Navigating Chemical Space with Latent Flows

Guanghao Wei [* 1] **Yining Huang** [* 2] **Chenru Duan** [3] **Yue Song** [† 4] **Yuanqi Du** [† 1]

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 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 (Sanchez-Lengeling & Aspuru-Guzik, 2018; Vamathevan et al., 2019). With the increasing interest in applying deep learning models in scientific problems (Wang et al., 2023; Zhang et al., 2023), 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 (Du et al., 2022a). Both of the approaches have demonstrated promising results in small molecule, protein, and materials design (Watson et al., 2023; Ingraham et al., 2023; Loeffler et al., 2023; Zeni et al., 2023). 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} (Bohacek et al., 1996; Lipinski et al., 2012). 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) (Goodfellow et al., 2016), variational autoencoders (VAEs) (Kingma & Welling, 2013), and denoising diffusion models (Ho et al., 2020)) in computer vision (Jahanian et al., 2019; Burgess et al., 2018; Härkönen et al., 2020; Kwon et al., 2022), 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 (Du et al., 2022b). 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 (Du et al., 2022c). 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,

---

[*]Equal contribution [†]Corresponding authors [1]Cornell University [2]Harvard University [3]Massachusetts Institute of Technology [4]University of Trento. Correspondence to: Yuanqi Du <yd392@cornell.edu>, Yue Song <yue.song@unitn.it>.

providing access to gradients for gradient-based optimization (Liu et al., 2018; Griffiths & Hernández-Lobato, 2020; Eckmann et al., 2022). 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 (Gómez-Bombarelli et al., 2018), which is also widely studied and used as a priori in computer vision (Shen et al., 2020; Shen & Zhou, 2021). ChemSpace (Du et al., 2023) 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 $x \in \mathcal{X}$ into an (often) low-dimensional and continuous space (*i.e.* latent space) while the decoder maps the latent vectors $z \in \mathcal{Z}$ 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 (Jin et al., 2018; Madhawa et al., 2019; Cao & Kipf, 2018; Vignac et al., 2023). 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 (Zang & Wang, 2020). 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:

$$\begin{aligned} \mathrm{d}z_t &= -\nabla_z h(z_t)\mathrm{d}t \\ z_t &= z_{t-1} - \nabla_z h(z_{t-1})\mathrm{d}t \end{aligned} \quad (1)$$

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 generative models (Gómez-Bombarelli et al., 2018). Since the direction is assumed to be linear, it can be found easily. ChemSpace (Du et al., 2023) learns a linear classifier that defines the separation boundary of the molecular properties. Then the normal direction of the boundary provides a linear direction $n \in \mathcal{Z}$ for traversing the latent space:

$$z_t = z_0 + nt \quad (2)$$

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 $\partial^2 z_t / \partial^2 z_{t-1} = \partial^2 z_t / \partial^2 t = 0$ 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 $x(t) \in \mathbb{R}^n$ 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 $x'(t) = -\nabla \mathcal{F}(x(t))$. 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 & Brenier (2000), the commonly used L^2 Wasserstein metric induces a dynamic formulation of optimal transport:

$$\begin{aligned} W_2(\mu, \nu)^2 = \min_{v, \rho} \Big\{ \int \int \frac{1}{2} \rho(t, x)|v(t, x)|^2 \\ dt\, dx : \partial_t \rho(t, x) = -\nabla \cdot (v(t, x)\rho(t, x)) \Big\} \end{aligned} \quad (3)$$

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 (Jordan et al., 1996).

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 $\phi^k = \mathtt{MLP}_{\theta^k}(t, z) \in \mathbb{R}$ and use the learned flow $\nabla_z\phi$ 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 (Rezende & Mohamed, 2015):

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

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.

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):

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

where the velocity field is defined as the flow $\nabla\phi$. The HJE can also be interpreted as mass transportation in fluid dynamics, *i.e.*, under the velocity field $\nabla\phi$, 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:

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

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 dynamic 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:

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

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 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 in different supervision settings.

