A Binded VAE for Inorganic Material Generation

Fouad Oubari
Centre Borelli, UMR9010, ENS Paris Saclay, Michelin
fouad.oubari@ens-paris-saclay.fr

Antoine De Mathelin
Centre Borelli, UMR9010, ENS Paris Saclay, Michelin
antoine.de-mathelin-de-papigny@michelin.com

Rodrigue Décatoire
Michelin
rodrigue.decatoire@michelin.com

Mathilde Mougeot
Centre Borelli, UMR9010, ENS Paris Saclay, ENSIIE
mmougeot@ens-paris-saclay.fr

Abstract

Designing new industrial materials with desired properties can be very expensive and time consuming. The main difficulty is to generate compounds that correspond to realistic materials. Indeed, the description of compounds as vectors of components' proportions is characterized by discrete features and a severe sparsity. Furthermore, traditional generative model validation processes as visual verification, FID and Inception scores are tailored for images and cannot then be used as such in this context. To tackle these issues, we develop an original Binded-VAE model dedicated to the generation of discrete datasets with high sparsity. We validate the model with novel metrics adapted to the problem of compounds generation. We show on a real issue of rubber compound design that the proposed approach outperforms the standard generative models which opens new perspectives for material design optimization.

1 Introduction

The design of new drugs and materials with desired properties is an expensive and time consuming process. It traditionally involved laborious trial-and-error procedures. Evolutionary methods such as genetic algorithms [14, 1] accelerated the process, for instance by modifying molecules structures to obtain molecules with desired properties. However, these methods need hand-crafted rules and constraints to avoid unrealistic generations, which might be hard or even impossible to express mathematically. To bypass this issue, a good alternative are deep generative methods that are able to learn the data space distribution without the need of constraint definition.

In the past few years, deep generative models such as GANs [4] and VAEs [11] have been successfully applied to different tasks ranging from classification to generation [9, 19] or even image translation [20]. However, as observed in our experiments (cf Section 3), these two models do not provide satisfactory solutions for the generation of new materials. The difficulties encountered are the severe sparsity of the data combined with the diversity of components' ratios from one compound to another.

NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications

Moreover, GANs and VAEs require their generations to be differentiable and thus fail to generate discrete compounds [5]. Their inability to directly generate discrete data appears to be an open problem.

Furthermore, unlike when generating images, visual inspection of each generated compound is more complex and requires an expert's validation. Finally, the traditional metrics used to evaluate a generative model are based on the internal features of a deep network trained on ImageNet [12] as the Inception network [18]; therefore, they cannot be used as such for material design.

To overcome all the above-mentioned issues, several works are carried out in the field of new compound generation [15, 6, 13, 3]. For instance when working with GANs for inorganic material generation, the Negative Dice Loss [16] is adopted to enforce the discontinuity of the generations by enforcing the overlap between the real and the generated data [2]. Our work is in line with this research field and proposes the two following main contributions:

- First, to deal with the sparsity and the diversity of components' ratios, we introduce a Binded-VAE model (BVAE) which learns to jointly generate the raw data mask (a binary vector encoding whether or not a component is present in the compound) and the ratio of the components in the compound description (cf Section 2.1).
- Second, we propose three metrics tailored to the problem of compound generation. The first one is an adaptation of the FID. The second one is the KL divergence computed between the distributions of real compound properties and predictions of these properties for generated compounds. The last one is a metric based on the agreement of "expert rules" (cf Section 2.2).

We show, on a real dataset of rubber compound, that the proposed BVAE provides outperforming results compared to the ones obtained with standard VAEs and GANs (cf Section 3). Thus, the combination of our model with an optimization algorithm can be considered as a promising tool for industrials wishing to generate new compounds with desired properties. Furthermore, the developed evaluation metric based on "expert rules" agreement appears to be a suitable validation process for specific issues as industrial ones. We are convinced that this metric can easily be extended to other industrial problems as long as expert rules can be defined on the dataset.

2 Methodology

We consider the problem of compound generation, where the learner has access to a sample of n real compounds with p components, and denote $X \in \mathbb{R}_+^{n \times p}$ the corresponding design matrix. Each compound $x \in \mathbb{R}_+^p$ is represented by a p multidimensional vector denoted by $x = (x_1, ..., x_p)$ where $x_i \in \mathbb{R}_+$ is the ratio of the ith component in the compound such that $\sum_{i=1}^n x_i = 1$. In practice, each vector is very sparse and only a few x_i are not null. The goal of this work is to build a model able to simulate a dataset of new compounds $\widetilde{X}$ distributed in the same way than X. We denote $\widetilde{x} = (\widetilde{x}_1, ..., \widetilde{x}_p)$ the multidimensional vector corresponding to $\widetilde{X}$.

