EXTRAMIX: DATA AUGMENTATION FOR REGRESSION USING GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

The primary objective of material science is discovery of novel materials. Because an unseen region can have a high probability of target materials (molecules), high predictive accuracy in out-of-distribution and few-shot regions is essential. However, limited data are available in material science because of high labeling costs. To overcome these difficulties, numerous techniques have been proposed for image and text domains. However, applying these techniques to material data is difficult because the data consists of combinatorial (non-Euclidean) input and continuous labels. In particular, in mixup-based methods, mixed labels are clustered in the middle range of the training set, which renders structured samples invalid. In this study, a novel data augmentation method is proposed for non-Euclidean input with regression tasks. (1) A mixup technique capable of extrapolation is defined to broaden not only the structure but also the label distribution. In contrast to existing mixup-based methods, the proposed method minimizes label imbalance. (2) The proposed method optimizes pseudo-label from the mixup-based approaches using decoder's knowledge of generative models. We proved that the proposed method generates high-quality pseudo data for the ZINC database. Furthermore, the phosphorescent organic light-emitting diode was used to prove that the method is effective in real problems with large-sized and highly complex properties. Moreover, this method can improve property prediction models.

1 INTRODUCTION

Because deep learning technologies have achieved excellent results in various fields, including vision and linguistic tasks, machine learning is increasingly being incorporated in chemistry (Baum et al., 2021). Although numerous deep learning techniques are applied to drugs and material domains (Carracedo-Reboredo et al., 2021; Popova et al., 2018; Vamathevan et al., 2019), applying some specific techniques is difficult because of inadequate data. The high cost of acquiring data points is the primary cause of inadequacy. For example, the synthesis of a phosphorescent organic light-emitting diode (OLED) molecule requires $1-3$ months and costs approximatly $20,000. Therefore, simulation, such as Density Function Theory (DFT) (Sholl & Steckel, 2011), is performed. However, simulation requires considerable time and resources (Patel & Ong, 2019). Moreover, because the mechanisms to express some key physical properties are unknown in many cases, calculating these parameters is difficult and results in a high calculation error. Furthermore, unlike vision and linguistic tasks, manual labeling is impossible in specific target.

When training with a small dataset, the predictive models do not generalize satisfactorily and may exhibit overfitting. To solve the small-data problem, various methods, including adding regularization terms, applying dropout techniques, adding batch normalization layers, and using transfer learning models, have been proposed (Srivastava et al., 2014; Santurkar et al., 2018; Rice et al., 2020). In contrast to aforementioned methods, data augmentation approaches have been used to fundamentally solve the overfitting problem by increasing the number of training samples (Van Dyk & Meng, 2001). With the increase in the number of original datasets, the model should learn more information and exhibit improved accuracy and stability. Image data augmentation techniques, such as rotation, translation, flipping, cropping, color space, and noise injection, have been applied to deep convolutional networks. These augmentation methods have achieved considerable improvement in several major computer vision tasks and improved generalization capability (Shorten & Khoshgoftaar, 2019). Data augmentation methods have been applied to other fields such as natural language processing and speech recognition (Morris et al., 2020; Feng et al., 2021; Ko et al., 2017; Park et al., 2019).

Mixup is a data augmentation method in which virtual examples are constructed by using convex combinations of examples and their labels (Zhang et al., 2018; Verma et al., 2019). This method improves the performance and robustness of neural network architectures in various fields such as image classification (Zhang et al., 2018; Verma et al., 2019; Yun et al., 2019; Lamb et al., 2019; Guo et al., 2019b), text classification (Guo et al., 2019a; Guo, 2020; Jindal et al., 2020), and graph classification (Wang et al., 2021; Guo & Mao, 2021; Han et al., 2022; Park et al., 2022). However, the use of mixup in regression tasks is limited because mixup typically generates examples with labels in the middle range of the train dataset. Remix (Chou et al., 2020) has been proposed to alleviate these issues, but it is not suitable for regression tasks. Furthermore, in material science, the primary interest is to discover new materials with excellent properties. These properties typically lie in the tail part of the label distribution.

In this study, we introduced new data augmentation methods for combinatorial (non-Euclidean) input with regression tasks (Fig. 1). The contributions of this study can be summarized as follows:

1. A mixup technique capable of extrapolation is proposed (*ExtraMix*). This broadens the continuous label's distribution. Unlike the existing mixup methods, *ExtraMix* has fewer label-imbalancing problems.
2. Pseudo-label from the mixup methods can be optimized by decoder's knowledge of conditional variational auto-encoder (cVAE). This method not only improves the accuracy of the pseudo-label, but also adjusts them so that the probability of the joint occurrence of multi-label is maximized.
3. We proposed suitable mixup-based techniques for the molecular domain. We achieved performance improvement even in difficult cases (phosphorescent OLED), large-sized molecules with complex properties.

