Keller, N. Sebe, and M. Welling. Latent traversals in generative models as potential flows. arXiv preprint arXiv:2304.12944, 2023.

[55] Y. Song, J. Zhang, N. Sebe, and W. Wang. Householder projector for unsupervised latent semantics discovery. In ICCV, 2023.

12

[56] Y. Song, T. A. Keller, Y. Yue, P. Perona, and M. Welling. Unsupervised representation learning from sparse transformation analysis. arXiv preprint arXiv:2410.05564, 2024.

[57] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer, et al. Applications of machine learning in drug discovery and development. Nature reviews Drug discovery, 18(6):463–477, 2019.

[58] C. Vignac, I. Krawczuk, A. Siraudin, B. Wang, V. Cevher, and P. Frossard. Digress: Discrete denoising diffusion for graph generation. In The Eleventh International Conference on Learning Representations, 2023.

[59] A. Voynov and A. Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In ICML, 2020.

[60] H. Wang, T. Fu, Y. Du, W. Gao, K. Huang, Z. Liu, P. Chandak, S. Liu, P. Van Katwyk, A. Deac, et al. Scientific discovery in the age of artificial intelligence. Nature, 620(7972):47–60, 2023.

[61] J. L. Watson, D. Juergens, N. R. Bennett, B. L. Trippe, J. Yim, H. E. Eisenach, W. Ahern, A. J. Borst, R. J. Ragotte, L. F. Milles, et al. De novo design of protein structure and function with rfdiffusion. Nature, 620 (7976):1089–1100, 2023.

[62] Y. Xie, C. Shi, H. Zhou, Y. Yang, W. Zhang, Y. Yu, and L. Li. Mars: Markov molecular sampling for multi-objective drug discovery. ArXiv, abs/2103.10432, 2021.

[63] X. Yang, J. Zhang, K. Yoshizoe, K. Terayama, and K. Tsuda. ChemTS: an efficient python library for de novo molecular generation. Science and technology of advanced materials, 18(1):972–976, 2017.

[64] J. You, B. Liu, R. Ying, V. Pande, and J. Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In NIPS, 2018.

[65] J. You, B. Liu, R. Ying, V. S. Pande, and J. Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In Neural Information Processing Systems, 2018.

[66] Y. You, R. Zhou, J. Park, H. Xu, C. Tian, Z. Wang, and Y. Shen. Latent 3d graph diffusion. In The Twelfth International Conference on Learning Representations, 2024.

[67] C. Zang and F. Wang. Moflow: an invertible flow model for generating molecular graphs. In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pages 617–626, 2020.

[68] B. Zdrazil, E. Felix, F. Hunter, E. J. Manners, J. Blackshaw, S. Corbett, M. de Veij, H. Ioannidis, D. M. Lopez, J. F. Mosquera, M. P. Magariños, N. Bosc, R. Arcila, T. Kizilören, A. Gaulton, A. P. Bento, M. F. Adasme, P. Monecke, G. A. Landrum, and A. R. Leach. The chembl database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. Nucleic Acids Research, 52:D1180 – D1192, 2023.

[69] C. Zeni, R. Pinsler, D. Zügner, A. Fowler, M. Horton, X. Fu, S. Shysheya, J. Crabbé, L. Sun, J. Smith, et al. Mattergen: a generative model for inorganic materials design. arXiv preprint arXiv:2312.03687, 2023.

[70] X. Zhang, L. Wang, J. Helwig, Y. Luo, C. Fu, Y. Xie, M. Liu, Y. Lin, Z. Xu, K. Yan, et al. Artificial intelligence for science in quantum, atomistic, and continuum systems. arXiv preprint arXiv:2307.08423, 2023.

[71] Z. Zhou, S. M. Kearnes, L. Li, R. N. Zare, and P. F. Riley. Optimization of molecules via deep reinforcement learning. Scientific Reports, 9, 2018.

[72] X. Zhu, C. Xu, and D. Tao. Learning disentangled representations with latent variation predictability. In ECCV, 2020.

13

Appendix for ChemFlow

A **Wasserstein Gradient Flow**   **15**

B **PDE-regularized Latent Space Learning**   **16**

C **Extended Related Work**   **17**

    C.1 Machine Learning for Molecule Generation 17

    C.2 Goal-oriented Molecule Generation 17

    C.3 Image Editing in the Latent Space 17

D **Experiments Details**   **18**

    D.1 Baselines 18

    D.2 Training Dataset 18

    D.3 Molecule Properties 18

    D.4 Training and inference 19

    D.5 Experiments Setup 19

    D.6 Evaluation Metrics 20

    D.7 Additional Baselines 21

    D.8 More Experiment Results 22

E **Latent Space Visualization and Analysis**   **24**

F **Qualitative Evaluations**   **25**

14

A Wasserstein Gradient Flow

As shown in the main paper, based on the dynamic formulation of optimal transport [2], the L_2 Wasserstein distance can be re-written as:

$$W_2(\mu_0, \mu_1) = \min_{\rho, v} \sqrt{\int \int \rho_t(\boldsymbol{z})|v_t(\boldsymbol{z})|^2 \, d\boldsymbol{z}dt} \tag{16}$$

where $v_t(\boldsymbol{z})$ is the velocity of the particle at position $\boldsymbol{z}$ and time t, and $\rho_t(\boldsymbol{z})$ is the density $d\mu(\boldsymbol{z}) = \rho_t(\boldsymbol{z})d\boldsymbol{z}$. The distance can be optimized by the gradient flow of a certain function on space and time. Consider the functional $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}$ that takes the following form:

$$\mathcal{F}(\mu) = \int U(\rho_t(\boldsymbol{z})) \, d\boldsymbol{z} \tag{17}$$

The curve is considered as a gradient flow if it satisfies $\nabla\mathcal{F} = -\frac{d}{dt}\rho_t(\boldsymbol{z})$ [1]. Moving the particles leads to:

$$\frac{d}{dt}\mathcal{F}(\mu) = \int U'(\boldsymbol{z})\frac{d\,\rho_t(\boldsymbol{z})}{dt} \, d\boldsymbol{z} \tag{18}$$

The velocity vector satisfies the continuity equation:

$$\frac{d\,\rho_t(\boldsymbol{z})}{dt} = -\nabla \cdot \left(v_t(\boldsymbol{z})\rho_t(\boldsymbol{z}) \right) \tag{19}$$

where $-\nabla \cdot \left(v_t(\boldsymbol{z})\rho_t(\boldsymbol{z}) \right)$ is the tangent vector at point $\rho_t(\boldsymbol{z})$. Eq. (18) can be simplified to:

$$\begin{aligned} \frac{d}{dt}\mathcal{F}(\mu) &= \int -U'(\rho_t(\boldsymbol{z}))\nabla \cdot \left(v_t(\boldsymbol{z})\rho_t(\boldsymbol{z}) \right) \, d\boldsymbol{z} \\ &= \int \nabla\left(U'(\rho_t(\boldsymbol{z}))\right)v_t(\boldsymbol{z})\rho_t(\boldsymbol{z}) \, d\boldsymbol{z} \end{aligned} \tag{20}$$

On the other hand, the calculus of differential geometry gives

$$\frac{d}{dt}\mathcal{F}(\mu) = \mathrm{Diff}\mathcal{F}|_{\rho_t}(-\nabla \cdot \left(v_t(\boldsymbol{z})\rho_t(\boldsymbol{z}) \right)) = \langle\nabla\mathcal{F}, -\nabla \cdot \left(v_t(\boldsymbol{z})\rho_t(\boldsymbol{z}) \right)\rangle_f \tag{21}$$

where $\langle,\rangle_f$ is a Riemannian distance function which is defined as:

$$\langle-\nabla \cdot \left(w_1(\boldsymbol{z})\rho_t(\boldsymbol{z}) \right), -\nabla \cdot \left(w_2(\boldsymbol{z})\rho_t(\boldsymbol{z}) \right)\rangle_f = \int w_1(\boldsymbol{z})w_2(\boldsymbol{z})f(\boldsymbol{z}) \, d\boldsymbol{z} \tag{22}$$

