JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

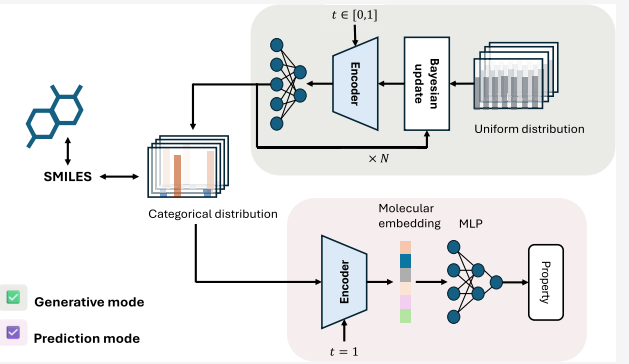
Bayesian Flow Network Framework for Chemistry Tasks

Nianze Tao* and Minori Abe*

Cite This: https://doi.org/10.1021/acs.jcim.4c01792 Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: In this work, we introduce ChemBFN, a language model that handles chemistry tasks based on Bayesian flow networks working with discrete data. A new accuracy schedule is proposed to improve sampling quality by significantly reducing reconstruction loss. We show evidence that our method is appropriate for generating molecules with satisfied diversity, even when a smaller number of sampling steps is used. A classifier-free guidance method is adapted for conditional generation. It is also worthwhile to point out that after generative training, our model can be fine-tuned on regression and classification tasks with state-of-the-art performance, which opens the gate of building all-in-one models in a single module style. Our model has been open sourced at https://github.com/Augus1999/bayesian-flow-network-for-chemistry.



INTRODUCTION

Autoregressive models (ARs) including SMILES-based or fragment-based models[1−9] that leverage the power of language models (LMs) and reinforcement learning[7−9] and graph-based models[10−15] coupled with advanced techniques such as Monte Carlo tree search[11−13] have proved their success in several de novo design benchmarks[6,16] consisting drug-like molecules. The constraint of ARs, i.e., the number of sampling steps, is the size of generated object, which limits the potential of generating large molecules. Conversely, the recently emerging denoising-diffusion models[17] (DMs) offer a way to generate objects of any size within a fixed sequence of sampling process. However, it has been pointed out in the research of Vignac et al.[18] that SMILES-based models generally worked better than graph DMs even when a dedicatedly designed discrete diffusion method was applied.

Bayesian flow networks[19] (BFNs) are in a different category of generative models that decouple the sampling process with the size of generated objects as well. Different from DMs, BFNs directly work on the parameters of data distributions, which naturally enable them to handle both continuous (including discretized) and discrete data without any data preprocessing or change of (mathematical) framework. Although the authors of BFN showed evidence in the original paper[19] that BFN had an advantage over discrete DMs on discrete data generating, e.g., text generation, the recent research considering de novo molecule design only successfully employed it on continuous and discretized data, e.g., 3D molecular conformation generation[20] rather than language-like representations such as SMILES[21] or SELFIES.[22] One potential reason discouraging the application to text generation is the lack of exact analytical expression for the accuracy schedule $\beta(t)$, one critical component of BFNs, in the discrete case, while the speculated quadratic $\beta(t)$ in the original paper is, as admitted by the authors,[19] suboptimal.

In this paper, we introduce ChemBFN, a Bayesian Flow Network framework for chemistry tasks, that leverages our newly proposed accuracy schedule and transformer[23] encoder model to generate 1D language-like molecular representations, e.g., SMILES and SELFIES. The experiments demonstrated that models with our accuracy schedule outperform those with the quadratic accuracy schedule. Besides, generative training of the BFN method can be a powerful pretraining strategy for downstream tasks in molecular property predictions, including regressions and classifications and reaction yield predictions.

METHODS

Discrete Bayesian Flow Networks. A functional BFN consists of a neural network (NN) model that converts the input distribution $\boldsymbol{p}_I(\boldsymbol{x}|\boldsymbol{\theta})$ into the output distribution $\boldsymbol{p}_O(\boldsymbol{x}|\boldsymbol{\theta}; t)$ and a Bayesian update process that updates the pervious input distribution to the current state according to a sender distribution $\boldsymbol{p}_S(\boldsymbol{y}|\boldsymbol{x}; \alpha)$, where $\boldsymbol{\theta}$ is the parameter of data $\boldsymbol{x}$, and $\boldsymbol{y}$ is a sample of $\boldsymbol{x}$.[19] The none-negative monotonic increasing function α, namely, accuracy rate, guides the sender distribution

Received: September 30, 2024
Revised: January 5, 2025
Accepted: January 8, 2025

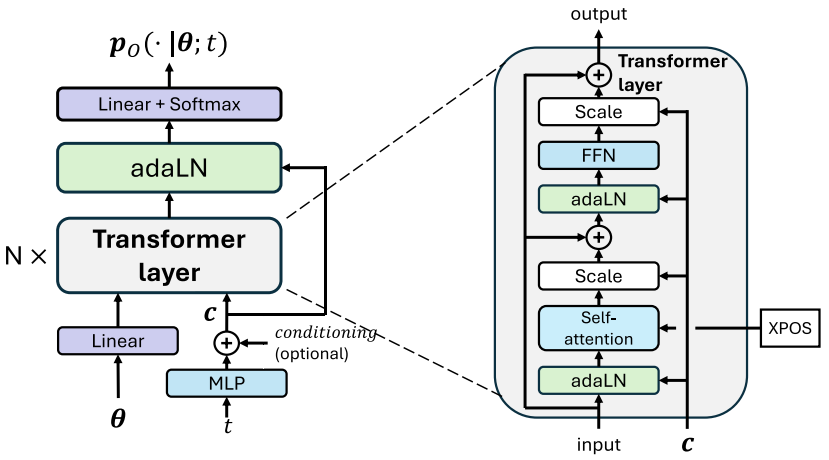


Figure 1. Visualized scheme of our model. The architecture is inspired by DiT.[24] The multihead self-attention layers did not use causal masking, which is the same as BERT[28] while we replaced the commonly used positional embedding method (absolute positional embedding used in DiT, BERT, and RoBERTa[29] models) with the novel X POS[26] variation of rotary positional embedding.[27] Note that each FFN (feed-forward network) layer adsorbs a dropout layer.

to move in a more informative direction along with the time.[19] Since α can be either continuous or discretized, a continuous accuracy schedule $\beta(t)$ is defined instead, which generates α as

$$\alpha = \begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\beta(t), & \text{when } \alpha \text{ is continuous} \\ \beta(t_i) - \beta(t_{i-1}), & \text{when } \alpha \text{ is discretized} \end{cases} \quad (1)$$

In the discrete case, (1) all distributions are K-class categorical distributions; (2) the sample is defined as $y = \mathcal{N}(\alpha(Ke_x - 1), \alpha KI)$ when Gaussian sampling is utilized, where $\boldsymbol{e}_x$ is the one-hot representation of data $\boldsymbol{x}$; and (3) the Bayesian update function is defined as $h(\theta^{(d)}, y^{(d)}, \alpha) = e^{y^{(d)}}\theta^{(d)} / \sum_{k=1}^{K} e^{y_k^{(d)}}\theta_k^{(d)}$, where $^{(d)}$ is the d^{th} parameter.[19]

During the training stage, a receiver distribution $\boldsymbol{p}_{\mathrm{R}}(\hat{\mathbf{y}}|\boldsymbol{\theta}; t, \alpha)$ is drawn by sampling the output of NN with the same sampling method as sender distribution.[19] The model is optimized by minimizing the Kullback−Leibler divergence between the receiver distribution and the sender distribution, which is decoupled as an n-step loss (L^n) and a reconstruction loss (L^r) and only the first loss in practice is used.[19] The limit case, i.e., continuous time loss, $L^\infty = \lim_{n\to\infty} L^n$ has been proved to be more efficient.[19] During the sampling (generating) stage, since the receiver distribution has been trained to match the sender distribution, i.e., $p_{\mathrm{R}}(\hat{\mathbf{y}}|\boldsymbol{\theta}; t, \alpha) \sim p_{\mathrm{S}}(\mathbf{y}|\mathbf{x}; \alpha)$, the receiver distribution is used in Bayesian update process directly to update the input distribution (initialized as a uniform distribution over K categories) where a discretized α is employed.