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
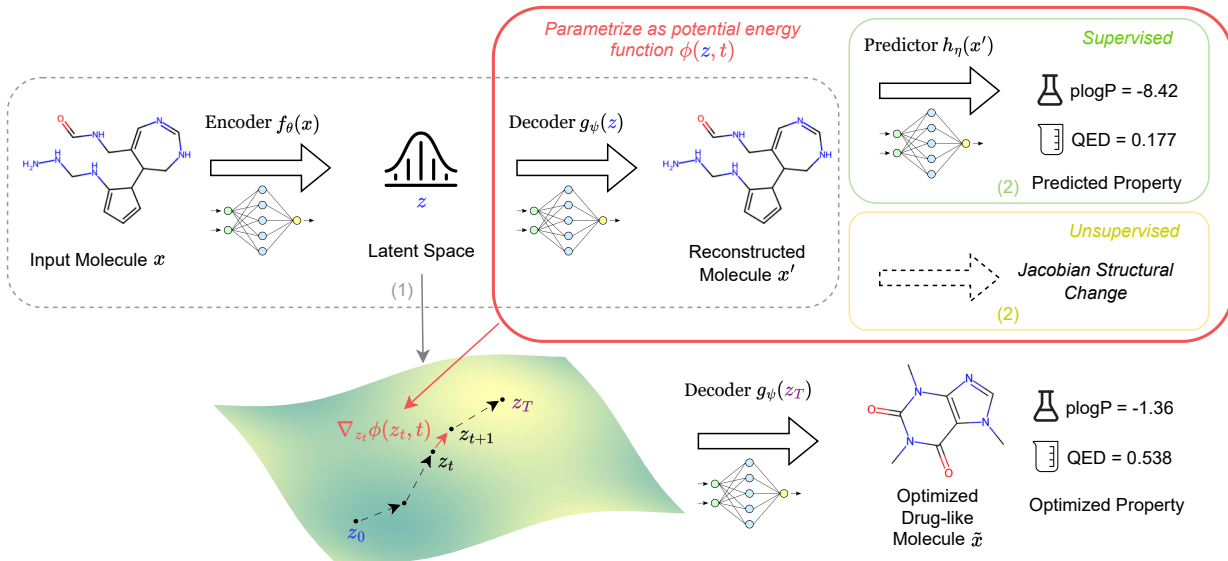
Figure 1. **ChemFlow** framework: (1) a pre-trained encoder $f_\theta(\cdot)$ and decoder $g_\psi(\cdot)$ that maps between molecules x and latent vectors z, (2) we use a property predictor $h_\eta(\cdot)$ (green box) or a "Jacobian control" (yellow box) as the guidance to learn a vector field $\nabla\phi(z_t, t)$ 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.

to learn flows that drive the increase of the property for the trajectory of the generated molecules.

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

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.

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 the 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. (2023b), 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. (2023b) 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.

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 for global optimization (Gardiner et al., 1985). Notably, the convergence of Langevin dynamics, particularly at low temperatures, tends to occur around the global minima of the potential energy function (Chiang et al., 1987). The continuous and discretized Langevin dynamics are as follows:

$$\mathrm{d}\boldsymbol{z}_t = -\nabla_z h_\eta(\boldsymbol{z}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{w}_t$$
$$\boldsymbol{z}_t = \boldsymbol{z}_{t-1} - \nabla_z h_\eta(\boldsymbol{z}_{t-1})\mathrm{d}t + \sqrt{2\mathrm{d}t}\mathcal{N}(0, I) \tag{15}$$

Proposition 3.1. *(Global Convergence of Langevin Dynamics, adapted from Gelfand & Mitter (1991)). Given a Langevin dynamics in the form of*

$$\boldsymbol{z}_t = \boldsymbol{z}_{t-1} - a_t(\nabla_z h_\eta(\boldsymbol{z}_{t-1}) + \boldsymbol{u}_t) + b_t \mathbf{w}_t$$

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 $\boldsymbol{u}_t$ is a set of random variables in $\mathbb{R}^n$ denoting noisy measurements of the energy function $h_\eta(\cdot)$. Under mild assumptions, $\boldsymbol{z}_t$ 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** (Polykovskiy et al., 2018), **ZINC250K** (Irwin & Shoichet, 2005), and **ChEMBL** (Zdrazil et al., 2023). Molecules are represented by SELFIES strings (Krenn et al., 2020). 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 (Huang et al., 2022) , *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. (2022) 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 (Brown et al., 2019). Various machine learning methods have accelerated this process (Du et al., 2022a). 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.

Baselines. For molecule optimization, we follow the same experiment procedure as in Eckmann et al. (2022)[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.3.1.

---

[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.

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

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.

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 OOD (Out-of-Distribution) generation.

Similarity-constrained Molecule Optimization. Adopting methodologies from JT-VAE (Jin et al., 2018) and LIMO (Eckmann et al., 2022), 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 and success rate of unsupervised methods decrease notably. The most considerable improvements of ChemSpace are observed when similarity

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)	**7.52 ± 6.29 (59.4)**	**5.70 ± 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)	**20.64 ± 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)	**9.70 ± 6.21 (94.4)**	6.14 ± 5.99 (70.9)	2.94 ± 4.34 (35.4)

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 6, Langevin dynamics also demonstrates a notably faster empirical convergence rate. We also report the performance in optimizing the QED property in Appendix D.7.

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 (Eckmann et al., 2022; Du
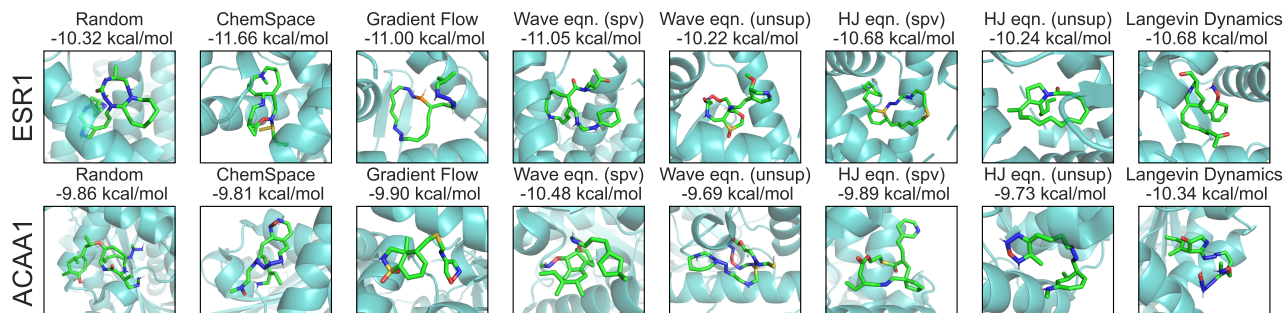
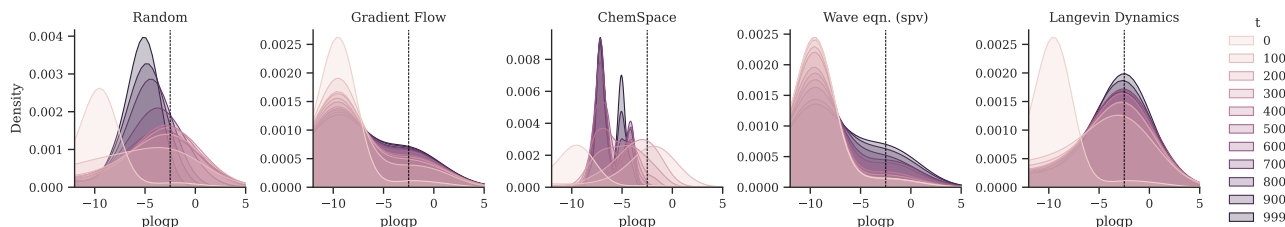*Figure 2.* **Visualization of generated ligands docked against target ESR1 and ACAA1.**

