Deterministic Decoding for Discrete Data in Variational Autoencoders

Daniil Polykovskiy
Insilico Medicine

Dmitry Vetrov
National Research University Higher School of Economics

Abstract

Variational autoencoders are prominent generative models for modeling discrete data. However, with flexible decoders, they tend to ignore the latent codes. In this paper, we study a VAE model with a deterministic decoder (DD-VAE) for sequential data that selects the highest-scoring tokens instead of sampling. Deterministic decoding solely relies on latent codes as the only way to produce diverse objects, which improves the structure of the learned manifold. To implement DD-VAE, we propose a new class of bounded support proposal distributions and derive Kullback-Leibler divergence for Gaussian and uniform priors. We also study a continuous relaxation of deterministic decoding objective function and analyze the relation of reconstruction accuracy and relaxation parameters. We demonstrate the performance of DD-VAE on multiple datasets, including molecular generation and optimization problems.

1 Introduction

Variational autoencoder (Kingma and Welling, 2013) is an autoencoder-based generative model that provides high-quality samples in many data domains, including image generation (Razavi et al., 2019), natural language processing (Semeniuta et al., 2017), audio synthesis (Hsu et al., 2019), and drug discovery (Zhavoronkov et al., 2019).

Variational autoencoders use stochastic encoder and decoder. An encoder maps an object x onto a distribution of the latent codes $q_\phi(z \mid x)$, and a decoder produces a distribution $p_\theta(x \mid z)$ of objects that correspond to a given latent code. In this paper, we analyze the impact
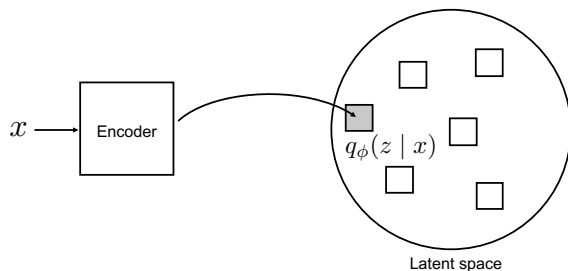

Figure 1: The encoder of DD-VAE outputs parameters of bounded support distribution. With Gaussian proposals, lossless auto-encoding is impossible, since the proposals of any two objects overlap.
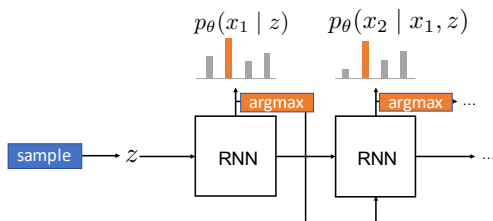

Figure 2: During sampling, the decoder selects $\arg\max$ of scores $p_\theta(x_i \mid x_{<i}, z)$. Hence, the only source of variation for the decoder is z. We propose a relaxed objective function to optimize through $\arg\max$.

of stochastic decoding on VAE models for discrete data and propose deterministic decoders as an alternative.

With complex stochastic decoders, such as PixelRNN (Oord et al., 2016), VAEs tend to ignore the latent codes, since the decoder is flexible enough to produce the whole data distribution $p(x)$ without using latent codes at all. Such behavior can damage the representation learning capabilities of VAE: we will not be able to use its latent codes for downstream tasks. A deterministic decoder, on the contrary, maps each latent code to a single data point, making it harder to ignore the latent codes, as they are the only source of variation.

One application of latent codes of VAEs is Bayesian optimization of molecular properties. Gómez-Bombarelli et al. (2018) trained a Gaussian process regressor on

the latent codes of VAE and optimized the latent codes to discover molecular structures with desirable properties. With stochastic decoding, a Gaussian process has to account for stochasticity in target variables, since every latent code corresponds to multiple molecular structures. Deterministic decoding, on the other hand, simplifies the regression task, leading to better predictive quality, as we show in the experiments.

Our contribution is three-fold:

- We formulate a model of deterministic decoder VAE (DD-VAE), derive its evidence lower bound and propose a convenient approximation with proven convergence to optimal parameters of non-relaxed objective;

- We show that lossless auto-encoding is impossible with full support proposal distributions and introduce bounded support distributions as a solution;

- We provide experiments on multiple datasets (synthetic, MNIST, MOSES, ZINC) to show that DD-VAE yields both a proper generative distribution and useful latent codes.

The code for reproducing the experiments is available at `https://github.com/insilicomedicine/DD-VAE`.

2 Deterministic Decoder VAE (DD-VAE)

In this section, we formulate a deterministic decoder variational autoencoder (DD-VAE). Next, we show the need for bounded support proposals and introduce them in Section 2.1. In Section 2.2, we propose a continuous relaxation of the DD-VAE's ELBO. In Section 2.3, we prove that the optimal solution of the relaxed problem matches the optimal solution of the original problem.

Variational autoencoder (VAE) consists of an encoder $q_\phi(z \mid x)$ and a decoder $p_\theta(x \mid z)$. The model learns a mapping of data distribution $p(x)$ onto a prior distribution of latent codes $p(z)$ which is often a standard Gaussian $\mathcal{N}(0, I)$. Parameters θ and ϕ are learned by maximizing a lower bound $\mathcal{L}(\theta, \phi)$ on log marginal likelihood $\log p(x)$. $\mathcal{L}(\theta, \phi)$ is known as an evidence lower bound (ELBO):

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{x \sim p(x)} \big[\mathbb{E}_{z \sim q_\phi(z|x)} \log p_\theta(x \mid z) \\ - \mathcal{KL}\left(q_\phi(z \mid x) \,\|\, p(z) \right) \big]. \tag{1}$$

The $\log p_\theta(x \mid z)$ term in Eq. 1 is a reconstruction loss, and the $\mathcal{KL}$ term is a Kullback-Leibler divergence that encourages latent codes to be marginally distributed as $p(z)$.

For sequence models, x is a sequence $x_1, x_2, \ldots, x_{|x|}$, where each token of the sequence is an element of a finite vocabulary V, and $|x|$ is the length of sequence x. A decoding distribution for sequences is often parameterized as a recurrent neural network that produces a probability distribution over each token x_i given the latent code and all previous tokens. The ELBO for such model is:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{x \sim p(x)} \Bigg[\mathbb{E}_{z \sim q_\phi(z|x)} \sum_{i=1}^{|x|} \log \pi_{x,i,x_i}^\theta(z) \\ - \mathcal{KL}\left(q_\phi(z \mid x) \,\|\, p(z) \right) \Bigg], \tag{2}$$

where $\pi_{x,i,s}^\theta(z) = p_\theta(x_i = s \mid z, x_1, x_2, \ldots, x_{i-1})$.
In deterministic decoders, we decode a sequence $\widetilde{x}_\theta(z)$ from a latent code z by taking a token with the highest score at each iteration:

$$\widetilde{x}_i = \underset{s \in V}{\arg\max}\, p_\theta(s \mid z, x_1, \ldots, x_{i-1}) = \underset{s \in V}{\arg\max}\, \pi_{x,i,s}^\theta(z) \tag{3}$$

To avoid ambiguity, when two tokens have the same maximal probability, $\arg\max$ is equal to a special "undefined" token that does not appear in the data. Such formulation simplifies derivations in the remaining of the paper. We also assume $\pi_{x,i,s}^\theta \in [0, 1]$ for convenience. After decoding $\widetilde{x}_\theta$, reconstruction term of ELBO is an indicator function which is one, if the model reconstructed a correct sequence, and zero otherwise:

$$p\left(x \mid \widetilde{x}_\theta\left(z \right) \right) = \begin{cases} 1, & \widetilde{x}_\theta(z) = x \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$\mathcal{L}_*(\theta, \phi) = \mathbb{E}_{x \sim p(x)} \big[\mathbb{E}_{z \sim q_\phi(z|x)} \log p\left(x \mid \widetilde{x}_\theta(z) \right) \\ - \mathcal{KL}\left(q_\phi(z \mid x) \,\|\, p(z) \right) \big]. \tag{5}$$

The $\mathcal{L}_*(\theta, \phi)$ is $-\infty$ if the model has non-zero reconstruction error rate, leading us to two questions: is $\mathcal{L}_*$ finite for some parameters (θ, ϕ) and how to optimize $\mathcal{L}_*$. We answer both questions in the following sections.

2.1 Proposal distributions with bounded support

In this section, we discuss bounded support proposal distributions $q_\phi(z \mid x)$ in VAEs and why they are crucial for deterministic decoders.

Variational Autoencoders often use Gaussian proposal distributions

$$q_\phi(z \mid x) = \mathcal{N}\left(z \mid \mu_\phi\left(x \right), \Sigma_\phi\left(x \right) \right), \tag{6}$$

where $\mu_\phi(x)$ and $\Sigma_\phi(x)$ are neural networks modeling the mean and the covariance matrix of the proposal distribution. For a fixed z, Gaussian density $q_\phi(z \mid x)$

is positive for any x. Hence, a lossless decoder has to decode every x from every z with a positive probability. However, a deterministic decoder can produce only a single data point $\widetilde{x}_\theta(z)$ for a given z, making reconstruction term of $\mathcal{L}_*$ minus infinity. To avoid this problem, we propose to use bounded support proposal distributions.