Model Architecture. Our model is an adaptation of the DiT[24] model. The differences in our implementation include the following: (1) the use of categorical distributions rather than image embeddings for input tokens because we are not dealing with images; (2) logit outputs that are then transformed into probabilities by the softmax function; (3) replacement of activation function with scaled exponential linear units (SELU)[25] function; (4) use of a 2-layer multilayer perceptron (MLP) to form time embedding since "time" in BFN is continuous from 0 to 1; and (5) employment of XPOS[26] variation of rotary positional embedding.[27] The architecture is shown in Figure 1.

Following the notations of the BFN paper,[19] the parameter of categorical distributions inputted into the NN is denoted by $\boldsymbol{\theta} = (\theta^{(1)}, \theta^{(2)}, ..., \theta^{(D)}) \in [0,1]^{KD}$ (K is the number of categories, D is the number of input data, and $\theta^{(d)}$ is the d^{th} parameter) and the output distribution at time step t is denoted by $\boldsymbol{p}_{\mathrm{O}}(\cdot|\boldsymbol{\theta}; t) \in [0,1]^{KD}$. We denote the sum of the time embedding vector and conditioning vector as c. A null conditioning ϕ is equivalent to zero vector $\mathbf{0}$.

In each experiment described in the later text, we employed the same hyperparameters of the model except category number K that depends on molecular representations. The two-layer MLP with SELU activation has the shape of [1, 256, 512]. We employed 12 transformer layers, which had 8 attention heads each, with the attention temperature $\tau = \sqrt{2d_{\mathrm{h}}}$ (d_{h} is the feature number of each attention head).[30] The dropout rate was 0.01, and the hidden feature number was 512. These settings lead to total learnable parameters of the model of the magnitude of 54 M.

New Accuracy Schedule. In the case of BFN, an accuracy schedule function $\beta(t)$ drives the expectation of entropy of the input distribution $\mathbb{E}_{\mathrm{p_F}(\boldsymbol{\theta}|\boldsymbol{x};t)} H[p_I(\boldsymbol{x}|\boldsymbol{\theta})]$ to decrease linearly with t, where $\boldsymbol{x}$ stands for the clear data, $\mathrm{p_F}(\boldsymbol{\theta}|\boldsymbol{x}; t)$ represents Bayesian flow distribution, and $\boldsymbol{p}_{\mathrm{I}}(\boldsymbol{x}|\boldsymbol{\theta})$ is the input distribution as denoted in the original paper.[19] The mathematical difficulty of deriving the expectation analytically in the discrete case compels us to speculate from intuition. The authors of BFN claimed that "$\beta(t) = t^2\beta(1)$ was a reasonable approximation" but disclosed later that finding a suitable value for the hyperparameter $\beta(1)$ was not an easy job.[19]

Here, we give our estimation of $\beta(t)$. If we estimate the expected entropy of the input distribution (denoted as $\boldsymbol{E}$ for short) as $\boldsymbol{E} \sim f(K)\mathrm{e}^{-K/4\beta(t)}$, then the relationship $E(t) = (1 - t)E(0) + tE(1)$ that eliminates the unknown factor $f(K)$ gives us

$$\beta(t) = -\frac{4}{K}\ln(1 - t + te^{-K/4\beta(1)}) \quad (2)$$

and the corresponding

$$\alpha(t) = \frac{\mathrm{d}\beta}{\mathrm{d}t} = \frac{4}{K}\frac{1 - e^{-K/4\beta(1)}}{1 - t + te^{-K/4\beta(1)}} \quad (3)$$

https://doi.org/10.1021/acs.jcim.4c01792
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

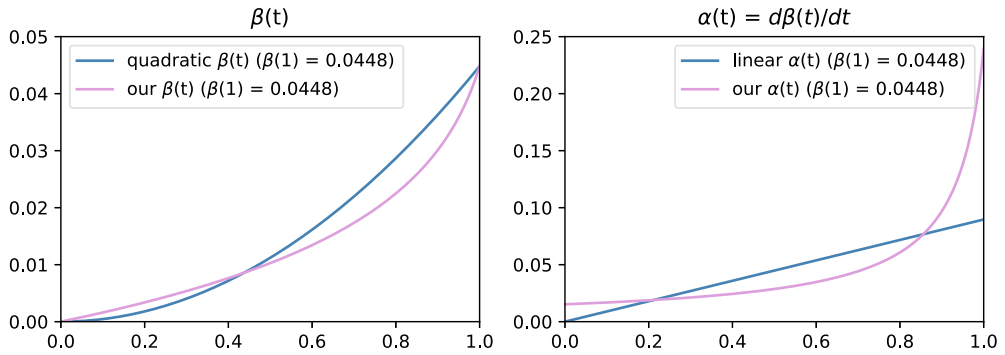


Figure 2. Comparing our accuracy schedule with quadratic accuracy schedule initialized with the same value of $\beta(1)$. (left) Accuracy schedules $\beta(t)$. (right) Accuracy rates $\alpha(t)$. Note that our $\beta(t)$ does not deviate too much from quadratic one, yet the rate (derivative) differs substantially as t goes to 1.

where $\beta(1)$ is still a hyperparameter. Equation 3 changes the continuous time loss L^{∞} to

$$L^{\infty}(x) = \frac{K}{2}\mathbb{E}_{t\sim U(0,1),\mathrm{p_F}(\theta|x;t)}(\alpha(t)\|e_x - \mathrm{e}(\hat{\theta};\, t)\|^2) \tag{4}$$

where e_x is the one-hot representation of data x while $\mathrm{e}(\hat{\theta};\, t)$ is the predicted categorical distribution of data x at time t. Note that when $\beta(1)$ is large, $\alpha(1)$ goes to extremely large. Therefore, we limit $\alpha(1) \leq 32\beta(1)$, from which

$$\beta(1)_{\max} \approx 20.4054/K \tag{5}$$

is obtained. An example of how our accuracy schedule looks different from the original one is plotted in Figure 2. We shall show in later experiments that our $\beta(t)$ in eq 2 works better than quadratic ones.

Data Sets and Benchmarks. Two benchmarks—MOSES[16] and GuacaMol[6]—were used to evaluate the generative performance, e.g., the similarity between generated molecules and training molecules, of ChemBFN. We reported the distribution-learning metrics of these benchmarks in Experiments and Results. A summary of these metrics is listed in Table 1.

Table 1. Brief Summary of Used Metrics of MOSES and GuacaMol Benchmarks

metrics	description
valid	fraction of valid molecules
unique	fraction of unique molecules
IntDiv_1 and IntDiv_2	internal diversities
novelty	fraction of generated unseen molecules compared with training data
FCD	Fréchet ChemNet distance[31]
SNN	Tanimoto similarity to a nearest neighbor
Frag	BRICS fragment[32] cosine similarity
Scaf	Bemis–Murcko scaffold[33] cosine similarity
filter	fraction of molecules that fit predefined constructions
KL divergence	Kullback–Leibler divergence

The QM9[34] data set was employed to study the capability of conditional generation of our method. We randomly selected 110,000 molecules, before which 3054 invalid data were removed, with the triple ($\in_{\mathrm{HOMO}}$, $\in_{\mathrm{LUMO}}$, $\Delta\in_{\mathrm{HOMO-LUMO}}$) as the conditioning label to form the training set.