Because of the sparsity in the raw data X, we observe that a standard VAE fails to generate a compound with a few non-zero components and adequate ratios (cf section 3.2). In order to overcome this problem, we decide to split the learning of the binary mask and the ratios into a Binded-VAE model (BVAE) as described in the next section.

2.1 BVAE model

The BVAE is composed of two chained Variational AutoEncoders (cf Figure 1). The first one, denoted "Mask-VAE" (VAE$_M$) receives as input a binary matrix $M \in \{0, 1\}^{n \times p}$ obtained by binarizing the real data X such that $M_{ij} = 1$ if $X_{ij} > 0$ and $M_{ij} = 0$ elsewhere. It generates a reconstructed binary mask $\widetilde{M}$. The second VAE referred as "Ratio-VAE" (VAE$_R$) is a Conditional VAE [17] that receives the concatenation of the real training data X and its corresponding reconstructed mask $\widetilde{M}$ generated by the Mask-VAE. It generates a reconstructed set of ratios $\widetilde{R}$. The final reconstruction of X is then obtained by multiplying the mask $\widetilde{M}$ with the output of the Ratio-VAE such that $\widetilde{X} = \widetilde{M} \times \widetilde{R}$.

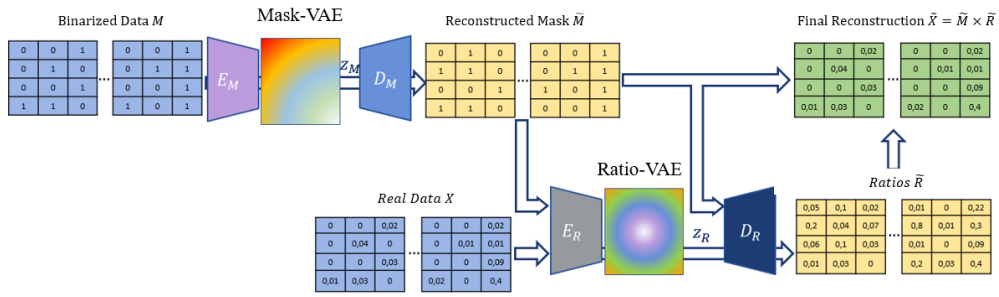

Figure 1: BVAE architecture. The mask $\widetilde{M}$ generated by the Mask-VAE contains the positions of the components in the generated compound. This mask is given as a condition to the Ratio-VAE, which then generates a vector $\widetilde{R}$ containing the ratio of each component.

The final reconstruction $\widetilde{X}$ is differentiable so that the involved gradient can be calculated. For the experiments in section 3 the generated data are truncated to the 4th decimal place.

The loss of the Ratio-VAE is a classical VAE loss:

$$\mathcal{L}_{\text{VAE}_R} = \lambda_R \, \text{BCE}(X, \widetilde{X}) + \text{KL}(\mathcal{N}(\mu_x, \sigma_x), \mathcal{N}(0,\, I)) \tag{1}$$

Where BCE is the Binary Cross-entropy defined as:

$$\text{BCE}(X, \widetilde{X}) = \frac{1}{n} \sum_{i=1}^{n} \widetilde{x}_i \times \log(x_i) + (1 - \widetilde{x}_i) \times \log(1 - x_i) \tag{2}$$

and KL the Kullback-Leibler divergence. μ_x, σ_x are the outputs of the VAE$_R$ encoder.

For the Mask-VAE, the overlap between the generated mask $\widetilde{M}$ and the binarized input M must be maximized. To this purpose, inspired by measures of regions overlap in image segmentation [8], we add to the classical VAE loss, the Negative Dice loss (NegDICE) [16] which has been recently adopted for inverse design of inorganic materials [2]. The NegDICE loss between two vectors $a, b \in [0, 1]^p$ with $a = (a_1, ..., a_p)$ and $b = (b_1, ..., b_p)$ is defined as follows:

$$\text{NegDICE}(a, b) = -2 \, \frac{a \cdot b}{||a||_1 + ||b||_1} = -2 \, \frac{\sum_{i=1}^{p} a_i b_i}{\sum_{i=1}^{p} a_i + \sum_{i=1}^{p} b_i} \tag{3}$$