As bounded support proposal distributions, we suggest to use factorized distributions with marginals defined using a kernel K:

$$q_\phi(z \mid x) = \prod_{i=1}^{d} \frac{1}{\sigma_i^\phi(x)} K\left(\frac{z_i - \mu_i^\phi(x)}{\sigma_i^\phi(x)}\right), \qquad (7)$$

where $\mu_i^\phi(x)$ and $\sigma_i^\phi(x)$ are neural networks that model location and bandwidth of a kernel K; the support of i-th dimension of z in $q_\phi(z \mid x)$ is a range $\left[\mu_i^\phi(x) - \sigma_i^\phi(x), \mu_i^\phi(x) + \sigma_i^\phi(x)\right]$. We choose a kernel such that we can compute $\mathcal{KL}$ divergence between $q(z \mid x)$ and a prior $p(z)$ analytically. If $p(z)$ is factorized, $\mathcal{KL}$ divergence is a sum of one-dimensional $\mathcal{KL}$ divergences:

$$\mathcal{KL}\left(q_\phi(z \mid x) \,\|\, p(z)\right) = \sum_{i=1}^{d} \mathcal{KL}\left(q_\phi(z_i \mid x) \,\|\, p(z_i)\right). \qquad (8)$$

In Table 1, we show $\mathcal{KL}$ divergence for some bounded support kernels and illustrate their densities in Figure 3. Note that the form of $\mathcal{KL}$ divergence is very similar to the one for a Gaussian proposal distribution—they only differ in a constant multiplier for σ^2 and an additive constant. For sampling, we use rejection sampling from $K(\epsilon)$ with a uniform proposal $K(0) \cdot \mathcal{U}[-1, 1]$ and apply a reparametrization trick to obtain a final sample: $z = \epsilon \cdot \sigma + \mu$. The acceptance rate in such sampling is $\frac{1}{2K(0)}$. Hence, to sample a batch of size N, we sample $N \cdot 2K(0)$ objects and repeat sampling until we get at least N accepted samples. We also store a buffer with excess samples and use them in the following batches.
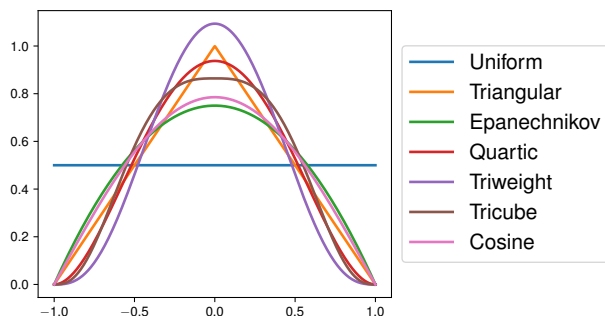

Figure 3: Bounded support proposals with $\mu = 0$ and $\sigma = 1$ for which we derived $\mathcal{KL}$ divergence.

With bounded support proposals, we can use a uniform distribution $U[-1, 1]^d$ as a prior in VAE as long as the support of $q_\phi(z \mid x)$ lies inside the support of a prior distribution. In practice, we ensure this by transforming μ and σ from the encoder into μ' and σ' using the following transformation:

$$\mu' = \frac{\tanh(\mu + \sigma) + \tanh(\mu - \sigma)}{2}, \qquad (9)$$

$$\sigma' = \frac{\tanh(\mu + \sigma) - \tanh(\mu - \sigma)}{2}. \qquad (10)$$

We report derived $\mathcal{KL}$ divergences for a uniform prior in Table 2.

For discrete data, with bounded support proposals we can ensure that for sufficiently flexible encoder and decoder, there exists a set of parameters (θ, ϕ) for which proposals $q_\phi(z \mid x)$ do not overlap for different x, and hence ELBO $\mathcal{L}_*(\theta, \phi)$ is finite. For example, we can enumerate all objects and map i-th object to a range $[i, i + 1]$.

2.2 Approximating ELBO

In this section, we discuss how to optimize a discontinuous function $\mathcal{L}_*(\theta, \phi)$ by approximating it with a smooth function. We also show the convergence of optimal parameters of an approximated ELBO to the optimal parameters of the original function in the next section.

We start by equivalently defining $\arg\max$ from Eq. 3 for some array r:

$$\mathbb{I}\left[i = \arg\max_j r_j\right] = \prod_{j \neq i} \mathbb{I}\left[r_i > r_j\right] \qquad (11)$$

We approximate Eq. 11 by introducing a smooth relaxation $\sigma_\tau(x)$ of an indicator function $\mathbb{I}[x > 0]$ parameterized with a temperature parameter $\tau \in (0, 1)$:

$$\mathbb{I}[x > 0] \approx \sigma_\tau(x) = \frac{1}{1 + \exp\left(-x/\tau\right)\left[\frac{1}{\tau} - 1\right]} \qquad (12)$$

Note that $\sigma_\tau(x)$ converges to $\mathbb{I}[x > 0]$ pointwise. In Figure 4 we show function $\sigma_\tau(x)$ for different values of τ. Substituting $\arg\max$ with the proposed relaxation, we get the following approximation of the evidence lower bound:

$$\mathcal{L}_\tau(\theta, \phi) = \mathbb{E}_{x \sim p(x)}\Bigg[$$

$$\mathbb{E}_{z \sim q_\phi(z|x)} \sum_{i=1}^{|x|} \sum_{s \neq x_i} \log \sigma_\tau\left(\pi_{x,i,x_i}^\theta(z) - \pi_{x,i,s}^\theta(z)\right)$$

$$- \mathcal{KL}\left(q_\phi(z \mid x) \,\|\, p(z)\right)\Bigg]$$

$$(13)$$

Table 1: $\mathcal{KL}$ divergence between a proposal q with support $|z - \mu| \leq \sigma$ and prior $\mathcal{N}(0,1)$.

Kernel	$q(z \mid \sigma, \mu)$	$\mathcal{KL}\left(q\left(z \mid \sigma, \mu\right) \| \mathcal{N}\left(0,1\right)\right)$
Uniform	$\frac{1}{2\sigma}$	$\frac{1}{2}\mu^2 + \frac{1}{6}\sigma^2 - \log\sigma + \frac{1}{2}\log(2\pi) - \log 2$
Triangular	$\frac{1}{\sigma}\left(1 - \left|\frac{z-\mu}{\sigma}\right|\right)$	$\frac{1}{2}\mu^2 + \frac{1}{12}\sigma^2 - \log\sigma + \frac{1}{2}\log(2\pi) - \frac{1}{2}$
Epanechnikov	$\frac{3}{4\sigma}\left(1 - \frac{(z-\mu)^2}{\sigma^2}\right)$	$\frac{1}{2}\mu^2 + \frac{1}{10}\sigma^2 - \log\sigma + \frac{1}{2}\log(2\pi) - \frac{5}{3} + \log 3$
Quartic	$\frac{15}{16\sigma}\left(1 - \frac{(z-\mu)^2}{\sigma^2}\right)^2$	$\frac{1}{2}\mu^2 + \frac{1}{14}\sigma^2 - \log\sigma + \frac{1}{2}\log(2\pi) - \frac{47}{15} + \log 15$
Triweight	$\frac{35}{32\sigma}\left(1 - \frac{(z-\mu)^2}{\sigma^2}\right)^3$	$\frac{1}{2}\mu^2 + \frac{1}{18}\sigma^2 - \log\sigma + \frac{1}{2}\log(2\pi) - \frac{319}{70} + \log 70$
Tricube	$\frac{70}{81\sigma}\left(1 - \frac{|z-\mu|^3}{\sigma^3}\right)^3$	$\frac{1}{2}\mu^2 + \frac{35}{486}\sigma^2 - \log\sigma + \frac{1}{2}\log\left(2\pi\right) + \frac{\pi\sqrt{3}}{2} - \frac{1111}{140} + \log 70\sqrt{3}$
Cosine	$\frac{\pi}{4\sigma}\cos\left(\frac{\pi(z-\mu)}{2\sigma}\right)$	$\frac{1}{2}\mu^2 + \left(\frac{1}{2} - \frac{4}{\pi^2}\right)\sigma^2 - \log\sigma + \frac{1}{2}\log\left(2\pi\right) - 1 + \log\frac{\pi}{2}$
Gaussian $(-\infty, \infty)$	$\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2\sigma^2}(z-\mu)^2}$	$\frac{1}{2}\mu^2 + \frac{1}{2}\sigma^2 - \log\sigma - \frac{1}{2}$

Table 2: $\mathcal{KL}$ divergence between a proposal q with support $|z - \mu| \leq \sigma$ and a uniform prior $\mathcal{U}[-1, 1]$ if the support of q lies in $[-1, 1]$.

Kernel	$\mathcal{KL}\left(q\left(z \mid \sigma, \mu\right) \| \mathcal{U}\left[-1, 1\right]\right)$
Uniform	$-\log\sigma$
Triangular	$-\frac{1}{2} + \log 2 - \log\sigma$
Epanechnikov	$-\frac{5}{3} + \log 6 - \log\sigma$
Quartic	$-\frac{47}{15} + \log 30 - \log\sigma$
Triweight	$-\frac{319}{70} + \log 140 - \log\sigma$
Tricube	$-\frac{1111}{140} + \frac{\pi\sqrt{3}}{2} + \log 140\sqrt{3} - \log\sigma$
Cosine	$-1 + \log\pi - \log\sigma$

Figure 4: Relaxation $\sigma_\tau(x)$ of an indicator function $\mathbb{I}[x > 0]$ for different τ.

A proposed $\mathcal{L}_\tau$ is finite for $0 < \tau < 1$ and converges to $\mathcal{L}_*$ pointwise. In the next section, we formulate the theorem that shows that if we gradually decrease temperature τ and solve maximization problem for ELBO $\mathcal{L}_\tau$, we will converge to optimal parameters of a non-relaxed ELBO $\mathcal{L}_*$.
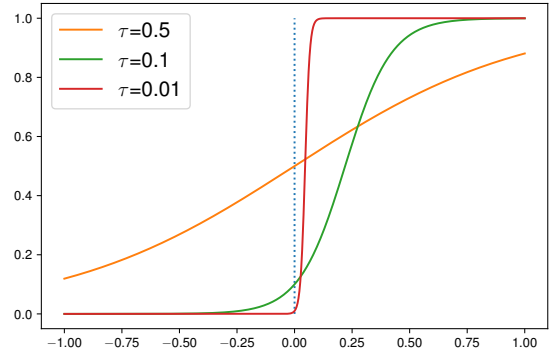
2.3 Convergence of optimal parameters of $\mathcal{L}_\tau$ to optimal parameters of $\mathcal{L}_*$

In this section, we introduce auxiliary functions that are useful for assessing the quality of the model and formulate a theorem on the convergence of optimal parameters of $\mathcal{L}_\tau$ to optimal parameters of $\mathcal{L}_*$.