This scalar product coincides with the W_2 distance according to Benamou and Brenier [2]. Then eq. (20) can be similarly re-written as:

$$\frac{d}{dt}\mathcal{F}(\mu) = \langle-\nabla \cdot \left(\nabla U'(\rho_t(\boldsymbol{z}))\rho_t(\boldsymbol{z}) \right), -\nabla \cdot \left(v_t(\boldsymbol{z})\rho_t(\boldsymbol{z}) \right)\rangle_f \tag{23}$$

So the relation arises as:

$$\nabla\mathcal{F} = -\nabla \cdot \left(\nabla U'(\rho_t(\boldsymbol{z}))\rho_t(\boldsymbol{z}) \right) \tag{24}$$

Since we have $\nabla\mathcal{F} = -\frac{d}{dt}\rho_t(\boldsymbol{z})$, the above equation can be re-written as

$$\frac{d}{dt}\rho_t(\boldsymbol{z}) = \nabla \cdot \left(\nabla U'(\rho_t(\boldsymbol{z}))\rho_t(\boldsymbol{z}) \right) \tag{25}$$

The above derivations can be alternatively made by the poineering JKO scheme [34]. This explicitly defines the relation between evolution PDEs of $\rho_t(\boldsymbol{z})$ and the internal energy U. For our method, we use the gradient of our scalar energy field $\nabla u(\boldsymbol{z}, t)$ to learn the velocity field which is given by $U'(\rho_t(\boldsymbol{z}))$. Interestingly, driven by certain specific velocity fields $\nabla u(\boldsymbol{z}, t)$, the evolution of $\rho(\boldsymbol{z}, t)$ would become some special PDEs. Here we discuss some possibilities:

15

Heat Equations. If we consider the energy function U as the weighted entropy:

$$U(\rho_t(z)) = \rho_t(z)\log(\rho_t(z)) \tag{26}$$

We would have exactly the heat equation:

$$\frac{d}{dt}\rho_t(z) - \frac{d}{dz^2}\rho_t(z) = 0 \tag{27}$$

Injecting the above equation back into the continuity equation leads to the velocity field $v_t(z)$ as

$$\begin{aligned}
\frac{d\,\rho_t(z)}{dt} &= -\nabla \cdot \Big(v_t(z)\rho_t(z)\Big) = \frac{d}{dz^2}\rho_t(z) \\
v_t(z) &= -\frac{\nabla\rho_t(z)}{\rho_t(z)} = -\nabla\log(\rho_t(z))
\end{aligned} \tag{28}$$

When our $\nabla u(z,t)$ learns the velocity field $-\nabla\log(\rho_t(z))$, the evolution of $\rho(z,t)$ would become heat equations.

Fokker Planck Equations. For the energy function defined as:

$$U(\rho_t(z)) = -A \cdot \rho_t(z) + \rho_t(z)\log(\rho_t(z)) \tag{29}$$

we would have the Fokker-Planck equation as

$$\frac{d}{dt}\rho_t(z) + \frac{d}{dz}[\nabla A\rho_t(z)] - \frac{d}{dz^2}[\rho_t(z)] = 0, \tag{30}$$

The velocity field can be similarly derived as

$$v_t(z) = \nabla A - \nabla\log(\rho_t(z)) \tag{31}$$

For the velocity field $\nabla A - \nabla\log(\rho_t(z))$, the movement of $\rho(z,t)$ is the Fokker Planck equation.

Porous Medium Equations. If we define the energy function as

$$U(\rho_t(z)) = \frac{1}{m-1}\rho_t^m(z) \tag{32}$$

Then we would have the porous medium equation where $m > 1$ and the velocity field:

$$\frac{d}{dt}\rho_t(z) - \frac{d}{dz^2}\rho_t^m(z) = 0, \; v_t(z) = -m\rho^{m-2}\nabla\rho \tag{33}$$

When the $\nabla u(z,t)$ learns the velocity $-m\rho^{m-2}\nabla\rho$, the trajectory of $\rho(z,t)$ becomes the porous medium equations.

B   PDE-regularized Latent Space Learning

Our framework can be extended to incorporate the PDE dynamics as part of the training procedure to encourage a more structured representation. To validate the effectiveness, we incorporate a PDE loss such that we expect any path in the latent space to follow specific dynamics (wave equation in our experiment). In addition to the initial setup outlined in Appendix D.5, we further fine-tune the VAE model by applying a PDE regularization loss term, defined as

$$\mathcal{L} = \mathcal{L}_{VAE} + \mathcal{L}_r + \mathcal{L}_\phi$$

that includes the velocity-matching objective $\mathcal{L}_r$ and boundary condition $\mathcal{L}_\phi$. This PDE-regularized latent space learning can also be adapted to other generative models by replacing $\mathcal{L}_{VAE}$. We further optimize the model and energy network for 10 epochs across the full training dataset using an AdamW optimizer with a 1e-4 learning rate and a cosine learning rate scheduler, without a warm-up period. All other training parameters remain consistent with those initially described in Appendix D.5.

During fine-tuning, the VAE loss drops from 0.2187 to 0.06288. The results of QED optimization surpass those of two other unsupervised methods, indicating potential areas for future research and refinement.

16

Table 5: **Single-objective Maximization with PDE-regularized Latent Space Learning** The results are the same as of Table 1, only unsupervised results are presented for fair comparison. (UNSUP denotes unsupervised scenarios). K represents the number of energy networks trained for unsupervised training with disentanglement regularization.

METHOD	K	PLOGP ↑			QED ↑		
		1ST	2ND	3RD	1ST	2ND	3RD
RANDOM	N/A	3.52	3.43	3.37	**0.940**	**0.933**	0.932
WAVE (UNSUP)	10	**5.30**	**5.22**	**5.14**	0.905	0.902	0.978
HJ (UNSUP)	10	4.26	4.10	4.07	0.930	0.928	0.927
WAVE (UNSUP FT)	1	3.71	3.58	3.46	0.936	**0.933**	**0.933**

C   Extended Related Work

C.1   Machine Learning for Molecule Generation

Molecules are highly discrete objects and two branches of methods are thus developed to design or search new molecules [8]. One idea is to leverage the advancement of deep generative models which approximate the data distribution from a provided dataset of molecules and then sample new molecules from the learned density. This idea inspires a line of work developing deep generative models from variational auto-encoders (VAE) [18, 31], generative adversarial networks (GAN) [21, 6], normalizing flows (NF) [43, 67] and more recently diffusion models [24, 58, 33]. However, to respect the combinatorial nature of molecules, another line of work leverage combinatorial optimization to search new molecules including genetic algorithm (GA) [30], Monte Carlo tree search (MCTS) [63], reinforcement learning (RL) [64], but often with sophisticated optimization objectives beyond simple valid molecules.

