Graph-to-String Variational Autoencoder for Synthetic Polymer Design

Gabriel Vogel
Department of Intelligent Systems
Delft University of Technology
Delft 2629 HZ, The Netherlands
g.vogel@tudelft.nl

Paolo Sortino
Deptartment of Computer Engineering
University of Palermo
Palermo, Italy
paolo.sortino@unipa.it

Jana M. Weber
Department of Intelligent Systems
Delft University of Technology
Delft 2629 HZ, The Netherlands
j.m.weber@tudelft.nl

Abstract

Generative molecular design is becoming an increasingly valuable approach to accelerate materials discovery. Besides comparably small amounts of polymer data, also the complex higher-order structure of synthetic polymers makes generative polymer design highly challenging. We build upon a recent polymer representation that includes stoichiometries and chain architectures of monomer ensembles and develop a novel variational autoencoder (VAE) architecture encoding a graph and decoding a string. Most notably, our model learns a latent space (LS) that enables de-novo generation of copolymer structures including different monomer stoichiometries and chain architectures.

1 Introduction

There is a constant demand for novel or improved polymers with superior properties. AI-assisted in-silico design of molecules is becoming an increasingly valuable approach to accelerate molecular discovery, yet, generative AI for synthetic polymers is still in its infancy [1]. One reason is that the overall amount of available, accessible, and sufficiently detailed polymer data is rather small compared to other domains [2, 3, 4]. Second, unlike small molecules, the definition of an accurate machine-readable data representation for synthetic polymers is difficult, due to their stochastic nature and different structural levels [4, 5, 6] (see Figure 1a). Besides monomer types, the latter includes stoichiometry of monomers, chain architecture, and linking structure, referred to as higher-order structural information in the following.

Past machine learning (ML) works in polymer informatics often relied on fingerprinting approaches (see an overview in [5]) which are not suitable for generative design as fingerprints cannot be mapped back to molecular structures. End-to-end learning making use of string [7, 8, 9] or graph [10, 11, 12] representations has recently shown to be successful for polymer property prediction and can also pave the way for generative design given appropriate representations are available. Walsh et al. [13] propose a data model that captures information from raw materials, over reaction conditions and processing steps, to material characterization and properties. Aldeghi and Coley [14] proposed a representation for polymers consisting of monomer graphs connected via weighted edges reflecting connectivity patterns. Very recently, Schneider et al. [15] introduced a text-based notation

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

(G-BigSMILES) building on BigSMILES [16] which extends the BigSMILES to include molecular weight distributions and connection probabilities of repeat units.

A generative model including higher-order structural information is currently lacking. Encoding higher-order structure of polymers in a continuous latent space allows for property-guided generation including connectivity patterns and further. Previous approaches either operate on the repeat unit level only [8, 17] or work with explicit representations of polymers, i.e. the exact sequence of repeat units [18]. We propose a Graph-to-String VAE for molecular ensembles that takes chain architecture, connection probabilities, and monomer stoichiometries into account. Concurrently, Dollar et al. [19] proposed for the first time a graph-to-string Transformer VAE for small molecular design.

2 Methods

Polymer representations and data set Our work combines a graph [14] and equivalent text-based representation, shown in Figure 1(a). The graph representation connects monomer graphs through weighted edges reflecting the probabilities of connections (chain architecture). The string representation encodes the stoichiometry and connection probabilities as numbers next to the monomer SMILES. Prospectively, including the linking structure besides stoichiometry and chain architecture will be important in future work and can be easily done for the string representation (e.g. using natural language descriptors) but requires more thought for the graph representation. We train our model on the data set from Aldeghi and Coley [14] which is built upon the polymer space of Bai et al. [20]. The data set combines eight A-monomers with 682 B-monomers through stoichiometries of 1:1, 1:3, and 3:1 and three chain architectures (alternating, random, block). This leads to a data set of 42966 copolymers (see Figure 1(c)) including the polymer properties ionization potential and electron affinity (DFT-calculated). The main model, discussed in this work, is trained on an augmented version of the data set, as described in Appendix A.2.