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

et al., 2023). 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.

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)

4.3. Molecule Manipulation

Molecule manipulation is a relatively new task proposed in Du et al. (2023) 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. (2023), we traverse 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 equa-

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

tion 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, as shown in Appendix Figure 4. 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 (Dudek et al., 2006), 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.

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.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Jensen, J. H. 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.

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

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

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

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

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

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

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

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

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

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

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

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

Madhawa, K., Ishiguro, K., Nakago, K., and Abe, M. Graph-nvp: An invertible flow model for generating molecular graphs. arXiv preprint arXiv:1905.11600, 2019.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Appendix for ChemFlow

A. Wasserstein Gradient Flow

As shown in the main paper, based on the dynamic formulation of optimal transport (Benamou & Brenier, 2000), the L_2 Wasserstein distance can be re-written as:

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

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 $\nabla \mathcal{F} = -\frac{d}{dt}\rho_t(\boldsymbol{z})$ (Ambrosio et al., 2005). 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:

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

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

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

On the other hand, the calculus of differential geometry gives

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

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

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

This scalar product coincides with the W_2 distance according to Benamou & Brenier (2000). Then eq. (20) can be similarly re-written as:

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

So the relation arises as:

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

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

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

The above derivations can be alternatively made by the poineering JKO scheme (Jordan et al., 1996). 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:

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

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

We would have exactly the heat equation:

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

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

$$\frac{d\,\rho_t(\boldsymbol{z})}{dt} = -\nabla \cdot \left(v_t(\boldsymbol{z})\rho_t(\boldsymbol{z})\right) = \frac{d}{d\boldsymbol{z}^2}\rho_t(\boldsymbol{z})$$
$$v_t(\boldsymbol{z}) = -\frac{\nabla\rho_t(\boldsymbol{z})}{\rho_t(\boldsymbol{z})} = -\nabla\log(\rho_t(\boldsymbol{z})) \tag{28}$$

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

Fokker Planck Equations. For the energy function defined as:

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

we would have the Fokker-Planck equation as

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

The velocity field can be similarly derived as

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

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

Porous Medium Equations. If we define the energy function as

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

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

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

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

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 $\uparrow$			QED $\uparrow$		
		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 (Du et al., 2022a). 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) (Gómez-Bombarelli et al., 2018; Jin et al., 2018), generative adversarial networks (GAN) (Guimaraes et al., 2017; Cao & Kipf, 2018), normalizing flows (NF) (Madhawa et al., 2019; Zang & Wang, 2020) and more recently diffusion models (Hoogeboom et al., 2022; Vignac et al., 2023; Jo et al., 2022). However, to respect the combinatorial nature of molecules, another line of work leverage combinatorial optimization to search new molecules including genetic algorithm (GA) (Jensen, 2019), Monte Carlo tree search (MCTS) (Yang et al., 2017), reinforcement learning (RL) (You et al., 2018a), 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 (Du et al., 2022a). 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 (Griffiths & Hernández-Lobato, 2020). For methods that do not explicitly reduce the dimensionality of data such as diffusion models, Schneuing et al. (2022) propose an evolutionary process to iteratively optimize the generated molecules. As it is observed that the learned latent space exhibits explicit structure (Gómez-Bombarelli et al., 2018), Du et al. (2023) 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 (Fu et al., 2022; Loeffler et al., 2023).

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 (Goetschalckx et al., 2019; Jahanian et al., 2020; Voynov & Babenko, 2020; Härkönen et al., 2020; Zhu et al., 2020; Peebles et al., 2020; Shen & Zhou, 2021; Song et al., 2022; 2023b;a;c). Here we highlight some representative supervised and unsupervised approaches. Supervised methods usually require pixel-wise annotations. InterfaceGAN (Shen et al., 2020) leverages face image pairs of different attributes to interpret disentangled latent representations of GANs. Jahanian et al. (2020) 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 & Babenko (2020) proposes to jointly learn a set of orthogonal directions and a classifier to learn the distinct interpretable directions. SeFa (Shen & Zhou, 2021) and HouseholderGAN (Song et al., 2023c) propose to use the eigenvectors of the (orthogonal) projection matrices as interpretable directions to traverse the latent space. More relevantly, Song et al. (2023b) proposes to use wave-like potential flows to model the spatiotemporal dynamics in the latent spaces of different generative models.