C.2   Goal-oriented Molecule Generation

In addition to simply generating valid molecules, a more realistic application is to generate molecules with desired properties [8]. For deep generative model-based methods, it is naturally combined with on-the-fly optimization methods such as gradient-based or Bayesian optimization (in low data regime) as it often maps data to a low-dimensional and smooth latent space thus more friendly for these optimization methods [20]. For methods that do not explicitly reduce the dimensionality of data such as diffusion models, Schneuing et al. [49] propose an evolutionary process to iteratively optimize the generated molecules. As it is observed that the learned latent space exhibits explicit structure [18], Du et al. [11] leverage such property to learn a linear classifier to find the latent direction to optimize the property of given molecules. In opposition to deep generative models, combinatorial optimization methods are often inherently associated with optimization, e.g. reward function in RL, selection criteria in GA, etc [14, 41].

C.3   Image Editing in the Latent Space

Beyond molecule generation, there is a vast literature on the study of the latent space of generative models on images for image editing and manipulation [17, 29, 59, 22, 72, 45, 50, 52, 54, 53, 55, 56]. Here we highlight some representative supervised and unsupervised approaches. Supervised methods usually require pixel-wise annotations. InterfaceGAN [51] leverages face image pairs of different attributes to interpret disentangled latent representations of GANs. Jahanian et al. [29] explores linear and non-linear walks in the latent space under the guidance of user-specified transformation. Compared to supervised methods, unsupervised ones mainly focus on discovering meaningful interpretable directions in the latent space through extra regularization. Voynov and Babenko [59] proposes to jointly learn a set of orthogonal directions and a classifier to learn the distinct interpretable directions. SeFa [50] and HouseholderGAN [55] propose to use the eigenvectors of the (orthogonal) projection matrices as interpretable directions to traverse the latent space. More relevantly, Song et al. [54] proposes to use wave-like potential flows to model the spatiotemporal dynamics in the latent spaces of different generative models.

17

D Experiments Details

D.1 Baselines

We compare with the following baselines:

- **Random**: we take a linear direction that is sampled from Multi-variant Gaussian distribution in the high dimensional latent space and normalized to unit length for all molecules across all time steps.

- **Random 1D**: we take a unit vector where only 1 randomly selected dimension is either 1 or -1 as the linear direction.

- **ChemSpace** [11]: a separation boundary of the training dataset in latent space w.r.t. the desired property is classified by an Support vector machine (SVM). Then we take the normal vector corresponding to the positive separation as the manipulation direction of control.

- **Gradient Flow (LIMO)** [13]: a VAE-based generative model that encodes the input molecules into SELFIES [36] and auto-regressive on the tokenized molecule. LIMO uses Adam optimizer to reverse optimize on the input latent vector z whereas Gradient Flow is equivalent to using an SGD optimizer for the same purpose.

- **Evolutionary Algorithm (EA)**: EA is a general framework that leverages random mutation and crossover to select better samples iteratively to search good solutions. In our scenario, we realize an EA-based baseline by perturbing the vector by the directions given by random, ChemSpace and gradient. The pseudocode can be found in Algorithm 3. The results are presented at Appendix D.7.

D.2 Training Dataset

- **ChEMBL** [68] is a database of 2.4M bioactive molecules with drug-like properties, including features like a Natural Product likeness score and annotations for chemical probes and bioactivity measurements.

- **MOSES** [46] is a benchmarking dataset derived from the ZINC [27] Clean Leads collection, containing 2M filtered molecules with specific physicochemical properties, organized into training, test, and unique scaffold test sets to facilitate the evaluation of model performance on novel molecular scaffolds.

- **ZINC250k** is a subset of the ZINC database containing $\sim$250,000 commercially available compounds for virtual screening.

D.3 Molecule Properties

We report the following metrics for our experiments:

- **Penalized logP/plogP**: Estimated octanol-water partition coefficient penalized by synthetic accessibility (SA) score and the number of atoms in the longest ring.

- **QED**: Quantitative Estimate of Drug-likeness, a metric that evaluates the likelihood of a molecule being a successful drug based on its pharmacophores and physicochemical properties.

- **SA**: Synthetic Accessibility, a score that predicts the ease of synthesis of a molecule, with lower values indicating easier synthesis.

- **DRD2 activity**: Predicted activity against the D2 dopamine receptor, using machine learning models trained on known bioactivity data.

- **JNK3 activity**: Predicted activity against the c-Jun N-terminal kinase 3, important for developing treatments for neurodegenerative diseases.

- **GSK3B activity**: Predicted activity against Glycogen Synthase Kinase 3 beta, which plays a crucial role in various cellular processes including metabolism and neuronal cell development.

- **ESR1 docking score**: Simulation-based score representing the binding affinity of a molecule to Estrogen Receptor 1, relevant in the context of breast cancer therapies.

18

- **ACAA1 docking score**: Simulation-based score representing the binding affinity of a molecule to Acetyl-CoA Acyltransferase 1, important for metabolic processes in cells.

D.3.1 Misalignment of normalization schemes for penalized logP

We notice that plogP is a commonly reported metric in recent molecule discovery literature but does not share the same normalization scheme. Following Gómez-Bombarelli et al. [18, Eq. 1], the SA scores and a ring penalty term were introduced into the calculation of penalized logP as the following

$$J^{\text{logP}}(m) = \text{logP}(m) - \text{SA}(m) - \text{ring-penalty}(m)$$

Each term of $\text{logP}(m)$, $\text{SA}(m)$, and $\text{ring-penalty}(m)$ are normalized to have zero mean and unit standard derivation across the training data. However, no sufficient details were included in their paper or their released source code on how the $\text{ring-penalty}(m)$ is computed. Specifically, 3 implementations are widely used in various works.

Penalized by the length of the maximum cycle without normalization $\text{ring-penalty}(m)$ is computed as the number of atoms on the longest ring minus 6 in their implementation. Neither $\text{logP}(m)$, $\text{SA}(m)$, or $\text{ring-penalty}(m)$ is normalized. MolDQN [71] reported their results in this scheme.

Penalized by the length of the maximum cycle with normalization $\text{ring-penalty}(m)$ is computed same as without normalization. MARS [62], HierVAE [32], GCPN [65], and ours report plogP using this scheme.

Penalized by number of cycles   As described by Jin et al. [31, page 7 footnote 3], $\text{ring-penalty}(m)$ is computed as the number of rings in the molecule that has more than 6 atoms. LIMO reports plogP using this metric.

D.4 Training and inference

We detail our training and inference workflows in Alg. 1 and Alg. D.4, respectively.

Algorithm 1 ChemFlow Training

Require: Pre-trained encoder f_θ, decoder g_ψ, (optional) classifier l_γ, timestamps T, # of potential functions K
1: Initialize $\phi^j(\cdot) \leftarrow \text{MLP}$ for $j = 1, \dots, K$
2: **repeat**
3:    Sampling: $z_0 = f_\theta(x_0), t \sim \texttt{Categorical}(T), k \sim \texttt{Categorical}(K)$
4:    **for** $i = 1, \dots, t$ **do**
5:        $z_{i+1} = z_i + \nabla_z \phi^k(i, z_i)$
6:    **end for**
7:    Decode: $x_t = g_\psi(z_t), x_{t+1} = g_\psi(z_{t+1})$
8:    **if** unsupervised **then**
9:        Classification: $\hat{k} = l_\gamma(x_t; x_{t+1})$
10:        Loss: $\mathcal{L} = \mathcal{L}_r + \mathcal{L}_\phi + \mathcal{L}_{\mathcal{J}} + \mathcal{L}_k$
11:    **else**
12:        Loss: $\mathcal{L} = \mathcal{L}_r + \mathcal{L}_\phi + \mathcal{L}_{\mathcal{P}}$
13:    **end if**
14:    Back-propagation through the Loss $\mathcal{L}$
15: **until** Convergence

