Re-balancing Variational Autoencoder Loss for Molecule Sequence Generation

Chaochao Yan  
The University of Texas at Arlington  
chaochao.yan@mavs.uta.edu

Sheng Wang  
The University of Texas at Arlington  
sheng.wang@mavs.uta.edu

Jinyu Yang  
The University of Texas at Arlington  
jinyu.yang@mavs.uta.edu

Tingyang Xu  
The University of Texas at Arlington  
tingyangxu@tencent.com

Junzhou Huang  
The University of Texas at Arlington  
jzhuang@uta.edu

ABSTRACT

Molecule generation is to design new molecules with specific chemical properties and further to optimize the desired chemical properties. Following previous work, we encode molecules into continuous vectors in the latent space and then decode the embedding vectors into molecules under the variational autoencoder (VAE) framework. We investigate the posterior collapse problem of the current widely-used RNN-based VAEs for the molecule sequence generation. For the first time, we point out that the underestimated reconstruction loss of VAEs leads to the posterior collapse, and we also provide both analytical and experimental evidences to support our findings. To fix the problem and avoid the posterior collapse, we propose an effective and efficient solution in this work. Without bells and whistles, our method achieves the state-of-the-art reconstruction accuracy and competitive validity score on the ZINC 250K dataset. When generating 10,000 unique valid molecule sequences from the random prior sampling, it costs the JT-VAE 1450 seconds while our method only needs 9 seconds on a regular desktop machine.

KEYWORDS

molecule generation, variational autoencoder, recurrent neural networks, posterior collapse

1 INTRODUCTION

Discovering new molecules that have the desired target properties is the key challenge of drug and material design. This can be considered as an optimization problem, and the goal is to search for molecules with the best desired property score [7]. However, exhaustive exploration in the molecule space is infeasible, as the total number of estimated drug-like molecules is in the order of $10^{60}$ [22]. Additionally, molecule synthesis [30] and validation are also time-consuming and expensive in practice, which makes the brute-forces impractical.

As deep learning models are making more achievements in various fields [20] [31] [19], they have also been applied for molecule generation. The majority of current molecule generation methods heavily rely on the variational autoencoder (VAE) which is a combination of a deep latent variable model and an accompanying variational learning technique [4] [23]. As shown in Figure 1, drug molecules can be first embedded by the encoder into the continuous latent space which can be further utilized for property prediction and optimization. After that, the decoder absorbs a continuous latent vector to reconstruct the input molecule. Thanks to the clustering ability of the VAE, the latent representations of semantically similar molecules (with similar chemical structures and properties) tend to be grouped together in the latent space. In consequence, it allows semantically meaningful sampling and smooth interpolation in the latent space. Therefore, new molecules can be generated by randomly sampling from the prior and can be further optimized by exploring the latent space. The key idea behind the optimization above is to search for molecules that maximize a property score objective, given latent representations of molecules as the input[7].

However, existing VAE models mainly suffer from the posterior collapse issue, where the decoder tends to ignore the latent vectors [1] [7]. This problem is more frequently observed in those models with RNN-based backbone [8]. As a result, the generated molecules tend to be in low diversity and are weakly relevant to the latent vectors [7] [15]. This phenomenon has also been observed in Natural Language Processing (NLP) tasks, such as the text generation [1]. To alleviate this problem, the major focus of the previous studies is to adopt various training strategies, such as KL cost annealing [1] or aggressively optimizing the decoder before each encoder update [8]. However, such methods can not be simply extended to molecule generation, mainly stemming from the fact that the molecule sequences are strictly structured and any mutations can result in invalid sequences. Motivated by the success of parse trees in the
NLP field and attribute grammars in the compiler design, recent work [15] [3] incorporate grammar or syntax rules to guarantee that syntactically valid SMILES sequences can be generated. As an alternative, a molecule can also be represented by a graph in order to mitigate the posterior collapse [17] [12].