2 Related Work, Background

2.1 Mixup for molecular data

Unlike data in the Euclidean space, interpolation of molecules is not simple. Because molecules can be expressed as graphs, graph mixup methods can be applied to molecular data. Wang et al. (2021) applied mixup in the latent space, whereas Guo & Mao (2021) and Park et al. (2022) directly applied mixup in the input space. By contrast, Yoshimori (2021) and Han et al. (2022) transformed a graph(s) in a class into a 2D matrix, and subsequently applied mixup. A unique characteristic of molecular data is **validity**. Because not all graphs are valid molecules, creating valid molecules through mixup is essential. Wang et al. (2021) and Yoshimori (2021) developed mixup latent variables that do not correspond to actual graphs in the input space. Guo & Mao (2021), Park et al. (2022), and Han et al. (2022) developed mixup graphs in the input space, but these are rarely valid in terms of molecules. However, using generative models such as variational autoencoder (VAE) (Kingma et al., 2014) to molecular mixup, can generate molecules with high validity (Kang & Cho, 2018; Kwon et al., 2022).

2.2 Deep Learning Approaches for Handling Molecules

To handle molecular structures with deep learning models, representation design is crucial (Raghunathan & Priyakumar, 2022). The graph structure is widely used to represent molecules because mapping the atoms and bonds of molecules to nodes and edges of graph, respectively, is intuitive. To learn graph representation, many graph neural networks (GNNs), such as GCN, GIN, GAT, and DimeNet (Kipf & Welling, 2017; Veličković et al., 2018; Xu et al., 2019; Klicpera et al., 2020), have been proposed. Molecular property prediction can be performed by applying the classification or regression layers on the representation. The graph generative models exhibit complex characteristics of graphs and generate likely graphs. With deep learning expected to directly design the target molecules based on the data, the application of the graph generative models has attracted considerable research attention (You et al., 2018; Liao et al., 2019; Yoo et al., 2020; Cao & Kipf, 2018; Simonovsky & Komodakis, 2018; Jo et al., 2022).

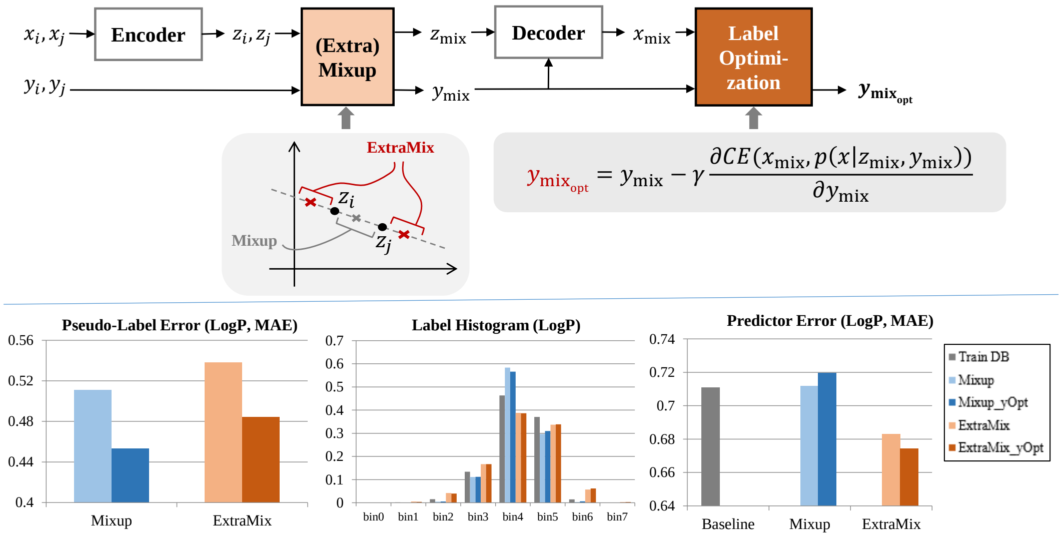

Figure 1: Proposed methods: from the latent points (z_i and z_j) of the two inputs obtained using the encoder of cVAE, a new latent point(z_{mix}) is generated in the outward direction by ExtraMix. The mixed label (y_{mix}) are optimized using the decoder's knowledge (yOPt) with the fixed z_{mix}. The bottom figures reveal performance comparisons. The left image presents a label error of generated samples (measurement: mean absolute error). The middle image is the label histogram of samples generated by each method. The right image denotes predictor's error when each generated dataset was utilized with the train dataset (from OOD-y set of ZINC DB). The baseline was trained by only the train dataset.

Simplified molecular input line entry system (SMILES) (Weininger, 1988), a string-based representation, is widley used to represent molecular structures. Because the method has limited expressive power, graph representation is increasingly being used, particularly when 3D coordinates of atoms are critical. However, it is easier to handle than graphs, and the powerful and fast models designed to deal with the sequence data, such as transformers (Vaswani et al.) can be directly used without any change. Moreover, in the domain in which 3D coordinates, which are time-consuming, are not available, the performance of SMILES-based models is similar to that of graph-based models. Becuase of such advantages, many researchers are still willing to use SMILES rather than graphs.

