LATENT OPTIMIZATION VARIATIONAL AUTOENCODER FOR CONDITIONAL MOLECULAR GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models such as variational autoencoder (VAE) and generative adversarial networks are widely used for image, audio and text generation tasks. However, in general, the training of VAE is still at risk of posterior collapsing especially for long sequential data. To alleviate this, modified evidence lower bounds (ELBOs) were proposed. However, these approaches heuristically control training loss using a hyper-parameter, and it is not way to solve the fundamental problem of vanilla VAE. In this paper, we propose a latent optimization VAE (LOVAE) to insert an optimization step of the latent variable and alternately update the encoder and decoder of conditional VAE for maximizing ELBOs. In experiments, we applied LOVAE on ZINC database, which contains 3D structural information of compound quality and molecular physical properties, for the molecular generation. We showed that LOVAE outperforms vanilla VAE in terms of ELBOs and the molecular generation. In addition, LOVAE showed better performance in target property satisfaction and target property maximization tasks compared to existing works.

1 INTRODUCTION

Deep neural networks (DNNs) have demonstrated a dramatic performance improvement in various applications. Text extraction from image recognition, language translation, speech recognition, natural language recognition, and personal identification by fingerprint and iris have already achieved high accuracy (Wu et al., 2016; Devlin et al., 2018; Awad, 2012; Nguyen et al., 2017). Recently, these applications became successful commercialized products.

For the purpose of generation of image, variational autoencoder (VAE) (Kingma & Welling, 2014), generative adversarial network (GAN) (Goodfellow et al., 2014), and reversible generative models (Kingma & Dhariwal, 2018) were proposed and showed much progress (Bleicher et al., 2003; Phatak et al., 2009; Grathwhohl et al., 2018). These generative models were initially studied in image data and showed better performance than previous models. Since then, it has been extended in research areas to generate new sentences (Iqbal & Qureshi, 2020) and to discover new drugs (Chen et al., 2018) and materials (Kim et al., 2018a).

Traditional materials research consists of four steps: molecule design, physical or chemical property prediction, molecular synthesis, and experimental evaluation. These steps are repeated until desired molecular properties of a molecular structure are satisfied. Until now, trial-and-error techniques based on human knowledge have been widely used. However, they are time consuming and very expensive. In order to improve the traditional method, research for a high-throughput computational screening (HTCS) (Bleicher et al., 2003) was conducted. However, this also had limitations such as high computational cost, predefined molecular structures by human knowledge, and low accuracy of simulation.

Unlike the traditional approach, inverse molecular design is an attempt to find novel molecules that satisfy desired properties from exploring a large chemical space (Sanchez-Lengeling & Aspuru-Guzik, 2018). It extracts knowledge of potential molecular structures and properties from accumulated molecular structure databases (PubChem, ZINC, etc.) and proposes new molecular structures that do not exist in their database (Bolton et al., 2008; Irwin et al., 2012). With the inverse molecular design, it is possible to save cost by conducting molecular synthesis and experimental evaluation only for molecular structures having desired properties instead of searching almost infinite chemical space.

With the development of machine learning techniques, the generative models such as GAN and VAE have been applied to inverse molecular design tasks in recent years (Sanchez-Lengeling & Aspuru-Guzik, 2018; Shi et al., 2020; Yang et al., 2020; Jin et al., 2018; Wang et al., 2020). In GAN, the discriminator tries to distinguish the molecular structure from the generator, and the generator tries to generate similar molecular structure from the database, simultaneously. In VAE, the encoder outputs latent variables using the molecular structure as an input, and the decoder generates the original molecular structure using the latent variables from the encoder.

However, there are difficulties in training generative models, such as poor convergence. In GAN, it is difficult to learn due to the alternative update method from the two player game situation (Goodfellow et al., 2014). Problems such as the two models oscillation without adversarial, or the fact that the parameters are no longer learned in a specific situation due to the mode collapsing phenomenon, are still problems to be solved. In VAE, evidence lower bound (ELBO) for training VAE models requires training both the reconstruction and the KL-divergence loss of the latent variables, but it can cause a phenomenon called posterior collapsing. In order to prevent the posterior collapse, β-VAE (Higgins et al., 2017), Re-balancing-VAE (Yan et al., 2019), KL-annealing (Bowman et al., 2016), and controlVAE (Shao et al., 2020) have been proposed.

In this paper, we propose a latent optimization VAE (LOVAE) that provides stable learning method by inserting the latent variable optimization technique in conditional VAE (cVAE) (Sohn et al., 2015). We apply two stages of the latent optimization to vanilla VAE. By training encoder first, the latent variable from the encoder has been optimized in the direction of reducing the training loss in the same data. After random sampling of the latent variables from the updated encoder, an additional latent optimization based gradient descent was applied. Our proposed method, LOVAE, was compared and verified in the inverse molecular design task, and a drug-like molecular structure (ZINC database (Sterling & Irwin, 2015)) was used as database. We show the proposed method outperforms vanilla VAE in terms of reconstruction loss and ELBO, which are training indicators. In addition, it showed a more improved appearance in molecular generation phase. Consequently, generating molecules of LOVAE showed higher uniqueness ratio, novelty ratio, and target property satisfaction than some of previous approaches. And, also, LOVAE generated molecules that showed higher value of penalized LogP property than the existing methods.