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** (Du et al., 2023): 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)** (Eckmann et al., 2022): a VAE-based generative model that encodes the input molecules into SELFIES (Krenn et al., 2019) 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.

D.2. Training Dataset

- **ChEMBL** (Zdrazil et al., 2023) 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** (Polykovskiy et al., 2018) is a benchmarking dataset derived from the ZINC (Irwin & Shoichet, 2005) 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 ~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.

- **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. (2018, 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 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 ring-penalty(m) is computed. Specifically, 3 implementations are widely used in various works.

Penalized by the length of the maximum cycle without normalization 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 ring-penalty(m) is normalized. MolDQN (Zhou et al., 2018) reported their results in this scheme.

Penalized by the length of the maximum cycle with normalization ring-penalty(m) is computed same as without normalization. MARS (Xie et al., 2021), HierVAE (Jin et al., 2020), GCPN (You et al., 2018b), and ours report plogP using this scheme.

Penalized by number of cycles As described by Jin et al. (2018, page 7 footnote 3), 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 $\phi^j(\cdot) \leftarrow$ MLP for $j = 1, \ldots, K$
 2: **repeat**
 3:     Sampling: $z_0 = f_\theta(x_0), t \sim \texttt{Categorical}(T), k \sim \texttt{Categorical}(K)$
 4:     **for** $i = 1, \ldots, t$ **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 (Misra, 2020) results 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 (Loshchilov & Hutter, 2017) 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

---

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**

---

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 90,000 random data and 10,000 unseen data for validation. For unsupervised settings, 10 disentangled potential energy functions are trained for 70 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.

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:

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

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:

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

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. 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.

Discrepancies of the supervised semantic guidance. Figure 4 illustrates the discrepancies between the output of the surrogate model and the ground truth as computed by the RDKit oracle. A large difference is observed between the two distributions. We believe that developing a more accurate surrogate model is essential for enhancing property prediction performance, which in turn would likely improve the efficacy of the optimization tasks. Addressing this improvement is designated as an area for future research.

Pearson correlation score for unsupervised diversity guidance with disentanglement regularization. Tables 6 and 7 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 5 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 6 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.

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 8 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
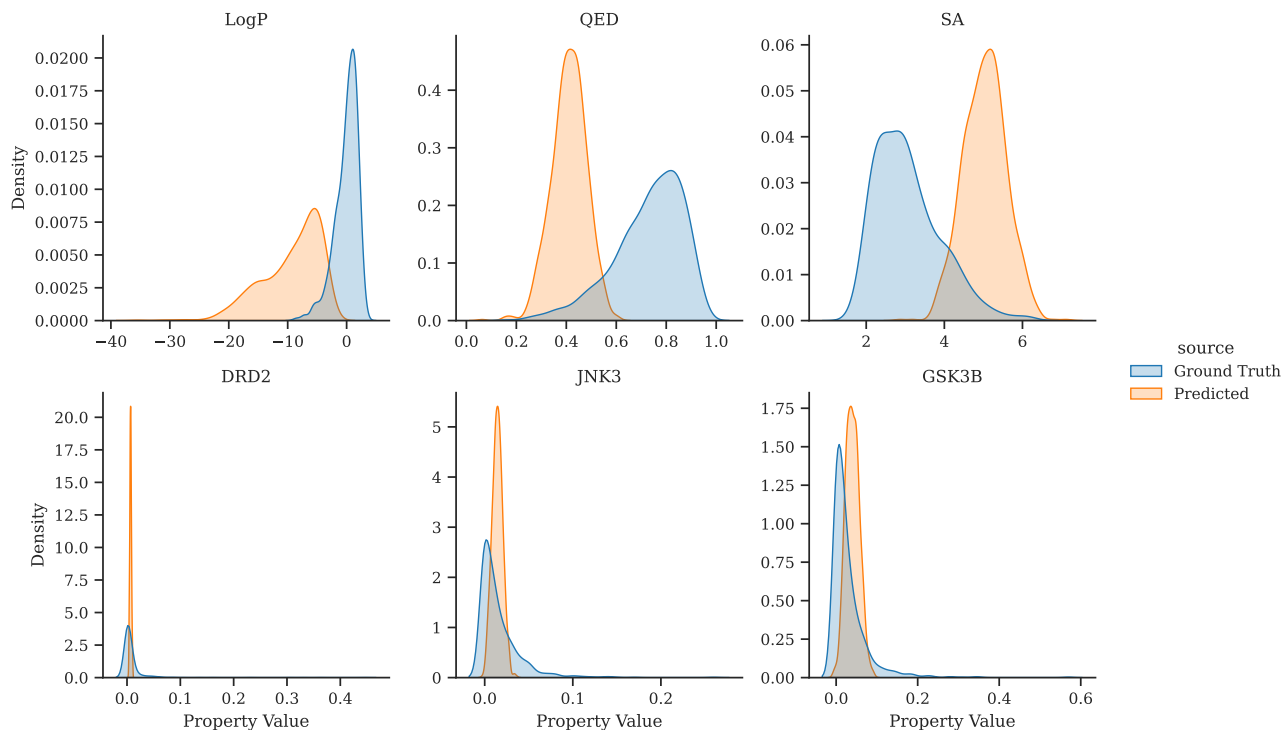
Figure 4. **Distribution for predicted properties and the ground truth.** We hypothesize there is a training and generalization errors in the surrogate model. We have observed the distribution of predicted and ground truth property values are different.