D.5 Experiments Setup

Pre-trained VAE   We follow the VAE architecture from LIMO consisting of a 128 dimension embedding layer, 1024 latent space size, 3-hidden-layer encoder, and 3-hidden-layer decoder both with 1D batch normalization and non-linear activation functions. The hidden layer sizes are $\{4096, 2048, 1024\}$ for the encoder and reversely for the decoder. We empirically find that replacing the ReLU activation function with its newer variant Mish activation function [44] results

19

Algorithm 2 ChemFlow Inference / Traversal

Require: Pre-trained encoder f_θ, pre-trained potential function ϕ, (optional) pre-trained proxy function h, timestamps T, step size α, LD strenth β
1: Sampling: $z_0 = f_\theta(x_0)$
2: **for** $t = 1, \ldots, T$ **do**
3:     **if** Langevin Dynamics **then**
4:         $z_t = z_{t-1} - \alpha \nabla_z h_\eta(z_{t-1}) + \beta\sqrt{2\alpha}\mathcal{N}(0, I)$
5:     **else if** Gradient Flow **then**
6:         $z_t = z_{t-1} - \alpha \nabla_z h_\eta(z_{t-1})$
7:     **else**
8:         $z_t = z_{t-1} + \alpha \nabla_z \phi(t - 1, z_{t-1})$
9:     **end if**
10: **end for**
---

in faster convergence and better validation loss. All the experiments reported in this paper use this Mish-activated variant of VAE.

The VAE is trained using an AdamW [42] optimizer, 0.001 initial learning rate, and 1,024 training batch size. To better prevent the model from being stacked at a sub-optimal local minimum, a cosine learning rate scheduler with a 1e-6 minimum learning rate with periodic restart is applied. The VAE is trained for 150 epochs with 4 restarts on 90% of the training data and validated with the rest 10% data. The checkpoint corresponding to the epoch with the lowest validation loss is selected. Training 150 epochs takes ∼8 hours on a single RTX 3090 desktop.

Surrogate Predictor   The performance of the surrogate predictor is crucial to the proposed latent traversal framework. To handle chemical properties of different magnitudes, we normalize all chemical properties in the training data to have zero mean and unit variance. Then we use a pre-activation-norm MLP with residual connections as the surrogate predictor. The predictor contains 3 residual blocks of size 1024 and the output dimension is 1. Similar to the LIMO setups, we find that the choice of optimizer and training hyperparameters like learning rate or learning rate scheduler is crucial for successful training. The predictor is trained for 20 epochs on 100,000 randomly generated samples and validated with 10,000 unseen data with SGD optimizer, 0.001 learning rate, and batch size 1000. The epoch with the best validation loss is selected. Training each predictor takes less than a minute.

Energy Network   We use an MLP structure to parameterize the energy function (the spatial derivative gives the velocity). The time input t is embedded with a sinusoidal positional embedding followed by a linear layer. The special input x is encoded with a linear layer and ReLU activation function. The training of the network uses 9,000 random data and 1,000 unseen data for validation. For unsupervised settings, 10 disentangled potential energy functions are trained for 310 epochs with a batch size of 100. The epoch with the best validation loss is selected. Training an energy network with 10 disentangled potential energy functions takes ∼40 minutes.

Reproducibility   All the experiments including baselines are conducted on one RTX 3090 GPU and one Nvidia A100 GPU. All docking scores are computed on one RTX 2080Ti GPU. The code implementation is available at https://github.com/garywei944/ChemFlow.

D.6   Evaluation Metrics

Success Rate   The success rate is used as the evaluation metric for the molecule manipulation task. It first randomly generates n molecules and traverses each of them in the latent space for k steps. The success rate is calculated as the percentage of k-step trajectories that are successful. In our case, we generate 1000 molecules and traverse for 10 steps. The manipulation is successful if the local change in molecular structure is smooth and molecular property is increased. Specifically, we showed two success rates: the strict success rate and the relaxed success rate.

20

For the strict success rate, manipulation is a success if the molecular property is monotonically increasing, molecular similarity with respect to the previous step is monotonically decreasing, and molecules are diversity on the manipulation trajectory. These constraints are formulated as follows:

$$
\begin{aligned}
C_{SP}(x, k, P) &= \mathbf{1}[\forall i \in [k], s.t. P(x_i) - P(x_{i+1}) \le 0], \\
C_{SS}(x, k, S) &= \mathbf{1}[\forall i \in [k], s.t. S(x_{i+1}, x_1) - S(x_i, x_1) \le 0], \\
C_{SD}(x, k) &= \mathbf{1}[\|x_t : \forall i \in [k]\| > 2], \\
SSR &= \frac{1}{|X|} \sum_{x \in X} \mathbf{1}[C_{SP}(x, k, P) \wedge C_{SS}(x, k, S) \wedge C_{SD}(x, k)]
\end{aligned}
\tag{34}
$$

where $\{x_i\}_{i=1}^{k}$ is one k-step manipulation trajectory, X contains n manipulation trajectories, C is the constraint, P is the property evaluation function, S is the structure similarity function (Taminoto similarity over Morgan fingerprints). The C_{SP} constraints that the property of molecules must monotonically increase. The C_{SS} constraints that the structure is similar in regard to the starting molecule must monotonically decrease. The C_{SD} constraint that the molecules must at least change twice during the manipulation. SSR calculates the percentage of trajectories that satisfy all success constraints.

The relaxed success rate relaxes some constraints by adding a tolerance interval. It is formulated as follows:

$$
\begin{aligned}
C_{SP}(x, k, P) &= \mathbf{1}[\forall i \in [k], s.t. P(x_t) - P(x_{t+1}) \le \epsilon], \\
C_{SS}(x, k, S) &= \mathbf{1}[\forall i \in [k], s.t. S(x_{t+1}, x_1) - S(x_t, x_1) \le \gamma], \\
C_{SD}(x, k) &= \mathbf{1}[\|x_t : \forall i \in [k]\| > 2], \\
SSR &= \frac{1}{|X|} \sum_{x \in X} \mathbf{1}[C_{SP}(x, k, P) \wedge C_{SS}(x, k, S) \wedge C_{SD}(x, k)]
\end{aligned}
\tag{35}
$$

The relaxed success rate does not require a monotonic increase of molecular property but sets a tolerance threshold ϵ. This tolerance threshold ϵ is defined as 5% of the range of property in the training dataset. It also does not require a monotonically decrease of structure similarity with a tolerance threshold γ of 0.1.

D.7 Additional Baselines

Evolutionary Algorithm. We present the Evolutionary Algorithm-based (EA) approach to optimize molecules in the latent space as an additional baseline for the experiment. The pseudocode is provided as Algorithm 3. Table 6 shows the performance of the evolutionary algorithm-based approach on unconstrained optimization. For a fair comparison, all methods in Table 6 have the same number of Oracle calls. The result shows that our methods outperform all EA approaches.

Algorithm 3 Evolutionary Algorithm Augmented Optimization

1: **Input:** n samples select k per iteration, pre-trained ChemSpace/Gradient direction l, step size α, pre-trained surrogate model h, decoder g_ψ
2: Randomly sample n latent vectors $\{z_i^0\}_{i=1}^{n}$ in latent space
3: **for** each iteration $t = 0$ to T **do**
4:      Evaluate sampled latent vector scores: $s_i^t = h(z_i^t)$ for $i = 1, \dots, n$
5:      Select top-k scored latent vectors: $\{z_{\text{top}}^t\}_{j=1}^{k}$
6:      **for** each selected vector $j = 0$ to k **do**
7:          Update $z_{\text{top},j}^{t+1} = z_{\text{top},j}^t + \alpha \cdot l + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, l the evolution direction guided by Random/ChemSpace/Gradient.
8:      **end for**
9:      Randomly sample $\frac{n}{k}$ samples around each $z_{\text{top},j}^t$ to generate n new latent vectors $\{z_i^{t+1}\}_{i=1}^{n}$
10: **end for**
11: Decode latent vectors: $x_i^T = g_\psi(z_i^T)$ for $i = 1, \dots, n$

21

Table 6: **Unconstrained plogP, QED maximization of Evolutionary Algorithm.** (SPV denotes supervised scenarios, UNSUP denotes unsupervised scenarios). Random/ChemSpace/Gradient are the evolution directions of EA. Since EA (Gradient Flow) has converged to a single molecule when optimizing pLogP, only one value is reported.