Denote $\Delta(\widetilde{x}_\theta, \phi)$ a sequence-wise error rate for a given encoder and decoder:

$$\Delta(\widetilde{x}_\theta, \phi) = \mathbb{E}_{x \sim p(x)}\mathbb{E}_{z \sim q_\phi(z|x)}\mathbb{I}\left[\widetilde{x}_\theta(z) \neq x\right]. \quad (14)$$

For a given ϕ, we can find an optimal decoder and a corresponding sequence-wise error rate $\Delta(\phi)$ by rearranging the terms in Eq. 14 and applying importance sampling:

$$\Delta(\widetilde{x}_\theta, \phi) = 1 - \mathbb{E}_{z \sim p(z)}\mathbb{E}_{x \sim p(x)}\frac{q_\phi(z \mid x)}{p(z)}\mathbb{I}\left[\widetilde{x}_\theta(z) = x\right]$$

$$= 1 - \mathbb{E}_{z \sim p(z)}\frac{p(\widetilde{x}_\theta(z))q_\phi(z \mid \widetilde{x}_\theta(z))}{p(z)}$$

$$\geq 1 - \mathbb{E}_{z \sim p(z)}\frac{p(\widetilde{x}_\phi^*(z))q_\phi(z \mid \widetilde{x}_\phi^*(z))}{p(z)}$$

$$= \Delta(\widetilde{x}_\phi^*, \phi) = \Delta(\phi) \geq 0, \quad (15)$$

where $\widetilde{x}_\phi^*(z)$ is an optimal decoder given by:

$$\widetilde{x}_\phi^*(z) \in \underset{x \in \chi}{\text{Arg max}}\, p(x)q_\phi(z \mid x). \quad (16)$$

Here, χ is a set of all possible sequences. Denote Ω a set of parameters for which ELBO $\mathcal{L}_*$ is finite:

$$\Omega = \{(\theta, \phi) \mid \mathcal{L}_*(\theta, \phi) > -\infty\} \quad (17)$$

Theorem 1. *Assume that $\Omega \neq \emptyset$, length of sequences in χ is bounded ($\exists L : |x| \leq L, \forall x \in \chi$), and Θ and Φ*

are compact sets of possible parameter values. Assume that $q_\phi(z \mid x)$ is equicontinuous in total variation for any ϕ and x:

$$\forall \epsilon > 0, \exists \delta = \delta(\epsilon, x, \phi) > 0 :$$

$$\|\phi - \phi'\| < \delta \Rightarrow \int |q_\phi(z \mid x) - q_{\phi'}(z \mid x)| \, dz < \epsilon. \tag{18}$$

Let τ_n, ϕ_n, θ_n be such sequences that:

$$\lim_{n \to \infty} \tau_n = 0, \quad \tau_n \in (0, 1), \tag{19}$$

$$(\theta_n, \phi_n) \in \underset{\theta \in \Theta, \phi \in \Phi}{\text{Arg max}} \, \mathcal{L}_{\tau_n}(\theta, \phi), \tag{20}$$

sequence $\{\phi_n\}$ converges to $\widetilde{\phi}$, and for any ϕ such that $\Delta(\phi) = 0$ exists θ such that $\Delta(\widetilde{x}_\theta, \phi) = 0$. Let $\widetilde{\theta}$ be:

$$\widetilde{\theta} \in \underset{\theta \in \Theta}{\text{Arg max}} \, \mathcal{L}_*(\theta, \widetilde{\phi}). \tag{21}$$

Then the sequence-wise error rate decreases asymptotically as

$$\Delta(\widetilde{x}_{\theta_n}, \phi_n) = \mathcal{O}\left(\frac{1}{\log(1/\tau_n)}\right), \tag{22}$$

$\Delta(\widetilde{\phi}) = 0$, and final parameters $(\widetilde{\theta}, \widetilde{\phi})$ solve the optimization problem for $\mathcal{L}_*$:

$$\mathcal{L}_*(\widetilde{\theta}, \widetilde{\phi}) = \sup_{\theta \in \Theta, \phi \in \Phi} \mathcal{L}_*(\theta, \phi). \tag{23}$$

Proof. See Appendix A. □

The maximum length of sequences is bounded in the majority of practical applications. Equicontinuity assumption is satisfied for all distributions we considered in Table 1 if μ and σ depend continuously on ϕ for all $x \in \chi$. Ω is not empty for bounded support distributions when encoder and decoder are sufficiently flexible, as discussed in Section 2.1.

Eq. 21 suggests that after we finish training the autoencoder, we should fix the encoder and fine-tune the decoder. Since $\Delta(\widetilde{\phi}) = 0$, the optimal stochastic decoder for such ϕ is deterministic—any z corresponds to a single x except for a zero probability subset. In theory, we could learn $\widetilde{\theta}$ for a fixed $\widetilde{\phi}$ by optimizing a reconstruction term of ELBO from Eq. 2:

$$\mathcal{L}_{\text{rec}}(\theta) = \mathbb{E}_{x \sim p(x)} \mathbb{E}_{z \sim q_{\widetilde{\phi}}(z|x)} \sum_{i=1}^{|x|} \log \pi_{x,i,x_i}^\theta, \tag{24}$$

but since in practice we do not anneal the temperature exactly to zero, we found such fine-tuning optional.

3 Related Work

Autoencoder-based generative models consist of an encoder-decoder pair and a regularizer that forces encoder outputs to be marginally distributed as a prior distribution. This regularizer can take a form of a $\mathcal{KL}$ divergence as in Variational Autoencoders (Kingma and Welling, 2013) or an adversarial loss as in Adversarial Autoencoders (Makhzani et al., 2016) and Wasserstein Autoencoders (Tolstikhin et al., 2016). Besides autoencoder-based generative models, generative adversarial networks (Goodfellow et al., 2014) and normalizing flows (Dinh et al., 2015, 2017) were shown to be useful for sequence generation (Yu et al., 2017; van den Oord et al., 2018).

Variational autoencoders are prone to posterior collapse when the encoder outputs a prior distribution, and a decoder learns the whole distribution $p(x)$ by itself. Posterior collapse often occurs for VAEs with autoregressive decoders such as PixelRNN (Oord et al., 2016). Multiple approaches were proposed to tackle posterior collapse, including decreasing the weight β of a $\mathcal{KL}$ divergence (Higgins et al., 2017), or encouraging high mutual information between latent codes and corresponding objects (Zhao et al., 2019).

Other approaches modify a prior distribution, making it more complex than a proposal: a Gaussian mixture model (Tomczak and Welling, 2018; Kuznetsov et al., 2019), autoregressive priors (Chen et al., 2017), or training a deterministic encoder and obtaining prior with a kernel density estimation (Ghosh et al., 2020). Unlike these approaches, we conform to the standard Gaussian prior, and study the required properties of encoder and decoder to achieve deterministic decoding.

Deep generative models became a prominent approach in drug discovery as a way to rapidly discover potentially active molecules (Polykovskiy et al., 2018b; Zhavoronkov et al., 2019). Recent works explored feature-based (Kadurin et al., 2017), string-based (Gómez-Bombarelli et al., 2018; Segler et al., 2018), and graph-based (Jin et al., 2018; De Cao and Kipf, 2018; You et al., 2018) generative models for molecular structures. In this paper, we use a simplified molecular-input line-entry system (SMILES) (Weininger, 1970; Weininger et al., 1989) to represent the molecules—a system that represents a molecular graph as a string using a depth-first search order traversal. Multiple algorithms were proposed to exploit SMILES structure using formal grammars (Kusner et al., 2017; Dai et al., 2018).

4 Experiments

We experiment on four datasets: synthetic and MNIST datasets to visualize a learned manifold structure, on

(a) DD-VAE. U prior; U proposal   (b) DD-VAE. U prior; T proposal   (c) VAE. G prior; G proposal
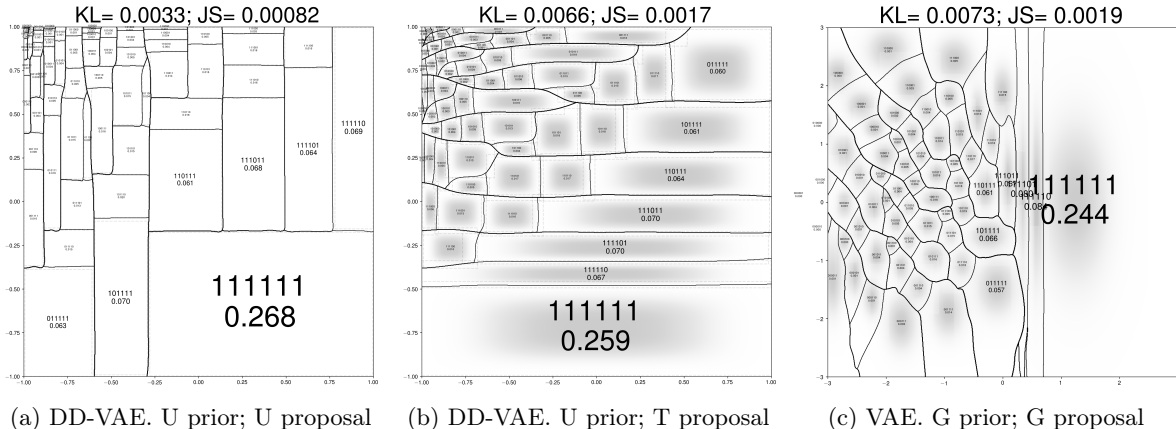
Figure 5: Learned 2D manifold on synthetic data. Dashed lines indicate proposal boundaries, solid lines indicate decoding boundaries. For each decoded string, we write its probability under deterministic decoding. Left and middle images: DD-VAE; Right image: VAE. U = Uniform, G = Gaussian, T = Tricube.

MOSES molecular dataset to analyze the distribution quality of DD-VAE, and ZINC dataset to see if DD-VAE's latent codes are suitable for goal-directed optimization. We describe model hyperparameters in Appendix B.