In order to evaluate the downstream performance, 40 M unique SMILES and 190 M unique SMILES strings were randomly selected from easily accessed ZINC15[35] database that formed two pretraining sets. The model trained on the 40 M set was finetuned on several regression (ESOL, FreeSolv, Lipo, etc.) and classification (BBBP, BACE, HIV, etc.) tasks, including the subsets of widely used MoleculeNet[36] benchmark. A brief description of the MoleculeNet tasks used is in Table 2. Each data set was split into training/validation/testing sets in the ratio of 80/10/10, following the scaffold splitting method proposed in DeepChem[37] project. We reported ROC-AUC (area under receiver operating characteristic curve) for classification tasks and RMSE (root-mean-squared error) for regression tasks in Experiments and Results. In addition to the tasks of MoleculeNet, two less biased data sets—the public ADME data set published by Fang et al.[38] consisting of 6 dedicatedly collected absorption, distribution, metabolism, and excretion (ADME) in vitro end points together with a kinase inhibitor data set prepared by Wu et al.[39] that contains bioactivities of total 141,086 compounds for 354 kinases—were employed to further benchmark our method in activity prediction. A brief summary of subtasks of the ADME data set is in Table 2. For the ADME data set, we employed the same split provided by Fang et al.;[38] for kinase inhibitor data set, we prepared a random split and a

Table 2. Brief Summary of Used MoleculeNet and Public ADME Tasks

name	no. molecules	no. tasks	label
ESOL	1128	1	aqueous solubility $\log_{10}$ (S/mol·L^{-1})
freeSolv	642	1	experimental hydration free energy/kcal/mol
Lipo	4200	1	octanol/water distribution coefficient $\log_{10}D_{7.4}$
HLM	3087	1	logarithm of human liver microsomal stability/mL· min^{-1} kg^{-1}
RLM	3054	1	logarithm of rat liver microsomal stability/mL· min^{-1} kg^{-1}
hPPB	1808	1	logarithm of human plasma protein binding (percent unbound)
rPPB	884	1	logarithm of rat plasma protein binding (percent unbound)
MDR1-MDCK ER	2642	1	$\log_{10}$ (MDR1-MDCK efflux ratio)
solubility	2173	1	aqueous solubility $\log_{10}$ (S/μ mL^{-1}) at PH = 6.8
BBBP	2039	1	if a compound penetrates the blood–brain barrier
BACE	1513	1	if a compound inhibits BACE-1 protein
HIV	41,127	1	if a compound is an HIV inhibitor

https://doi.org/10.1021/acs.jcim.4c01792
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

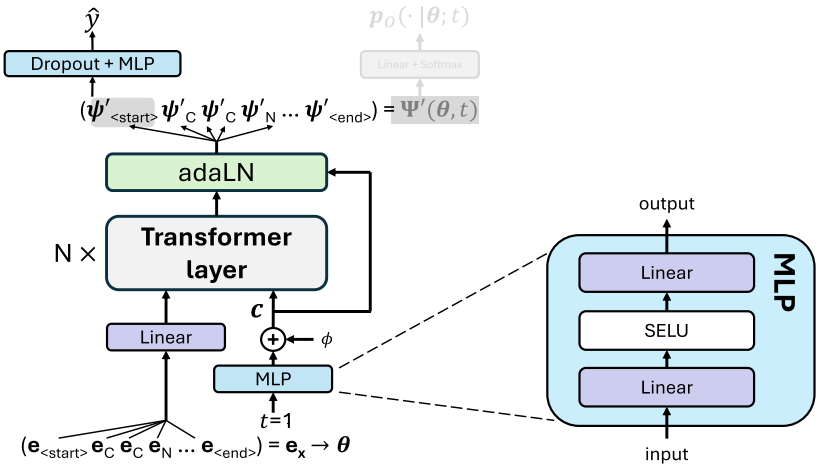


Figure 3. Fine-tuning strategy of our model. The predicted label $\hat{y} \in \mathbb{R}^n$ is mapped by a MLP from embedding of the ⟨start⟩ token $\psi'_{\langle start \rangle}$ restricted by $t = 1$. The MLP used here had 2 linear layers with a SELU activation function between them in a size of $[512, 256, n_{task}]$. Note that at prediction mode, the linear layer that maps latent vectors to output distributions is not activated; The conditioning is biased to null ϕ; all ⟨pad⟩ tokens are masked out in attention.

Table 3. Comparing Scores of MOSES Benchmark When Varying $\beta(1)$ Value of Different Accuracy Schedules[a]

	$\beta(1)$	valid ↑	FCD ↓	SNN ↑	Frag ↑	Scaf ↑	filters ↑	novelty ↑
quad	0.15	0.893 ± 0.001	3.438 ± 0.034	0.559 ± 0.000	0.985 ± 0.000	0.095 ± 0.001	0.982 ± 0.000	**0.900** ± 0.002
	0.0829	0.895 ± 0.001	3.772 ± 0.012	0.551 ± 0.001	0.984 ± 0.001	**0.096** ± 0.006	0.985 ± 0.001	**0.900** ± 0.002
	0.0448	0.899 ± 0.003	3.902 ± 0.045	0.561 ± 0.000	0.988 ± 0.000	0.089 ± 0.006	0.986 ± 0.001	0.887 ± 0.003
ours	0.0829	**0.900** ± 0.001	**2.731** ± 0.015	0.563 ± 0.000	**0.990** ± 0.000	0.091 ± 0.004	**0.987** ± 0.001	0.886 ± 0.000
	0.0448	**0.900** ± 0.001	3.580 ± 0.008	**0.568** ± 0.000	0.987 ± 0.000	0.075 ± 0.006	**0.987** ± 0.000	0.877 ± 0.001

[a]↑ indicates that higher is better and ↓ stands for the contrary. The best results are in bold. We used a sampling step of 1 k.

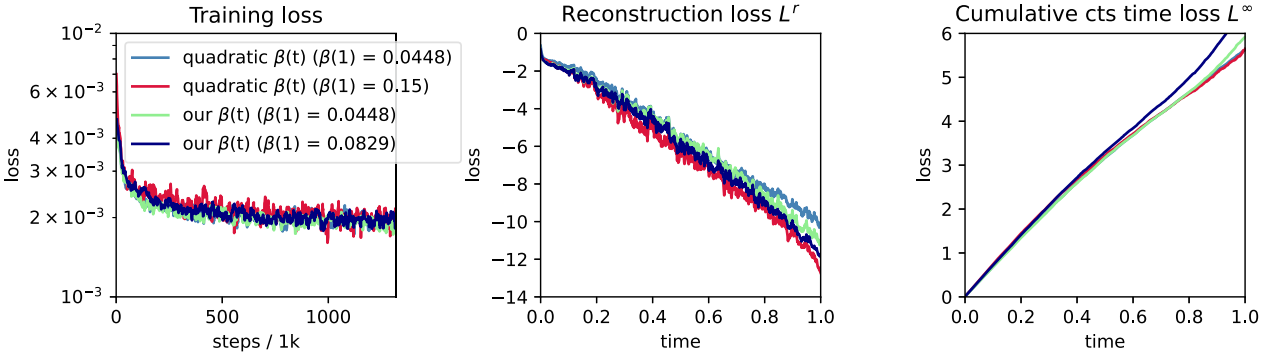


Figure 4. Visualization of the impact on training loss, reconstruction loss L^r, and continuous (cts) time loss L^∞ of different accuracy schedules with different values of $\beta(1)$. L^r and L^∞ were computed on 1 k discretized steps after training.

scaffold split (both had training/validation/testing = 80/10/10). The testing MAE, RMSE, Pearson's correlation coefficient (R value), and averaged ROC-AUC were reported in Experiments and Results.

The USPTO-50k[40] data set and Buchwald−Hartwig and Suzuki−Miyaura reaction yield data sets from high-throughput experiments (HTE) cleaned by Schwaller et al.[41] were employed to train the model to predict reaction yield. USPTO-50k that contains 50k reactions mined from patents were used to pretrain the model, while HTE data were used for fine-tuning. We report coefficient of determination (R^2 score) on testing sets in Experiments and Results.