METHOD	PLOGP ↑			QED ↑		
	1ST	2ND	3RD	1ST	2ND	3RD
EA (RANDOM)	2.29	1.64	1.52	0.836	0.801	0.794
EA (CHEMSPACE)	3.79	3.79	3.79	0.933	0.931	0.931
EA (GRADIENT FLOW)	3.53	/	/	0.930	0.929	0.929
WAVE (SPV)	4.76	3.78	3.71	**0.947**	0.934	0.932
WAVE (UNSUP)	**5.30**	**5.22**	**5.14**	0.905	0.902	0.978
HJ (SPV)	4.39	3.70	3.48	0.946	0.941	0.940
HJ (UNSUP)	4.26	4.10	4.07	0.930	0.928	0.927
LD	4.74	3.61	3.55	**0.947**	**0.947**	**0.942**

Unconstrained Molecular Optimization Additional Results   In addition to reporting the top 3 scores as presented in Table 1, we computed the mean and standard deviation for the top 100 molecules after unconstrained optimization in Table 7. The table shows that our methods have overall the best optimization performance. In addition, HJ exhibits better performance on mean and standard deviation than on top 3, showing that minimizing the kinetic energy is efficient in pushing the distribution to desired properties.

Table 7: **MSE of Unconstrained plogP, QED maximization, and docking score minimization.** (SPV denotes supervised scenarios, UNSUP denotes unsupervised scenarios). Each entry in the table follows the format mean ± std (median). Boldface highlights the highest-performing generation for each property within each rank.

METHOD	PLOGP ↑	QED ↑	ESR1 DOCKING ↓	ACAA1 DOCKING ↓
RANDOM	2.345 ± 0.386 (2.259)	0.903 ± 0.014 (0.902)	-9.127 ± 0.360 (-9.015)	-8.454 ± 0.316 (-8.390)
GRADIENT FLOW	2.664 ± 0.382 (2.537)	0.910 ± 0.012 (0.908)	-9.452 ± 0.338 (-9.365)	-8.735 ± 0.337 (-8.650)
CHEMSPACE	2.580 ± 0.406 (2.446)	0.907 ± 0.014 (0.906)	-9.523 ± 0.409 (-9.395)	-8.749 ± 0.356 (-8.640)
WAVE (SPV)	2.536 ± 0.439 (2.388)	0.903 ± 0.015 (0.898)	**-9.630 ± 0.399 (-9.525)**	-8.764 ± 0.344 (-8.650)
WAVE (UNSUP)	1.736 ± 0.401 (1.610)	0.845 ± 0.014 (0.840)	-9.074 ± 0.329 (-9.000)	**-8.813 ± 0.265 (-8.745)**
HJ (SPV)	2.482 ± 0.397 (2.382)	0.899 ± 0.017 (0.894)	-9.544 ± 0.322 (-9.460)	-8.792 ± 0.332 (-8.675)
HJ (UNSUP)	**3.405 ± 0.254 (3.377)**	**0.911 ± 0.009 (0.909)**	-9.132 ± 0.321 (-9.090)	-8.668 ± 0.243 (-8.630)
LD	2.463 ± 0.388 (2.399)	0.905 ± 0.014 (0.903)	-9.400 ± 0.360 (-9.300)	-8.709 ± 0.372 (-8.585)

D.8   More Experiment Results

We conduct more experiments to analyze the performance of the proposed methods systematically. They are referred to and discussed in the main paper.

Pearson correlation score for unsupervised diversity guidance with disentanglement regularization.   Tables 8 and 9 present the Pearson correlation scores for the trained energy networks of wave equations and Hamilton-Jacobi equations with disentanglement regularization, respectively. For properties other than synthetic accessibility (SA), we select the network with the highest correlation score to maximize these properties. Conversely, for SA, the network with the lowest correlation score (most negative score) is chosen.

Distribution shift and convergence for plogP optimization.   Figure 4 illustrates the distribution shift in plogP optimization, complementing the analysis in Figure 3. Similar to the findings discussed in Section 4.2, both unsupervised methods encounter out-of-distribution (OOD) issues after 400 steps, consistent with those observed with ChemSpace. The Random 1D method does not achieve the expected distribution shift, as it manipulates only one dimension of the latent vector. Figure 5 depicts the convergence trends of each method's improvements. Consistent with the predictions in Proposition 3.1, Langevin Dynamics demonstrates the fastest and most effective convergence among all methods.

22

Table 8: **Pearson Correlation of trained Wave PDE Energy Network.** The average Pearson correlation between the sequence of real properties and sequence of time steps along the manipulation trajectory following a learned potential function $\phi^k(t, z)$ using wave equations.

	PLOGP	SA	QED	DRD2	JNK3	JSK3B
0	0.019	0.003	0.016	0.029	0.015	**0.051**
1	**0.160**	**-0.451**	**0.275**	-0.074	-0.153	-0.272
2	0.035	-0.003	0.011	0.002	-0.006	0.017
3	0.072	-0.096	0.065	0.011	-0.017	-0.028
4	0.042	0.003	0.039	-0.010	**0.025**	0.018
5	-0.036	-0.022	0.150	0.009	-0.017	0.008
6	0.032	-0.045	0.006	0.002	-0.011	0.002
7	0.023	-0.023	0.054	-0.002	-0.013	-0.017
8	0.075	-0.085	0.064	-0.040	-0.007	-0.054
9	0.013	0.020	-0.011	**0.031**	0.005	0.014
INDEX	1	1	1	9	4	0

Table 9: **Pearson Correlation of trained Hamilton-Jacobian PDE Energy Network.** The average Pearson correlation between the sequence of real properties and sequence of time steps along the manipulation trajectory following a learned potential function $\phi^k(t, z)$ using Hamilton-Jacobi equations.

	PLOGP	SA	QED	DRD2	JNK3	GSK3B
0	**0.345**	**-0.453**	0.141	-0.210	-0.127	-0.350
1	0.257	-0.289	0.057	-0.154	-0.121	-0.276
2	0.203	-0.284	0.050	**0.020**	-0.044	-0.164
3	-0.029	0.041	0.034	-0.001	**0.001**	0.002
4	0.304	-0.343	0.066	-0.225	-0.179	-0.336
5	0.008	-0.036	0.029	-0.009	-0.021	0.015
6	0.316	-0.370	0.173	-0.208	-0.163	-0.309
7	0.305	-0.429	0.046	-0.291	-0.191	-0.352
8	0.003	-0.011	0.016	-0.021	-0.035	**0.026**
9	0.311	-0.386	**0.209**	-0.222	-0.209	-0.342
INDEX	0	0	9	2	3	8

Similarity-constrained QED optimization and distribution shift. To further explore optimization tasks on additional properties, we also attempted to enhance the QED of molecules. Table 10 presents the results of efforts to maximize the QED of 800 molecules from the ZIC250K dataset that initially had the lowest QED scores. Despite encountering OoD issues with ChemSpace and two unsupervised methods, as illustrated in Figure 6, the performance of Langevin Dynamics surpasses other methods across various similarity levels, which is consistent with the result from Table 2.