2.3 Conditional Variational Auto-encoder

The cVAE (Kang & Cho, 2018; Kingma et al., 2014) is designed to generate data given certain conditions such as classes or labels. In the cVAE, input x is assumed to be generated from $p_\theta(x|y, z)$ conditioned on labels y and latent variable z. The prior distribution of z is assumed to be Gaussian distribution, that is, $p(z) = \mathcal{N}(z|0, I)$. We use variational inference to approximate the posterior distribution of z, given x and y, as follows

$$q_\phi(z|x, y) = \mathcal{N}(z|\mu_\phi(x, y), \text{diag}(\sigma_\phi(x, y))). \tag{1}$$

From the perspective of the autoencoder, $q_\phi(z|x, y)$ and $p_\theta(x|y, z)$ are called encoder and decoder, respectively. To handle string data, a structure, such as gated recurrent unit (GRU) (Cho et al., 2014) or long short-term memory (LSTM) (Sak et al., 2014), is typically used. In this study, because the transformer outperformed GRU and LSTM in cVAE, transformer encoders were used for $\mu_\phi(x, y)$ and $\sigma_\phi(x, y)$ and a transformer decoder was used for $p_\theta(x|y, z)$. The objective of the cVAE is to maximize evidence lower bound (ELBO), which is a lower bound of the marginal log-likelihood:

$$\log p_\theta(x|y) \geq \mathbb{E}_{q_\phi(z|x,y)}[\log p_\theta(x|y, z)] - \text{KLD}(q_\phi(\cdot|x, y)||p(\cdot)).$$

We define $\mathcal{L}_{\text{recon}} = -\mathbb{E}_{q_\phi(z|x,y)}[\log p_\theta(x|y,z)]$ because it can be considered a reconstruction loss. $\mathcal{L}_{\text{reg}} = \text{KLD}(q_\phi(\cdot|x,y)||p(\cdot))$ behaves like a regularizing term. In summary, parameters θ and ϕ are optimized to minimize $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{recon}} + \mathcal{L}_{\text{reg}}$. Given target labels y, a new sample x having these labels is generated as follows:

$$z \sim p(z),\ x \sim p_\theta(x|y,z). \tag{2}$$

3 Extrapolatable Mixup and Label Optimization

Generally, mixup-based approaches for combinatorial input are performed on the latent space. However, if labels are continuous values, mixed labels are not used because they are unreliabe. According to our study, cVAE can be used to obtain somewhat accurate mixed labels for regression. Herein, we propose two techniques for more accurate data generation using cVAE.

3.1 ExtraMix

General mixup approaches are as follows:

$$x_{mix} = \lambda x_i + (1-\lambda)x_j, \qquad y_{mix} = \lambda y_i + (1-\lambda)y_j, \tag{3}$$

where $\lambda \in [0,1]$. Two points, (x_i, y_i) and (x_j, y_j) are randomly sampled from train (or real) samples. Generally, labels y are one-hot vector. In these approaches, the features are assumed to have a somewhat locally linear appearance in the input space-like convex set. Thus, the vicinity of a specific x is assumed to have features similar to the x, and this result will be the same from the label perspective. For instance, the regions in the direction of x_j from x_i have higher probability of having the feature x_j than the regions in other directions. With the same assumption, regions in the direction opposite to x_j in x_i will have an increased probability of having the opposite features to x_j. If the form of label is a binary class, the meaning of the mixup could be difficult to interpret in the opposite direction; however, it is clearer in the continuous value. (In Wei et al. (2022), negative label smoothing for class was found to enhance classifiers.) Furthermore, a label distribution expansion is critical for continuous labels than for class labels. The label distribution expansion is similar to the generation of new classes.

To obtain diverse pseudo-data, we propose an extrapolatable mixup approach (*ExtraMix*). In this study, because we focused on a non-Euclidean input, the mixup process was performed in the latent space from the encoder of cVAE ($z_i = \text{Encoder}(x_i), z_j = \text{Encoder}(x_j)$). Two randomly sampled data can be mixed as follows:

$$z_{\text{mix}} = (1+\lambda)z_i - \lambda z_j, \qquad y_{\text{mix}} = (1+\lambda)y_i - \lambda y_j, \tag{4}$$

where $\lambda \in [0,1]$. If the range of λ is changed to $[-1,0]$, the equation 4 is similar to the equation 3. After *ExtraMix* of two points, the decoder of cVAE generates x_{mix} using z_{mix} and y_{mix}: $p_\theta(x_{\text{mix}}|y_{\text{mix}}, z_{\text{mix}})$. Here, y_{mix} is both condition for cVAE and a pseudo-label of x_{mix}. This simple method provides reliable distribution extensions of the label and input. Thus, *ExtraMix* clearly reduces undesirable oscillations when predicting outside the train dataset's label range.

3.2 Label Optimization using Decoder of cVAE (yOpt)

Although cVAE generates some plausible mixed labels, the label error can be large as the number of label types increases, and the labels do not exhibit a linear characteristic in the latent space. In the cVAE, the decoder is a probability distribution of input x for given z and y. Here, y is generally called a condition, but at some perspective it can be viewed as a deterministic part of z. After cVAE was trained, the log-likelihood of whether specific z_{mix} and y_{mix} fit the decoder knowledge can be checked. Instead of using log-likelihood, we can use the cross-entropy (CE) with the generated input x_{mix} as follows

$$\text{CE}(x_{\text{mix}}, x) = -\sum x_{\text{mix}} \log p_\theta(x|y_{\text{mix}}, z_{\text{mix}}) \tag{5}$$

where x_{mix} is one-hot vector from the decoder. Here, y_{mix} is a label that will be used for the train of predictors. Therefore, we are more interested in y_{mix} than in z_{mix} if x_{mix} is valid. To obtain more accurate y_{mix} for x_{mix}, y_{mix} can be optimized by the gradient of equation 5.

$$y_{\text{mix}_{\text{opt}}} = y_{\text{mix}} - \gamma \frac{d\text{CE}(x_{\text{mix}}, x)}{dy_{\text{mix}}}, \tag{6}$$

where γ indicates a learning rate of gradient descent. From this process, an initial mixed label y_{mix} can be close to answer label. This method also reflects a relation information between the label types if the cVAE condition is multi-dimensional ($p_\theta(x|y_{\text{mix}_1}, y_{\text{mix}_2}, ..., y_{\text{mix}_K}, z_{\text{mix}})$, K: the number of conditions). Not only y_{mix} but also z_{mix} can be optimized. For example, if an initial mixed label is to be maintained, z_{mix} can be optimized to minimize the entropy of predictions (=logit) of each token.