4.1 Synthetic data

This dataset provides a proof of concept comparison of standard VAE with a stochastic decoder and a DD-VAE model with a deterministic decoder. The data consist of 6-bit strings, a probability of each string is given by independent Bernoulli samples with a probability of 1 being 0.8. For example, a probability of string "110101" is $0.8^4 \cdot 0.2^2 \approx 0.016$.

In Figure 5, we illustrate the 2D latent codes learned with the proposed model. As an encoder and decoder, we used a 2-layer gated recurrent unit (GRU) (Cho et al., 2014) network with a hidden size 128. We provide illustrations for a proposed model with a uniform prior and compare uniform and tricube proposals. For a baseline model, we trained a β-VAE with Gaussian proposal and prior. We used $\beta = 0.1$, as for larger β we observed posterior collapse. For our model, we used $\beta = 1$, which is equivalent to the described model.

For a baseline model, we observe an irregular decision boundary, which also behaves unpredictably for latent codes that are far from the origin. Both uniform and tricube proposals learn a brick-like structure that covers the whole latent space. During training, we observed that the uniform proposal tends to separate proposal distributions by a small margin to ensure there is no overlap between them. As the training continues, the width of proposals grows until they cover the whole space. For the tricube proposal, we observed a similar

behavior, although the model tolerates slight overlaps.

4.2 Binary MNIST

To evaluate the model on imaging data, we considered a binarized MNIST (LeCun and Cortes, 2010) dataset obtained by thresholding the original 0 to 1 gray-scale images by a threshold of 0.3. The goal of this experiment is to visualize how DD-VAE learns 2D latent codes on moderate size datasets.

For this experiment, we trained a 4-layer fully-connected encoder and decoder with structure $784 \rightarrow 256 \rightarrow 128 \rightarrow 32 \rightarrow 2$. In Figure 6, we show learned latent space structure for a baseline VAE with Gaussian prior and proposal and compare it to a DD-VAE with uniform prior and proposal. Note that the uniform representation evenly covers the latent space, as all points have the same prior probability. This property is useful for visualization tasks. The learned structure better separates classes, although it was trained in an unsupervised manner: K-nearest neighbor classifier on 2D latent codes yields 87.8% accuracy for DD-VAE and 86.1% accuracy for VAE.

4.3 Molecular sets (MOSES)

In this section, we compare the models on a distribution learning task on MOSES dataset (Polykovskiy et al., 2018a). MOSES dataset contains approximately 2 million molecular structures represented as SMILES strings (Weininger, 1970; Weininger et al., 1989); MOSES also implements multiple metrics, including Similarity to Nearest Neighbor (SNN/Test) and Fréchet ChemNet Distance (FCD/Test) (Preuer et al., 2018). SNN/Test is an average Tanimoto similarity of generated molecules to the closest molecule from

Table 3: Distribution learning with deterministic decoding on MOSES dataset. We report generative modeling metrics: FCD/Test (lower is better) and SNN/Test (higher is better). Mean ± std over multiple runs. G = Gaussian proposal, T = Triweight proposal.

METHOD	FCD/TEST ($\downarrow$)			SNN/TEST ($\uparrow$)		
	70%	80%	90%	70%	80%	90%
VAE (G)	0.205 ± 0.005	0.344 ± 0.003	0.772 ± 0.007	0.550 ± 0.001	0.525 ± 0.001	0.488 ± 0.001
VAE (T)	0.207 ± 0.004	0.335 ± 0.005	0.753 ± 0.019	0.550 ± 0.001	0.526 ± 0.001	0.490 ± 0.000
DD-VAE (G)	0.198 ± 0.012	0.312 ± 0.011	0.711 ± 0.020	$\mathbf{0.555} \pm \mathbf{0.001}$	0.531 ± 0.001	0.494 ± 0.001
DD-VAE (T)	$\mathbf{0.194} \pm \mathbf{0.001}$	$\mathbf{0.311} \pm \mathbf{0.010}$	$\mathbf{0.690} \pm \mathbf{0.010}$	$\mathbf{0.555} \pm \mathbf{0.000}$	$\mathbf{0.532} \pm \mathbf{0.001}$	$\mathbf{0.495} \pm \mathbf{0.001}$
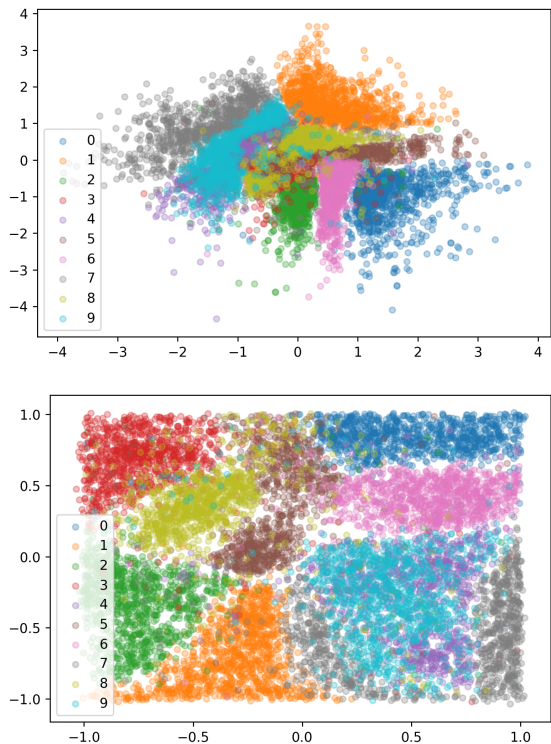
Figure 6: Learned 2D manifold on binarized MNIST obtained as proposal means on the test set. **Top:** VAE with Gaussian prior and Gaussian proposal; **Bottom:** DD-VAE with uniform prior and uniform proposal.

the test set. Hence, SNN acts as precision and is high if generated molecules lie on the test set's manifold. FCD/Test computes Fréchet distance between activations of a penultimate layer of ChemNet for generated and test sets. Lower FCD/Test indicates a closer match of generated and test distributions.

In this experiment, we monitor the model's behavior for high reconstruction accuracy. We trained a 2-layer GRU encoder and decoder with 512 neurons and a latent dimension 64 for both VAE and DD-VAE. We pretrained the models with such β that the sequence-

wise reconstruction accuracy was approximately 95%. We monitored FCD/Test and SNN/Test metrics while gradually increasing β until sequence-wise reconstruction accuracy dropped below 70%.

In the results reported in Table 3, DD-VAE outperforms VAE on both metrics. Bounded support proposals have less impact on the target metrics, although they slightly improve both FCD/Test and SNN/Test.

4.4 Bayesian Optimization

A standard use case for generative molecular autoencoders for molecules is Bayesian Optimization (BO) of molecular properties on latent codes (Gómez-Bombarelli et al., 2018). For this experiment, we trained a 1-layer GRU encoder and decoder with 1024 neurons on ZINC with latent dimension 64. We tuned hyperparameters such that the sequence-wise reconstruction accuracy on train set was close to 96% for all our models. The models showed good reconstruction accuracy on test set and good validity of the samples (Table 4). We explored the latent space using a standard two-step validation procedure proposed in (Kusner et al., 2017) to show the advantage of DD-VAE's latent codes. The goal of the Bayesian optimization was to maximize the following score of a molecule m:

$$\text{score}(m) = \log\text{P}(m) - \text{SA}(m) - \text{cycle}(m), \quad (25)$$

where $\log\text{P}(m)$ is octanol-water partition coefficient of a molecule, $\text{SA}(m)$ is a synthetic accessibility score (Ertl and Schuffenhauer, 2009) obtained from RDKit package (Landrum, 2006), and $\text{cycle}(m)$ penalizes the largest ring $R_{\max}(m)$ in a molecule if it consists of more than 6 atoms:
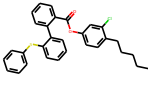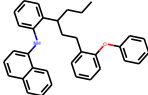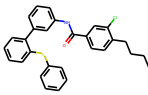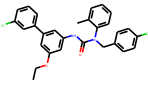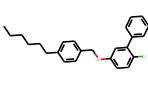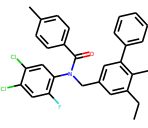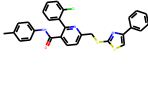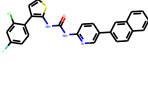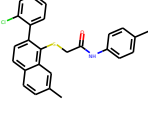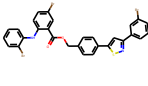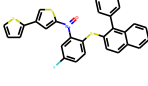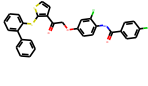
$$\text{cycle}(m) = \max(0, |R_{\max}(m)| - 6). \quad (26)$$

Each component in $\text{score}(m)$ is normalized by subtracting mean and dividing by standard deviation estimated on the training set. Validation procedure consists of two steps. First, we train a sparse Gaussian process (Snelson and Ghahramani, 2006) on latent codes of DD-VAE trained on approximately 250,000 SMILES

Table 4: Reconstruction accuracy (sequence-wise) and validity of samples on ZINC dataset; Predictive performance of sparse Gaussian processes on ZINC dataset: Log-likelihood (LL) and Root-mean-squared error (RMSE); Scores of top 3 molecules found with Bayesian Optimization. G = Gaussian proposal, T = Tricube proposal.

METHOD	RECONSTRUCTION	VALIDITY	LL	RMSE	TOP1	TOP2	TOP3
CVAE	44.6%	0.7%	-1.812 ± 0.004	1.504 ± 0.006	1.98	1.42	1.19
GVAE	53.7%	7.2%	-1.739 ± 0.004	1.404 ± 0.006	2.94	2.89	2.80
SD-VAE	76.2%	43.5%	-1.697 ± 0.015	1.366 ± 0.023	4.04	3.50	2.96
JT-VAE	76.7%	100.0%	-1.658 ± 0.023	1.290 ± 0.026	5.30	4.93	4.49
VAE (G)	87.01%	78.32%	-1.558 ± 0.019	1.273 ± 0.050	5.76	5.74	**5.67**
VAE (T)	90.3%	73.52%	-1.562 ± 0.022	1.265 ± 0.051	5.41	5.38	5.35
DD-VAE (G)	89.39%	63.07%	-1.481 ± 0.020	1.199 ± 0.050	5.13	4.84	4.80
DD-VAE (T)	89.89%	61.38%	$\mathbf{-1.470 \pm 0.022}$	$\mathbf{1.186 \pm 0.053}$	**5.86**	**5.77**	5.64

Table 5: Best molecules found using Bayesian Optimization.