Table 6. **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 $\phi^k(z, t)$ using wave equations.

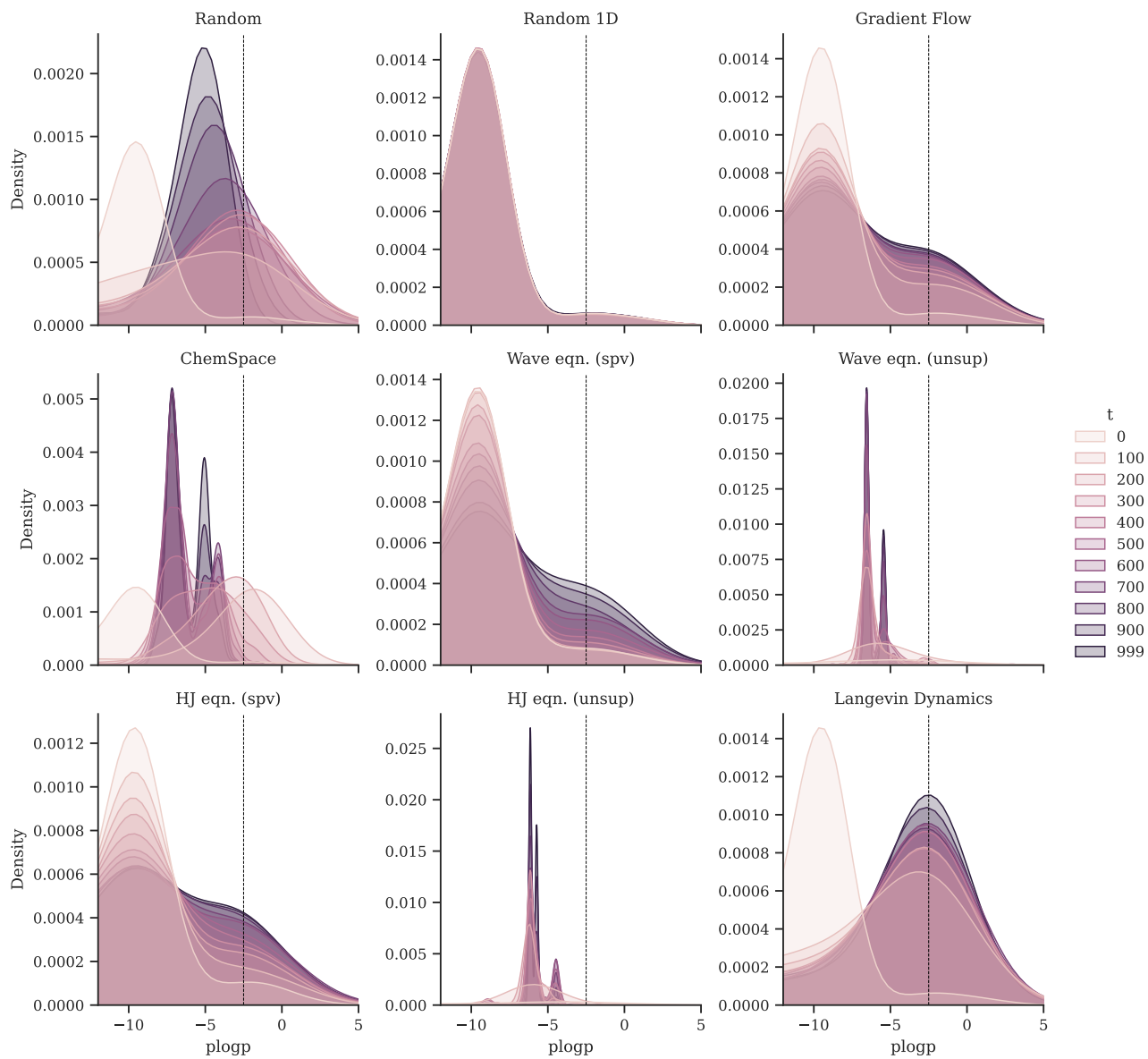|	PLOGP	SA	QED	DRD2	JNK3	JSK3B
0	0.019	0.003	0.016	0.029	0.015	0.051
1	0.160	-0.451	0.275	-0.074	-0.153	-0.272
2	0.035	-0.003	0.011	0.002	-0.006	0.017
3	0.072	-0.096	0.065	0.011	-0.017	-0.028
4	0.042	0.003	0.039	-0.010	0.025	0.018
5	-0.036	-0.022	0.150	0.009	-0.017	0.008
6	0.032	-0.045	0.006	0.002	-0.011	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