Inspired by the essential pitfalls of the contemporary RNN-based VAE models in the molecule generation, here, we propose a new method to alleviate the posterior collapse issue. To achieve this goal, we first analyze the posterior collapse of vanilla VAE model for SMILES sequence generation. For the first time, we find that the posterior collapse is largely triggered by the underestimated reconstruction loss. We propose a novel loss function to leverage the trade-off between the reconstruction loss and the KL loss in VAE training. Without making any changes on the VAE network structures or introducing additional computational complexity, our method is extremely simple yet effective in preventing posterior collapse in molecule generation. We also provide a detailed analysis of our method [1], and empirically demonstrate its state-of-the-art (SOTA) reconstruction accuracy and competitive validity score on the ZINC 250K dataset. Our primary contributions can be summarized as:

- We diagnose the main reason causing the posterior collapse within the RNN-based VAE model for molecule sequence generation with detailed analysis.
- We propose an effective and efficient method to eliminate the posterior collapse by leveraging the associations between the reconstruction loss and the KL loss.
- Extensive empirical studies demonstrate our method’s superiority over SOTA molecule generation approaches on the ZINC 250K dataset.

2 BACKGROUND INFORMATION

2.1 The Variational Autoencoder

The VAE [4] [23] is a specially regularized version of the standard autoencoder (AE). It is appealing because it can learn complex distribution in an unsupervised manner and later act as a generative model defined by a prior \( p(z) \) and a conditional distribution \( p_{\theta}(x|z) \). Since the true data likelihood is usually intractable, so the VAE instead optimizes an evidence lower bound (ELBO) which is a valid lower bound of the true data log likelihood:

\[
\mathcal{L}(x; \theta, \phi) = \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)] - \text{KL}(q_{\phi}(z|x) \parallel p(z)) \leq \log p(x).
\]

where the encoder \( q_{\phi}(z|x) \) is parameterized with \( \phi \) and learns to map the input \( x \) to a variational distribution, and the decoder \( p_{\theta}(x|z) \) parameterized with \( \theta \) tries to reconstruct the input \( x \) given the latent vector \( z \) from the learned distribution. Usually, \( q_{\phi}(z|x) \) is modeled as a Gaussian distribution and optimized to approximate the true posterior \( p_{\theta}(z|x) \).

The VAE is optimized to maximize ELBO (1), where (i) negative reconstruction loss \( \mathbb{E}_{q_{\phi}(z|x)}[\log p_{\theta}(x|z)] \) enforces the encoder to generate meaningful latent vector \( z \), so that the decoder can reconstruct the input \( x \) from the \( z \) and (ii) the KL regularization loss \( \text{KL}(q_{\phi}(z|x) \parallel p(z)) \) minimizes the KL divergence between the approximate posterior \( q_{\phi}(z|x) \) and the prior \( p(z) \sim \mathcal{N}(0, I) \).

3 RELATED WORK

3.1 Text Generation with VAE

Motivated by the ubiquitous posterior collapse problems observed in VAE-based models for text generation, various methods have been proposed and investigated recently [1] [32] [9] [13]. Bowman et al. [1] propose to anneal the KL loss weight to enable the model to learn meaningful encoding before applying KL loss to cluster the encoding. They also weaken the decoder with word dropout and historyless decoding to force the decoder rely on the latent vectors. [8] concludes that the posterior collapse mainly attributes to the lagging encoder network’s inability in approximating the true posterior. To overcome this limitation, they propose a novel training strategy which aggressively optimize the encoder network. Inspired by [1], [6] proposes a cyclical annealing strategy which repetitively starts training from a pretrained model resulted in the previous cycle. They claim this procedure can make the model learn more meaningful latent representations progressively.

3.2 Molecule Generation

Thanks to the development of NLP text generation, the VAE model is applied for molecule generation for the first time in CVAE [7]. They build a VAE encoder and decoder with GRU layers, representing molecules in the SMILES sequences. However, their model suffers from generating invalid SMILES sequences which makes their model impracticable. To improve the prior validity, context-free grammars for SMILES are introduced in GVAE [15] to represent a molecule in the sparse tree. However, the validity score is still unsatisfactory. Inspired by this method, Syntax-directed VAE (SD-VAE) [3] incorporates extra semantic rules to ensure generated SMILES valid, and it achieves the best performance among all SMILES-based methods. However, these models did not solve model posterior collapse problem and there is a large space to improve.

