STRING-BASED MOLECULE GENERATION VIA MULTI-DECODER VAE

Kisoo Kwon\textsuperscript{1}, Kuhwan Jeong\textsuperscript{1}, Junghyun Park\textsuperscript{1}, Hwidong Na\textsuperscript{1}, Jinwoo Shin\textsuperscript{2}

\textsuperscript{1} Samsung Advanced Institute of Technology, Samsung Electronics, Korea
\textsuperscript{2} Korea Advanced Institute of Science and Technology, Daejeon, Korea

\section*{ABSTRACT}

In this study, we investigate the problem of string-based molecular generation via variational autoencoders (VAEs) that have served a popular generative approach for various tasks in artificial intelligence. Our main idea is to maintain multiple decoders while sharing a single encoder, i.e., it is a type of ensemble techniques. Here, we first found that training each decoder independently may not be effective as the bias of the ensemble decoder increases severely under its auto-regressive inference. To alleviate this issue, our proposed technique is two-fold: (a) a different latent variable is sampled for each decoder (from estimated mean and variance offered by the shared encoder) to encourage diverse characteristics of decoders and (b) a collaborative loss is used during training to control the aggregated quality of decoders using different latent variables. In our experiments, the proposed VAE model particularly performs well for generating a sample from out-of-domain distribution.

\textbf{Index Terms}— Ensemble, generative models, out-of-distribution, auto-regressive, novelty

\section{1. INTRODUCTION}

For the material (molecular) discovery via molecular generation, several approaches based on machine learning are actively adopting [1, 2, 3, 4]. Instead of human experts that design novel molecules using the domain knowledge, such a machine learning model can do a similar job by generating candidate molecules based on existing data annotated with physical properties. The ability to generate novel molecules is essential for successful molecular discovery. For generating string representation of novel molecules, we investigate variational autoencoders (VAEs) [5] with auto-regressive decoders, which is of popular choice as a deep generative model for such a generation task.

We aim for providing an orthogonal simple, yet effective idea. To this end, we use ‘ensemble’ techniques that are arguably the most trustworthy techniques for boosting the performance of machine learning models. They have played a critical role in the machine learning community to obtain better predictive performance than what could be obtained from any of the constituent learning models alone, e.g., Bayesian model/parameter averaging [6], boosting [7] and bagging [8]. However, under the VAE framework, researchers draw less attention to ensemble techniques as it is not straightforward how to aggregate predictions of multiple encoders and decoders of VAE for improving them.

In this paper, we design an ensemble version of VAE by utilizing multiple decoders in VAE, while sharing the encoder. In particular, our proposed idea ensembles logits from decoders in an auto-regressive generation process. Here, we found that training each decoder independently is not effective as the bias of the ensemble model increases under its auto-regressive inference.\footnote{For example, in our experiment, multiple decoders trained independently showed a higher reconstruction loss than vanilla VAE with the similar model size.} To maintain both small bias and variance of the ensemble model, each decoder is trained not only individually, but also in a way to collaborate with the others. Furthermore, to encourage diverse characteristics of decoders, we also propose to sample different latent variables for each decoder (from estimated mean and variance offered by the shared encoder) given a molecule during training.

We conducted experiments on a publicly available dataset on the molecular generation task for demonstrating the superiority of the proposed multi-decoder VAE model (MD-VAE). MD-VAE achieved lower training loss and higher reconstruction accuracy than baselines including VAE with k-annealing and Controllable VAE [9, 10]. The efficiency of generation for OOD-domain is more important than that for in-domain for the molecular discovery. We emphasize that MD-VAE achieved 31.4\% higher relative generative efficiency than Controllable VAE (ControlVAE) for this perspective. This is remarkable as VAEs are often reported to be poor for generating OOD-domain samples [11]. Although we apply MD-VAE to the molecular generation task, we believe that it can be applied to a broader range of domains, e.g., image generation with desired styles or text generation with desired sentiments.