2 Related Work

Inverse molecular design The inverse molecular design based on human-knowledge is very time consuming and relies on the intuition of the researcher, so many researchers recently tried to solve it through simulation method and others. HTCS (Bleicher et al., 2003), One of major simulation method, is an automated computational process that can rapidly identify active compounds, antibodies or genes, and the results provide starting points for many chemical works such as drug and material design (Chen et al., 2018; Kim et al., 2018a). HTCS uses a kind of brute-force method for searching and analyzing the desired chemical characteristics of molecule using a combination of hundreds or tens of millions of active compounds, but this can be disadvantage because the absolutely large amount of resources used to find the desired goals (Phatak et al., 2009). Recently, to solve the problems, artificial intelligence approaches have been widely applied in the field of the molecular design under the name of the inverse molecular design. The VAE and GAN have been applying to the inverse molecular design field (Sanchez-Lengeling & Aspuru-Guzik, 2018; Jin et al., 2020; Yang et al., 2020; Simm et al., 2020).

For the inverse molecular design, various formats for simple representation of molecular structure information have been defined instead of atom's xyz coordinates. First, MDL format that represents 3D coordinate information and binding information between adjacent atoms together (Dalby et al., 1992). Extended Connectivity Fingerprint (ECFP) (Rogers & Hahn, 2010) and Simplified Molecular Input Line Entry System (SMILES) (Weininger et al., 1989) are another representation for the molecular structure as a sequential character string. And recently, the method of representing the molecular structure as a graph structure is also being researched. Among them, SMILES, string representation of molecules, is relatively easy to handle and has been showing a good performance.

Generative models The generative models such as GAN and VAE are very sensitive to the latent variables. In order words, according to ways of dealing with latent variables, difficult problems have been occurred such as posterior collapsing and mode collapsing in the training. As solutions for

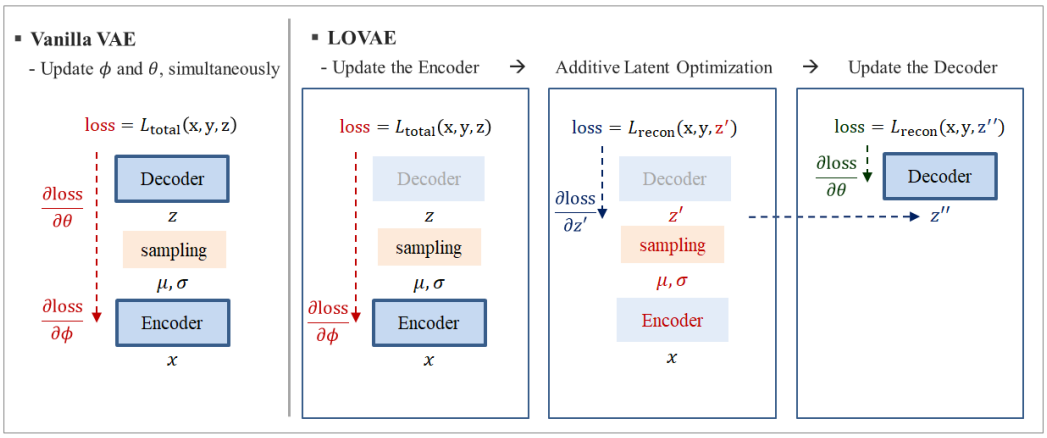

Figure 1: Comparison of the vanilla (conditional) VAE and LOVAE

these phenomenons, methods to adjust a KL loss weight have been proposed in VAE (Higgins et al., 2017; Bowman et al., 2016; Shao et al., 2020). In Kim et al. (2018b), latent variable from encoder is optimized according to maximizing of ELBO, and both of the encoder and decoder parameters are updated as the optimized latent variable. In Zhang et al. (2020), the reconstruction loss on the latent space is additive to the previous loss term. In GAN, there are several works applying the latent variable optimization. The method in Bojanowski et al. (2017) optimizes the latent variable by assuming that the latent variable is learnable noise. In Wu et al. (2019), randomly sampled latent variables are performed a gradient descent in the direction of reducing the loss of GAN. After that, the parameters of the discriminator and generator are updated with the loss from the optimized latent variable z. All of these approaches resulted in a training stability and performance improvement in GAN.