Table 10: **Similarity-constrained QED maximization.** For each method with minimum similarity constraint δ, the results in reported in format mean ± standard derivation (success rate %) of absolute improvement, where the mean and standard derivation are calculated among molecules that satisfy the similarity constraint.

Method	$\delta = 0$	$\delta = 0.2$	$\delta = 0.4$	$\delta = 0.6$
Random	0.36 ± 0.15 (98.0)	0.19 ± 0.14 (78.4)	0.11 ± 0.11 (54.4)	0.08 ± 0.08 (29.9)
Random 1D	0.13 ± 0.11 (40.1)	0.12 ± 0.10 (38.1)	0.09 ± 0.07 (29.2)	0.07 ± 0.06 (18.0)
Gradient Flow	0.48 ± 0.13 (99.2)	0.25 ± 0.16 (84.9)	0.13 ± 0.12 (53.8)	0.10 ± 0.10 (24.9)
ChemSpace	0.47 ± 0.13 (99.8)	0.29 ± 0.18 (90.1)	0.18 ± 0.15 (62.9)	0.12 ± 0.12 (33.5)
Wave (spv)	0.38 ± 0.16 (97.9)	0.23 ± 0.15 (84.9)	0.13 ± 0.11 (62.6)	0.08 ± 0.08 (35.0)
Wave (unsup)	**0.54 ± 0.18 (99.1)**	0.09 ± 0.10 (51.0)	0.05 ± 0.06 (27.4)	0.03 ± 0.04 (14.0)
HJ (spv)	0.21 ± 0.16 (74.2)	0.17 ± 0.14 (69.4)	0.12 ± 0.11 (57.0)	0.08 ± 0.09 (35.0)
HJ (unsup)	0.52 ± 0.19 (98.5)	0.11 ± 0.11 (60.4)	0.05 ± 0.05 (28.6)	0.03 ± 0.03 (13.4)
LD	0.53 ± 0.12 (99.8)	**0.31 ± 0.17 (96.2)**	**0.16 ± 0.12 (77.8)**	**0.10 ± 0.09 (49.5)**

23
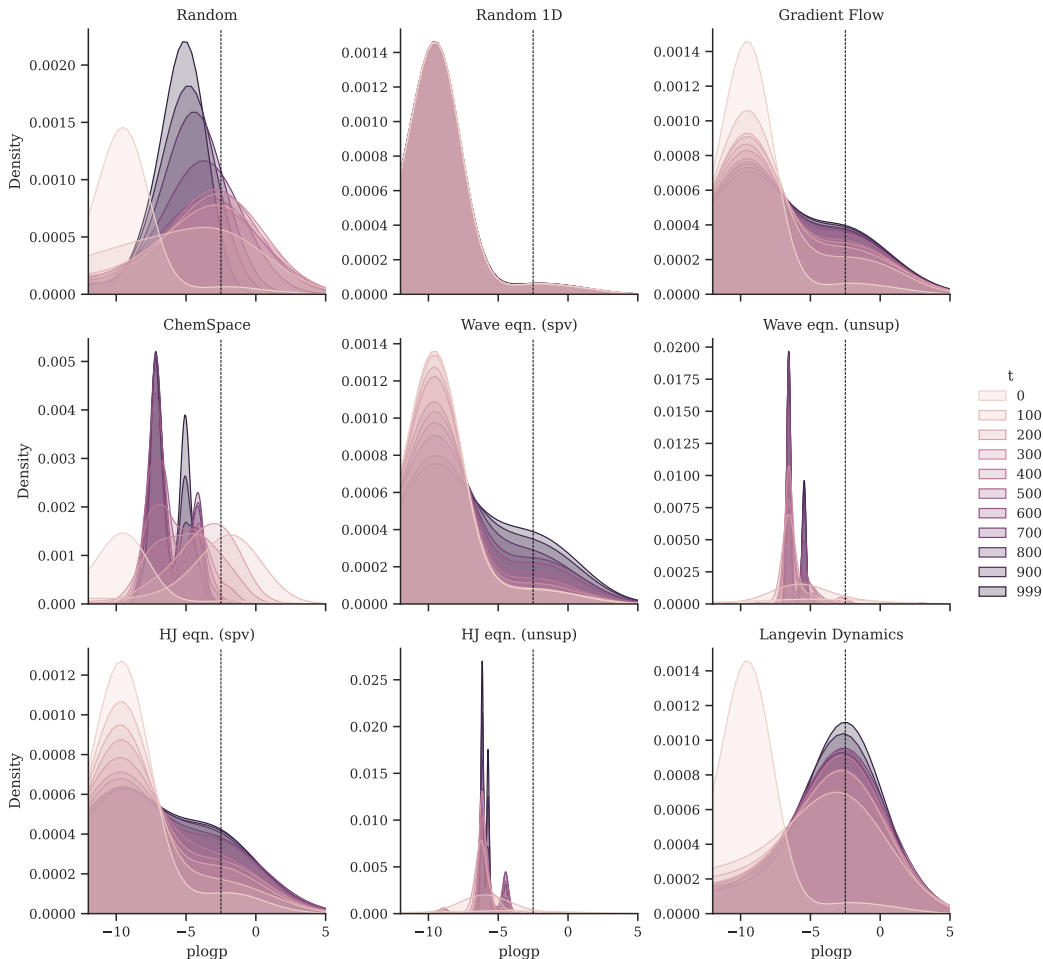
Figure 4: **Distribution shift for plogP optimization**

E Latent Space Visualization and Analysis

As observed in experiments such that random directions perform surprisingly well on molecule manipulation and optimization tasks, we look into the learned latent space to understand its structure. As the prior of a VAE is an isotropic Gaussian distribution, we first verify if the learned variational poster also follows a Gaussian distribution and we find that it does learn so from the evidence shown in Figure 7, where the norm of the molecule projected to the latent space concentrate around 32 which is around $\sqrt{d}$ such that the latent dimension d is 1024. We also visualize in Figure 8 how the properties of the molecules in the training dataset are related to their latent vector norms. Surprisingly, we find a strong correlation between almost all molecular properties and their latent norms. Combining these two evidences, it is not surprising that a random latent vector taking a random direction will change the molecular property smoothly and monotonically. In addition, we further plot when we traverse along a random direction in the latent space, how the change of the norm may correspond to the change of a certain property. Among them, we find that SA is particularly in strong positive correlation with the traversal in Figure 9. Though the emergence of the structure in the latent space is interesting and suggests that better algorithms can be developed to exploit the structure, we leave this to future work.