4 Mixup Strategy for Molecular Domain

The mixup in molecule has many differences from the image domains. (1) Because the input is one-hot vectors, and a valid structure is required, the latent space of generative models should be used. (2) A generated molecule x_{mix} does not map a unique point z_{mix}. Thus, the method is not one-to-one mapping. For instance, even if $z_{\text{mix}} = 0.9z_i + 0.1z_j$, x_{mix} having the same structure as that of x_i can be generated. This result can be removed by a novelty check. (3) For the same reason as (2), a possibility that duplicate molecules with different y_{mix} are generated during the mixup process exists. To address this issue, the duplicated molecules are removed and an average of all duplicated molecule's mixed labels becomes the mixed label by ensemble ($y_{\text{mix,ens}}$). According to our experimental result, rather than randomly de-duplicating, using $y_{\text{mix,ens}}$ resulted in a slight performance improvement. As such, unlike the image domain, a data augmentation efficiency is somewhat lower because of novelty, uniqueness, and validity.

5 Experiment

5.1 Database and Target Task

For non-Euclidean input with regression tasks, we selected ZINC 250K database (DB) (Kusner et al., 2017). The input of molecule is SMILES (Weininger, 1988) and the vocabulary for SMILES contains 39 symbols. The minimum, median, and maximum lengths of a SMILES string in ZINC250K are 9, 44, and 120, respectively. It has three-type properties, namely molecule weight (molwt), log Partition-coefficient (LogP) and Qualitative Estimate of Drug-likeness (QED), which are continuous values. These properties' average values of ZINC 250k DB are [331.4, 2.45, 0.73]. To analyze the performance in a small-data situation, we justified two-type evaluation sets. (1) **Random**$_{5k}$ and **Random**$_{25k}$. We randomly sampled 5,000 and 25,000 samples for the training set from ZINC 250K DB, and each training set had validation and test sets. (2) **OOD-y**$_{5k}$ and **OOD-y**$_{25k}$: Out-of-distribution (OOD) domain set for property LogP. To justify the OOD set, 2,500 samples with the largest and smallest LogP value were defined as the testing set. Next, the validation set consisted of 2,500 samples with the largest and smallest LogP value of the remaining data. After defining the validation and testing set, 5,000 and 25,000 samples of the remaining data were randomly sampled as training set. The range of LogP of each set does not overlap. The specification of OLED DB is described in Section 5.8.

5.2 Model Structure and Training Phase

As reported in many papers, controlling the KL and reconstruction losses of VAE is critical. To reduce a catastrophic forgetting, we applied controllable VAE (Shao et al., 2020) and empirically found that setting the KL loss to approximately 20 is adequate in the molecular domain. The encoder and decoder of VAE were implemented with the transformer structure, and each part had three layers. The number of the latent variables is 100, and each VAE model was selected based on the ELBO. For a high level of reconstruction power, we used a multi-decoder VAE (Kwon et al., 2022). The results revealed approximately 40% smaller reconstruction loss compared with those of single-decoder VAE. After VAE or conditional VAE models were trained, we generated new samples using encoder and decoder. For a stable mixup of latent variables, the weight of mix λ was sampled from Beta(0.2, 1.2).[1] The learning rate for latent optimization γ was 0.001, and gradient descent was performed in 50 steps.

[1] We experimented to detail the change of several metrics, such as validity, uniqueness, and novelty, according to λ. As λ increased, validity decreased, but uniqueness and novelty increased. The label error of Mixup was the smallest when Beta(0.2,1.2), and a valid ratio was 0.83.

Table 1: Label error of generated molecules: yOpt+ indicates filtered database of deleting 5% samples with the largest reconstruction loss from the decoder. (30,000 samples for each method)

MAE		trainDB: **Random**$_{5k}$			trainDB: **Random**$_{25k}$			trainDB: **OOD-y** $_{25k}$		
	Methods	**molwt**	**LogP**	**QED**	**molwt**	**LogP**	**QED**	**molwt**	**LogP**	**QED**
VAE	Mixup	31.861	1.317	0.100	21.014	1.020	0.090	24.739	1.033	0.124
cVAE	Mixup	11.021	0.563	0.077	7.237	0.467	**0.070**	8.475	0.511	0.087
	Mixup$_{\text{yOpt}}$	9.485	0.513	0.076	6.609	0.446	0.076	7.253	0.453	0.084
	Mixup$_{\text{yOpt+}}$	**9.337**	**0.504**	**0.075**	**6.464**	**0.438**	0.074	**7.035**	**0.449**	**0.083**
	ExtraMix	13.171	0.668	0.100	7.852	0.499	0.079	9.042	0.538	0.098
	ExtraMix$_{\text{yOpt}}$	11.449	0.618	0.102	6.943	0.469	0.083	7.744	0.484	0.096
	ExtraMix$_{\text{yOpt+}}$	**11.228**	**0.592**	**0.098**	**6.765**	**0.463**	**0.082**	**7.394**	**0.471**	**0.093**