3 Conditional VAE (cVAE)

In the cVAE (Kang & Cho, 2018; Kingma et al., 2014), the input variable $\mathbf{x}$ is assumed to be generated from a generative distribution $p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z})$ conditioned on the output variable $\mathbf{y}$ and latent variable $\mathbf{z}$. The prior distribution of $\mathbf{z}$ is assumed to be $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$. We use variational inference to approximate the posterior distribution of $\mathbf{z}$ given $\mathbf{x}$ and $\mathbf{y}$ by

$$q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}, \mathbf{y}), \text{diag}(\boldsymbol{\sigma}_\phi(\mathbf{x}, \mathbf{y}))). \tag{1}$$

From the perspective of the auto-encoder, $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$ and $p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z})$ are called as an encoder and a decoder, respectively. Feed-forward neural networks are used for $\boldsymbol{\mu}_\phi(\mathbf{x}, \mathbf{y})$ and $\boldsymbol{\sigma}_\phi(\mathbf{x}, \mathbf{y})$.

The objective of the cVAE is to maximize ELBO, which is a lower bound of the marginal log-likelihood:

$$\log p_\theta(\mathbf{x}, \mathbf{y}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})} \log p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z}) - \text{KL}(q_\phi(\cdot|\mathbf{x}, \mathbf{y})||p(\cdot)), \tag{2}$$

up to an additive constant, where KL denotes the Kullback-Leibler divergence. Given a random sample $\mathbf{z}$ generated from the encoder $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$, define the total loss as

$$\mathcal{L}_{\text{total}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = -\log p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z}) + \text{KL}(q_\phi(\cdot|\mathbf{x}, \mathbf{y})||p(\cdot)). \tag{3}$$

Then, $-\mathcal{L}_{\text{total}}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is Monte Carlo approximation of ELBO equation 2. We define $\mathcal{L}_{\text{recon}}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = -\log p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z})$, because it can be regarded as a reconstruction loss. In vanilla cVAE, parameters θ and ϕ are jointly optimized to minimize $\mathcal{L}_{\text{total}}$.

A string representation of a molecule called SMILES is widely used to analyze molecular data (Higgins et al., 2017; Yan et al., 2019; Kang & Cho, 2018). To deal with string data like SMILES in cVAE, recurrent neural networks (Yan et al., 2019; Kang & Cho, 2018) are used for the decoder $p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z})$. Given a target molecular property $\mathbf{y}$, a new molecule $\mathbf{x}$ having this property is generated in the following way:

$$\mathbf{z} \sim p(\mathbf{z}), \ \mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z}). \tag{4}$$

Algorithm 1 An update of the encoder and decoder in LOVAE

Update the Encoder
Generate $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$.
Calculate $\mathcal{L}_{\text{total}}(\mathbf{x}, \mathbf{y}, \mathbf{z})$.
Update $\phi' \leftarrow \phi$ using $\mathcal{L}_{\text{total}}(\mathbf{x}, \mathbf{y}, \mathbf{z})$.

Additive Latent Optimization
Generate $\mathbf{z}' \sim q_{\phi'}(\mathbf{z}'|\mathbf{x}, \mathbf{y})$.
Update $\mathbf{z}'' \leftarrow \mathbf{z}' - \frac{\alpha}{\beta + ||g||^2} g$ where $g = \frac{\partial \mathcal{L}_{\text{recon}}(\mathbf{x}, \mathbf{y}, \mathbf{z}')}{\partial \mathbf{z}'}$.

Update the Decoder
Calculate $\mathcal{L}_{\text{recon}}(\mathbf{x}, \mathbf{y}, \mathbf{z}'')$.
Update $\theta' \leftarrow \theta$ using $\mathcal{L}_{\text{recon}}(\mathbf{x}, \mathbf{y}, \mathbf{z}'')$.

4 Latent Optimization VAE (LOVAE)

Since the latent variable $\mathbf{z}$ generated from the encoder is inputted to the decoder to calculate $\mathcal{L}_{\text{total}}$, the encoder and decoder are closely related. If the encoder generates non-informative or highly volatile $\mathbf{z}$, then the decoder is difficult to reconstruct $\mathbf{x}$ from $\mathbf{z}$. In this case, it may not be optimal to update the encoder and decoder simultaneously. The decoder is very sensitive to the encoder results, $\mathbf{z}$, and the accuracy of the decoder training may vary according to the encoder results. In other words, the result of the encoder can be optimized in advance in direction in which the decoder trains well. That is, better encoder and better latent variable can make the decoder even better.