TOP1	TOP2	TOP3
VAE, GAUSSIAN		

VAE, TRICUBE

DD-VAE, GAUSSIAN

DD-VAE, TRICUBE

strings from ZINC database, and report predictive performance of a Gaussian process on a ten-fold cross validation in Table 4. We compare DD-VAE to the following baselines: Character VAE, CVAE (Gómez-Bombarelli et al., 2018); Grammar VAE, GVAE (Kusner et al., 2017); Syntax-Directed VAE, SD-VAE (Dai et al., 2018); Junction Tree VAE, JT-VAE (Jin et al., 2018).

Using a trained sparse Gaussian process, we iteratively sampled 60 latent codes using expected improvement acquisition function and Kriging Believer Algorithm (Cressie, 1990) to select multiple points for the batch. We evaluated selected points and added reconstructed objects to the training set. We repeated training and sampling for 5 iterations and reported molecules with the highest score in Table 4 and Table 5. We also report top 50 molecules for our models in Appendix D.

5 Discussion

The proposed model outperforms the standard VAE model on multiple downstream tasks, including Bayesian optimization of molecular structures. In the ablation studies, we noticed that models with bounded support show lower validity during sampling. We suggest that it is due to regions of the latent space that are not covered by any proposals: the decoder does not visit these areas during training and can behave unexpectedly there. We found a uniform prior suitable for downstream classification and visualization tasks since latent codes evenly cover the latent space.

DD-VAE introduces an additional hyperparameter τ that balances reconstruction and $\mathcal{KL}$ terms. Unlike $\mathcal{KL}$ scale β, temperature τ changes loss function and its gradients non-linearly. We found it useful to select starting temperatures such that gradients from $\mathcal{KL}$ and reconstruction term have the same scale at the beginning of training. Experimenting with annealing schedules, we found log-linear annealing slightly better than linear annealing.

Acknowledgements

References

Chen, X., Kingma, D. P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., Sutskever, I., and Abbeel, P. (2017). Variational Lossy Autoencoder. *International Conference on Learning Representations*.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Cressie, N. (1990). The origins of kriging. *Mathematical geology*, 22(3):239–252.

Dai, H., Tian, Y., Dai, B., Skiena, S., and Song, L. (2018). Syntax-directed variational autoencoder for molecule generation. In *Proceedings of the International Conference on Learning Representations*.

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

Dinh, L., Krueger, D., and Bengio, Y. (2015). NICE: Non-linear Independent Components Estimation. *International Conference on Learning Representations Workshop*.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density Estimation Using Real NVP. *International Conference on Learning Representations*.

Ertl, P. and Schuffenhauer, A. (2009). Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *Journal of cheminformatics*, 1(1):8.

Ghosh, P., Sajjadi, M. S. M., Vergari, A., Black, M., and Scholkopf, B. (2020). From variational to deterministic autoencoders. In *International Conference on Learning Representations*.

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. (2018). Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. pages 2672–2680.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6.

Hsu, W.-N., Zhang, Y., Weiss, R. J., Zen, H., Wu, Y., Wang, Y., Cao, Y., Jia, Y., Chen, Z., Shen, J., et al. (2019). Hierarchical generative modeling for controllable speech synthesis. *International Conference on Learning Representations*.

Jin, W., Barzilay, R., and Jaakkola, T. (2018). Junction tree variational autoencoder for molecular graph generation. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2323–2332, Stockholmsmässan, Stockholm Sweden. PMLR.

Kadurin, A., Aliper, A., Kazennov, A., Mamoshina, P., Vanhaelen, Q., Khrabrov, K., and Zhavoronkov, A. (2017). The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget*, 8(7):10883.

Kingma, D. P. and Welling, M. (2013). Auto-Encoding Variational Bayes. *International Conference on Learning Representations*.

Kusner, M. J., Paige, B., and Hernández-Lobato, J. M. (2017). Grammar variational autoencoder. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1945–1954. JMLR. org.

Kuznetsov, M., Polykovskiy, D., Vetrov, D. P., and Zhebrak, A. (2019). A prior of a googol gaussians: a tensor ring induced prior for generative models. In *Advances in Neural Information Processing Systems*, pages 4104–4114.

Landrum, G. (2006). Rdkit: Open-source cheminformatics. *Online). http://www. rdkit. org. Accessed*, 3(04):2012.

LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database.

Makhzani, A., Shlens, J., Jaitly, N., and Goodfellow, I. (2016). Adversarial autoencoders.

Oord, A. V., Kalchbrenner, N., and Kavukcuoglu, K. (2016). Pixel recurrent neural networks. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1747–1756, New York, New York, USA. PMLR.

Polykovskiy, D., Zhebrak, A., Sanchez-Lengeling, B., Golovanov, S., Tatanov, O., Belyaev, S., Kurbanov, R., Artamonov, A., Aladinskiy, V., Veselov, M., Kadurin, A., Nikolenko, S., Aspuru-Guzik, A., and Zhavoronkov, A. (2018a). Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *arXiv preprint arXiv:1811.12823*.

Polykovskiy, D., Zhebrak, A., Vetrov, D., Ivanenkov, Y., Aladinskiy, V., Bozdaganyan, M., Mamoshina, P., Aliper, A., Zhavoronkov, A., and Kadurin, A. (2018b). Entangled conditional adversarial autoencoder for de-novo drug discovery. *Molecular Pharmaceutics*.

Preuer, K., Renz, P., Unterthiner, T., Hochreiter, S., and Klambauer, G. (2018). Fréchet ChemNet distance: A metric for generative models for molecules in drug discovery. *J. Chem. Inf. Model.*, 58(9):1736–1741.

Razavi, A., Oord, A. v. d., and Vinyals, O. (2019). Generating diverse high-fidelity images with vq-vae-2. *Advances In Neural Information Processing Systems*.

Segler, M. H. S., Kogej, T., Tyrchan, C., and Waller, M. P. (2018). Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent Sci*, 4(1):120–131.

Semeniuta, S., Severyn, A., and Barth, E. (2017). A hybrid convolutional variational autoencoder for text generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 627–637, Copenhagen, Denmark. Association for Computational Linguistics.

Snelson, E. and Ghahramani, Z. (2006). Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. (2016). Wasserstein auto-encoders.

Tomczak, J. and Welling, M. (2018). Vae with a vampprior. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1214–1223, Playa Blanca, Lanzarote, Canary Islands. PMLR.

van den Oord, A., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., van den Driessche, G., Lockhart, E., Cobo, L., Stimberg, F., Casagrande, N., Grewe, D., Noury, S., Dieleman, S., Elsen, E., Kalchbrenner, N., Zen, H., Graves, A., King, H., Walters, T., Belov, D., and Hassabis, D. (2018). Parallel WaveNet: Fast high-fidelity speech synthesis. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3918–3926, Stockholmsmässan, Stockholm Sweden. PMLR.

Weininger, D. (1970). Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. 17:1–14.

Weininger, D., Weininger, A., and Weininger, J. L. (1989). Smiles. 2. algorithm for generation of unique smiles notation. *Journal of chemical information and computer sciences*, 29(2):97–101.

You, J., Ying, R., Ren, X., Hamilton, W., and Leskovec, J. (2018). GraphRNN: Generating realistic graphs with deep auto-regressive models. In *International Conference on Machine Learning*, pages 5694–5703.

Yu, L., Zhang, W., Wang, J., and Yu, Y. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Zhao, S., Song, J., and Ermon, S. (2019). Infovae: Balancing learning and inference in variational autoencoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5885–5892.

Zhavoronkov, A., Ivanenkov, Y., Aliper, A., Veselov, M., Aladinskiy, V., Aladinskaya, A., Terentiev, V., Polykovskiy, D., Kuznetsov, M., Asadulaev, A., Volkov, Y., Zholus, A., Shayakhmetov, R., Zhebrak, A., Minaeva, L., Zagribelnyy, B., Lee, L., Soll, R., Madge, D., Xing, L., Guo, T., and Aspuru-Guzik, A. (2019). Deep learning enables rapid identification of potent ddr1 kinase inhibitors. *Nature biotechnology*, pages 1–4.

A Proof of Theorem 1

We prove the theorem using five lemmas.

Lemma 1. $\mathcal{L}_\tau$ *convergences to* $\mathcal{L}_*$ *pointwise when* τ *converges to 0 from the right:*

$$\forall(\theta, \phi) \quad \lim_{\tau \to 0+} \mathcal{L}_\tau(\theta, \phi) = \mathcal{L}_*(\theta, \phi) \qquad (27)$$

Proof. To prove Eq. 27, we first show that our approximation in Eq.10 from the main paper converges pointwise to $\mathbb{I}[x > 0]$. $\forall x \in \mathbb{R}$:

$$\lim_{\tau \to 0+} \sigma_\tau(x) = \lim_{\tau \to 0+} \frac{1}{1 + e^{-x/\tau}\left[\frac{1}{\tau} - 1\right]} = \mathbb{I}[x > 0] \qquad (28)$$

If x is negative, both $e^{-x/\tau}$ and $1/\tau$ converge to $+\infty$, hence $\sigma_\tau(x)$ converges to zero. If x is zero, then $\sigma_\tau(x) = \tau$ which also converges to zero. Finally, for positive x we apply L'Hôpital's rule to compute the limit:

$$\lim_{\tau \to 0+} \frac{e^{-x/\tau}}{\tau} = \lim_{\tau \to 0+} \frac{(1/\tau)'}{(e^{x/\tau})'} = \lim_{\tau \to 0+} \frac{e^{-x/\tau}}{x} = 1 \quad (29)$$

To prove the theorem, we consider two cases. First, if $(\theta, \phi) \notin \Omega$, then for some x, i, and $x \neq s$,

$$\mathbb{E}_{z \sim q_\phi(z|x)} \mathbb{I}\left[\widetilde{\pi}^\theta_{x,i,x_i}(z) \leq \widetilde{\pi}^\theta_{x,i,s}(z)\right] > 0. \qquad (30)$$

From the equation above follows that for given parameters the model violates indicators with positive probability. For those z, a smoothed indicator function takes values less than τ, so the expectation of its logarithm tends to $-\infty$ when $\tau \to 0+$.