Therefore, the loss of the Mask-VAE is written as follows:

$$\mathcal{L}_{\text{VAE}_M} = \lambda_M \, \text{BCE}(M, \widetilde{M})) + \text{KL}(\mathcal{N}(\mu_x, \sigma_x), \mathcal{N}(0, I)) + \gamma \, \text{NegDICE}(M, \widetilde{M}) \tag{4}$$

Finally, the total loss of the BVAE is:

$$\mathcal{L}_{\text{BVAE}} = \beta \, \mathcal{L}_{\text{VAE}_M} + \mathcal{L}_{\text{VAE}_R} \tag{5}$$

With λ_R,λ_M,γ and β positive parameters selected with gridsearch.

2.2 Evaluation Metrics

As mentioned in Section 1, standard metrics used to evaluate generative models are tailored for images and thus are not suited for compound generation. We then propose the followings metrics:

- **Custom FID (FID*).** The FID is the most widespread score used to evaluate the quality of the images of a generative model [7] (cf equation 6). The FID compares the first two moments of the real and generated distributions in an embedding given by the penultimate layer of a pretrained deep network. The FID score is then computed as follows:

$$FID(R, G) = \|\mu_R - \mu_G\| + Tr(\Sigma_R + \Sigma_G - 2(\Sigma_R \cdot \Sigma_G)^{\frac{1}{2}}) \tag{6}$$

With μ_R,μ_G and Σ_R,Σ_G the first and the second moments of two distributions R, G. The deep network usually used to compute the FID is the Inception network [18] trained on ImageNet [12]. This network is trained on images and is then not suited for our problem. Consequently, we adapt the computation of the FID to our problem by using a fully-connected neural network trained on the real compound dataset to predict properties of interest associated to each compound. We denote this custom metric FID*.

- **KL divergence.** Following the same idea, we use the aforementioned predictor to predict the properties for the generated data. We then compute the KL divergence between the predictor outputs of the generated data and the real distribution of the material properties.
- **"Expert Rules Agreement" metric (ERA).** As our data represents compounds of real materials, they must verify specific rules given by experts as for instance, bounds on the total proportion of several components in the compound. We then define the ERA metric as the percentage of generated data that match the expert rules.

3 Experiments

In our experiments, we compare different generative models trained on a real rubber compound dataset. In this chapter, we start by presenting the experimental setup, then the dataset used for training the models, and lastly we present and discuss the results obtained.

3.1 Setup

Baselines. To perform our experiments we use different generative models: a GAN, a Vanilla VAE, our BVAE model, and an "Unbinded-VAE" model built with a "Mask-VAE" and a "Ratios-VAE" trained successively and separately.

The models used in our experiments are composed of fully connected layers with the architectures given in Appendix A.1. We use the Adam optimizer [10] for all trainings.

Hyper-parameters selection is performed through gridsearch. The range for each hyper-parameter gridsearch are: $[10^{-1}, 1, 10]$ for λ_R,λ_M,γ and β (cf section 2.1), $[32, 64, 128, 256]$ for the latent dimensions and $[10^{-3}, 10^{-2}, 10^{-1}]$ for the learning rates. The selected hyper-parameters are provided in Appendix A.1 Table 1.

Dataset. To train the models, we use a dataset of **rubber** compound of size $\sim 12k$. The dataset is made of 1D vectors of components ratios, with a size of 99 and with an average sparsity of $\sim 80\%$. We train all the models for 200 epochs, with a batch size of 128. The evaluation metrics presented in Section 2.2 are computed at each epoch. The predictor used to compute the FID* and KL scores is trained using a dataset containing the values of three properties of interest for each compound.

3.2 Results and Discussion

In this section, we discuss the experiments results. We start by comparing the heatmaps of the generations and the real data; then, we compare the models with respect to the custom metrics introduced in section 2.2. As mentioned in section 2.1, generations are truncated to the 4th decimal place.