Except for SMILES representations, molecules can also be represented in graphs. [17] employs graph-structured representations for molecules and models the probabilistic dependencies among a graph’s nodes and edge with graph neural networks. Molecules are generated node by node by their model. Chemical sub-graphs instead of atoms are used as the basic building blocks in JT-VAE.
their methods can incrementally generate molecules to ensure chemical validity at each step. JT-VAE is the SOTA model for molecule generation.

4 PROBLEM AND OUR SOLUTION

4.1 Posterior Collapse Problem

Prior work [1] [32] [9] [13] on NLP text generation has observed the posterior collapse phenomenon, in which the decoder tends to ignore $z$ when training the VAE model. When posterior collapse happens, the model training falls into the the ill-posed region of the ELBO objective (1), in which the variational posterior $q_{θ}(z|x)$ naively mimics the model prior $p(z)$. Note that the KL loss in the ELBO objective can be further decomposed [11] as:

$$
\mathbb{E}_{p_{θ}(x)}[D_{KL}(q_{θ}(z|x)||p(z))] = I_q + D_{KL}(q_{θ}(z)||p(z)),
$$

where $I_q$ is the mutual information between $x$ and $z$ given $q_{θ}(z|x)$, and $p_{θ}(x)$ is the empirical data distribution. When posterior collapse occurs, the KL loss decreases nearly to zero so that $I_q$ is also close to zero (both items on the right-hand side in (2) are non-negative) during the VAE model training process. It is especially evident when modelling discrete data with a strong auto-regressive network such as LSTM [10] and GRU [2], which is exactly our case. This is undesirable since the VAE model fails to learn meaningful latent representations for input sequences.

For NLP text generation task, the posterior collapse problem has been mainly attributed to the low quality of latent representations $z$ at the early stage of model training [1] [8] [6]. To be more specific, the decoder $p_{θ}(x|z)$ falls behind the encoder $q_{θ}(z|x)$ at the initial training procedure, and $q_{θ}(z|x)$ generates low-quality latent representations so that it is very hard for $p_{θ}(x|z)$ to recover the input sequences. In consequence, the model is forced to ignore $z$. Many solutions have been proposed to solve the problem and they have demonstrated satisfactory improvement on various NLP datasets.

However, molecule generation is a quite different scenario though it appears to be same as the NLP text generation. First of all, its token size is far less than the NLP text generation. The token size of NLP text data is usually tens of thousands or even more, while it is less than 100 for chemical molecule data. The smaller token size makes the molecule reconstruction task easier. Second, the molecule sequence is composed strictly following the SMILES grammar rules, and the reconstructed sequence must be exactly the same as the input to be matched successfully. Any token mutations can result in a completely different sequence. However, there are no such rigid grammar rules applied to the NLP text and the exact match is not required.

We experimentally confirm the existing solutions [8] [6] for NLP text generation does not work well for the chemical molecule generation. This motivates us to propose such a solution for molecule sequence generation.

4.2 The Trick in Previous Solutions

To avoid the posterior collapse, which will disable the reconstruction ability of VAE models, previous SMILES-based methods CVAE [7], GVAE [15], and SD-VAE [3] reduce the standard deviation $\sigma$ of prior Gaussian distribution to a small value 0.01 (can be found in their public implementations CVAE$^2$, GVAE$^3$, SD-VAE$^4$), which makes their models more like AEs instead of VAEs. This is why CVAE and GVAE have a decent reconstruction accuracy but extremely low validity scores as shown in Table 1. If we set $\sigma=1$, all these models will suffer from the posterior collapse and can not reconstruct inputs faithfully (similar to the vanilla VAE in Figure 2(e)). In the following our analysis and experiments, we will strictly keep $\sigma=1$.

4.3 Underestimated Reconstruction Loss

To investigate the cause of posterior collapse within the VAE for molecule generation, we conduct convincing analysis and investigation into the posterior collapse. We hypothesize it is the underestimated reconstruction loss that causes posterior collapse of a VAE during training process. Both analytical analysis and experimental support are provided to verify our hypothesis.