AqSolDB,[42] a more challenging solubility data set containing more species than ESOL, was used to investigate the effect of the size of pretraining data. A training/validation/testing (80/10/10) split was generated using the scaffold splitting method. Testing MAE (mean absolute error) and RMSE are reported in Experiments and Results.

For SMILES representation, we developed a universal tokenizer that generates a fixed number (specifically $K = 246$) of unique vocabulary for any collection of molecules. A similar strategy was not applicable to SELFIES strings, which were translated from SMILES via the official selfies[22] package, whereby the vocabulary should be computed separately for each data set and the category number K varies. Note that we include the three special tokens ⟨start⟩, ⟨end⟩, and ⟨pad⟩ in the vocabulary.

Fine-Tuning Strategy. Similar to the strategy of ChemBERTa models,[43,44] embedding, denoted as $\psi'_{\langle start \rangle}$, of the ⟨start⟩ token at time $t = 1$ was used as a fingerprint for downstream tasks. A 2-layer MLP absorbing a dropout layer was used as the prediction head. We replaced the input distribution in

https://doi.org/10.1021/acs.jcim.4c01792
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Table 4. Scores of MOSES and GuacaMol Benchmarks When Different Padding Strategies Were Used during Training[a]

strategy	valid ↑	FCD ↓	SNN ↑	frag ↑	scaf ↑	filters ↑	novelty ↑
MOSES							
dynamic	0.900 ± 0.001	2.731 ± 0.015	0.563 ± 0.000	**0.990** ± 0.000	0.091 ± 0.004	**0.987** ± 0.001	**0.886** ± 0.000
global	**0.916** ± 0.001	**2.730** ± 0.014	**0.565** ± 0.001	**0.990** ± 0.000	**0.094** ± 0.002	**0.987** ± 0.001	0.880 ± 0.002

	valid ↑	unique ↑	novelty ↑	KL divergence ↑	FCD ↑
GuacaMol					
dynamic	0.799 ± 0.003	0.815 ± 0.002	**0.975** ± 0.000	**0.810** ± 0.001	0.370 ± 0.003
global	**0.807** ± 0.003	**0.818** ± 0.001	**0.975** ± 0.001	0.808 ± 0.010	**0.399** ± 0.002

[a]↑ for higher is better and ↓ for the contrary. The best results are in bold. We used a sampling step of 1 k.

Table 5. Testing Metrics on MOSES Test Set Compared with SOTA Models[a]

	method	valid ↑	unique@1 k ↑	unique@10 k ↑	IntDiv1 ↑	IntDiv2 ↑	novelty ↑
Ars	JTN-VAE[5]	**1.0** ± 0.0	**1.0** ± 0.0	**1.0** ± 0.0	0.855 ± 0.003	0.850 ± 0.004	0.913 ± 0.006
	LatentGAN[3]	0.897 ± 0.003	**1.0** ± 0.0	0.997 ± 0.000	**0.857** ± 0.001	**0.851** ± 0.001	0.950 ± 0.001
	GraphINVENT[10]	0.964	**1.0**	0.998	**0.857**	**0.851**	
	MolGPT[4]	0.994		**1.0**	**0.857**	**0.851**	0.797
DMs	DiGress[18]	0.857		**1.0**			**0.950**
BFNs	ChemBFN	0.916 ± 0.001	**1.0** ± 0.0	0.998 ± 0.000	0.836 ± 0.000	0.830 ± 0.000	0.880 ± 0.002
	ChemBFN[b]	**1.0** ± 0.0	**1.0** ± 0.0	**1.0** ± 0.0	0.847 ± 0.000	0.841 ± 0.000	0.940 ± 0.001

[a]The metrics of all other models were copied from the original paper. ↑ indicates that higher is better. We sampled the molecules for 1000 steps. The best results are in bold. [b]For SELFIES version.

Table 6. Metrics on MOSES Scaffold Test Set[a]

	method	FCD ↓	SNN ↑	Frag ↑	Scaff ↑	filters ↑
ARs	JTN-VAE[5]	0.938 ± 0.053	0.519 ± 0.007	0.995 ± 0.000	0.101 ± 0.011	0.976 ± 0.002
	LatentGAN[3]	**0.828** ± 0.012	0.513 ± 0.000	**0.997** ± 0.001	0.107 ± 0.010	0.974 ± 0.001
	GraphINVENT[10]	1.223	0.539	0.986	0.127	0.950
DMs	DiGress[18]	1.19	0.52		**0.148**	0.971
BFNs	ChemBFN	2.730 ± 0.014	**0.565** ± 0.001	0.990 ± 0.000	0.094 ± 0.002	**0.987** ± 0.001
	ChemBFN[b]	4.473 ± 0.058	0.524 ± 0.001	0.976 ± 0.000	0.141 ± 0.008	0.962 ± 0.001

[a]Settings are the same as Table 5 while ↓ indicates that lower is better. [b]For SELFIES version.

Table 7. Testing Metrics on GuacaMol Distribution-Learning Tasks[a]

	method	valid ↑	unique ↑	novelty ↑	KL divergence ↑	FCD ↑
ARs	MolGPT[4]	0.981	0.998	**1.0**	**0.992**	0.907
	SMILES LSTM[6]	0.959	**1.0**	0.912	0.991	**0.913**
	VGAE-MCTS[12]	**1.0**	**1.0**	**1.0**	0.659	0.009
DMs	DiGress[18]	0.852	**1.0**	0.999	0.929	0.680
BFNs	ChemBFN	0.807 ± 0.003	0.818 ± 0.001	0.975 ± 0.001	0.808 ± 0.010	0.399 ± 0.002
	ChemBFN[b]	**1.0** ± 0.0	0.850 ± 0.003	0.994 ± 0.001	0.811 ± 0.002	0.142 ± 0.003

[a]Settings are the same as Table 5. [b]For SELFIES version.

generative mode with the one-hot representation of data (token), i.e., $\boldsymbol{\theta} \leftarrow \boldsymbol{e}_{\boldsymbol{x}} = (\boldsymbol{e}_{\langle \text{start} \rangle}, \ldots, \boldsymbol{e}_{\langle \text{end} \rangle}) \in \{0,1\}^{KD}$ in this stage. A visualized scheme is shown in Figure 3.

EXPERIMENTS AND RESULTS

Unconditional Generation. We first evaluate the effect of different $\beta(t)$ with different values of $\beta(1)$ using the MOSES data set. We reported the validity, FCD on scaffold set, SNN on scaffold set, Frag on scaffold set, Scaf on scaffold set, Filters, and Novelty scores computed by MOSES program in Table 3 together with reconstruction loss $L^{\mathrm{r}} = -\mathbb{E}_{\mathrm{p_F}(\boldsymbol{\theta}|\boldsymbol{x};t)} \ln p_{\mathrm{O}}(x|\theta; t)$ and continuous time loss L^{∞} in Figure 4. It is clear that raising $\beta(1)$ in both quadratic and our schedules did not have obvious influence on training loss but lowered L^{r}, while our schedule lead to a lower loss when $\beta(1)$ was the same. The effect on L^{∞} was subtle. However, after we calculated the R^2 values of the cumulative L^{∞} curves, we found that while using quadratic $\beta(t)$ the curve became more distorted when $\beta(1)$ was larger ($R^2|_{\beta(1)=0.0448} = 0.995$ while $R^2|_{\beta(1)=0.15} = 0.992$); after switching to our $\beta(t)$, the curves were more linear (i.e., L^{∞} was more uniform) and the linearity was not affected by the value of $\beta(1)$ ($R^2|_{\beta(1)=0.0448} = R^2|_{\beta(1)=0.0829} = 0.997$). The metrics in Table 3 provide more quantitative evidence that our $\beta(t)$ is more optimal. It is notable that a larger $\beta(1)$ value usually results in better scores. Therefore, we conclude here that our proposed $\beta(t)$ with $\beta(1) = \beta(1)_{\max} = 20.4054/\mathrm{K}$ is a more optimal solution in discrete BFNs.