In this section, we propose LOVAE in which the encoder and decoder are alternatively trained. To be more specific, the encoder is updated as usual with the decoder while the decoder is fixed, and an additive optimization of the latent variable from the encoder follows. Finally, the decoder is updated with the optimized latent variable from the updated encoder. Updating encoder first has an effect similar to latent optimization. By this updating, a more suitable $\mathbf{z}$ can be created and this latent variable not only reduces the loss but also depends on the input data $\mathbf{x}$. A brief comparison of the vanilla VAE and LOVAE is described in Figure 1. First, the encoder parameter ϕ is updated to ϕ' in the direction of reducing the total loss $\mathcal{L}_{\text{total}}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ where $\mathbf{z}$ is generated using the current encoder parameter ϕ. Secondly, z' is generated using the updated encoder parameter ϕ', and then z' is updated to z'' in the direction of reducing the reconstruction loss $\mathcal{L}_{\text{recon}}(\mathbf{x}, \mathbf{y}, \mathbf{z}')$ using a single step of natural gradient descent method (Wu et al., 2019). Lastly, the decoder parameter θ is updated to θ' in the direction of reducing the reconstruction loss $\mathcal{L}_{\text{recon}}(\mathbf{x}, \mathbf{y}, \mathbf{z}'')$ using the optimized z''.

Since the optimized z'' is used in the update of the decoder parameter, LOVAE is expected to achieve bigger ELBO and show stable convergence. This will be verified with numerical results in Section 5. Note that latent optimization is only applied in the training, and the inference of LOVAE remains same with the vanilla one. In summary, the whole process of LOVAE is detailed in Algorithm 1.

5 Experimental Results

5.1 Experiment Setup

ZINC database (Sterling & Irwin, 2015) is a database that organizes information about various compounds drug-like molecules. ZINC contains 3D structural information of compound quality and molecular physical properties such as molecular weight (molwt), partition coefficient (LogP), and quantitative estimation of drug-likeness (QED). In addition, this information is provided in conformity with various molecular expression formats such as SMILES, mol2, and 3D SDF used in the chemical field. SMILES is a string representation to express chemical substances, such as molecules, in the form of ASCII string. It is possible to express complex graph-structured chemicals in the sequence form using simple rules. Figure 2 shows an example molecule of SMILES in ZINC DB, and we used two types of DB, ZINC310K (Yan et al., 2019; Kang & Cho, 2018) and ZINC250K (Kusner et al., 2017). The vocabulary contains 39 different symbols {1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, =, #, (,), [,], H, B, C, N, O, F, Si, P, S, Cl, Br, Sn, I, c, n, o, p, s, \, /, @, @@}. The minimum, median, and maximum

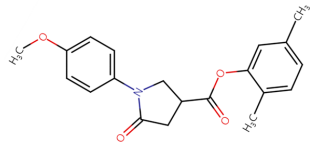

Figure 2: Example of SMILES in ZINC database: COc1ccc(N2CC(C(=O)Oc3cc(C)ccc3C)CC2=O)cc1

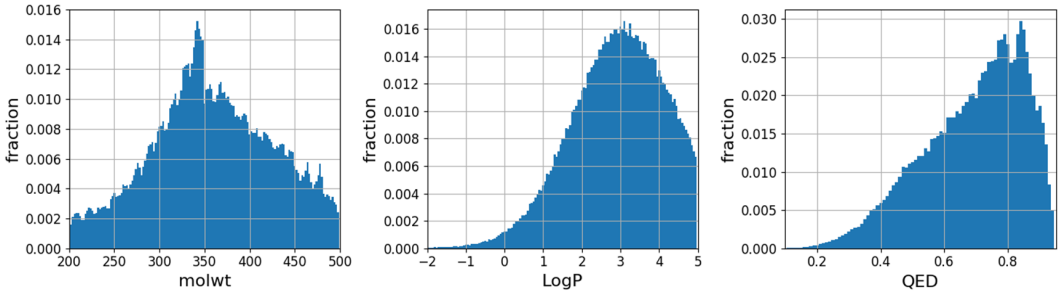

Figure 3: Distribution of target properties

lengths of SMILES strings of ZINC310K are 8, 42, and 86, respectively. (In case of ZINC250K, 9, 44, and 120) The distribution of target properties in our sampled database, ZINC310K, is shown in Figure 3. Median values of molwt, LogP, and QED are 359.02, 2.91, and 0.70, respectively.

Among the existing VAE variants, β-VAE (Higgins et al., 2017), re-balancing VAE (Yan et al., 2019), KL-annealing VAE (Bowman et al., 2016), and controlVAE (Shao et al., 2020) control the weight of $\mathcal{L}_{\mathrm{KL}}$ in equation 3 to achieve their own purpose, such as the disentanglement of latent variables, the avoidance of the posterior collapse, and the stability of the training. The proposed latent optimization technique is also applicable with those methods.

Unlike LOVAE, semi-amortized VAE (SA-VAE) in Kim et al. (2018b) optimizes latent variables without the encoder update phase and applied a momentum based optimization multiple times. After that, decoder and encoder parameters are updated according to the optimized latent variable. According to Kim et al. (2018b), SA-VAE utilized the encoder and latent optimization parts even in test phase. On the other hand, LOVAE does not utilize the additive latent optimization and the encoder in the inference (molecular generation) phase.