Data visualization. Contrarily to image generation, inspecting each individual generated rubber compound would require an expert. However, the presence of patterns in the data provides a visual means to verify the generations. We propose to visualize the whole generated datasets as a 2D heatmap where zeros appear in dark and higher values are lightened. As we can observe in Figure 2, this qualitative evaluation clearly shows that the BVAE generates a rubber compound dataset presenting much more realistic patterns than the one produced by the GAN. This qualitative observation is confirmed by quantitative metrics (cf Figure 3). By using this kind of visual verification, one can easily discriminate between models and quickly spot poor generations. However, selecting the best generative model needs finer evaluation metrics as the ones developed in the following. **FID* and KL scores.** Figure 3 presents the evolution of FID* and KL score through epochs for the different generative models. The two scores are computed on 10 000 generations. The KL score is computed by averaging the KLs obtained for the three properties. In addition we plot the test scores computed

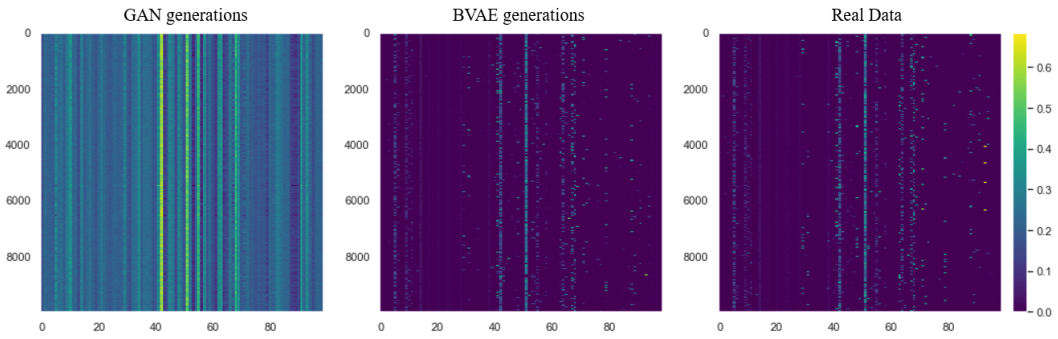

Figure 2: Generations and real data heatmaps. The 10 000 generations and real data vectors displayed are of size 99.

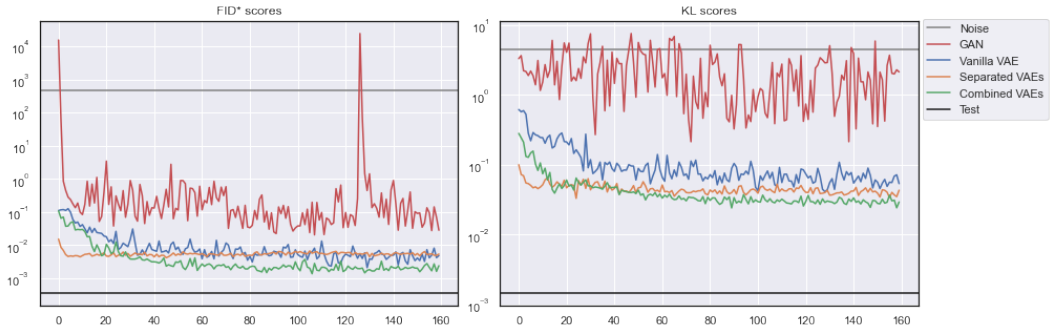

Figure 3: Evolution of FID* and KL scores by epoch. The two metrics are in coherence in terms of models scores. The test score is computed between two subsets of the real data, and the noise score is computed between the real data and a random Gaussian noise.

between two subsets of the real data, and the noise scores computed between the real data and a random Gaussian noise.

We can observe that the GAN performs poorly compared to the other models and is unstable during training, but sill has generally lower scores than the random noise. We can also observe that the two metrics are coherent and give the same score ranking for all models. We notice that the Unbinded-VAE model has better scores in the first epochs before being overrun by the BVAE, which is due to the fact that for the Unbinded-VAE, the mask generation is already pretrained.

The most notable point is the improved performance of the BVAE against the Unbinded-VAE. This fact shows that the learning of the ratios by the Ratio-VAE has a positive influence on the learning of the mask by the Mask-VAE. Indeed, some parts of the mask might be more important than others. For instance if a component's ratio is always high when this component is present in the compound, a good prediction of the mask for this component will be more valuable than a good mask prediction for a component that always appears with small ratios.

"Expert Rules Agreement" score. Looking at the ERA score is even more instructive. We observe that only the Unbinded-VAE and the BVAE models provide generated data that match all the rules that a rubber compound must at least verify to be considered plausible. It means that the data produced by the GAN and the Vanilla VAE would not be considered as valid rubber compound by experts. This highlights one limitation of the FID* and the KL divergence. Indeed, according to these metrics Vanilla VAE was close from the two others but as shown here, only 1% of the data generated can be considered as realistic.