In the above experiments, we used a dynamic padding strategy, i.e., each batch was padded to the maximum length of that batch, to reduce the training time. In the following experiments, a global padding strategy, i.e., padding all batches

https://doi.org/10.1021/acs.jcim.4c01792
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

Table 8. Scores of MOSES and GuacaMol Benchmarks When Different Numbers of Sampling Steps Were Used[a]

step	valid ↑	FCD ↓	SNN ↑	frag ↑	scaf ↑	filters ↑	novelty ↑
MOSES							
10	0.835 ± 0.003	2.768 ± 0.035	0.533 ± 0.000	0.988 ± 0.000	0.145 ± 0.004	0.976 ± 0.001	0.921 ± 0.002
100	0.911 ± 0.002	2.604 ± 0.040	0.562 ± 0.001	0.991 ± 0.000	0.103 ± 0.005	0.985 ± 0.001	0.884 ± 0.002
1 k	0.916 ± 0.001	2.730 ± 0.014	0.565 ± 0.001	0.990 ± 0.000	0.094 ± 0.002	0.987 ± 0.001	0.880 ± 0.002
10[b]	1.0 ± 0.0	11.79 ± 0.09	0.422 ± 0.001	0.965 ± 0.001	0.118 ± 0.0164	0.806 ± 0.001	0.991 ± 0.000
100[b]	1.0 ± 0.0	4.802 ± 0.045	0.517 ± 0.000	0.976 ± 0.001	0.141 ± 0.008	0.955 ± 0.001	0.947 ± 0.001
1k[b]	1.0 ± 0.0	4.473 ± 0.058	0.524 ± 0.001	0.976 ± 0.000	0.141 ± 0.008	0.962 ± 0.001	0.940 ± 0.001

	valid ↑	unique ↑	novelty ↑	KL divergence ↑	FCD ↑
GuacaMol					
10[b]	1.0 ± 0.0	0.853 ± 0.002	1.0 ± 0.0	0.451 ± 0.001	0.000 ± 0.000
100[b]	1.0 ± 0.0	0.846 ± 0.003	0.994 ± 0.000	0.803 ± 0.003	0.110 ± 0.003
1k[b]	1.0 ± 0.0	0.850 ± 0.003	0.994 ± 0.001	0.811 ± 0.002	0.142 ± 0.003

[a]↑ For higher is better and ↓ for the contrary. [b]For SELFIES version.

to a global maximum length, was employed to compare with a dynamic strategy on both MOSES and GuacaMol benchmarks. The results are summarized in Table 4. We found that the global padding method benefited the performance. In the following experiment, we therefore employed the global padding method in generative tasks.

Finally, we trained models by applying the above optimal settings (i.e., $\beta(1) = 20.4054$/K and global padding) on MOSES and GuacaMol data sets. Both the SMILES and SELFIES versions were implemented. The comparison with published state-of-the-art (SOTA) models[3−6,10,12,18] are summarized in Tables 5−7. Table 8 shows the effect of reduced sampling. We found that (1) except FCD, metrics of both SMILES version and SELFIES version were close to SOTA performance; (2) number of sampling step, as expected, affected the validity of the generated molecules (for SMILES version only because SELFIES always gives valid molecules[22]), but dropping from 1 k steps to 100 steps did not degrade the performance a lot. If lower validity is acceptable, only sampling 10 steps significantly reduces the computational time without much impact on other qualities. Larger FCD (lower FCD score in terms of GuacaMol, where FCD score = $e^{-0.2\text{FCD}}$) is a hint that BFNs learn the grammar of molecules rather than the way of combining characters within the data set.

Conditional Generation of Small Molecules. The classifier-free guidance[45] method is easily adapted into BFN, where only the computation of output distribution needs changing during the sampling process. The pseudocode for computing discrete output distribution is presented in algorithm 1. In the experiment, we jointly trained a model conditionally and unconditionally on the QM9 data set with the unconditional rate $p_{\text{uncond}} = 0.2$. In the sampling stage, w was set to 4. We sampled 10 molecules using the label $[-0.249, 0.0615, 0.3105]$ that was transformed to $\boldsymbol{y}$ via a trained 2-layer MLP. Ten unconditioned samples were generated as a control group. RDKit[46] was employed to generate the 3D conformations, then the geometry optimizations and energy calculations were performed via PySCF[47] at B3LYP/6-31G(2df,p) level of accuracy. The results of the MAE between calculated values and labels are presented in Table 9. The conditioned samples are displayed in Figure 5.

Algorithm 1 Invoking classifier-free guidance into output distribution

Require: $w \in \mathbb{R}$, conditioning vector $\boldsymbol{y}$
  function DISCRETE_OUTPUT_DISTRIBUTION($\boldsymbol{\theta} \in [0,1]^{KD}$, $t \in [0,1]$, $\boldsymbol{y} \in \mathbb{R}^f$)
    Input $(\boldsymbol{\theta}, t, \boldsymbol{y})$ to network, receive $\boldsymbol{\Psi}(\boldsymbol{\theta}, t, \boldsymbol{y})$ as output
    if in training stage or $\boldsymbol{y}$ is ϕ **then**
      $\boldsymbol{p}_O(\cdot|\boldsymbol{\theta}; t) \leftarrow \text{softmax}(\boldsymbol{\Psi}(\boldsymbol{\theta}, t, \boldsymbol{y}))_{dim=-1}$
    else
      Input $(\boldsymbol{\theta}, t, \phi)$ to network, receive $\boldsymbol{\Psi}(\boldsymbol{\theta}, t, \phi)$ as output
      $\boldsymbol{p}_O(\cdot|\boldsymbol{\theta}; t) \leftarrow \text{softmax}((1+w)\boldsymbol{\Psi}(\boldsymbol{\theta}, t, \boldsymbol{y}) - w\boldsymbol{\Psi}(\boldsymbol{\theta}, t, \phi))_{dim=-1}$
    end if
    return $\boldsymbol{p}_O(\cdot|\boldsymbol{\theta}; t)$
  end function

Table 9. MAE on QM9 Data Set w/ and w/o Classifier-Free Guidance Generation[a]

	ϵ_{HOMO}/a.u.	ϵ_{LUMO}/a.u.	$\Delta\epsilon$/a.u.
conditional	**0.00724**	**0.00981**	**0.01329**
unconditional	0.01901	0.04076	0.04104

[a]Smaller errors are in bold.

Molecular Scaffold Extension. Here, we show a simple inpaint strategy can extend molecular scaffolds by using ChemBFN. In every sampling step, parameters of input distributions are modified as $\boldsymbol{\theta} \leftarrow \boldsymbol{M} \odot \boldsymbol{e}_x + (1 - \boldsymbol{M}) \odot \boldsymbol{\theta}$ before being inputted into the network, where $\boldsymbol{M}$ is the mask and $\boldsymbol{e}_x$ is the one-hot representation of the scaffold. Figure 6 shows an example of extending scaffold 'Cc1cc(OC5)cc(C6)c1.' by a model trained on MOSES SAFE[48] version, a variation of SMILES. We found that inpainting sampling for 10 to 100 steps was sufficient to generate complex molecules.

Finetuning on Prediction Tasks. In this section, we compare our model with SOTA models,[43,44,49−55] including graph-based and language-based, which could be further classified as smaller scale natural language processing models and large language models on subsets of MoleculeNet benchmark. As shown in Table 10, our method outperformed SOTA LMs on several tasks, especially BBBP. It is notable that ChemBERTa[43] and ChemBERTa-2,[44] which had a similar model size as ours, were pretrained on 77 M molecules but had worse scores on half of the tasks than ours. This indicated that BFN-style generative pretraining is a better strategy than masked language modeling and multitask regression pretraining. A similar observation applied to the $\text{CaR}_{\text{RoBERTa}}$ model that coupled the knowledge of ChatGPT[56] (which is far larger in scale than ours and is believed to have seen more chemical texts) and the distillation capability of the RoBERTa[29] method: our model outperformed $\text{CaR}_{\text{RoBERTa}}$ on 3 out of 4 tasks. However,

https://doi.org/10.1021/acs.jcim.4c01792
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

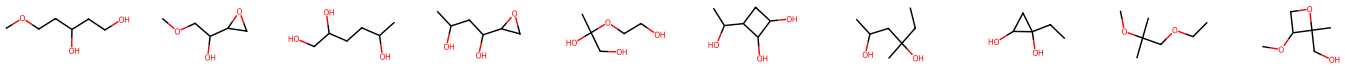


Figure 5. Conditioned samples on QM9. The number of sampling steps was 1 k. Since QM9 exhaustively included stable small molecules made up of CHONF, only 4 conditioned samples and 5 unconditioned samples are novel.