\section{2. CONDITIONAL VAE}

The conditional VAE (cVAE) [12, 13] is designed to generate data given certain conditions such as classes or labels. In the cVAE, SMILES (string form) [14] of molecule $x$ is assumed to be generated from $p\theta(x|y, z)$ conditioned on target...
In this study, we propose a neural architecture for VAE, which we name MD-VAE. Our primary idea is to use multiple decoders while sharing a single encoder, as illustrated in Figure 1 (a). In particular, we consider an auto-regressive model (e.g., recurrent neural network or transformer). Then, to aggregate outputs of multiple decoders, the decoder’s logit values are averaged to predict the next token in an auto-regressive manner. Here, each decoder has its own separate parameters, and produces a different logit value. One may expect that such ensemble of different logits provides more robust prediction than that from an individual decoder. However, to the best of our knowledge, such ensemble version of VAE has not been explored in literature.

This is because a naive ensemble of independently trained decoders (while sharing a single encoder) may increase the model bias (although it may decrease the model variance), as it can boost up significantly under VAEs using auto-regressive decoders. To mitigate such an negative effect, we propose to
optimize the following additional collaborative loss to train our model:

\[
L_{\text{col}} = -\mathbb{E}_z \left[ \log \frac{1}{K} \sum_k p_{\phi_k}(x|y,z) \right].
\] (3)

Herein, \( K \) indicates the number of decoders of MD-VAE. We remark that such a collaborative loss may not be effective for ensemble of non-auto-regressive models, e.g., the standard classification or regression model, as it increases the model variance. However, in our case using auto-regressive models, it is effective as reducing the model bias is more crucial than doing the model variance. Nevertheless, even for our model, to reduce the model variance, we suggest to sample a different latent variable from the shared encoder for each decoder during training. It encourage for decoders to produce diverse outputs, and hence reduces the variance of the ensemble model. Specifically, each latent variable is sampled from the approximation of the posterior distribution of \( z \) given \( x \) and \( y \), and there are \( K \) sampled latent variables \( z_1, z_2, \ldots, z_K \) where

\[
z_k \sim \mathcal{N}(z_k|\mu_\phi(x,y), \text{diag}(\sigma_\phi(x,y))).
\] (4)

In summary, we integrate two ideas to train the proposed MD-VAE model. The collaborative loss promotes small bias of the ensemble prediction over multiple decoders, while sampling different latent variables for decoders does small variance of the ensemble model. Namely, we consider the following additional loss to train decoders in MD-VAE:

\[
L_{\text{dif, col}} = -\mathbb{E}_{z_1,\ldots,z_K} \left[ \log \frac{1}{K} \sum_k p_{\phi_k}(x|y,z_k) \right].
\] (5)

The only difference in Eq. (3) and Eq. (5) is to use the latent variable \( z \) or \( z_k \). The total reconstruction loss is a linear interpolation between collaborative and individual loss functions:

\[
L_{\text{MD}}^{\text{recon}} = \alpha L_{\text{dif, col}} - \frac{1 - \alpha}{K} \sum_{k=1}^{K} \log p_{\phi_k}(x|y,z_k),
\] (6)

where the second term indicates an individual loss of each decoder, and \( \alpha \) is a hyper-parameter to control a ratio of the collaborative loss. Eq. (6) is minimized together with the KLD loss in Eq. (2):

\[
L_{\text{total}}^{\text{MD}} = L_{\text{recon}} - \text{KLD}(q_\phi(\cdot|x,y)||p(\cdot)).
\] (7)

4. EXPERIMENTS

4.1. Experiment Setup

ZINC [18] is a database comprising information on various drug-like molecules. ZINC contains 3D structural information of molecules and molecular physical properties such as molecular weight (molWt), partition coefficient (LogP), and quantitative estimation of drug-likeness (QED). We used two subsets of ZINC: ZINC250K [19] and ZINC310K [20, 12]. The vocabulary for SMILES contains 39 different symbols, such as 1, 2, 3, [, ], H, B, C, N, O. The minimum, median, and maximum lengths of a SMILES string in ZINC250K are 9, 44, and 120, respectively. The average values of molwt, LogP, and QED are about 330, 2.457, and 0.7318 in ZINK250K, respectively. The three quantities are 313, 1.9029, 0.7527 in ZINK310K, respectively.