Figure 5. **Distribution shift for plogP optimization**

Table 7. **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 $\phi^k(z,t)$ 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
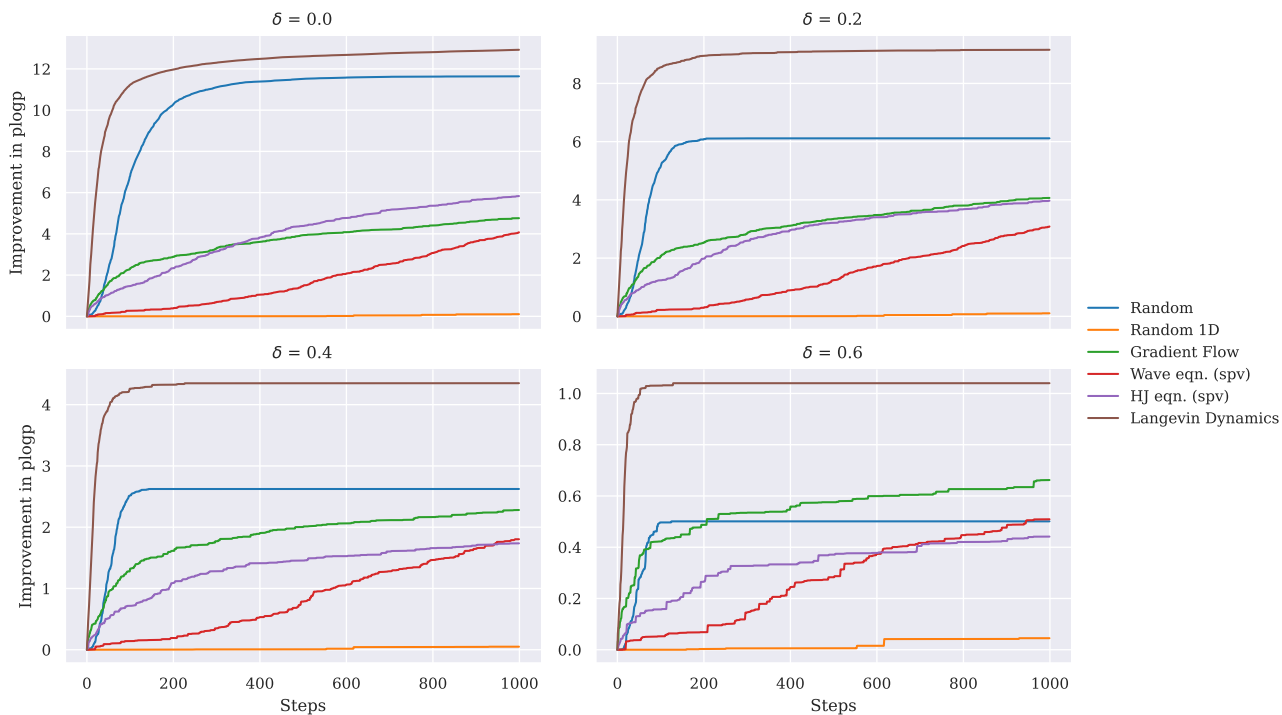
Figure 6. **Optimization Convergence** Langevin Dynamics shows faster convergence and achieves greater improvement in plogP.

issues with ChemSpace and two unsupervised methods, as illustrated in Figure 7, the performance of Langevin Dynamics surpasses other methods across various similarity levels, which is consistent with the result from Table 2.

Table 8. **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)**

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 8, 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 9 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 10. 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.