The second case is $(\theta, \phi) \in \Omega$. Since $\mathcal{L}_*(\theta, \phi) > -\infty$, indicators are violated only with probability zero, which will not contribute to the loss neither in $\mathcal{L}_*$, nor in $\mathcal{L}_\tau$. For all x, i and s, consider a distribution of a random variable $\delta = \widetilde{\pi}^\theta_{x,i,x_i}(z) - \widetilde{\pi}^\theta_{x,i,s}(z)$ obtained from a distribution $q_\phi(z \mid x)$. Let $\delta_{\max} \leq 1$ be the maximal value of δ. We now need to prove that

$$\lim_{\tau \to 0+} \mathbb{E}_{\delta \sim p(\delta)} \log \sigma_\tau(\delta) = 0 \qquad (31)$$

For any $\epsilon > 0$, we select $\delta_0 > 0$ such that $p(\delta < \delta_0) < \epsilon$. For the next step we will use the fact that $\sigma_\tau(\delta_{1/2}) = 0.5$, where $\delta_{1/2} = \tau \log\left(\frac{1}{\tau} - 1\right)$. By selecting τ small enough such that $\delta_{1/2} < \delta_0$, we split the integration limit for δ in expectation into three segments: $(0, \delta_{1/2}]$, $(\delta_{1/2}, \delta_0]$, $(\delta_0, \delta_{\max})$. A lower bound on $\log \sigma_\tau(\delta)$ in each segment is given by its value in the left end: $\log \tau$, $\log 1/2$, $\log \sigma_\tau(\delta_0)$. Also, since $p(\delta \leq 0) = 0$ and δ is continuous on compact support of $q_\phi(z \mid x)$, density $p(\delta)$ is bounded by some constant M. Such estimation gives us the final lower bound using pointwise convergence

of $\sigma_\tau(\delta)$:

$$0 \geq \mathbb{E}_{\delta \sim p(\delta)} \log \sigma_\tau(\delta) \geq$$
$$M \cdot \underbrace{\log \tau \cdot \delta_{1/2}}_{\lim_{\tau \to 0+} \cdots = 0} + \epsilon \cdot \log 1/2$$
$$+ M \cdot \underbrace{\log \sigma_\tau(\delta_0)}_{\lim_{\tau \to 0+} \cdots = 0} \cdot (\delta_{\max} - \delta) \to_{\tau \to 0+} \epsilon \cdot \log 1/2. \qquad (32)$$

We used $\lim_{\tau \to 0+} \log \tau \cdot \delta_{1/2} = 0$ which can be proved by applying the L'Hôpital's rule twice. $\qquad \square$

Proposition 1. *For our model,* $\mathcal{L}_*$ *is finite if and only if a sequence-wise reconstruction error rate is zero:*

$$(\theta, \phi) \in \Omega \Leftrightarrow \Delta(\widetilde{x}_\theta, \phi) = 0 \qquad (33)$$

Lemma 2. *Sequence-wise reconstruction error rate* $\Delta(\phi)$ *is continuous.*

Proof. Following equicontinuity in total variation of $q_\phi(z \mid x)$ at ϕ for any x and finiteness of χ, for any $\epsilon > 0$ there exists $\delta > 0$ such that for any $x \in \chi$ and any ϕ' such that $\|\phi - \phi'\| < \delta$

$$\int |q_\phi(z|x) - q_{\phi'}(z|x)| \, dz < \epsilon. \qquad (34)$$

For parameters ϕ and ϕ', we estimate the difference in Δ function values

$$\Delta(\phi) - \Delta(\phi')$$
$$= \underbrace{\Delta(\widetilde{x}^*_\phi, \phi) - \Delta(\widetilde{x}^*_{\phi'}, \phi)}_{\leq 0} + \Delta(\widetilde{x}^*_{\phi'}, \phi) - \Delta(\widetilde{x}^*_{\phi'}, \phi')$$
$$\leq \mathbb{E}_{x \sim p(x)} \underbrace{\int (q_\phi(z|x) - q_{\phi'}(z|x)) \, \mathbb{I}\left[\widetilde{x}^*_{\phi'}(z) \neq x\right] dz}_{< \epsilon}$$
$$\leq \epsilon \qquad (35)$$

Symmetrically, $\Delta(\phi') - \Delta(\phi) \leq \epsilon$, resulting in $\Delta(\phi)$ being continuous. $\qquad \square$

Lemma 3. *Sequence-wise reconstruction error rate* $\Delta(\phi_n)$ *converges to zero:*

$$\lim_{n \to +\infty} \Delta(\phi_n) = \Delta(\widetilde{\phi}) = 0. \qquad (36)$$

The convergence rate is $\mathcal{O}(\frac{1}{\log(1/\tau_n)})$.

Proof. Since Ω is not empty, there exists $(\widehat{\theta}, \widehat{\phi}) \in \Omega$. From pointwise convergence of $\mathcal{L}_\tau$ to $\mathcal{L}_*$ at point $(\widehat{\theta}, \widehat{\phi})$, for any $\epsilon > 0$ exists N such that for any $n > N$:

$$\underbrace{\mathcal{L}_{\tau_n}(\theta_n, \phi_n) \geq \mathcal{L}_{\tau_n}(\widehat{\theta}, \widehat{\phi})}_{\text{from the definition of } (\theta_n, \phi_n)} \geq \mathcal{L}_*(\widehat{\theta}, \widehat{\phi}) - \epsilon. \qquad (37)$$

Next, we derive an upper bound on $\mathcal{L}_{\tau_n}(\theta_n, \phi_n)$ using the fact that $\log \sigma_\tau(x) < 0$ if $x > 0$, and $\log \sigma_\tau(x) \leq \log \tau_n$ if $x \leq 0$:

$$\mathcal{L}_{\tau_n}(\theta_n, \phi_n) \leq \mathbb{E}_{x \sim p(x)} \Bigg[\mathbb{E}_{z \sim q_\phi(z|x)} \sum_{i=1}^{|x|} \sum_{s \neq x_i} \log \tau_n \cdot$$

$$\mathbb{I}\left[\pi_{x,i,x_i}(z) \leq \pi_{x,i,s}(z) \right] \underbrace{-\mathcal{KL}q_\phi(z \mid x)p(z)}_{\leq 0} \Bigg]$$

$$\leq |V| L \cdot \log \tau_n \cdot \Delta(\widetilde{x}_{\theta_n}, \phi_n). \tag{38}$$

Combining Eq. 37 and Eq. 38 together we get

$$|V| L \cdot \underbrace{\log \tau_n}_{<0} \cdot \Delta(\widetilde{x}_{\theta_n}, \phi_n) \geq \mathcal{L}_*(\theta^*, \phi^*) - \epsilon \tag{39}$$

Adding the defintion of $\Delta(\phi)$, we obtain

$$0 \leq \Delta(\phi_n) \leq \Delta(\widetilde{x}_{\theta_n}, \phi_n) \leq \frac{\epsilon - \mathcal{L}_*(\theta^*, \phi^*)}{|V| L \cdot \log(1/\tau_n)} \tag{40}$$

The right hand side goes to zero when n goes to infinity and hence $\lim_{n \to +\infty} \Delta(\widetilde{x}_{\theta_n}, \phi_n) = 0$ and $\lim_{n \to +\infty} \Delta(\phi_n) = 0$ with the convergence rate $\mathcal{O}(\frac{1}{\log(1/\tau_n)})$. Since $\Delta(\phi_n)$ is continuous, $\Delta(\widetilde{\phi}) = 0$. $\square$

Lemma 4. $\mathcal{L}_*(\theta, \phi)$ *attains its supremum:*

$$\exists \theta^* \in \Theta, \phi^* \in \Phi : \mathcal{L}_*(\theta^*, \phi^*) = \sup_{\theta \in \Theta, \phi \in \Phi} \mathcal{L}_*(\theta, \phi). \tag{41}$$

Proof. From Lemma 3, $\Delta(\widetilde{\phi}) = 0$. Hence, for a choice of $\widetilde{\theta}$ from the theorem statement, $\Delta(\widetilde{\theta}, \widetilde{\phi}) = 0$. Equivalently, $(\widetilde{\theta}, \widetilde{\phi}) \in \Omega$.

Note that since $\Delta(\phi) \geq 0$ is continuous on a compact set, $\Phi_0 = \{\phi \mid \Delta(\phi) = 0\}$ is a compact set. Also, $\mathcal{L}_*(\theta, \phi)$ is constant with respect to θ on Ω. From the theorem statement, for any ϕ such that $\Delta(\phi) = 0$, there exists $\theta(\phi)$ such that $(\theta(\phi), \phi) \in \Omega$. Combining all statements together,

$$\sup_{\phi \in \Phi_0} \mathcal{L}_*(\theta(\phi), \phi) = \sup_{\theta \in \Theta, \phi \in \Phi} \mathcal{L}_*(\theta, \phi) \tag{42}$$

In Ω, $\mathcal{L}_*$ is a continuous function: $\forall(\theta, \phi) \in \Omega$,

$$\mathcal{L}_*(\theta, \phi) = -\mathcal{KL}(\phi) = -\mathbb{E}_{x \sim p(x)} \mathcal{KL}\left(q_\phi(z|x) \,\|\, p(z) \right) \tag{43}$$

Hence, continuous function $\mathcal{L}_*(\theta(\phi), \phi)$ attains its supremum on a compact set Φ at some point (θ^*, ϕ^*), where $\theta^* = \theta(\phi^*)$. $\square$

Lemma 5. *Parameters $(\widetilde{\theta}, \widetilde{\phi})$ from theorem statement are optimal:*

$$\mathcal{L}_*(\widetilde{\theta}, \widetilde{\phi}) = \sup_{\theta \in \Theta, \phi \in \Phi} \mathcal{L}_*(\theta, \phi). \tag{44}$$

Proof. Assume that $\mathcal{L}_*(\widetilde{\theta}, \widetilde{\phi}) < \mathcal{L}_*(\theta^*, \phi^*)$. Since $(\widetilde{\theta}, \widetilde{\phi}) \in \Omega$ and $(\theta^*, \phi^*) \in \Omega$, $\mathcal{L}_*(\widetilde{\theta}, \widetilde{\phi}) = -\mathcal{KL}(\widetilde{\phi})$ and $\mathcal{L}_*(\theta^*, \phi^*) = -\mathcal{KL}(\phi^*)$. As a result, from our assumption, $\mathcal{KL}(\phi^*) < \mathcal{KL}(\widetilde{\phi})$.