The basic model structure in this paper follows a general VAE structure for sequential data. The encoder has a bi-directional RNN structure, and the decoder is a uni-directional RNN structure (Yan et al., 2019; Kang & Cho, 2018). Each RNN structure was composed of three layers of GRU cells, and the dimension of the latent variable was set to 100. The hidden size of each GRU cells is 250, and the dimension of the properties is 3. A 103-dimensional vector in which a 100-dimensional latent variable and a 3-dimensional property are concatenated is inputted to the decoder. Vanilla VAE and LOVAE have the exactly same model structure, but only training strategy differs.

We used the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-6}$, and a polynomial-based learning rate decay was applied. The initial and end learning rate for the learning rate decay were 0.001 and 0.0, respectively. During the training, we normalize each property value to have a mean 0 and standard deviation 1. For the additive latent optimization of LOVAE, $\alpha = 0.001$ and $\beta = 5$ were applied.

5.2 Evaluation: Training Loss

As defined in Section 3, $\mathcal{L}_{\mathrm{total}}$ of VAE is the sum of $\mathcal{L}_{\mathrm{recon}}$ and $\mathcal{L}_{\mathrm{KL}}$. It can be thought that $\mathcal{L}_{\mathrm{total}}$ is primarily important, and the importance of $\mathcal{L}_{\mathrm{recon}}$ and $\mathcal{L}_{\mathrm{KL}}$ can be determined depending on the purpose. For the evaluation, models initialized with 5 different random seeds were trained in each algorithm, vanilla VAE and LOVAE. The loss of LOVAE was measured without the additive latent

Table 1: Training loss of ZINC310K and ZINC250K (Figures in round brackets: when $\alpha = 0.0$)

	vanilla VAE			**LOVAE**		
original ELBO	recon. loss	KL loss	total loss	recon. loss	KL loss	total loss
ZINC310K	20.86	4.38	25.23	17.30 (18.09)	6.51 (5.99)	23.80 (24.08)
ZINC250K	16.69	6.13	22.81	14.76 (15.24)	7.34 (7.00)	22.10 (22.24)
	vanilla VAE			**LOVAE**		
reduced KL term	recon. loss	KL loss	total loss	recon. loss	KL loss	total loss
ZINC310K	14.90	9.31	24.21	7.72	15.35	23.07
ZINC250K	5.93	16.63	22.55	8.99	12.52	21.50

Table 2: Generative efficiency at each property (ZINC310K, 3 target conditions for each property)

	molwt	LogP	QED	average
ssVAE (Kang & Cho, 2018)	0.760	0.866	0.901	0.842
vanilla **VAE**	0.800	0.868	0.880	0.849
LOVAE	0.897	0.924	0.937	0.920

optimization. Table 1 shows the training results for vanilla VAE and LOVAE. We tried each method five times and calculated the mean of the training losses. 'Original ELBO' utilized equation 3 as the total loss, and 'reduced KL term' reduced the weight of $\mathcal{L}_{\mathrm{KL}}$ of equation 3 like Yan et al. (2019). In our experiments, when the weight of KL loss was from 0.7 to 0.8, each method showed a smaller $\mathcal{L}_{\mathrm{total}}$. Figures in round brackets are losses of LOVAE without the additive latent optimization during training ($\alpha = 0.0$) (to verify the updating encoder first only). It can be seen that LOVAE is better than vanilla VAE in both $\mathcal{L}_{\mathrm{recon}}$ and $\mathcal{L}_{\mathrm{total}}$. In the results of 'original ELBO', $\mathcal{L}_{\mathrm{recon}}$ of LOVAE is on average 2.75 less than that of vanilla VAE. In terms of $\mathcal{L}_{\mathrm{KL}}$, LOVAE is 1.67 greater, but $\mathcal{L}_{\mathrm{total}}$ of LOVAE is 1.08 smaller. In case of 'reduced KL term', LOVAE also shows a smaller $\mathcal{L}_{\mathrm{total}}$. The relative improvement of $\mathcal{L}_{\mathrm{total}}$ is about 4.45% in the case of 'original ELBO'. LOVAE with $\alpha = 0.0$ outperformed vanilla VAE about 0.86 in terms of $\mathcal{L}_{\mathrm{total}}$.

5.3 Evaluation: Target Molecular Generation

The generated molecules can be evaluated according to three criteria before considering their property values. The first one is a validity. This means that the generated molecule has a sound structure. This can be determined using the RDKit package (Landrum). The second one is a novelty. The purpose of the inverse molecular design is to find new molecules that have not discovered yet. A molecule is said to be novel if it is not in the train database. The third one is an uniqueness which refers to how molecules are generated without overlapping. Generally, the more uniqueness, the better the generative model. The generated molecules are more meaningful if they satisfy all three criteria, validity, novelty, and uniqueness. In this paper, the ratio that satisfies all three criteria is defined as a generative efficiency. For example, if a generative model attempts to generate 1,000 molecules and has 600 molecules that satisfy validity, novelty, and uniqueness at once, the generative efficiency is 0.6.