The predictor's structure is almost similar to that of BERT (Devlin et al., 2018) and consists of an eight-layer; each regression part of the property has two-layer multi-layer perceptron. Because we consider molecular data, some differences exist. General molecules have many identical atom types. Thus, many duplicates of the same vocab exist. Furthermore, each atom of a molecule is strongly influenced by neighboring atoms. Therefore, we removed a position embedding part and added a relative positional encoding (Shaw et al., 2018) ($k = 5$).

All models were trained using the Adam optimizer (Ruder, 2016) with β_1 = 0.9, β_2 = 0.999, and = 10^{-6}, and we applied a polynomial decay (end learning rate = 0.0). Each model's batch size was 128. Initial learning rates for cVAE and predictors were 10^{-4} and 10^{-3}, respectively.

5.3 Evaluation: Label Accuracy

In data augmentation, (pseudo-) label accuracy is critical. However, checking the label accuracy in real world is difficult when labels are obtained through high cost experiments. By contrast, ZINC DB's labels are simulation values, so the simulated result from RDkit (Landrum) can be considered the answer label. In this subsection, the simulation value is assumed to be the correct answer. Two baselines, namely VAE-Mixup and cVAE-Mixup, were used. VAE-Mixup indicates samples that were generated by VAE with interpolation-based mixup. cVAE-Mixup denotes samples that were generated by conditional VAE with interpolation-based mixup. As seen in Table 1, cVAE-Mixup is considerably better than VAE-Mixup. Among cVAE results, except for QED property, *yOpt* has always revealed better results. *Extramix* results revealed that the inside region of two points on manifold can represent better than the outer region of two points on manifold. Despite these results, the reason why *Extramix* is preferred for regression models will be analyzed in the following subsections.

5.4 Evaluation: Chemical and Label Distributions

The latent variables were transfer to 2-dimensional value (Van der Maaten & Hinton, 2008). The left part of Fig. 2 shows distributions of each database. Samples from Mixup are more close to the train samples than samples from *ExtraMix*. Because *ExtraMix* generates some OOD samples, its mixed label accuracy can be worse than that of Mixup.

The expansion of the label distribution is also very important for the predictive models. The range of pseudo-labels and answer labels can be analyzed through the **OOD-y**$_{5k}$ set. The right part of Fig. 2 reveals the label range of each database. Mixup cannot generate samples outside the label range of the train set. As Mixup is not perfect, some generated samples show OOD-y value according to the answer value by RDkit. From this analysis, we can confirm that *ExtraMix*'s coverage is considerably wider in terms of labels. Thus, the samples generated by *ExtraMix* are more helpful in predicting OOD samples than those obtained from Mixup if they exhibit a similar pseudo-label accuracy.

5.5 Evaluation: Label-balancing

In general, many samples are present in a specific region (=many-shot region). In this case, the predictive accuracy of the many-shot region is excellent. However, it is likely to be low in the region with small number of samples (few-shot region). In a domain, such as new material discovery,

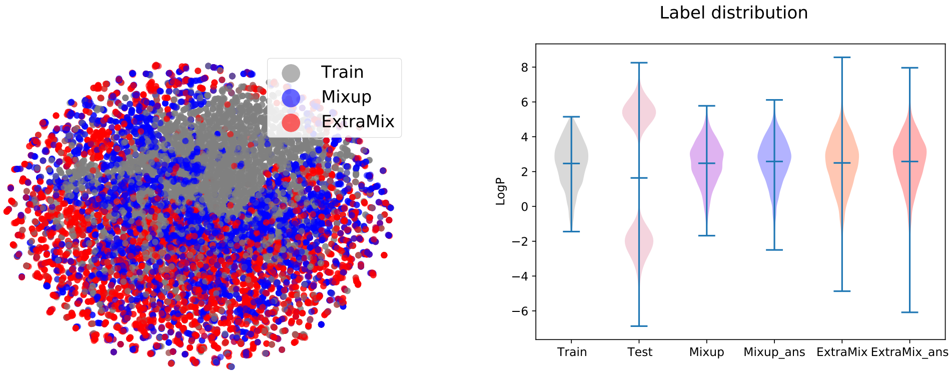

Figure 2: Left: Latent space of training and generated samples(2-dimensional t-SNE), Right: Label range of the generated samples (OOD-y in in Sec. 5.1)