Additionally, we visualize the traversal trajectory for each different property using both supervised and unsupervised wave flow in Figure 10. The plot shows that almost all trajectories grow towards a unique direction in the t-SNE plot, which implies the disentanglement of learned directions and, thus, molecular properties. In addition, the figures display sinusoidal wave-shape trajectories, indicating the

24
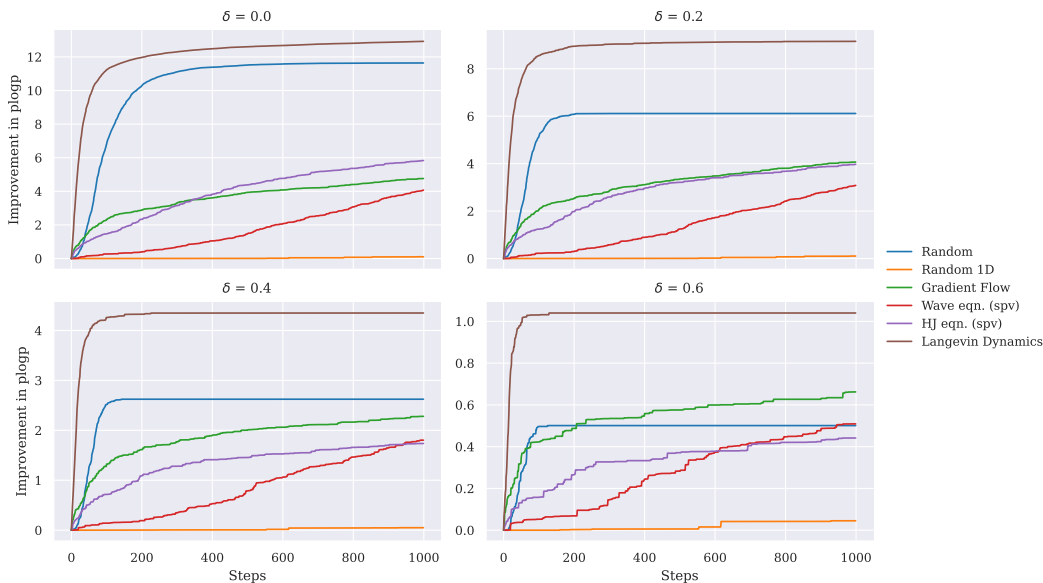
Figure 5: **Optimization Convergence** Langevin Dynamics shows faster convergence and achieves greater improvement in plogP.

flow follows wave-like dynamics. In the unsupervised t-SNE plot, the trajectories of some properties overlap, such as plogP and SA. It is because some properties correlate with the same disentangled direction, so their traversal follows the same direction, thus the same trajectories.

F Qualitative Evaluations

In addition to quantitative evaluations, we demonstrate some qualitative evaluations in this section. We showcase three manipulation trajectories in Figures 11 to 13. Each of these paths is a 6-step manipulation for different molecular properties using different flows. Specifically, we can see that the supervised wave flow and gradient flow improves the molecular property by the conversion of large heterocyclic rings for better synthesizability. We also show three optimization trajectories in Figures 14 to 16. From left to right, each molecule is a snapshot selected from a trajectory of 1000-step optimization. It is notable that the supervised Hamilton-Jacobi flow optimizes the property by reducing the number of nitrogen atoms. This leads to a more chemically stable molecule, whereas the original molecule, with a large number of nitrogen atoms, is unstable and potentially explosive. The supervised wave flow optimizes the molecular property by simplifying the poly-cyclic molecule which enhances its synthesizability and overall stability.

25
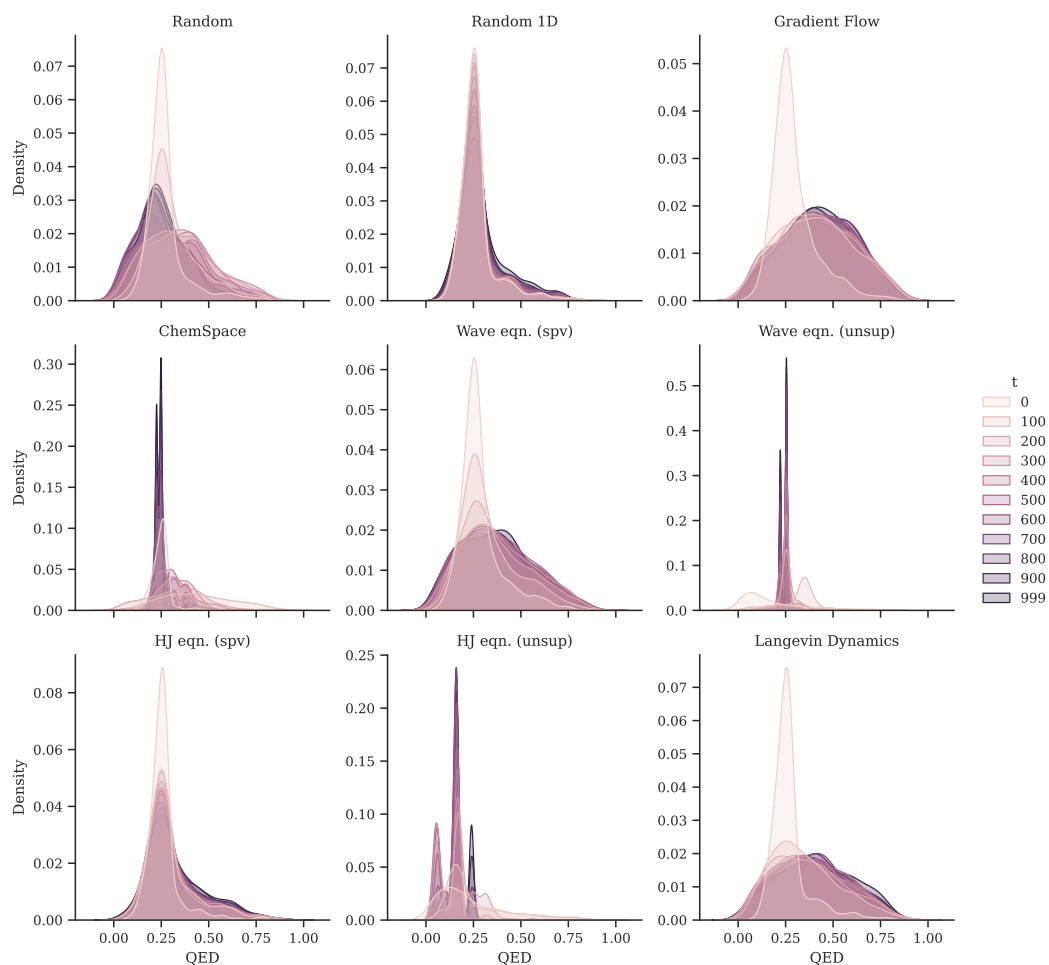
Figure 6: **Distribution shift for QED optimization**
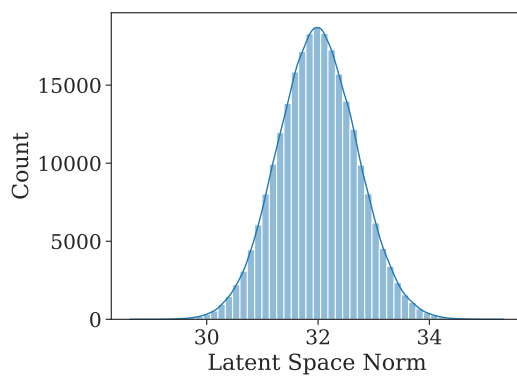
Figure 7: **Latent Vector Norm.** Distribution of the norm of the latent vectors projected from the training dataset onto the learned latent space.