For evaluation, three values were determined for each property as a condition for the generative model, and ZINC310K was used. It was determined to be close to the median, lower 10%, and upper 10% value in our train set. In case of molwt, 360.0, 260.0, and 460.0 were used as the condition. For LogP and QED, {3.0, 1.5, 4.5} and {0.7, 0.5, 0.9} were chosen, respectively (Kang & Cho, 2018). Among three target properties, the other two except one of the target property were sampled from a conditional prior distribution given the target value. The molecular generation was attempted 3,500 times in each condition value, and 31,500 molecules were generated in a total of 9 conditions. For the analysis, 'original ELBO' models were used. The results of the generative efficiency are shown in Table 2. It can be seen that LOVAE has good performance of the generative efficiency in all properties.

Table 3: Probability that the property value of the generated molecule falls within 10% and 5% of the error range of the condition value (ZINC310K, 3 target conditions for each property)

	vanilla VAE				**LOVAE**			
	molwt	LogP	QED	average	molwt	LogP	QED	average
within 10%	0.791	0.577	0.616	0.662	0.893	0.703	0.630	0.742
within 5%	0.789	0.377	0.413	0.526	0.879	0.450	0.414	0.581

Table 4: Probability that the property value of the generated molecule falls within target range (ZINC250K)

	LogP		**molwt**	
target range	-2.5 $\sim$ -2.0	5.0 $\sim$ 5.5	150 $\sim$ 200	500 $\sim$550
JT-VAE (Jin et al., 2018)	0.113	0.076	0.007	0.160
ORGAN (Guimaraes et al., 2017)	0.000	0.002	0.151	0.001
GCPN (You et al., 2018a)	0.855	0.547	0.761	0.741
LOVAE	0.316	0.606	0.992	0.976

5.4 Evaluation: Target Property Satisfaction

A target property satisfaction is how many molecules with properties being close to the target condition can be generated. For this, evaluation of property satisfaction was conducted based on two criteria. The first is the percentage that the property value of the generated molecule falls within 10% of the error range of the condition value. For example, if the target value of QED is 0.7, it measures the percentage of the generated molecules whose property value lies between 0.63 and 0.77. The results of the experiment are shown in Table 3. The target values and the number of generation are the same to Sec. 5.3, and each property of generated molecules was computed using RDkit. In all three properties, LOVAE showed higher property satisfaction. At 10% and 5% error property satisfaction, LOVAE showed a relatively 12.1% and 10.5% improvement, respectively.

For the comparison with previous works using ZINC250K, we referred to You et al. (2018b). In You et al. (2018b), a property targeting task was performed, and specific ranges of molwt and LogP were considered. In our case, if the target range of molwt is from 150 to 200, we conditioned LOVAE as 175. There are four target ranges like Table 4. Except for the target range -2.5 $\leqq$ LogP $\leqq$ -2.0, LOVAE showed the best performance. Since percentages of the training data in the range of -2.5 $\leqq$ LogP $\leqq$ -2.0 and 5.0 $\leqq$ LogP $\leqq$ 5.5 are 0.28% and 1.30%, respectively, the first target range, -2.5 $\leqq$ LogP $\leqq$ -2.0, can be a bit more difficult than 5.0 $\leqq$ LogP $\leqq$ 5.5 for cVAE structures. In this respect, the result of LOVAE at the target range -2.5 $\leqq$ LogP $\leqq$ -2.0 seems to make sense.

Table 5: Properties of the top three generated molecules trained on ZINC250K

	penalized LogP			**QED**		
	Top 1	Top 2	Top 3	Top 1	Top 2	Top 3
GrammarVAE (Kusner et al., 2017)	2.94	2.89	2.80	-	-	-
RevalancingVAE (Yan et al., 2019)	5.32	5.28	5.23	-	-	-
JT-VAE (Jin et al., 2018)	5.30	4.93	4.49	0.925	0.911	0.910
GCPN (You et al., 2018a)	7.98	7.85	7.80	0.948	0.947	0.946
molecularRNN (Popova et al., 2019)	10.34	10.19	10.14	0.948	0.948	0.947
graphAF (Shi et al., 2020)	12.23	11.29	11.5	0.948	0.948	0.948
AllSMILES (Nigam et al., 2020)	16.42	16.32	16.21	0.948	0.948	0.948
LOVAE	20.59	18.26	16.39	0.948	0.948	0.948
LOVAE with plogP	15.47	15.46	15.45	-	-	-