Each encoder and decoder of the VAE model consists of three layers with self-attention like transformer, and the dimension of the latent variable is set to 100. We used the Adam optimizer [21] with \( \beta_1 = 0.9, \beta_2 = 0.999, \) and \( \epsilon = 10^{-8} \), and the initial learning rate was 0.001. Each model was trained during 100 epochs, and batch size was 128. We compared six methods and its detailed information is as follows:

- **Base**: vanilla VAE with k-annealing [9]
- **ControlVAE**: controllable VAE [10]. It controls a weight of KLD term in Eq. (2) to have desired KLD value. According to the our experimental results, when the KLD loss was controlled as about 15, ControlVAE showed a better performance than Base.
- **MD**: ControlVAE with multi-decoder with only the individual loss and the same sampled \( z \).
- **MD_{col}**: ControlVAE with multi-decoder and collaborative loss, \( (\alpha = 0.5) \)
- **MD_{dif}**: ControlVAE with multi-decoder and multi-\( z \) sampling. Each decoder is trained by different \( z \) from the same input data.
- **MD_{dif,col}**: ControlVAE with multi-decoder, collaborative loss and multi-\( z \) sampling.

4.2. Evaluation: Training Phase and Reconstruction Success Rate

ZINC250K was used for the training data, and the reconstruction loss of each model was measured for the evaluation metric. In the cases of single decoder based VAE (Base, ControlVAE), the reconstruction loss is the sum of the cross-entropy for each token between the decoder’s output and the true label. For MDs (MD, MD_{col}, MD_{dif} and MD_{dif,col}), an ensemble of decoders’ outputs is used as an output for the recon-
Table 2. The conditional satisfaction in terms of MAE between a condition and simulation value of the top 1 molecules generated by each model (the lower the better): For a conditional molecular generation, the quality of generated molecules is one of the essential factor. For each condition (property), the best generated molecules are compared. For each case, molecular generation was tried 2,000 times by each model. (QED = 1.2861 is an improper condition because it was a physically absent region. The maximum QED from RDKit is 0.948.)

| Top1 MAE | In-domain | Out-of-distribution domain |
|---------|-----------|---------------------------|
| Property | molWt | LogP | QED | molWt | LogP | QED |
| Condition | 434, 330, 230 | 4.816, 2.457, 0.098 | 0.9598, 0.7318, 0.5038 | 580 | 84 | -3.281 |
| Base | 0.1520 | 0.0008 | 0.0041 | 0.0810 | 0.0780 | 0.0013 |
| ControlVAE | 0.1177 | 0.0005 | 0.0041 | 0.0800 | 0.0740 | 0.0005 |
| MD | 0.0940 | 0.0030 | 0.0040 | 0.1740 | 0.0980 | 0.0003 |
| MD_{col} | 0.0497 | 0.0013 | 0.0042 | 0.0760 | 0.0740 | 0.0111 |
| MD_{dif} | 0.0797 | 0.0007 | 0.0041 | 0.0470 | 0.0074 | 0.0051 |
| MD_{dif,col} | 0.0513 | 0.0004 | 0.0041 | 0.0620 | 0.1140 | 0.0005 |

4.3. Evaluation: Molecular Generative Efficiency and Conditional Satisfaction

The generative efficiency is a rate of the generated molecule satisfying validity, uniqueness, and novelty. The validity means that the generated molecule has a sound structure determined using the RDKit package [22]. For example, if the generative efficiency is 0.9, 90 molecules are sound, distinct and novel when the molecular generation is tried 100 times. To generate novel molecules using eVAE, latent variables are sampled from a normal distribution, and it is necessary to decide the condition that generated molecules should have. Each molecule of ZINC dataset has three properties: molWt, LogP and QED. One of the properties were determined manually as desired value, and the other properties were sampled from the conditional probability distribution of training dataset [12]. We specified two types of conditions: in-domain and y-extrapolation (OOD). In the in-domain case, the mean μ and lower and upper limits of 90% confidence interval (μ ± 1.645σ) of train dataset were used (molWt ∈ {330, 434, 230}, LogP ∈ {2.457, 4.816, 0.098}, QED ∈ {0.9598, 0.7318, 0.5038}). Meanwhile, in the OOD case, outlier values (μ ± 4σ) were used (molWt ∈ {580, 84}, LogP ∈ {8.194, −3.281}, QED ∈ {1.2861, 0.1775}). For each condition, 2,000 molecules were generated by each method. According to our experiments, ControlVAE and MD_{dif,col} showed 0.363 and 0.477 in terms of the molecular generative efficiency on OOD case. Furthermore, MD_{dif,col} (0.909) outperformed ControlVAE (0.880) on in-domain case.