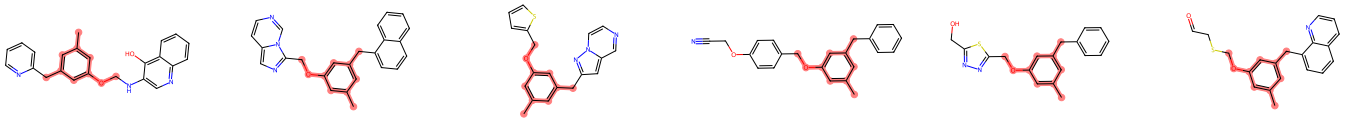


Figure 6. Example of extended molecular scaffold. The scaffold is highlighted in red.

Table 10. Testing Metrics on Sub-Tasks of MoleculeNet Benchmark with Scaffold Splitting Compared with SOTA Models[a]

	Method	ROC-AUC ↑ BBBP	BACE	HIV	RMSE ↓ ESOL	FreeSolv	Lipo
GNNs	Uni-Mol[49]	72.9 ± 0.6	**85.7** ± 0.2	**80.8** ± 0.3	**0.788** ± 0.029	1.480 ± 0.048	**0.603** ± 0.010
	MolKD[50]	74.8 ± 2.3	80.1 ± 0.8	74.9 ± 1.7	–	–	–
	GEM[51]	72.4 ± 0.4	85.6 ± 1.1	80.6 ± 0.9	0.798 ± 0.029	1.877 ± 0.094	0.660 ± 0.008
	Mole-BERT[52]	71.9 ± 1.6	80.8 ± 1.4	78.2 ± 0.8	1.015 ± 0.030	–	0.676 ± 0.017
LLMs	$CaR_{RoBERTa}$[53]	81.99 ± 4.19	80.73 ± 1.42	–	0.96 ± 0.09	–	1.02 ± 0.06
NLPs	ChemBERTa[43]	64.3	–	62.2	–	–	–
	ChemBERTa-2[44]	74.2	79.9	–	–	–	0.744
	AGBT[55]	76.3	–	–	–	–	–
	SMILES Transformer[54]	70.4	70.1	72.9	–	–	–
	ChemBFN (ours)	**95.74** ± 0.70	73.56 ± 1.22	79.37 ± 1.66	0.884 ± 0.003	**1.418** ± 0.067	0.746 ± 0.001
	$\Delta_{GNNs_{best}}$	+28%	-14%	-2%	+12%	-4%	+24%
	$\Delta_{LMs_{best}}$	+17%	-9%	+9%	-8%	–	0%

[a]The metrics of all other models were copied from their original paper. ↑ indicates that the higher is better and ↓ stands for the contrary. The best results are in bold. The best results within the same category (graph-based or language-based) are underlined. Percentages in the last two rows show the performance changes w.r.t the best models, and the color represents whether our model was better (in red) or not (in blue).

Table 11. MAE, RMSE, and Pearson's Correlation Coefficient on the Public ADME Data Set

	HLM	RLM	hPPB	rPPB	MDR1-MDCK ER	solubility
MAE	0.359 ± 0.005	0.428 ± 0.001	0.365 ± 0.006	0.408 ± 0.006	0.337 ± 0.003	0.411 ± 0.008
RMSE	0.474 ± 0.004	0.556 ± 0.005	0.479 ± 0.012	0.549 ± 0.009	0.466 ± 0.007	0.630 ± 0.013
R	0.653 ± 0.002	0.685 ± 0.007	0.771 ± 0.001	0.700 ± 0.003	0.765 ± 0.007	0.588 ± 0.003

Table 12. R^2 Scores on Different Testing Sets of HTE Buchwald−Hartwig and Suzuki−Miyaura Reaction Data Sets[a]

		method			
data set	split	MFF[57]	yield-BERT[41]	yield-BERT-DA[58]	ChemBFN (ours)
Buchwald−Hartwig	rand 70/30	0.927 ± 0.007	0.951 ± 0.005	**0.969** ± 0.004	0.952 ± 0.008
	test 1	**0.85**	0.84 ± 0.010	0.82 ± 0.01	0.844 ± 0.002
	test 2	0.71	0.84 ± 0.03	0.90 ± 0.01	**0.910** ± 0.001
	test 3	0.64	0.75 ± 0.04	0.63 ± 0.05	**0.787** ± 0.034
	test 4	0.18	0.49 ± 0.05	0.43 ± 0.07	**0.633** ± 0.082
	avg. 1−4	0.60	0.73 ± 0.15	0.69 ± 0.19	**0.794** ± 0.118
Suzuki−Miyaura	rand 70/30		**0.81** ± 0.02		0.796 ± 0.011

[a]The scores of all other models were copied from the original paper. The best results are in bold. The score of "rand 70/30" split was the 10-fold average value. Test 1−4 were out-of-sample splits.

when comparing with graph neural networks our model performed averagely worse, especially on regression tasks.

We further benchmarked our model on the public ADME data set[38] (regression task) and kinase inhibitor data set[39] (classification task). The results for the public ADME data set are summarized in Table 11. For the kinase inhibitor data set, the averaged ROC-AUC over 354 assays tested on the random split was (87.93 ± 14.05)% and the averaged ROC-AUC tested on the scaffold split was (79.35 ± 18.71)%.

Reaction Yield Prediction. In order to predict the reaction yield, we first trained the generative model to understand the chemical reaction by learning to predict the products. We developed an in-context style guidance that during training stage only the parameters of product in reaction SMILES were predicted. This was achieved by always masking the input

https://doi.org/10.1021/acs.jcim.4c01792
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

distribution of reactant/reagent and ≫ tokens that were converted to the corresponding one-hot representation, i.e., $\boldsymbol{\theta} \leftarrow \boldsymbol{M}_{rr} \odot \boldsymbol{e}_x + (1 - \boldsymbol{M}_{rr}) \odot \boldsymbol{\theta}$, where $\boldsymbol{M}_{rr}$ is the mask for reactant, reagent, and ≫ token.

The generative model was first pretrained on a USPTO-50k data set then post-trained on Buchwald−Hartwig and Suzuki−Miyaura coupling data sets before the whole prediction model was fine-tuned. The testing scores compared with previous research[41,57,58] were reported in Table 12. It is notable that the Yield-BERT series[41,58] was based on a pretrained RXNFP[59] model, which had been pretrained on over 2 M reactions while our model was pretrained on 50 k reactions. Despite the disadvantage of limited access of pretraining data, the performance of our method was still close to that of a largely pretrained model on random-split sets and significantly better on out-of-sample predictions.

Is Larger Pretrain Data Set Better? We have seen that our model, although was pretrained on 40 M molecules, outperformed the models pretrained on larger data sets on several prediction tasks. Here rises a question: does a larger pretraining data set benefit our method? To answer this, three models were trained on AqSolDB data set, of which one was trained from scratch, one was pretrained on 40 M molecules from ZINC15 database, and the third one was pretrained on 190 M molecules from ZINC15. We summarize the testing results in Table 13.