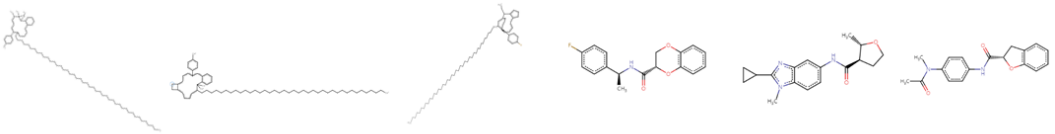

Figure 4: Samples of generated molecules of LOVAE

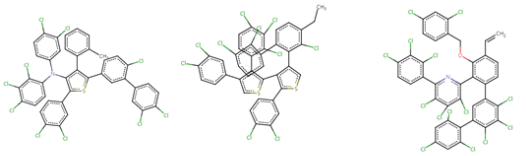

Figure 5: Samples of generated molecules of LOVAE with condition of plogP

5.5 Evaluation: property maximization (penalized LogP and QED)

In the many previous papers, a property maximization task was performed and it was evaluated on penalized LogP (pLogP) and QED (Kusner et al., 2017). plogP has an unbounded range, while QED has a range of [0, 1] by definition. pLogP is a LogP penalized by the synthetic accessibility score (SA) and the number of large rings (cycle), pLogP = logP - SA - cycle, and it can be calculated by RDKit. It can be thought as an extrapolation task because generative models have to create a new molecule that is not in the range of the property values of the training database (The maximum pLogP in ZINC250K = 5.072). In order to find new molecules which show the highest property, some of previous approaches utilized a reward function or a property regressors with Bayesian optimization (Kusner et al., 2017; Shi et al., 2020; Gómez-Bombarelli et al., 2018). LOVAE, which is cVAE structure, is not very suitable for this task. In addition, the maximizing pLogP task is a little less important because it generates 'unusual' molecules with high probability like long aliphatic chains (Winter et al., 2019; Zhou et al., 2019). Despite the above reasons, for comparison with the existing works, we just conditioned LOVAE by a high value such as 30.0 and 0.98 for LogP and QED, respectively, and 20k of molecular generation were tried.

Table 5 shows the property maximization results using ZINC250K. LOVAE generated the new molecules shown sufficiently high pLogP and QED properties. It is noteworthy that LOVAE showed good performance only with LogP condition without an additive part like the reward function or Bayesian optimization on latent space. In other words, LOVAE, which is a cVAE structure, works properly even in the extrapolation task. The top 3 molecules of each property are represented in Fig. 4. In addition, LOVAE with "pLogP" as condition was trained and verified. As the conditions of cVAE changed from [LogP, molwt, QED] to [pLogP, SA, cycle], it seems that LOVAE with pLogP learned a different $\mathbf{z}$ space from LOVAE. The performance was worse than LOVAE (with LogP), but since the number of large rings and SA can also be given as condition, the trend of the generated molecules was different (Fig. 5). As a result, LOVAE with pLogP generated 'usual' molecules, such as not including long aliphatic chains.

6 Conclusion

In this paper, LOVAE applying latent optimization technique to conditional VAE algorithm is proposed. By training the encoder first, the latent variable $\mathbf{z}$ has an updated distribution in the direction of reducing the train loss. Additionally, a reparameterized $\mathbf{z}$ is updated in the direction of reducing the reconstruction loss before training the decoder. The training of decoder becomes more efficient by utilizing our optimized $\mathbf{z}$. This was applied and verified in the inverse molecular design task in ZINC database, and confirmed that it showed a better appearance in the train loss, ELBO, and molecular generative performance than those of vanilla VAE and previous works.

REFERENCES

Ali Ismail Awad. Machine learning techniques for fingerprint identification: A short review. In *International Conference on Advanced Machine Learning Technologies and Applications*, pp. 524–531. Springer, 2012.

Konrad H Bleicher, Hans-Joachim Böhm, Klaus Müller, and Alexander I Alanine. Hit and lead generation: beyond high-throughput screening. *Nature reviews Drug discovery*, 2(5):369–378, 2003.

Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space of generative networks. In *International Conference on Machine Learning*, 2017.

Evan E Bolton, Yanli Wang, Paul A Thiessen, and Stephen H Bryant. Pubchem: integrated platform of small molecules and biological activities. In *Annual reports in computational chemistry*, volume 4, pp. 217–241. Elsevier, 2008.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10–21, 2016.

Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. The rise of deep learning in drug discovery. *Drug discovery today*, 23(6):1241–1250, 2018.

Arthur Dalby, James G. Nourse, W. Douglas Hounshell, Ann. K Gushurst, David L. Grier, Burton A. Leland, and John Laufer. Description of several chemical structure file formats used by computer programs developed at molecular design limited. *Journal of chemical information and computer sciences*, 32(3):244–255, 1992.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Will Grathwhohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Padro Luis Cunha Farias, and Aian Aspuru-Guzik. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *arXiv preprint arXiv:1705.10843*, 2017.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *Advances in neural information processing systems*, volume 2, pp. 6, 2017.