Other important indicator is a conditional satisfaction for novel molecules. The discovery of new molecules with properties close to the target properties is essential. For this reason, we measured a mean absolute error (MAE) of the top 1 molecule having the smallest MAE to the input conditions, as shown in Table 2. Three properties of generated molecules were all calculated using RDKit. In the table, the dash (-) means that the model failed to generate even a single valid molecule. In the case of OOD, MD_{dif,col} outperformed ControlVAE except the condition MolWt = 84. In particular, MDs showed a sensational performance when the LogP’s condition was -3.281. In a nutshell, the proposed methods averagely showed a lower MAE than ControlVAE and Base.

5. CONCLUSION

In this study, we propose a simple method for molecular generation task. Our primary idea is to maintain multiple decoders while sharing a single encoder. To facilitate synergy between decoders, we introduce a collaborative loss function. We also propose to utilize a different latent variable for each decoder in order to diversify decoders while they collaborate with others to achieve a common goal. In our experiments, especially for out-of-distribution conditions, MD-VAE is far better than baselines with respect to generative efficiency and conditional satisfaction of generated molecules.
6. REFERENCES

[1] Benjamin Sanchez-Lengeling and Alán Aspuru-Guzik, “Inverse molecular design using machine learning: Generative models for matter engineering.” Science, vol. 361, no. 6400, pp. 360–365, 2018.

[2] Chence Shi, Minkai Xu, Zhaoyang 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.

[3] Wengong Jin, Regina Barzilay, and Tommi Jaakkola, “Junction tree variational autoencoder for molecular graph generation,” in International Conference on Machine Learning, 2018.

[4] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamin 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, vol. 4, no. 2, pp. 268–276, 2018.

[5] Diederik P. Kingma and Max Welling, “Auto-encoding variational bayes,” in Proceedings of the International Conference on Learning Representations (ICLR), 2014.

[6] Pedro Domingos, “Bayesian averaging of classifiers and the overfitting problem,” International Conference on Machine Learning (ICML), 2000.

[7] Robert Schapire Yoav Freund and Naoki Abe, “A short introduction to boosting,” Journal-Japanese Society For Artificial Intelligence, vol. 14, pp. 771–780, 1999.

[8] Leo Breiman, “Bagging predictors,” Machine learning, vol. 24, no. 2, pp. 123–140, 1996.

[9] 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, 2016, pp. 10–21.

[10] 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.

[11] Milton Llera Montero, Casimir JH Ludwig, Rui Ponte Costa, Gaurav Malhotra, and Jeffrey Bowers, “The role of disentanglement in generalisation,” in Proceedings of the International Conference on Learning Representations (ICLR), 2020.

[12] Seokho Kang and Kyunghyun Cho, “Conditional molecular design with deep generative models,” Journal of chemical information and modeling, vol. 59, no. 1, pp. 43–52, 2018.

[13] 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, 2014, pp. 3581–3589.

[14] David Weininger, Arthur Weininger, and Joseph L Weininger, “Smiles. 2. algorithm for generation of unique smiles notation,” Journal of chemical information and computer sciences, vol. 29, no. 2, pp. 97–101, 1989.

[15] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.

[16] Hasim Sak, Andrew W. Senior, and Françoise Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” arXiv preprint arXiv:1402.1128, 2014.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems, Jun 2017, vol. 2017-Decem, pp. 5999–6009.

[18] Teague Sterling and John J Irwin, “Zinc 15–ligand discovery for everyone,” Journal of chemical information and modeling, vol. 55, no. 11, pp. 2324–2337, 2015.

[19] 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. JMLR. org, 2017, pp. 1945–1954.

[20] 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.

[21] Sebastian Ruder, “An overview of gradient descent optimization algorithms,” arXiv preprint arXiv:1609.04747, 2016.

[22] Greg Landrum, “Rdkit: Open-source cheminformatics,”