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.
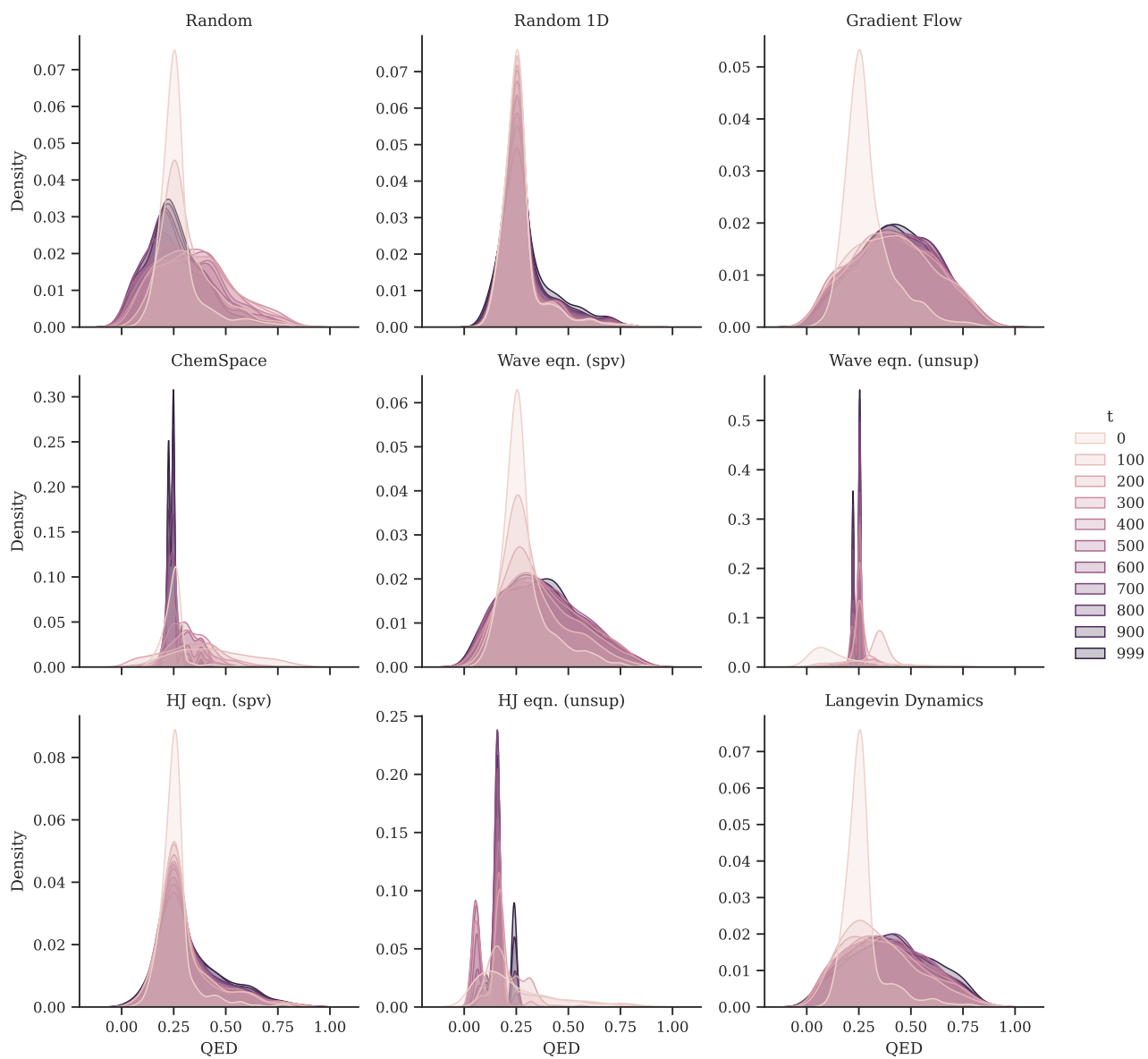
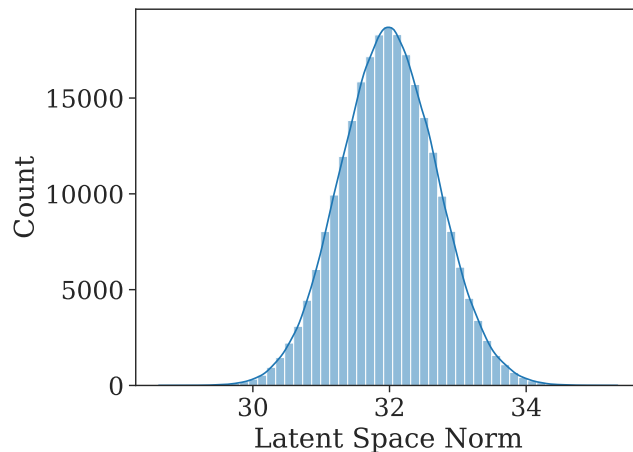*Figure 7.* **Distribution shift for QED optimization**

Figure 8. **Latent Vector Norm.** Distribution of the norm of the latent vectors projected from the training dataset onto the learned latent space.
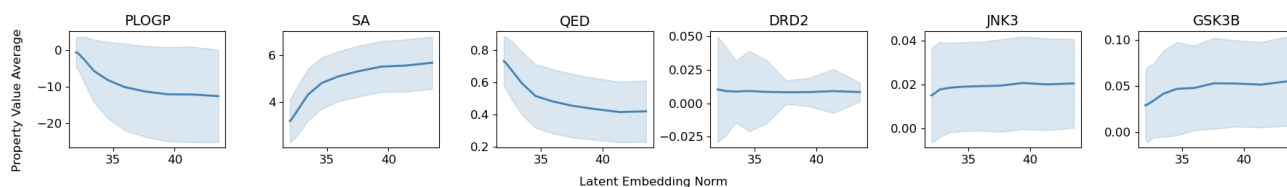

Figure 9. **Embedding Norm against Property Value of each path.** Norm and property value of molecules along the direction of latent traversal with a random direction. The middle curve shows the mean property value and latent embedding norm for all paths. The shaded area is the standard deviation of property value.
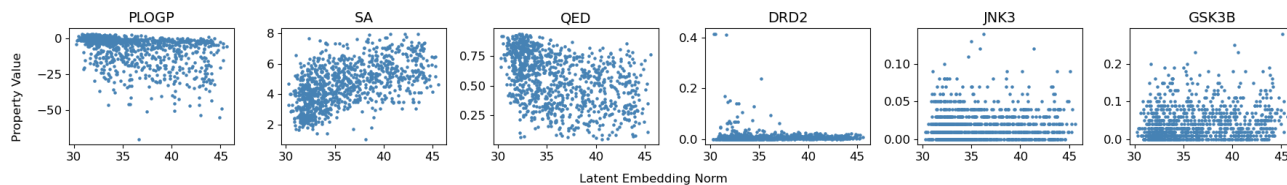

Figure 10. **Embedding Norm against Property Value of each Molecule.** Scatter plot of norm and property value of individual molecules in the training set encoded in the latent space.
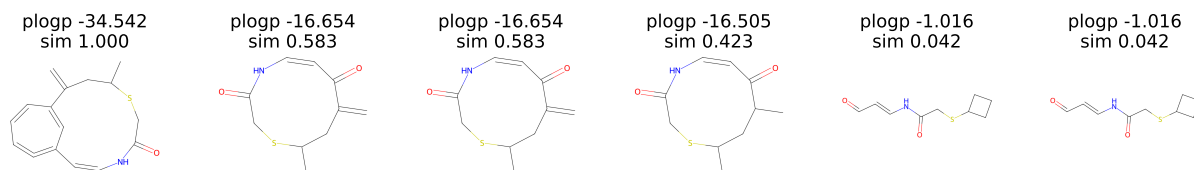