The reconstruction loss term $\mathbb{E}_{q_{θ}(z|x)}[\log p_{θ}(x|z)]$ in (1) measures the reconstruction ability of the decoder given the latent vector $z$. The decoder should only receive information from $z$ and tries to reconstruct the full sequence accurately from the given $z$. However, the RNN models are usually optimized with the teacher forcing [28] in practice, in which the RNN input at each step is the ground truth instead of the prediction from a prior time step. We can rewrite the reconstruction loss term in (1) as:

$$
\mathbb{E}_{q_{θ}(z|x)} \sum_{t=1}^{T} \log p_{θ}(x_t|z, \tilde{x}_{<t}),
$$

where the $T$ is the maximum time step, $\tilde{x}_{<t}$ is the prediction prefix before time $t$ and the current input is the previous time step output $\tilde{x}_{t-1}$, and $\tilde{x}_0$ is the predefined start symbol.

With teacher forcing, the actual reconstruction loss now is:

$$
\mathbb{E}_{q_{θ}(z|x)} \sum_{t=1}^{T} \log p_{θ}(x_t|z, x_{<t}),
$$

where $x_{<t}$ is the ground-truth prefix before time $t$ and the ground-truth token of previous time step $x_{t-1}$ is the RNN input at each time step, and $x_0$ is also the predefined start symbol.

Since the ground-truth information is incorporated additionally in (4) when training the VAE, which can make the decoder’s prediction task easier, we can expect that the reconstruction ability of the decoder is largely overestimated compared with (3). Therefore, we can assume the reconstruction loss term is underestimated, which will potentially breaks the balance between reconstruction loss and KL loss in (1). We will verify the assumption and also demonstrate quantitatively how much the reconstruction loss is underestimated in the experiment section. Let us agree on the claim for now.

4.4 Our Solution: Re-balancing VAE Loss

Since the reconstruction loss is underestimated during training, and it breaks the balance with KL loss, which leads to the posterior collapse finally. We can recover the balance by applying a

\[3\] https://github.com/mkusner/grammarVAE
\[4\] https://github.com/Hanjun-Dai/sdvae
reconstruction loss weight $\alpha$ to the ELBO (1):

$$L(x; \theta, \phi) = \alpha \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x) || p(z)), \alpha > 1, \quad (5)$$

where $\alpha$ can be estimated using Monte Carlo sampling in every training iteration. Specifically, we can sample a batch of data as input and run a VAE with/without teacher forcing, respectively. Since the reconstruction loss without teacher forcing can be regarded as the “true” reconstruction loss (the reconstruction loss it should be), we approximate $\alpha$ as the ratio of reconstruction loss without teacher forcing to that with teacher forcing. However, estimating $\alpha$ in every training iteration is too expensive. We can set $\alpha$ as a hyper-parameter for simplicity and efficiency.

Inspired by the $\beta$-VAE [9] formulation, we can instead reduce KL loss weight $\beta$, which is equivalent to increasing reconstruction loss weight $\alpha$. It is more natural and convenient to search for the optimal value of hyper-parameter $\beta$ since increasing $\beta$ from 0 is a gradual transition from the autoencoder (AE) to VAE. So we can have a similarly modified VAE loss formulation:

$$L(x; \theta, \phi) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \beta D_{KL}(q_\phi(z|x) || p(z)), 0 \leq \beta < 1. \quad (6)$$

Note that in our case $\beta < 1$, while the $\beta$-VAE requires the KL weight $\beta > 1$. The $\beta$-VAE is proposed in [9] to learn disentangled representation of generative factors by enforcing a larger penalty on KL loss, since they postulate that $\beta > 1$ could place a stronger constraint on the latent representation to drive the VAE to learn more efficient latent representation of input $x$. While we have a completely different motivation and goal of fixing the imbalanced VAE loss by reducing KL weight since we find reconstruction loss is underestimated in ELBO (1).

Except for the above analysis, our method can also be explained from an intuitive perspective. In previous methods CVAE, GVAE, and SD-VAE, when sampling latent vectors $z$ they have to reduce the standard deviation $\sigma$ to a small value 0.01 otherwise the model will collapse and lose the reconstruct ability. Instead of reducing sampling $\sigma$, we can anneal the KL loss weight $\beta$ to make the model transform from AE to VAE gradually [1] since the AE has a strong reconstruction ability. Different from [1], we restrict $\beta$ to be smaller than 1. By searching for the optimal $\beta$, we can arrive a trade-off between the reconstruction accuracy and validity score.