26
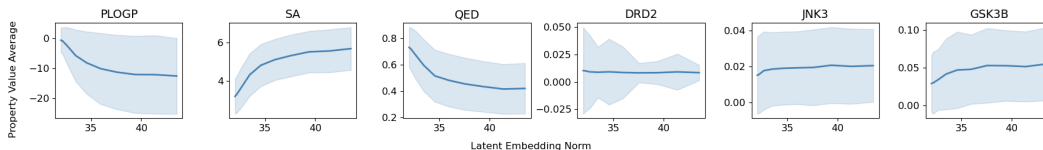
Figure 8: **Embedding Norm against Property Value of each path.** Norm and property value of molecules along the direction of latent traversal with a random direction. The middle curve shows the mean property value and latent embedding norm for all paths. The shaded area is the standard deviation of property value.
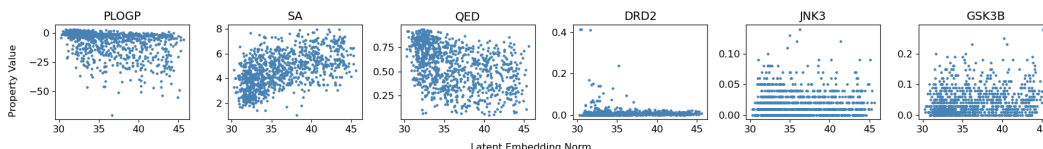
Figure 9: **Embedding Norm against Property Value of each Molecule.** Scatter plot of norm and property value of individual molecules in the training set encoded in the latent space.
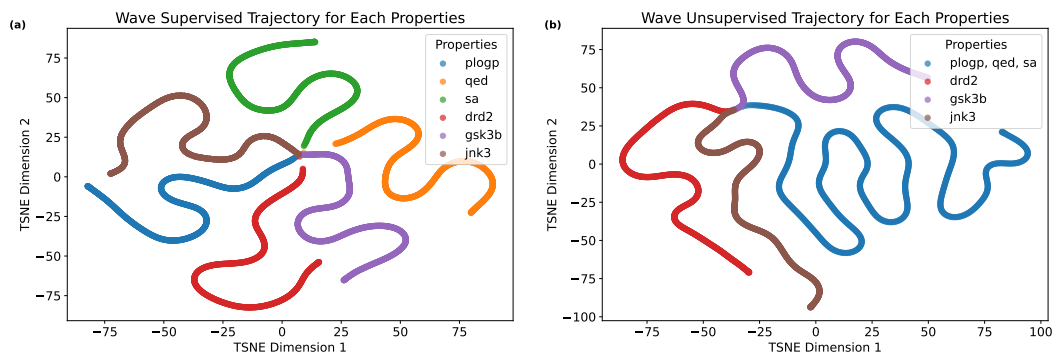
Figure 10: **t-SNE Visualization of Optimization Trajectories.** Optimization Trajectories following supervised and unsupervised wave flow visualized using t-SNE.
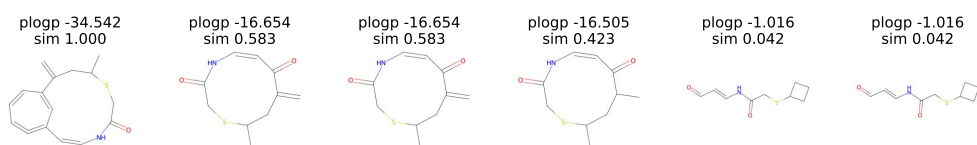
Figure 11: **Molecule Manipulation Trajectory** Molecule manipulation by chemspace on plogP.
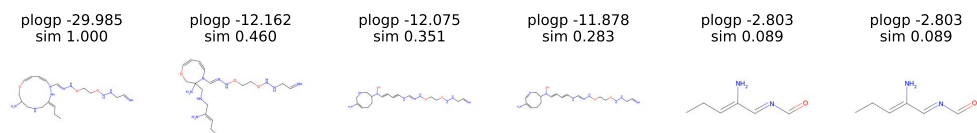
Figure 12: **Molecule Manipulation Trajectory** Molecule manipulation by gradient flow on plogP.
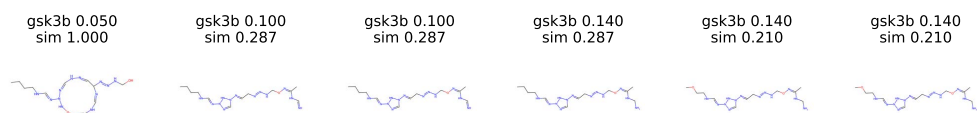
Figure 13: **Molecule Manipulation Trajectory** Molecule manipulation by supervised wave flow on GSK3B.

27
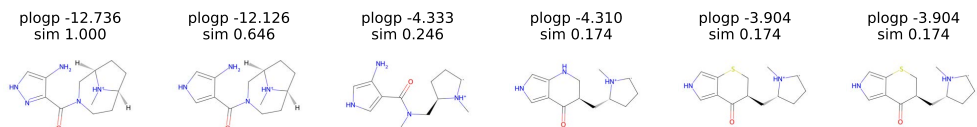
Figure 14: **Molecule Optimization Trajectory** Molecule optimization by random direction on plogP.
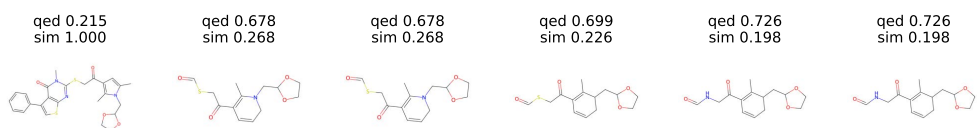
Figure 15: **Molecule Optimization Trajectory** Molecule optimization by supervised wave flow on QED.
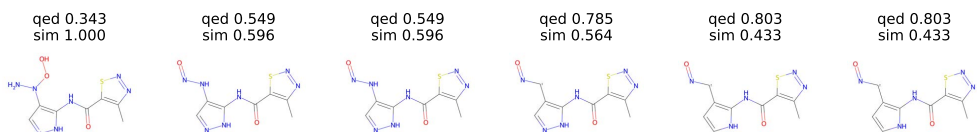
Figure 16: **Molecule Optimization Trajectory** Molecule optimization by supervised Hamilton-Jacobi flow on QED.

28

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

    Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

    Answer: [Yes]

    Justification: The claims in the abstract and introduction reflect the scope and contributions of the paper, including the development of the ChemFlow framework for navigating chemical latent spaces and its validation across multiple molecule manipulation and optimization tasks.

    Guidelines:
    - The answer NA means that the abstract and introduction do not include the claims made in the paper.
    - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
    - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
    - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

    Question: Does the paper discuss the limitations of the work performed by the authors?

29

Answer: [Yes]

Justification: The paper discusses limitations such as scaling with billions of available molecules in large databases and generalizing beyond small molecules.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The paper does not invent any new theory but replies on existing theoretical results. The details including assumptions and derivations (referenced in Section 3 and Appendix A), are discussed.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

30

Justification: All experimental setups are clearly described, including data sources, model parameters, and evaluation criteria. The source code is released at https://github.com/garywei944/ChemFlow. This should allow for the reproducibility of the results presented.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All the data used in our experiments are open-sourced and we include our codes and instructions to run with README.md in https://github.com/garywei944/ChemFlow.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.

31

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specifies all necessary experimental details, such as data splits, hyperparameters, and types of optimizers, which are essential for understanding and reproducing experimental results, referred in Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Statistical significance is discussed with appropriate measures (std) and success rate provided for key results, ensuring the reliability of the findings.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

32

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The paper details the computational resources used, including the type of GPUs and the estimated compute time for experiments in Appendix D.5, which aids in the replication and understanding of the resource requirements.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: The research adheres to the NeurIPS Code of Ethics, and ethical considerations, especially concerning dual-use risks and responsible AI practices, are discussed in Section 5.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Both positive and negative societal impacts are considered, with discussions on how the ChemFlow framework could impact drug discovery and potential misuse scenarios.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

33

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not introduce new models or datasets with high risks of misuse at this moment; thus, specific safeguards are not necessary. However, general practices for responsible AI are discussed.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper correctly credits all used assets, including datasets and codes, with appropriate references and discusses the terms of use and licenses where applicable.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not introduce new datasets or tools that require additional documentation or licensing information.

34

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The research does not involve human subjects or crowdsourcing, making this question not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No human subject research is conducted; hence IRB approval is not required.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

35