Table 13. Testing Metrics of Models with Different Pretrain Data Sizes (0, 40, and 190 M) on AqSolDB Data Set

	from scratch	pretrained on 40 M	pretrained on 190 M
MAE	0.978 ± 0.016	0.837 ± 0.005	0.851 ± 0.021
RMSE	1.309 ± 0.014	1.131 ± 0.008	1.145 ± 0.034

Interestingly, the errors did not shrink when the pretraining data grew from 40 to 190 M. However, compared with zero pretraining, an improvement in performance of ≥12.5% can be confirmed.

Training Details. For all generative tasks, the models were trained for 100 epochs with a batch size of 120 molecule/batch. The learning rate (lr) was 5.0×10^{-5} that was linearly increased (warm-up) from 10^{-8} during the first 1000 training steps.

We pretrained one model on 40 M SMILES for 15 epochs with the batch-size of 512 on single A100 GPU and one model on 190 M SMILES for 5 epochs with the effective batch-size of 1024 (2×512) on 2× A100 GPUs. The warm-up strategy and lr were the same as those mentioned above.

During fine-tuning stages, models were trained for 100 epochs on labeled data sets. The batch-size, both for training and validation, was 32 on the MoleculeNet benchmark, AqSolDB data set, public ADME data set, and kinase inhibitor data set; the training batch-size was 16 for reaction yield prediction. lr_{max} was 10^{-4} that was warmed up from 10^{-7} during the first 1000 steps for regression tasks and 100 steps for classification tasks. After the warm-up stage, lr decreased by 0.2 after the validation metrics stopped improving for 20 epochs unless the learning rate had reached 10^{-6}. The dropout rate of prediction MLP head was fine-tuned for each case, and we recommend to try from {0.0, 0.1, 0.5, 0.7}. The validation metrics for regression and classification tasks were MAE and inverted accuracy (i.e., 1 − accuracy), respectively.

We employed AdamW[60] with default hyperparameters implemented in PyTorch[61] as the optimizer for all tasks.

CONCLUSIONS

ChemBFN, a Bayesian flow network framework for chemistry tasks of both generation and prediction, was developed in this work. The new accuracy schedule helped ChemBFN achieve the competitive performance of discrete diffusion models and autoregressive models on generating large molecules. We proposed a BFN-style generative pretraining strategy that surpassed existing language-based transformer models for several classification and regression tasks. We believe this work provides a tool that can accelerate research of both drug designing and filtering and give helpful information for synthesis planning. However, there are still gaps left between graph-based models in prediction tasks, which we shall keep for future research.

ASSOCIATED CONTENT

Data Availability Statement

The code, pretrained models, and instructions necessary to reproduce the results of this study are available for download at https://github.com/Augus1999/bayesian-flow-network-for-chemistry.

AUTHOR INFORMATION

Corresponding Authors

Nianze Tao − *Department of Chemistry, Graduate School of Advanced Science and Engineering, Hiroshima University, Higashi-Hiroshima 739-8524, Japan;* orcid.org/0009-0004-2601-3616; Email: tao-nianze@hiroshima-u.ac.jp

Minori Abe − *Department of Chemistry, Graduate School of Advanced Science and Engineering, Hiroshima University, Higashi-Hiroshima 739-8524, Japan;* orcid.org/0000-0001-7420-9454; Email: minoria@hiroshima-u.ac.jp

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.4c01792

Funding

The authors claim that there is no funding related to this research.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

We express our gratitude to the Research Center for Computational Science (RCCS) in Okazaki, Japan, and its maintenance team for providing computing resources, including A100 GPUs. This work is under project 24-IMS-C043 of RCCS. We also thank Dr. Maho Nakata who kindly lent us his own RTX 3080 GPU and Prof. Kazumasa Okada for discussion.

REFERENCES

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

(2) Amabilino, S.; Pogány, P.; Pickett, S. D.; Green, D. V. Guidelines for recurrent neural network transfer learning-based molecular generation of focused libraries. *J. Chem. Inf. Model.* **2020**, *60*, 5699−5713.

(3) Prykhodko, O.; Johansson, S. V.; Kotsias, P.-C.; Arús-Pous, J.; Bjerrum, E. J.; Engkvist, O.; Chen, H. A de novo molecular generation method using latent vector based generative adversarial network. *J. Cheminf.* **2019**, *11*, 74.

https://doi.org/10.1021/acs.jcim.4c01792
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