From continuity of $\mathcal{KL}(\phi)$ divergence, for any $\epsilon > 0$, exists $\delta > 0$ such that if $\|\widetilde{\phi} - \phi\| < \delta$,

$$\mathcal{KL}(\phi) > \mathcal{KL}(\widetilde{\phi}) - \epsilon = \mathcal{L}_*(\widetilde{\theta}, \widetilde{\phi}) - \epsilon \tag{45}$$

From the convergence of ϕ_n to $\widetilde{\phi}$ and convergence of τ_n to zero, there exists N_1 such that for any $n > N_1$, $\|\widetilde{\phi} - \phi_n\| < \delta$.

From pointwise convergence of $\mathcal{L}_{\tau_n}$ at point (θ^*, ϕ^*) to $\mathcal{L}_*(\theta^*, \phi^*)$, for any $\epsilon > 0$, exists N_2 such that for all $n > N_2$, $\mathcal{L}_{\tau_n}(\theta^*, \phi^*) > \mathcal{L}_*(\theta^*, \phi^*) - \epsilon$. Also, $\mathcal{L}_{\tau_n}(\theta_n, \phi_n) \leq -\mathcal{KL}(\phi_n)$ from the definition of $\mathcal{L}_{\tau_n}$ as a negative $\mathcal{KL}$ divergence plus some non-positive penalty for reconstruction error.

Taking $n > \max(N_1, N_2)$, we get the final chain of inequalities:

$$\mathcal{L}_{\tau_n}(\theta_n, \phi_n) \leq -\mathcal{KL}(\phi_n) < -\mathcal{KL}(\widetilde{\phi}) + \epsilon$$
$$= \mathcal{L}_*(\widetilde{\theta}, \widetilde{\phi}) + \epsilon < \mathcal{L}_{\tau_n}(\theta^*, \phi^*) - \epsilon + \epsilon$$
$$= \mathcal{L}_{\tau_n}(\theta^*, \phi^*) \tag{46}$$

Hence, $\mathcal{L}_{\tau_n}(\theta_n, \phi_n) < \mathcal{L}_{\tau_n}(\theta^*, \phi^*)$, which contradicts $(\theta_n, \phi_n) \in \text{Arg max of } \mathcal{L}_{\tau_n}$. As a result, $\mathcal{L}_*(\widetilde{\theta}, \widetilde{\phi}) = \mathcal{L}_*(\theta^*, \phi^*)$. $\square$

B Implementation details

For all experiments, we provide configuration files in a human-readable format in the supplementary code. Here we provide the same information for convenience.

B.1 Synthetic data

Encoder and decoder were GRUs with 2 layers of 128 neurons. The latent size was 2; embedding dimension was 8. We trained the model for 100 epochs with Adam optimizer with an initial learning rate $5 \cdot 10^{-3}$, which halved every 20 epochs. The batch size was 512. We fine-tuned the model for 10 epochs after training by fixing the encoder and learning only the decoder. For a proposed model with a uniform prior and a uniform proposal, we increased $\mathcal{KL}$ weight β linearly from 0 to 0.1 during 100 epochs. For the Gaussian and tricube proposals, we increased $\mathcal{KL}$ weight β linearly from 0 to 1 during 100 epochs. For all three experiments, we pretrained the autoencoder for the first two epochs with $\beta = 0$. We annealed the temperature from 10^{-1} to 10^{-3} during 100 epochs of training in a log-linear scale. For a tricube proposal, we annealed the temperature to 10^{-2}.
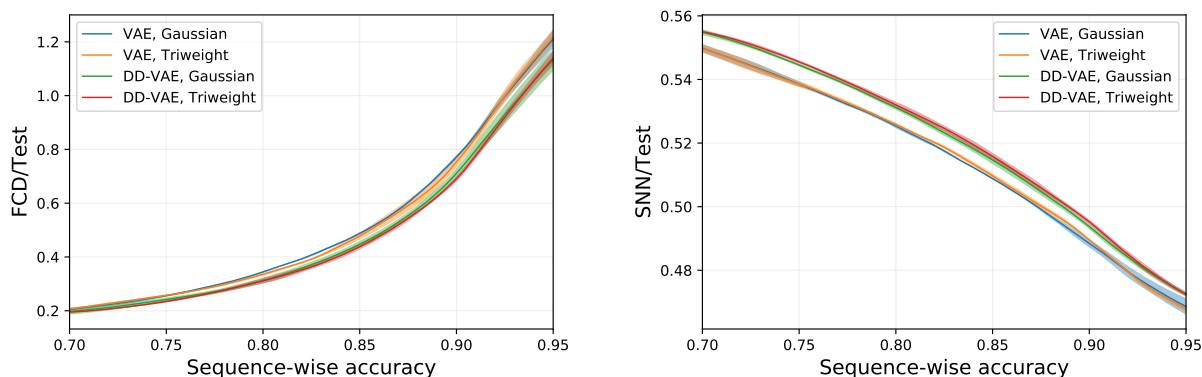
Figure 7: Distribution learning with deterministic decoding on MOSES dataset: FCD/Test (lower is better) and SNN/Test (higher is better). Solid line: mean, shades: std over multiple runs.

B.2 Binary MNIST

We binarized the dataset by thresholding original MNIST pixels with a value of 0.3. We used a fully connected neural network with layer sizes $784 \rightarrow 256 \rightarrow 128 \rightarrow 32 \rightarrow 2$ with LeakyReLU activation functions. We trained the model for 150 epochs with a starting learning rate $5 \cdot 10^{-3}$ that halved every 20 epochs. We used a batch size 512 and clipped the gradient with value 10. We increased β from 10^{-5} to 0.005 for VAE and 0.05 for DD-VAE. We decreased the temperature in a log scale from 0.01 to 0.0001.

B.3 MOSES

We used a 2-layer GRU network with a hidden size of 512. Embedding size was 64, the latent space was 64-dimensional. We used a tricube proposal and a Gaussian prior. We pretrained a model with a fixed β for 20 epochs and then linearly increased β for 180 epochs. We halved the learning rate after pretraining. For DD-VAE models, we decreased the temperature in a log scale from 0.2 to 0.1. We linearly increased β divergence from 0.0005 to 0.01 for VAE models and from 0.0015 to 0.02.

B.4 ZINC

We used a 1-layer GRU network with a hidden size of 1024. Embedding size was 64, the latent space was 64-dimensional. We used a tricube proposal and a Gaussian prior. We trained a model for 200 epochs with a starting learning rate $5 \cdot 10^{-4}$ that halved every 50 epochs. We increased divergence weight β from 10^{-3} to 0.02 linearly during the first 50 epochs for DD-VAE models, from 10^{-4} to $5 \cdot 10^{-4}$ for VAE model, and from 10^{-4} to $8 \cdot 10^{-4}$ for VAE model with a tricube proposal. We decreased the temperature log-linearly from 10^{-3}

to 10^{-4} during the first 100 epochs for DD-VAE models. With such parameters we achieved a comparable train sequence-wise reconstruction accuracy of 95%.

C MOSES distribution learning

In Figure 7, we report detailed results for the experiment from Section 4.3.

D Best molecules found for ZINC

In Figure 8, Figure 9, Figure 10, and Figure 11 we show the best molecules found with Bayesian optimization during 10-fold cross validation.