We acknowledge that previous methods have empirically tried to reduce the KL loss weight to avoid the posterior collapse [3] [8] [6]. The $\beta$-VAE ($\beta = 0.4$) alleviates the problem and achieves competitive performance on density estimation for NLP text datasets [8], which proves that reducing $\beta$ is viable for NLP text task. It is also indicated setting $\beta = 1/\text{LatentDimension}$ could lead to better results [15] [3]. But none of these methods provided any analysis or explanation.

We are the first to recognize underestimated reconstruction loss leads to posterior collapse in VAE molecule generation, and further we officially propose to reduce KL weight to overcome the posterior collapse with detailed analysis and solid experimental support.

5 EXPERIMENTS

Our proposed solution to the VAE model posterior collapse is simple but extremely effective and efficient. We do not need to modify the
network architecture and only adjust the training loss slightly, without introducing much extra computation. In this section, we will first train a vanilla VAE model and track the occur of model collapse, as well as experimentally verify that the reconstruction loss is underestimated. Then we will conduct extensive experiments to demonstrate the effectiveness of our proposed method.

5.1 Experimental Setting Up
We build our VAE model based on GRU. The VAE encoder is composed of two layers of bi-directional GRU which is good at capturing sequence representation [24], and hidden size of each layer is 512. The decoder is made up of four layers of uni-directional GRU with the same hidden size 512. Following previous work [7] [12], we use unit Gaussian prior and set the latent vector dimension to be 56. The ELBO objective is optimized with Adam [14] and learning rate is 0.0001. The model is trained with teacher forcing and KL loss annealing. We train the model for 150 epochs and report the performance of the final model. Experiments are conducted on a machine with an Intel Core i7-5930K@3.50GHz CPU and a GTX 1080 Ti GPU.

We conduct all our experiments on ZINC 250K dataset [15] which is a subset of the ZINC [27]. Molecule sequences are tokenized with the regular expression from [25]. We use the same training and testing split as previous work [15] [12], and have 10K hold-out data out of the training as the validation data. From now on, we will use the same experimental setting in all our experiments unless explicitly stated.

As for the model evaluation metrics, we report the reconstruction accuracy and validity score like previous work. Following [12], we encode each molecule from test dataset 10 times, and then decode 10 times for each latent vector. The reconstruction accuracy is defined to be the ratio of successfully reconstructed molecule sequences to the total tried reconstruction. The reconstructed SMILES must be exactly the same as the input to be counted as successful. To calculate validity, 1000 latent vectors are randomly sampled from the prior distribution, and each is decoded 100 times. The validity is the portion of chemically valid reconstruction SMILES to the total decoded sequences. We use the open source tool RDkit [16] to check if a SMILES is valid.

5.2 VAE Training Dynamics
We track the training process of a vanilla VAE model for SMILES sequences, as well as that of our proposed method. By investigating training dynamic like the KL loss weight, KL loss, reconstruction loss, mutual information, as well as the reconstruction accuracy and validity score, we conclude that the underestimated reconstruction loss causes the posterior collapse of vanilla VAE for molecule sequence during the training process. Mutual information \( I_q \) can be calculated using Monte Carlo sampling as proposed in [11] [5]:

\[
I_q = \mathbb{E}_{p_d(x)} [D_{KL}(q_\phi(z|x)||p(z))] - D_{KL}(q_\phi(z)||p(z)),
\]

which is actually the same as the (2). We approximate the aggregated posterior \( q_\phi(z) = \mathbb{E}_{p_d(x)}[q_\phi(z|x)] \) using Monte Carlo sampling. \( D_{KL}(q_\phi(z)||p(z)) \) can also be estimated by the Monte Carlo, and we can obtain samples from \( q_\phi(z) \) by ancestral sampling. More details about \( I_q \) computation can be found in [11].