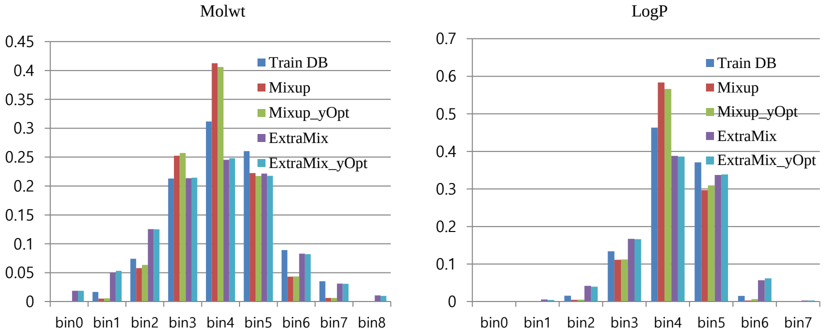

Figure 3: Histograms of labels (Left: molwt, Right: LogP): Mixup has increased the label imbalancing issue. *ExtraMix* generated some samples in a few-shot region.

prediction accuracy in a few-shot region is more crucial than that in many-shot region. To analyze this, we prepared histograms of molwt and LogP, as shown in Fig. 3 (**Random**$_{5k}$). Mixup reveals severe label imbalance than the training set by generating samples in almost many-shot regions. However, *ExtraMix* produced samples in diverse areas. Table 2 confirms this result. *ExtraMix* shows higher standard deviation than that of Mixup. Thus, the labels of *ExtraMix* are more balanced than those of Mixup.

5.6 Evaluation: Examples of Mixup Data

Examples of generated molecules are displayed in Fig. 4. Two molecules were randomly sampled, and five molecules were generated by two real samples. The leftmost molecule has “Br” as in Sample 1, but the rest of the generated molecules do not have “Br”. In the three generated molecules close to Sample 1, “O” exists at the end part, as in Sample 1. By contrast, the three molecules close to Sample 2 have a ring structure with two “N” as in Sample 2. The tendency of the structure change according to the change in the latent value was observed in our experiments.

5.7 Evaluation: Property predictor - ZINC DB

To utilize the generated samples with pseudo-labels, we used a regression loss of the generated DB as pre-training. The baseline was trained by only the train DB, and the others were sequentially trained using each generated DB and the train DB. All predictive models were trained by molwt, LogP and QED, simultaneously (multi-task learning). Table 3 shows each method’s predictive error through mean absolute error (MAE). Each figure in Table 3 is the average value of five

Table 2: Mean and standard deviations of labels.

	molwt		LogP		QED	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**
Train set	331.40	61.48	2.45	1.43	0.73	0.14
Mixup	323.31	47.59	2.40	1.07	0.72	0.11
ExtraMix$_{yOpt}$	310.87	**78.66**	2.42	**1.85**	0.76	**0.18**

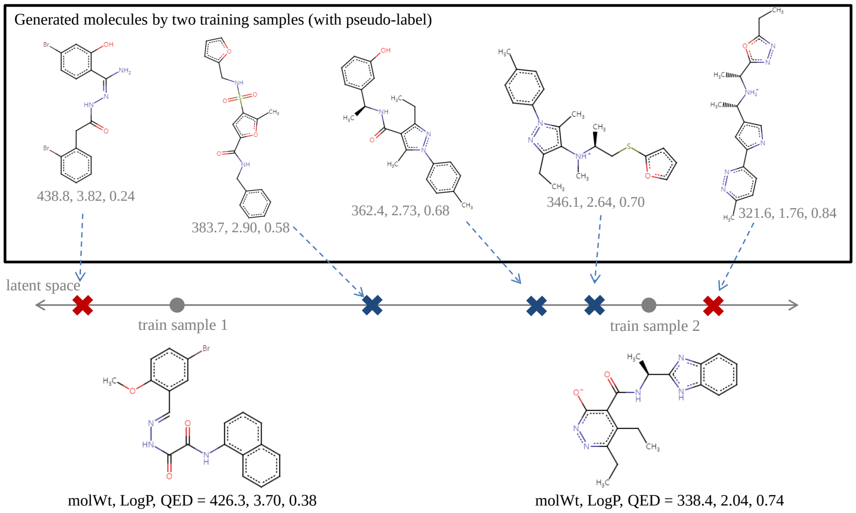
Generated molecules by two training samples (with pseudo-label)
438.8, 3.82, 0.24
383.7, 2.90, 0.58
362.4, 2.73, 0.68
346.1, 2.64, 0.70
321.6, 1.76, 0.84
latent space
train sample 1
train sample 2
molWt, LogP, QED = 426.3, 3.70, 0.38
molWt, LogP, QED = 338.4, 2.04, 0.74

Figure 4: Example molecules generated by Mixup and *ExtraMix*: We selected two samples from **Random**$_{5k}$. Upper molecules were generated from the weighted sum of two latent variable. Red points denote the results of *ExtraMix*.

trials. Because molecules generated by VAE exhibit very low label accuracy (Table 1), they achieved poor performance when used in predictive model training (**Random**$_{5k}$, MAE: molwt 31.156, LogP 1.133, QED 0.103). *ExtraMix*$_{yOpt}$ outperformed the other methods. *[Mixup+ExtraMix]*$_{yOpt}$ is the result when both *Mixup*$_{yOpt}$ and *ExtraMix*$_{yOpt}$ were used to pre-train. Table 3 also reveals a result of OOD-y cases. In case of **OOD-y**$_{25k}$, although Mixup and *Mixup*$_{yOpt}$ achieved worse performance than that of Baseline, *ExtraMix* and *ExtraMix*$_{yOpt}$ outperformed Baseline. Because masked language model (MLM) loss can be applied to SMILES, pre-trained model using MLM loss was prepared by Mixup and *ExtraMix* samples (**Random**$_{25k}$). When *[Mixup+ExtraMix]*$_{yOpt}$ was trained from the pre-trained model using MLM loss, the predictor performed better than before (LogP MAE 0.0594 $\rightarrow$ 0.0423).