(4) Bagal, V.; Aggarwal, R.; Vinod, P.; Priyakumar, U. D. MolGPT: molecular generation using a transformer-decoder model. *J. Chem. Inf. Model.* **2022**, *62*, 2064−2076.
(5) Jin, W.; Barzilay, R.; Jaakkola, T. *Junction Tree Variational Autoencoder for Molecular Graph Generation. International Conference on Machine Learning*, 2018; pp 2323−2332.
(6) Brown, N.; Fiscato, M.; Segler, M. H.; Vaucher, A. C. GuacaMol: benchmarking models for de novo molecular design. *J. Chem. Inf. Model.* **2019**, *59*, 1096−1108.
(7) Loeffler, H. H.; He, J.; Tibo, A.; Janet, J. P.; Voronov, A.; Mervin, L. H.; Engkvist, O. Reinvent 4: Modern AI−driven generative molecule design. *J. Cheminf.* **2024**, *16*, 20.
(8) Guo, J.; Knuth, F.; Margreitter, C.; Janet, J. P.; Papadopoulos, K.; Engkvist, O.; Patronov, A. Link-INVENT: generative linker design with reinforcement learning. *Digital Discovery* **2023**, *2*, 392−408.
(9) Popova, M.; Isayev, O.; Tropsha, A. Deep reinforcement learning for de novo drug design. *Sci. Adv.* **2018**, *4*, No. eaap7885.
(10) Mercado, R.; Rastemo, T.; Lindelöf, E.; Klambauer, G.; Engkvist, O.; Chen, H.; Jannik Bjerrum, E. Graph networks for molecular design. *Mach. learn.: sci. technol.* **2021**, *2*, 025023.
(11) Jensen, J. H. A graph-based genetic algorithm and generative model/Monte Carlo tree search for the exploration of chemical space. *Chem. Sci.* **2019**, *10*, 3567−3572.
(12) Iwata, H.; Nakai, T.; Koyama, T.; Matsumoto, S.; Kojima, R.; Okuno, Y. VGAE-MCTS: A New Molecular Generative Model Combining the Variational Graph Auto-Encoder and Monte Carlo Tree Search. *J. Chem. Inf. Model.* **2023**, *63*, 7392−7400.
(13) Yang, X.; Zhang, J.; Yoshizoe, K.; Terayama, K.; Tsuda, K. ChemTS: an efficient python library for de novo molecular generation. *Sci. Technol. Adv. Mater.* **2017**, *18*, 972−976.
(14) Li, Y.; Zhang, L.; Liu, Z. Multi-objective de novo drug design with conditional graph generative model. *J. Cheminf.* **2018**, *10*, 33.
(15) Atance, S. R.; Diez, J. V.; Engkvist, O.; Olsson, S.; Mercado, R. De novo drug design using reinforcement learning with graph-based deep generative models. *J. Chem. Inf. Model.* **2022**, *62*, 4863−4872.
(16) Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovanov, S.; Tatanov, O.; Belyaev, S.; Kurbanov, R.; Artamonov, A.; Aladinskiy, V.; Veselov, M.; et al. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *arXiv* **2020**, arXiv:1811.12823.
(17) Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 2020; Vol. 33, pp 6840−6851
(18) Vignac, C.; Krawczuk, I.; Siraudin, A.; Wang, B.; Cevher, V.; Frossard, P. DiGress: Discrete Denoising diffusion for graph generation. *arXiv* **2023**, arXiv:2209.14734.
(19) Graves, A.; Srivastava, R. K.; Atkinson, T.; Gomez, F. Bayesian Flow Networks. *arXiv* **2024**, arXiv:2308.07037.
(20) Song, Y.; Gong, J.; Zhou, H.; Zheng, M.; Liu, J.; Ma, W.-Y. *Unified Generative Modeling of 3D Molecules with Bayesian Flow Networks. The Twelfth International Conference on Learning Representations*, 2024.
(21) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput.* **1988**, *28*, 31−36.
(22) Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn.: Sci. Technol.* **2020**, *1*, 045024.
(23) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; Polosukhin, I. Attention is All you Need. *Advances in Neural Information Processing Systems* 2017.
(24) Peebles, W.; Xie, S. Scalable Diffusion Models with Transformers. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023; pp 4195−4205.
(25) Klambauer, G.; Unterthiner, T.; Mayr, A.; Hochreiter, S. Self-Normalizing Neural Networks. *Advances in Neural Information Processing Systems* 2017.
(26) Sun, Y.; Dong, L.; Patra, B.; Ma, S.; Huang, S.; Benhaim, A.; Chaudhary, V.; Song, X.; Wei, F. A Length-Extrapolatable Transformer. *arXiv* **2022**, arXiv:2212.10554.
(27) Su, J.; Ahmed, M.; Lu, Y.; Pan, S.; Bo, W.; Liu, Y. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing* **2024**, *568*, 127063.
(28) Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.
(29) Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
(30) Zhang, S.; Zhang, X.; Bao, H.; Wei, F. *Attention Temperature Matters in Abstractive Summarization Distillation*; ACL 2022, 2022.
(31) Preuer, K.; Renz, P.; Unterthiner, T.; Hochreiter, S.; Klambauer, G. Fréchet ChemNet Distance: A Metric for Generative Models for Molecules in Drug Discovery. *J. Chem. Inf. Model.* **2018**, *58*, 1736−1741.
(32) Degen, J.; Wegscheid-Gerlach, C.; Zaliani, A.; Rarey, M. On the Art of Compiling and Using 'Drug-Like' Chemical Fragment Spaces. *ChemMedChem* **2008**, *3*, 1503−1507.
(33) Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39*, 2887−2893.
(34) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **2014**, *1*, 140022.
(35) Sterling, T.; Irwin, J. J. ZINC 15−ligand discovery for everyone. *J. Chem. Inf. Model.* **2015**, *55*, 2324−2337.
(36) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.* **2018**, *9*, 513−530.
(37) Ramsundar, B.; Eastman, P.; Walters, P.; Pande, V.; Leswing, K.; Wu, Z. *Deep Learning for the Life Sciences*; O'Reilly Media, 2019. https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837.
(38) Fang, C.; Wang, Y.; Grater, R.; Kapadnis, S.; Black, C.; Trapa, P.; Sciabola, S. Prospective validation of machine learning algorithms for absorption, distribution, metabolism, and excretion prediction: An industrial perspective. *J. Chem. Inf. Model.* **2023**, *63*, 3263−3274.
(39) Wu, J.; Chen, Y.; Wu, J.; Zhao, D.; Huang, J.; Lin, M.; Wang, L. Large-scale comparison of machine learning methods for profiling prediction of kinase inhibitors. *J. Cheminf.* **2024**, *16*, 13.
(40) Schneider, N.; Stiefl, N.; Landrum, G. A. What's What: The (Nearly) Definitive Guide to Reaction Role Assignment. *J. Chem. Inf. Model.* **2016**, *56*, 2336−2346.
(41) Schwaller, P.; Vaucher, A. C.; Laino, T.; Reymond, J.-L. Prediction of chemical reaction yields using deep learning. *Mach. learn.: sci. technol.* **2021**, *2*, 015016.
(42) Sorkun, M. C.; Khetan, A.; Er, S. AqSolDB, a curated reference set of aqueous solubility and 2D descriptors for a diverse set of compounds. *Sci. Data* **2019**, *6*, 143.
(43) Chithrananda, S.; Grand, G.; Ramsundar, B. ChemBERTa: Large-Scale Self-Supervised Pretraining for Molecular Property Prediction. *arXiv* **2020**, arXiv:2010.09885.
(44) Ahmad, W.; Simon, E.; Chithrananda, S.; Grand, G.; Ramsundar, B. ChemBERTa-2: Towards Chemical Foundation Models. *arXiv* **2022**, arXiv:2209.01712.
(45) Ho, J.; Salimans, T. Classifier-Free Diffusion Guidance. *arXiv* **2022**, arXiv:2207.12598.
(46) *RDKit: Open-source cheminformatics*. https://www.rdkit.org, (Accessed 19 06, 2024).
(47) Sun, Q.; Zhang, X.; Banerjee, S.; Bao, P.; Barbry, M.; Blunt, N. S.; Bogdanov, N. A.; Booth, G. H.; Chen, J.; Cui, Z.-H.; et al. Recent developments in the PySCF program package. *J. Chem. Phys.* **2020**, *153*, 024109.
(48) Noutahi, E.; Gabellini, C.; Craig, M.; Lim, J. S. C.; Tossou, P. Gotta be SAFE: a new framework for molecular design†. *Digital Discovery* **2024**, *3*, 796−804.

https://doi.org/10.1021/acs.jcim.4c01792
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX

(49) Zhou, G.; Gao, Z.; Ding, Q.; Zheng, H.; Xu, H.; Wei, Z.; Zhang, L.; Ke, G. Uni-Mol: A Universal 3D Molecular Representation Learning Framework. *The Eleventh International Conference on Learning Representations*, 2023.
(50) Zeng, L.; Li, L.; Li, J. M.K. D. Distilling Cross-Modal Knowledge in Chemical Reactions for Molecular Property Prediction. *arXiv* **2023**, arXiv:2305.01912.
(51) Fang, X.; Liu, L.; Lei, J.; He, D.; Zhang, S.; Zhou, J.; Wang, F.; Wu, H.; Wang, H. Geometry-enhanced molecular representation learning for property prediction. *Nat. Mach. Intell.* **2022**, *4*, 127−134.
(52) Xia, J.; Zhao, C.; Hu, B.; Gao, Z.; Tan, C.; Liu, Y.; Li, S.; Li, S. Z. Mole-bert: Rethinking Pre-training Graph Neural Networks for Molecules. *The Eleventh International Conference on Learning Representations*, 2022.
(53) Qian, C.; Tang, H.; Yang, Z.; Liang, H.; Liu, Y. Can Large Language Models Empower Molecular Property Prediction? *arXiv* **2023**, arXiv:2307.07443.
(54) Honda, S.; Shi, S.; Ueda, H. R. SMILES Transformer: Pre-trained Molecular Fingerprint for Low Data Drug Discovery. *arXiv* **2019**, arXiv:1911.04738.
(55) Chen, D.; Gao, K.; Nguyen, D. D.; Chen, X.; Jiang, Y.; Wei, G.-W.; Pan, F. Algebraic graph-assisted bidirectional transformers for molecular property prediction. *Nat. Commun.* **2021**, *12*, 3521.
(56) Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S. others GPT-4 Technical Report. *arXiv* **2024**, arXiv:2303.08774.
(57) Sandfort, F.; Strieth-Kalthoff, F.; Kühnemund, M.; Beecks, C.; Glorius, F. A Structure-Based Platform for Predicting Chemical Reactivity. *Chem.* **2020**, *6*, 1379−1390.
(58) Schwaller, P.; Vaucher, A. C.; Laino, T.; Reymond, J.-L. Data augmentation strategies to improve reaction yield predictions and estimate uncertainty. *ChemRxiv* **2020**.
(59) Schwaller, P.; Probst, D.; Vaucher, A. C.; Nair, V. H.; Kreutter, D.; Laino, T.; Reymond, J.-L. Mapping the space of chemical reactions using attention-based neural networks. *Nat. Mach. Intell.* **2021**, *3*, 144−152.
(60) Loshchilov, I.; Hutter, F. Fixing Weight Decay Regularization in Adam. *arXiv* **2019**, arXiv:1711.05101.
(61) Paszke, A.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems* 2019.

https://doi.org/10.1021/acs.jcim.4c01792
J. Chem. Inf. Model. XXXX, XXX, XXX−XXX