Figure 4: Evolution of the ERA scores by epoch. The BVAE is able to get closer to the real data score, whereas the GAN and Vanilla VAE perform very poorly with a 0% score for the GAN.

4 Conclusion

The poor performances of standard generative models like GANs or VAEs applied to industrial material generations such as rubber compound, makes them unusable and not very popular amongst industrial experts. This work proposes a VAE-based generative model able to overcome the difficulties encountered by standard generative models, and to generate plausible industrial compound-like data. We believe that this work should contribute to the popularization of deep generative methods in the industry, and demonstrates the potential that these methods can bring to this field.

Acknowledgments

Part of this research was funded by Manufacture Française des Pneumatiques **Michelin** and the Industrial Data Analytics and Machine Learning chair of **Centre Borelli** from **ENS Paris Saclay**.

References

[1] N Chakraborti. Genetic algorithms in materials design and processing. *International Materials Reviews*, 49(3-4):246–260, 2004.

[2] Yabo Dan, Yong Zhao, Xiang Li, Shaobo Li, Ming Hu, and Jianjun Hu. Generative adversarial networks (gan) based efficient sampling of chemical composition space for inverse design of inorganic materials. *npj Computational Materials*, 6(1):1–7, 2020.

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

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[5] Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jieping Ye. A review on generative adversarial networks: Algorithms, theory, and applications. *arXiv preprint arXiv:2001.06937*, 2020.

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

[7] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[8] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7. IEEE, 2020.

[9] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[10] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[11] Diederik P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[13] Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *International Conference on Machine Learning*, pages 1945–1954. PMLR, 2017.

[14] Eric-Wubbo Lameijer, Joost N Kok, Thomas Bäck, and Ad P IJzerman. The molecule evoluator. an interactive evolutionary algorithm for the design of drug-like molecules. *Journal of chemical information and modeling*, 46(2):545–552, 2006.

[15] Łukasz Maziarka, Agnieszka Pocha, Jan Kaczmarczyk, Krzysztof Rataj, Tomasz Danel, and Michał Warchoł. Mol-cyclegan: a generative model for molecular optimization. *Journal of Cheminformatics*, 12(1):1–18, 2020.

[16] Reuben R Shamir, Yuval Duchin, Jinyoung Kim, Guillermo Sapiro, and Noam Harel. Continuous dice coefficient: a method for evaluating probabilistic segmentations. *arXiv preprint arXiv:1906.11031*, 2019.

[17] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28:3483–3491, 2015.

[18] C Szegedy, V Vanhoucke, S Ioffe, J Shlens, and ZB Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, volume 2016, pages 2818–2826. IEEE, 2016.

[19] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*, 2020.

[20] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

A Appendices

A.1 Architectures and hyper-parameters

Hyper-parameter	Mask-VAE	Ratio-VAE	Vanilla VAE	GAN
Latent dimension	64	32	128	1024
Learning rate	0.001	0.01	0.001	0.001
λ	1	10	-	-
γ	5 only concerns the BVAE		-	-
β	10 only concerns the BVAE		-	-

Table 1: Selected hyper-parameters. Exceptionally for the GAN's latent dimension we went up to 1024, exceeding the gridsearch rang limit due to its poor generations quality.

Model	Layer size	Activation
Encoder	512	Relu
	1024	Relu
	256	Relu
	2×latent dim	Linear
Decoder	512	Relu
	1024	Relu
	256	Relu
	99	Sigmoid
Loss function	$\mathcal{L}_{\text{VAE}}$ for the Vanilla VAE and $\mathcal{L}_{\text{VAE}_M}$ for Mask-VAE	

Table 2: Vanilla VAE and Mask-VAE architecture

Model	Layer size	Activation
Encoder	256	Relu
	256	Relu
	2×latent dim	Linear
Decoder	256	Relu
	256	Relu
	99	Sigmoid
Loss function	$\mathcal{L}_{\text{VAE}_R}$	

Table 3: Ratio-VAE architecture

Model	Layer size	Activation
Generator	512	LeakyReLU(α=0.2)
	1024	LeakyReLU(α=0.2)
	256	LeakyReLU(α=0.2)
	99	Linear
Discriminator	512	LeakyReLU(α=0.2)
	1024	LeakyReLU(α=0.2)
	256	LeakyReLU(α=0.2)
	1	Sigmoid
Loss function	BCE	

Table 4: GAN architecture