Figure 8: DD-VAE with Tricube proposal. The best molecules found with Bayesian optimization during 10-fold cross validation and their scores.
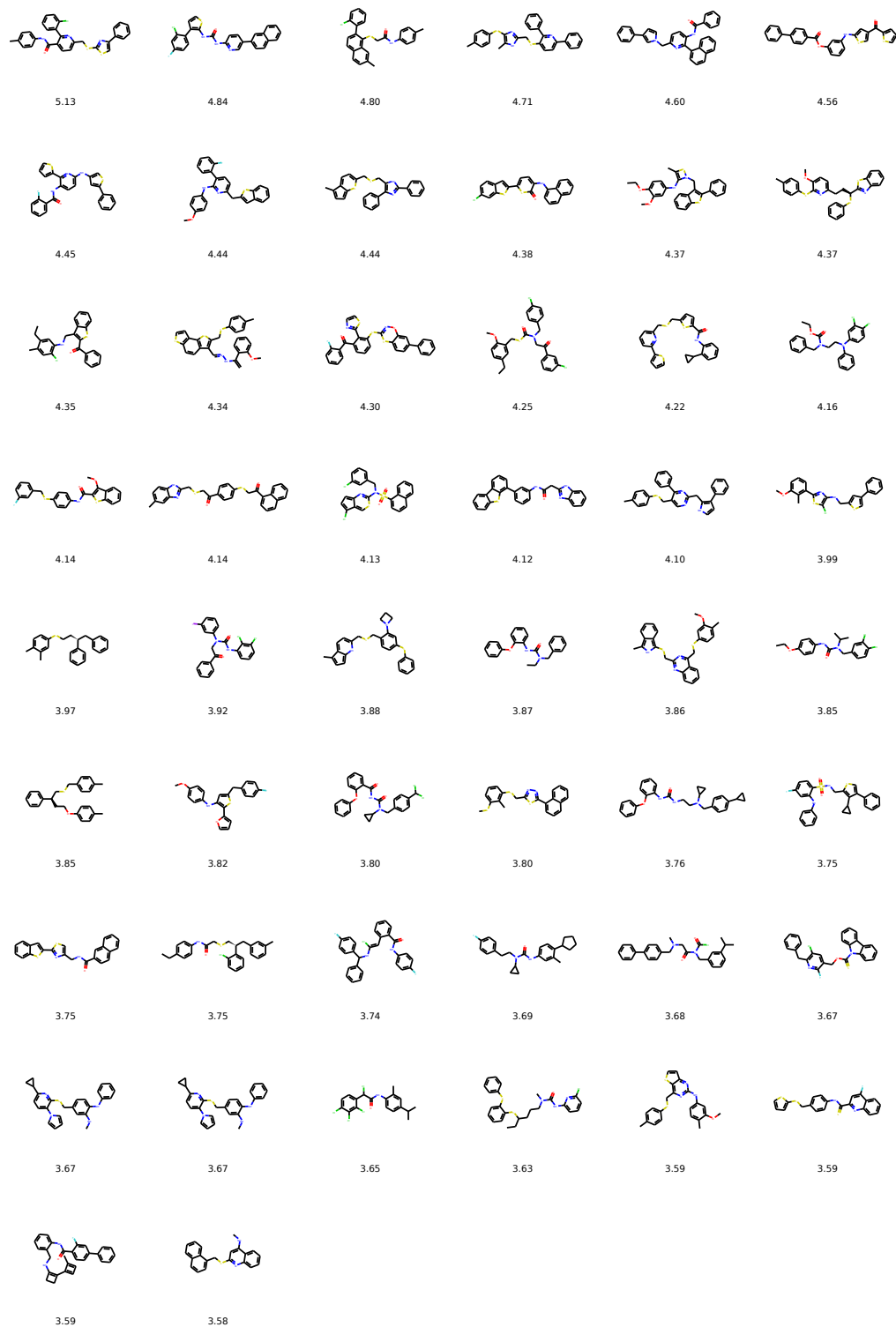
Figure 9: DD-VAE with Gaussian proposal. The best molecules found with Bayesian optimization during 10-fold cross validation and their scores.

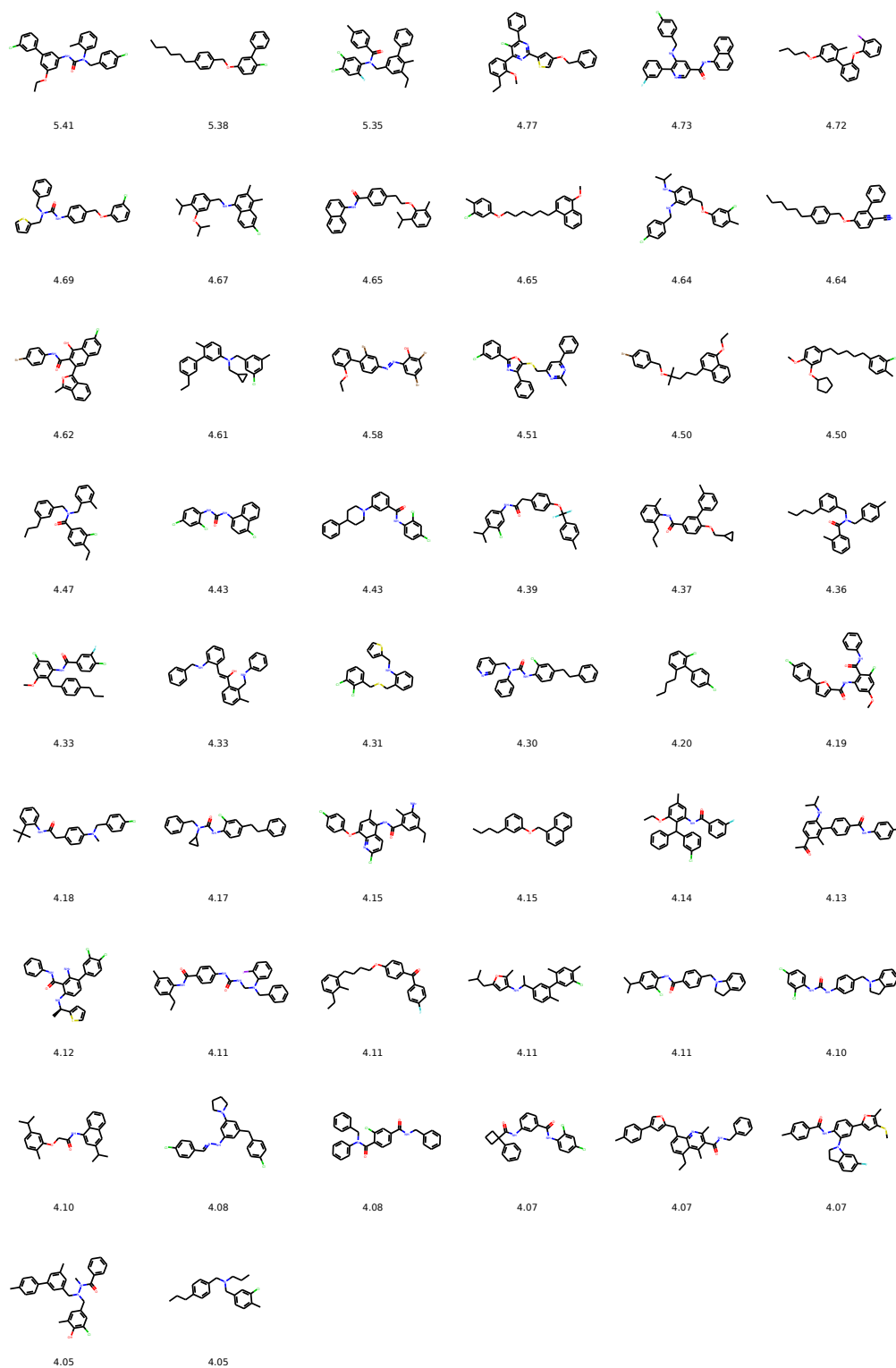Figure 10: VAE with Tricube proposal. The best molecules found with Bayesian optimization during 10-fold cross validation and their scores.
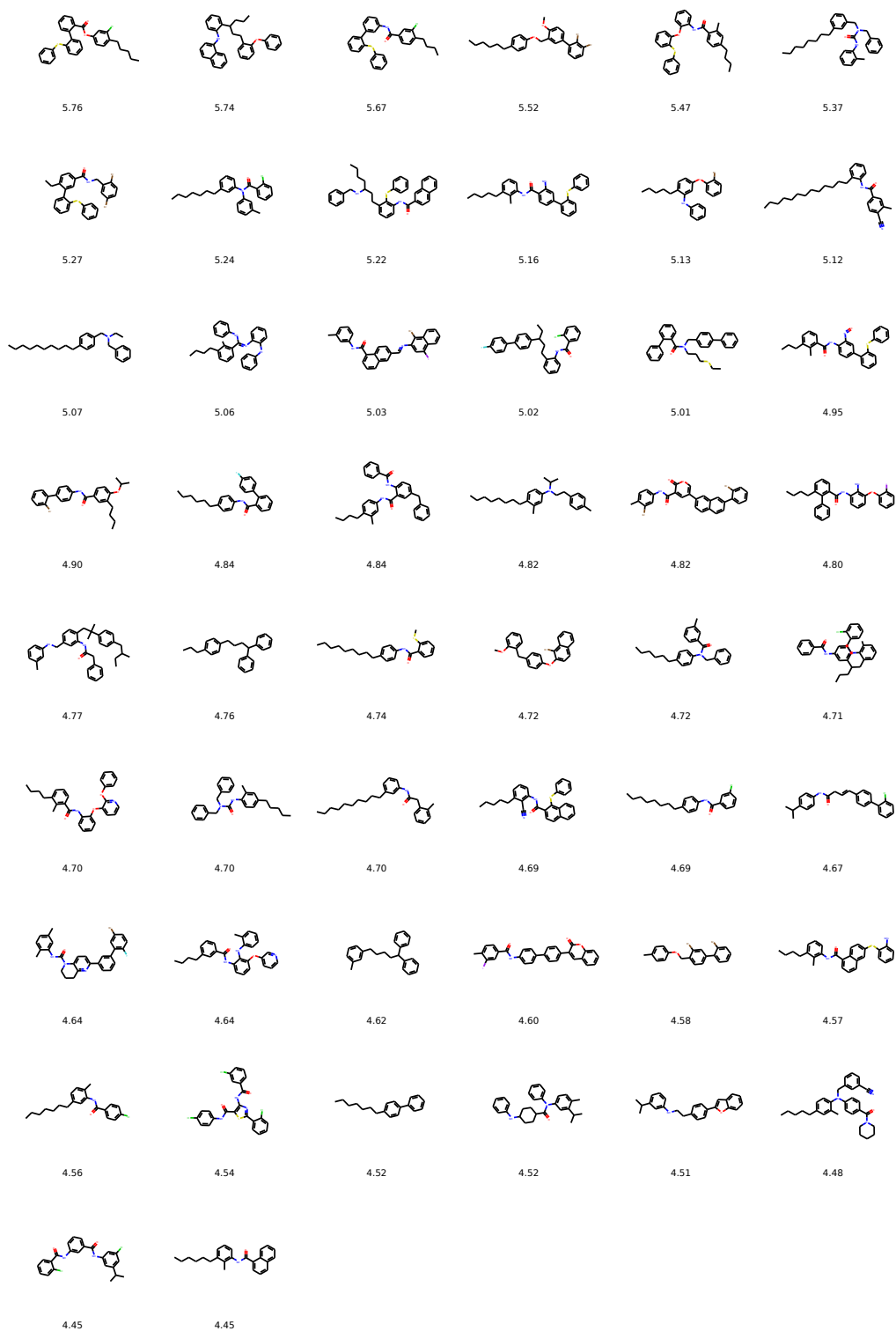
Figure 11: VAE with Gaussian proposal. The best molecules found with Bayesian optimization during 10-fold cross validation and their scores.