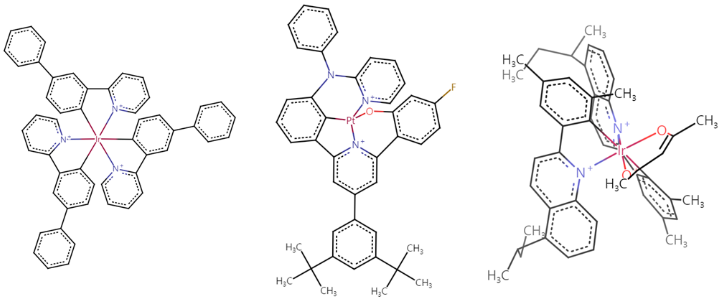

Figure 5: Examples of phosphorescent OLED

Table 3: Predictive error (MAE): Average of five trials presented in each figure. All mixup-based methods used cVAE.

trainDB	**Random$_{5k}$**			**Random$_{25k}$**			**OOD-y$_{5k}$**	**OOD-y$_{25k}$**
Methods	**molwt**	**LogP**	**QED**	**molwt**	**LogP**	**QED**	**LogP**	**LogP**
Baseline	1.5990	0.1519	0.0294	0.9505	0.0718	0.0108	0.9476	0.7110
Mixup	1.5789	0.1420	0.0288	0.9472	0.0611	0.0088	0.9292	0.7119
Mixup$_{\text{yOpt}}$	1.6489	0.1371	0.0279	0.9744	0.0612	0.0090	0.8950	0.7194
ExtraMix	1.6318	0.1370	0.0287	0.9546	0.0624	0.0091	0.9106	0.6830
ExtraMix$_{\text{yOpt}}$	1.5856	0.1347	0.0280	**0.8884**	**0.0589**	**0.0082**	**0.8857**	**0.6742**
[Mixup+ExtraMix]$_{\text{yOpt}}$	**1.5745**	**0.1323**	**0.0270**	0.9038	0.0594	0.0085	0.8955	0.6881

Table 4: Predictive accuracy (R^2) of phosphorescent OLED: Each method used the same pre-trained model which trained masked language model loss.

OLED dopant			R^2		
Methods	**Area**	**Peak**	**Ts**	**Orientation**	**average**
Baseline	0.6267	**0.9118**	0.5917	0.7184	0.7121
Mixup	0.6487	0.9029	0.6151	0.7128	0.7199
[Mixup+ExtraMix]$_{\text{yOpt}}$	**0.6540**	0.9117	**0.6632**	**0.7269**	**0.7389**

5.8 Evaluation: Property predictor - Phosphorescent OLED

Phosphorescent OLED is widely used in the current industry, and it is on averagely about 3-time bigger size than ZINC 250k DB. Because the material has one metallic atom, such as Iridium or Platinum, its physical mechanism is more complex than that of other organic molecules. Furthermore, calculating relevant properties is difficult or impossible with existing simulators. Moreover, because these materials are expensive to synthesize and evaluate, limited data are available. These data are usually private, and therefore, they are not publicly available. Only a few samples, in Fig. 5, can be published. The approximate specifications of our OLED DB are as follows. Approximate 1,000 $\sim$ 2,000 samples, which have over ten types of properties, were obtained from synthesis and evaluation in the laboratory. In addition, approximately 100,000 $\sim$ 200,000 samples with only simulated properties, such as homo, and lumo, were considered. This simulated DB was used for the pre-training stage. The average and maximum number of atom without hydrogen were 64.7 and 108, respectively (the maximum lengths of SMILES is 314). Among several properties, we focused on four properties, namely emission spectra shape (normalized Area), central wavelength (Peak), sublimation temperature (Ts), and Orientation. Area and Peak are related to a efficiency and color. Ts is the temperature required to deposit phosphorescent OLED on the OLED device, and Orientation is a property related to luminous efficiency. To summarize, in this subsection, we compared Baseline, Mixup and *[Mixup+ExtraMix]$_{yOpt}$*. All methods used the same pre-trained model by simulated DB. The measurement is R^2 because inter molecular trends are more crucial than accurate predictions in this domain. Table 4 details the property prediction's accuracy in terms of R^2. On average, Mixup without *yOpt* outperformed Baseline. However the improvement was 0.0078, which was smaller than that of *[Mixup+ExtraMix]$_{yOpt}$*, which was 0.0268. In particular, the proposed method exhibited excellent performance improvement in Ts. From this evaluation, we confirmed that our proposed method works well in large-sized molecules with highly complex properties.

6 Conclusion

In this study, novel data augmentation methods were introduced for non-Euclidean data with regression tasks. These methods are based on the mixup approach. The proposed methods can be used to expand data distribution to outside the training set with a proper continuous label, alleviate the data imbalance, and enable data augmentation in the OOD domain or the few shot regions. Moreover, the proposed methods can optimize an initial mixed label using decoder knowledge of generative models. We can also use the relation information between multi-condition (multi-label) from the decoder. The methods exhibited several meaningful results. In particular, the predictive models outperformed the interpolation-based mixup. Moreover, we confirmed that the methods are effective even in high-complexity domains, such as phosphorescent OLED.