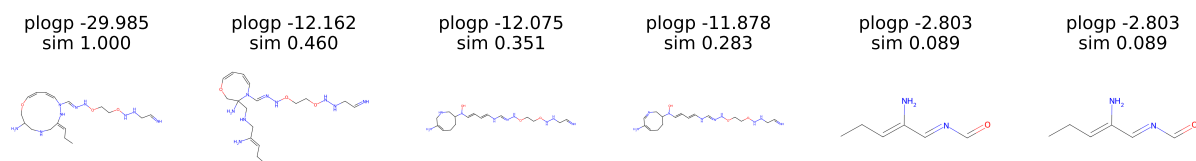*Figure 11.* **Molecule Manipulation Trajectory** Molecule manipulation by chemspace on plogP.

plogp -29.985
sim 1.000

plogp -12.162
sim 0.460

plogp -12.075
sim 0.351

plogp -11.878
sim 0.283

plogp -2.803
sim 0.089

plogp -2.803
sim 0.089

Figure 12. **Molecule Manipulation Trajectory** Molecule manipulation by gradient flow on plogP.

gsk3b 0.050
sim 1.000

gsk3b 0.100
sim 0.287

gsk3b 0.100
sim 0.287

gsk3b 0.140
sim 0.287

gsk3b 0.140
sim 0.210

gsk3b 0.140
sim 0.210
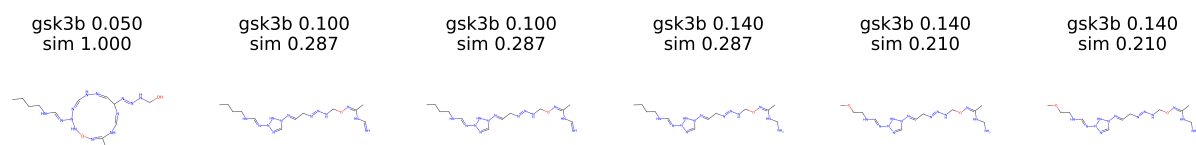
Figure 13. **Molecule Manipulation Trajectory** Molecule manipulation by supervised wave flow on GSK3B.

plogp -12.736
sim 1.000

plogp -12.126
sim 0.646

plogp -4.333
sim 0.246

plogp -4.310
sim 0.174

plogp -3.904
sim 0.174

plogp -3.904
sim 0.174
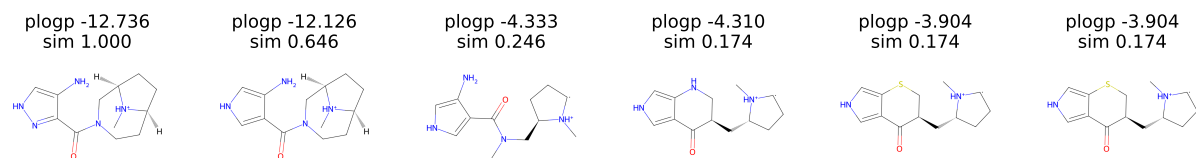
Figure 14. **Molecule Optimization Trajectory** Molecule optimization by random direction on plogP.

qed 0.215
sim 1.000

qed 0.678
sim 0.268

qed 0.678
sim 0.268

qed 0.699
sim 0.226

qed 0.726
sim 0.198

qed 0.726
sim 0.198
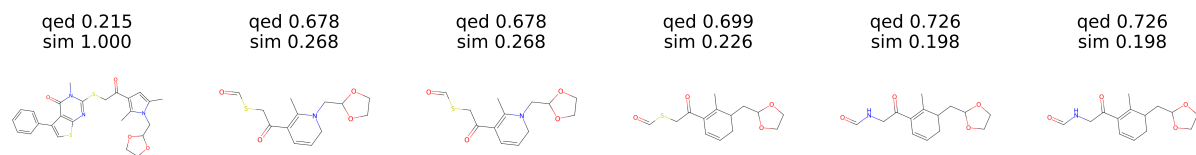
Figure 15. **Molecule Optimization Trajectory** Molecule optimization by supervised wave flow on QED.

qed 0.343
sim 1.000

qed 0.549
sim 0.596

qed 0.549
sim 0.596

qed 0.785
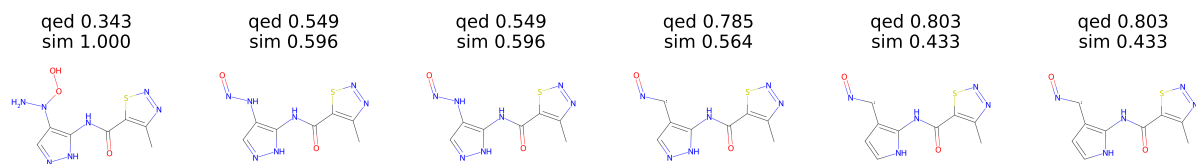sim 0.564

qed 0.803
sim 0.433

qed 0.803
sim 0.433

Figure 16. **Molecule Optimization Trajectory** Molecule optimization by supervised Hamilton-Jacobi flow on QED.