As a comparison, we also illustrate the training dynamic when our proposed method is applied. For our method, we set KL weight \( \beta = 0.1 \) which is explained and derived from the next section. We keep all the other experimental settings the same as the vanilla VAE to make a fair comparison.

Results of two models run are plot in the Figure 2. The vanilla VAE model performs well on the validation data at the early stage of the KL weight annealing. As the KL weight increases, KL loss drops quickly as expected since more penalty is added to the KL loss term, while the small reconstruction loss starts to rise at the same time. The mutual information \( I_q \) decreases to 0.65 at the end, which means the decoder does not absorb much information from the latent vectors when generating the output. This evidence indicates the posterior collapse has happened. When looking at the model performance on validation data, we can notice that the reconstruction accuracy is close 0 while the validity score is almost 0. As a comparison, our method maintains the mutual information to be around 4.8, which proves the superiority of our method.

5.3 Proof of Loss Underestimation
We hypothesize that introducing ground-truth information into the decoder will result in underestimated reconstruction loss, and have provided our detailed analysis previously. In this section, we will experimentally verify that the reconstruction loss is indeed...
underestimated during the training. We can estimate how much the reconstruction loss has been underestimated using Monte Carlo Sampling. Specifically, we can sample a batch of data, then run the model with and without the teacher forcing, respectively. The underestimated ratio can be approximated by the ratio of reconstruction loss with teacher forcing to that without teacher forcing.

We track the reconstruction loss on the validation dataset when the teacher forcing is applied and removed, respectively. Results are shown in the Figure 3(a). When teacher forcing is applied, the reconstruction loss drops close to 1 quickly, while the loss is much larger (at least 7.5) without teacher forcing. This is expected since, without teacher forcing, any wrong prediction token as input may result in following prediction totally different from ground-truth sequences.

To figure out how much the reconstruction loss has been underestimated, we can compute the ratio as reconstruction loss w/ teacher forcing to that w/o teacher forcing at each time step. Results are shown in Figure 3(b). It confirms our claim that the reconstruction loss is underestimated. To recover a balanced VAE loss, we can set KL loss weight exactly as the underestimated ratio in each epoch. To be simplified, we set $\beta = 0.1$ and we find it works very well in practice.

### 5.4 Reconstruction Accuracy and Validity

We summarize the molecule reconstruction accuracy and validity on test dataset in the Table 1. Our method outperforms all previous models in reconstruction accuracy by a large margin (16% larger than the second-best model). In the meanwhile, our method achieves 90.7% validity, which is much better than previous SMILES-based methods.

Compared with other SMILES-based methods, our model is much more superior in both the reconstruction accuracy and prior validity, even if complex grammar or syntax rules are incorporated [15] [3]. Note that the JT-VAE model assembles molecules by adding sub-graphs step-by-step to make sure the generated molecule graphs are always valid. However, these sub-graphs are extracted from the training dataset, which limits the JT-VAE from generating molecules with unseen sub-graphs. While our method achieves competitive validity performance without any constraints, and is able to generate novel molecules that are not from the same distribution as the training data. That is one important reason why our method achieves the SOTA reconstruction accuracy, while JT-VAE suffers from reconstructing testing molecules [21]. Besides, our method is much more efficient than JT-VAE. When generating 10,000 unique valid SMILES from prior random sampling, JT-VAE\(^5\) (faster version) takes about 1450s while our method only needs 9s.

### 5.5 Error analysis and Visualization

Our model achieves 92.7% reconstruction accuracy and all reconstructed SMILES are valid. We investigate the reconstruction results further and find that our model can predict 97.3% of all tokens correctly, which is measured on the level of the token instead of the sequence. Besides, most of the unmatched sequences (62%) are valid, and it confirms the reconstruction ability of our model.

As for the validity, we also investigate the model outputs. We find our model can generate complicated and diverse molecules with multiple rings. As for the invalid sequences, from both the reconstruction and prior sampling, there are several typical errors: (1) unkekulized atoms, (2) valence error, (3) unclosed ring, and (4) parentheses error. We believe the grammar-based methods [15] [3] are complimentary to our method, and can be combined to reduce these errors.