Touseef Iqbal and Shaima Qureshi. The survey: Text generation models in deep learning. *Journal of King Saud University-Computer and Information Sciences*, 2020.

John J Irwin, Teague Sterling, Michael M Mysinger, Erin S Bolstad, and Ryan G Coleman. Zinc: a free tool to discover chemistry for biology. *Journal of chemical information and modeling*, 52(7): 1757–1768, 2012.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning*, 2018.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Hierarchical generation of molecular graphs using structural motifs. *International Conference on Machine Learning (ICML)*, 2020.

Seokho Kang and Kyunghyun Cho. Conditional molecular design with deep generative models. *Journal of chemical information and modeling*, 59(1):43–52, 2018.

Kyungdoc Kim, Seokho Kang, Jiho Yoo, Youngchun Kwon, Youngmin Nam, Dongseon Lee, Inkoo Kim, Youn-Suk Choi, Yongsik Jung, Sangmo Kim, et al. Deep-learning-based inverse design model for intelligent discovery of organic molecules. *npj Computational Materials*, 4(1):1–7, 2018a.

Yoon Kim, Sam Wiseman, Andrew C. Miller, David Sontag, and Alexander M. Rush. Semi-amortized variational autoencoders. *International Conference on Machine Learning (ICML)*, 2018b.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Durk. P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *In Advances in neural information processing systems*, pp. 10215–10224, 2018.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pp. 3581–3589, 2014.

Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1945–1954. JMLR. org, 2017.

Greg Landrum. Rdkit: Open-source cheminformatics. URL `http://www.rdkit.org`.

Kien Nguyen, Clinton Fookes, Arun Ross, and Sridha Sridharan. Iris recognition with off-the-shelf cnn features: A deep learning perspective. *IEEE Access*, 6:18848–18855, 2017.

AkshatKumar Nigam, Pascal Friederich, Mario Krenn, and Alan Aspuru-Guzik. Augmenting genetic algorithms with deep neural networks for exploring the chemical space. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

Sharangdhar S Phatak, Clifford C Stephan, and Claudio N Cavasotto. High-throughput and in silico screenings in drug discovery. *Expert opinion on drug discovery*, 4(9):947–959, 2009.

Mariya Popova, Mykhailo Shvets, Junier Oliva, and Olexandr Isayev. Molecularrnn: Generating realistic molecular graphs with optimized properties. *arXiv preprint arXiv:1905.13372*, 2019.

David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

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

Huajie Shao, Shuochao Yao, Dachun Sun, Aston Zhang, Liu Liu, Shengzhong, Dongxin, Jun Wang, and Tarek Abdelzaher. Controlvae: Controllable variational autoencoder. *International Conference on Machine Learning (ICML)*, 2020.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

Gregor NC Simm, Robert Pinsler, and José Miguel Hernández-Lobato. Reinforcement learning for molecular design guided by quantum mechanics. *International Conference on Machine Learning (ICML)*, 2020.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pp. 3483–3491, 2015.

Teague Sterling and John J Irwin. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.

Hui-Po Wang, Wen-Hsiao Peng, and Wei-Jan Ko. Learning priors for adversarial autoencoders. *APSIPA Transactions on Signal and Information Processing*, 9, 2020.

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

Robin Winter, Floriane Montanari, Andreas Steffen, Hans Briem, Frank Noé, and Djork-Arné Clevert. Efficient multi-objective molecular optimization in a continuous latent space. *Chemical science*, 10(34):8016–8024, 2019.

Yan Wu, Jeff Donahue, David Balduzzi, Karen Simonyan, and Timothy Lillicrap. Logan: Latent optimisation for generative adversarial networks. *arXiv preprint arXiv:1912.00953*, 2019.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Chaochao Yan, Sheng Wang, Jinyu Yang, Tingyang Xu, and Junzhou Huang. Re-balancing variational autoencoder loss for molecule sequence generation. *arXiv preprint arXiv:1910.00698*, 2019.

Kevin Yang, Wengong Jin, Kyle Swanson, Regina Barzilay, and Tommi Jaakkola. Improving molecular design by stochastic iterative target augmentation. *International Conference on Machine Learning (ICML)*, 2020.

Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. *In Advances in neural information processing systems*, 2018a.

Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in neural information processing systems*, pp. 6410–6421, 2018b.

Zijun Zhang, Ruixiang Zhang, Zongpeng Li, Yoshua Bengio, and Liam Paull. Perceptual generative autoencoders. *International Conference on Machine Learning (ICML)*, 2020.

Zhenpeng Zhou, Steven Kearnes, Li Li, Richard N. Zare, and Patrick Riley. Optimization of molecules via deep reinforcement learning. *Scientific reports*, 9(1):1–10, 2019.