REFERENCES

Zachary J. Baum, Xiang Yu, Philippe Y. Ayala, Yanan Zhao, Steven P. Watkins, and Qiongqiong Zhou. Artificial intelligence in chemistry: Current trends and future directions. *Journal of Chemical Information and Modeling*, 61:3197–3212, 2021.

Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs. In *ICML 2018 workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

Paula Carracedo-Reboredo, Jose Liñares-Blancoab, Nereida Rodríguez-Fernández, Francisco Cedróna, Francisco J. Novoa, Adrian Carballal, Victor Maojo, Alejandro Pazos, and Carlos Fernandez-Lozano. A review on machine learning approaches and trends in drug discovery. *Computational and Structural Biotechnology Journal*, 19:4538–4558, 2021.

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. *The 2022 Conference on Empirical Methods in Natural Language Processing*, 2014.

Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. Remix: rebalanced mixup. *In European Conference on Computer Vision, Springer, Cham*, pp. 95–110, 2020.

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.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*, 2021.

Hongyu Guo. Nonlinear mixup: Out-of-manifold data augmentation for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4044–4051, 2020.

Hongyu Guo and Yongyi Mao. ifmixup: Towards intrusion-free graph mixup for graph classification. *arXiv*, 2110, 2021.

Hongyu Guo, Yongyi Mao, and Richong Zhang. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*, 2019a.

Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3714–3722, 2019b.

Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. G-mixup: Graph data augmentation for graph classification. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 8230–8248. PMLR, 2022.

Amit Jindal, Arijit Ghosh Chowdhury, Aniket Didolkar, Di Jin, Ramit Sawhney, and Rajiv Shah. Augmenting nlp models using latent feature interpolations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6931–6936, 2020.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of sdes. In *International Conference on Machine Learning*, 2022.

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

Diederik 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.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *The International Conference on Learning Representations (ICLR)*, 2017.

Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *The International Conference on Learning Representations (ICLR)*, 2020.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, and Sanjeev Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5220–5224. IEEE, 2017.

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.

Kisoo Kwon, Kuhwan Jung, Junghyen Park, Hwidong Na, and Jinwoo Shin. String-based molecule generation via multi-decoder vae. *arXiv preprint arXiv:2208.10718*, 2022.

Alex Lamb, Vikas Verma, Juho Kannala, and Yoshua Bengio. Interpolated adversarial training: Achieving robust neural networks without sacrificing too much accuracy. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pp. 95–103, 2019.

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

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, William L. Hamilton, David K. Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient graph generation with graph recurrent attention networks. In *In Advances in Neural Information Processing Systems*, 2019.

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. *arXiv preprint arXiv:2005.05909*, 2020.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

Joonhyung Park, Hajin Shim, and Eunho Yang. Graph transplant: Node saliency-guided graph mixup with local structure preservation. In *Proceedings of the First MiniCon Conference*, 2022.

Prachi Patel and Shyue Ping Ong. Artificial intelligence is aiding the search for energy materials. *MRS Bulletin*, 44:162–163, 2019.

Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science Advances*, 4:1–14, 2018.

Shampa Raghunathan and U. Deva Priyakumar. Molecular representation for machine learning applications in chemistry. *International Journal of Quantum Chemistry*, 122:1–21, 2022.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

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.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? *Advances in neural information processing systems*, 31, 2018.

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

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018.

David S. Sholl and Janice A. Steckel. Density functional theory: a practical introduction. *John Wiley Sons*, 2011.

Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. *arXiv preprint arXiv:1802.03480*, 2018.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15:1929–1958, 2014.

Jessica Vamathevan, Dominic Clark, Paul Czodrowski, Ian Dunham, Edgardo Ferran, George Lee, Bin Li, Anant Madabhushi, Parantu Shah, Michaela Spitzer, and Shanrong Zhao. Applications of machine learning in drug discovery and development. *Nature reviews Drug discovery*, pp. 463–477, 2019.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

David A. Van Dyk and Xiao-Li Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *The International Conference on Learning Representations (ICLR)*, 2018.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 6438–6447. PMLR, 2019.

Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. Mixup for node and graph classification. In *Proceedings of the Web Conference 2021*, pp. 3663–3674, 2021.

Jiaheng Wei, Hangyu Liu, Tongliang Liu, Gang Niu, Masashi Sugiyama, and Yang Liu. To smooth or not? when label smoothing meets noisy labels. In *International Conference on Machine Learning*, 2022.

David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1), 1988.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *The International Conference on Learning Representations (ICLR)*, 2019.

Sanghyun Yoo, Young-Seok Kim, Kang Hyun Lee, Kuhwan Jeong, Junhwi Choi, Hoshik Lee, and Young Sang Choi. Graph-aware transformer: Is attention all graphs need? *arXiv preprint arXiv:12006.05213*, 2020.

Atsushi Yoshimori. Prediction of molecular properties using molecular topographic map. *Molecules*, 26(15):4475, 2021.

Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International Conference on Machine Learning*, 2018.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimizatio. In *The International Conference on Learning Representations (ICLR)*, 2018.