### 5.5 Error analysis and Visualization

Our model achieves 92.7% reconstruction accuracy and all reconstructed SMILES are valid. We investigate the reconstruction results further and find that our model can predict 97.3% of all tokens correctly, which is measured on the level of the token instead of the sequence. Besides, most of the unmatched sequences (62%) are valid, and it confirms the reconstruction ability of our model.

As for the validity, we also investigate the model outputs. We find our model can generate complicated and diverse molecules with multiple rings. As for the invalid sequences, from both the reconstruction and prior sampling, there are several typical errors: (1) unkekulized atoms, (2) valence error, (3) unclosed ring, and (4) parentheses error. We believe the grammar-based methods [15] [3] are complimentary to our method, and can be combined to reduce these errors.

### Table 1: Reconstruction accuracy and validity results. Baseline results are reported in [15] [3] [26] [12].

| Model          | Reconstruction | Validity |
|----------------|----------------|----------|
| SMILES-based   |                |          |
| CVAE           | 44.6%          | 0.7%     |
| GVAE           | 53.7%          | 7.2%     |
| SD-VAE         | 76.2%          | 43.5%    |
| Our Method     | 92.7%          | 90.7%    |
| Graph-based    |                |          |
| GraphVAE       | -              | 13.5%    |
| JT-VAE         | 76.7%          | 100.0%   |

---

**Figure 4:** Top–3 molecules and associated scores found by our model with Bayesian optimization.

5https://github.com/wengong-jin/icml18-jtnn
6.5 Bayesian Optimization

One of the important tasks in the drug molecule generation is to make molecules with desired chemical properties. We follow [15] for all the experimental setting, and the optimization target score is:

\[ y(m) = \log p(m) - SA(m) - cycle(m), \]

where \( \log p(m) \) is the octanol-water partition coefficients of molecule \( m \), \( SA(m) \) is synthetic accessibility score, and \( cycle(m) \) is number of large rings with more than six atoms.

We first associate each molecule with a latent vector which is the mean of the learned variational encoding distribution. The latent vector for each molecule will be treated as its feature and we train a Sparse Gaussian Process (SGP) to predict target score \( y(m) \) given its latent vector. After training SGP, five iterations of batched Bayesian optimization (BO) are performed with expected improvement heuristics.

We report SGP prediction performance when trained on latent representations learned by different models. We train the SGP with 10-fold cross validation and report the top-3 molecules found by the BO. As shown in Table 2, molecules found by our model are much better than that found by previous SMILES-based methods, and our method is even superior to the graph-based method JT-VAE. Figure 4 shows top-3 molecules found by our model.

6 DISCUSSION

Our method is very efficient and it works extremely well in the molecule generation, in which SMILES sequences are highly structured and grammatically organized. Our experimental results indicate that grammar and syntax rules are necessary to generate more valid SMILES sequences, and they are complementary to our method. Besides, SMILES-based methods and graph-based methods may also be combined together to boost the model performance further.

Though our primary focus is the VAE for molecule generation, our method can also help the NLP task as we mentioned at the end of section 4.4. Reducing KL loss weight can help the VAE model for NLP task avoid the posterior collapse as shown in [8] [6].

The latent representation learnt by our model can be applied to various downstream tasks, such as molecule property prediction [29] [18]. In the future we may explore more about this application.

7 ACKNOWLEDGMENT

This work was partially supported by US National Science Foundation IIS-1718853, the CAREER grant IIS-1553687 and Cancer Prevention and Research Institute of Texas (CPRIT) award (RP190107).

REFERENCES

[1] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating Sentences from a Continuous Space. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning. 10–21.

[2] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In NIPS 2014 Workshop on Deep Learning, December 2014.

[3] Hanjun Dai, Yingtao Tian, Bo Dai, Steven Skiena, and Le Song. 2018. Syntax-directed variational autoencoder for structured data. arXiv preprint arXiv:1802.08786 (2018).

[4] P Kingma Diederik, Max Welling, et al. 2014. Auto-encoding variational bayses. In Proceedings of the ICLR.

[5] Adji B Dieng, Yoon Kim, Alexander M Rush, and David M Blei. 2019. Avoiding Latent Variable Collapse with Generative Skip Models. In The 22nd International Conference on Artificial Intelligence and Statistics. 2397–2405.

[6] Hodon Fu, Chunyan Li, Xiaodong Liu, Jianfeng Gao, Asi Celikyilmaz, and Lawrence Carin. 2019. Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing. In NAACL.

[7] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, and et al. 2018. Automatic chemical design using a data-driven continuous representation of molecules. ACS central science 4, 2 (2018), 268–276.

[8] Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Lending Inference Networks and Posterior Collapse in Variational Autoencoders. In Proceedings of the ICLR.

[9] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, and et al. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In Proceedings of the ICLR.

[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural computation 9, 8 (1997), 1735–1780.

[11] Matthew D Hoffman and Matthew J Johnson. 2016. Elbo surgery: yet another way to curve up the variational evidence lower bound. In NIPS Workshop on Advances in Approximate Bayesian Inference.

[12] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. 2018. Junction Tree Variational Autoencoder for Molecular Graph Generation. In ICML. 2328–2337.

[13] Yoon Kim, Sam Wiseman, Andrew Miller, David Sontag, and Alexander Rush. 2018. Semi-Amortized Variational Autoencoders. In ICML. 2683–2692.

[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).

[15] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. 2017. Grammar variational autoencoder. In Proceedings of the 34th ICML-Volume 70. JMLR. org, 1945–1954.

[16] Greg Landrum et al. 2006. RDKit: Open-source cheminformatics.

[17] Yuzhe Li, Oriol Vinyals, Chris Dyer, Razvan Pascanu, and Peter Battaglia. 2018. Learning deep generative models of graphs. arXiv preprint arXiv:1803.03324 (2018).

[18] Heliuan Ma, Yu Rong, Wenbing Huang, Tingyang Xu, Weiyang Xie, Geyan Ye, and Junzhou Huang. 2020. Multi-View Graph Neural Networks for Molecular Property Prediction. arXiv preprint arXiv:2005.13607 (2020).

[19] Xin Miao, Xin Yuan, Yunchen Pu, and Vassilis Athitsos. [n.d.] Jambal-NET: Reconstruct Hyperspectral Images From a Snapshot Measurement. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV). IEEE, 4058–4068.

[20] Xin Miao, Xiantong Zhen, Xianglong Liu, Cheng Deng, Vassilis Athitsos, and Heng Huang. 2018. Direct shape regression networks for end-to-end face alignment. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 5040–5049.

[21] S Mohammadi, B O'Dowd, C Paulitz-Erdmann, and L Goerlitz. 2019. Penalized Variational Autoencoder for Molecular Design. ChemRxiv (2019).

[22] Pavel G Poližuch, Timur I Mashhadov, and Alexandre Varnek. 2013. Estimation of the size of drug-like chemical space based on GDB-17 data. Journal of computer-aided molecular design 27, 8 (2013), 675–679.

[23] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In ICML. 1278–1286.

[24] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. Neural computation 9, 11 (2015), 2324–2337.

[25] Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. Neural computation 1, 2 (1989), 270–280.

[26] Zheng Xu, Sheng Wang, Feiyuan Zhu, and Junzhou Huang. 2017. Seq2seq fingerprint: An unsupervised deep molecular embedding for drug discovery. In Proceedings of the 8th ACM international conference on bioinformatics, computational biology, and health informatics. 285–294.

[27] Zichao Yang, Jun Qianggang Dong, Peilin Zhao, Shuangjia Zheng, Jinyu Yang, Yang Yu, and Junzhou Huang. 2020. Xpert: Decompose Retrosynthesis Prediction like A Chemist. ChemRev (2020).

[28] Zichao Yang, Zhihui An, Sheng Wang, Xinliang Zhu, Chaoxiao Yan, and Junzhou Huang. 2020. Label-Driven Reconstruction for Domain Adaptation in Semantic Segmentation. arXiv preprint arXiv:2001.04614 (2020).

[29] Chaoxiao Yan, Zichao Yang, Huolan Salakhdinov, and Taylor Berg-Kirkpatrick. 2017. Improved variational autoencoders for text modeling using dilated convolutions. In Proceedings of the 34th ICML-Volume 70. JMLR. org, 3881–3890.