Model architecture and hyperparameters The model is a graph-to-string VAE, encoding the polymer graph to a latent representation $\boldsymbol{z}$ and decoding it to the corresponding polymer string. The information content in each representation is equivalent. As graph encoder we use a re-implementation [21] of the edge-centered message passing graph neural network (wDMPNN) as in [14] and for decoding we develop a transformer. The encoder is composed of one wDMPNN layer (with k=3) and subsequently two parallel wDMPNNs, one for each $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ with a node feature dimension of 300 and global mean pooling. $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are then compressed and reparametrized [22] to the latent bottleneck $\boldsymbol{z}$ with a dimension of 32 which is fed to the transformer decoder consisting of four sequential layers with each four attention heads. Notably, in addition to the encoder-decoder attention, calculated between $\boldsymbol{z}$ and previously generated token embeddings, we concatenated $\boldsymbol{z}$ with each token embedding after the positional encoding (in the following called improved latent space feeding) slightly different to Fang et al. [23] who added it element wise. We vary between two tokenization strategies, a simple SMILES-based tokenization (SM-tokenization) and one with an extended vocabulary based on the Regression Transformer (RT-tokenization)[24]. More details are provided in Appendix A.3. We further train the VAE in a normal setup (with $\beta = 1$) and with cyclical KL annealing [25] (β-schedule). We also test the impact of the weighted cross entropy over normal cross entropy as reconstruction loss function, as frequencies of tokens are uneven.
In addition, we train a supervised model variation using property information on the original labelled data set. We add a feed-forward neural network for both polymer properties (two output neurons) using the latent space as input and add an additional loss term (see Appendix A.4).

Evaluation metrics The evaluation of different model architectures and training procedures is based on the common quantitative metrics: reconstruction, validity, novelty, and uniqueness, as outlined in Table 2 in the Appendix. During inference, i.e. the novel generation of molecules from latent codes, we use beam search with $beam\_size = 5$ to decode the polymer strings. While the reconstruction is evaluated on the test set, the other metrics are evaluated for 16000 polymers which we sampled from Gaussian noise (sampled set). Furthermore, we qualitatively compared LS structures using UMAP dimensionality reduction to evaluate the structure of the encoded hierarchical information [26].

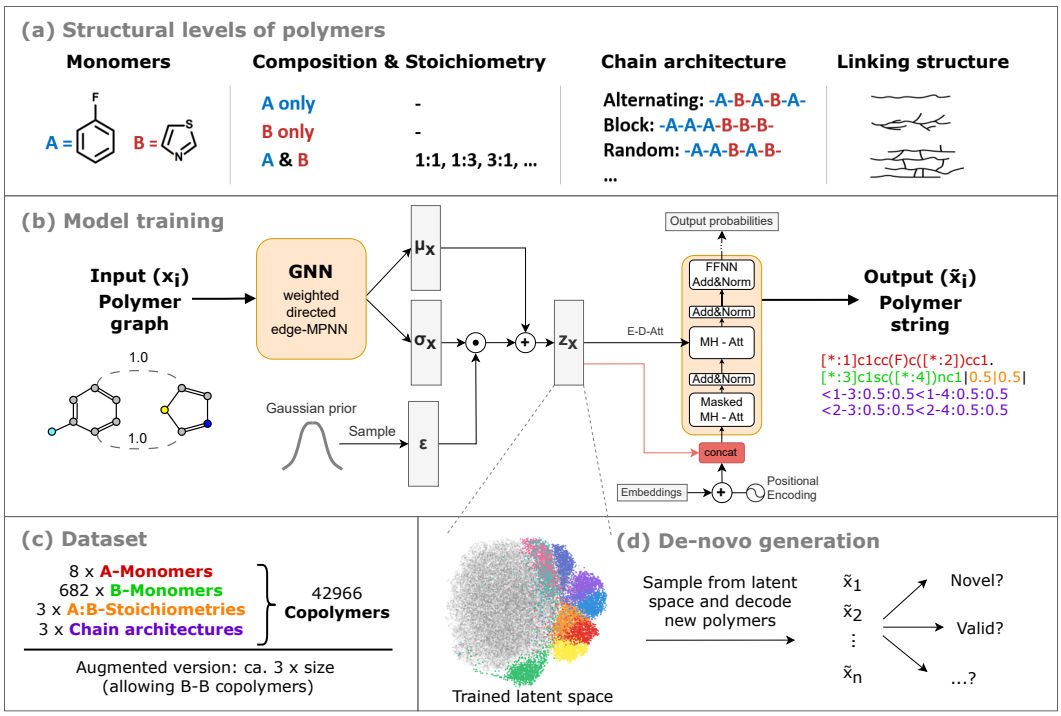

Figure 1: (a) Structural levels of polymers: Monomers, combination of monomers and stoichiometry, chain architecture, chain linking structure (not considered in our work) (b) The graph-2-string VAE during training encodes a graph [14] to a continuous representation z which is fed to a transformer decoder to reconstruct the polymer as string. (c) The model is trained and tested on the data set in [14] and an augmented version. (d) In inference mode, we sample from the trained latent space to decode novel polymers.

3 Results

We perform several experiments on the model architecture and training procedure (see Table 1), using five seeds each and reporting mean and standard deviation. Our best performing model leads to accurately reconstructed polymers (92%) including their higher-order structural information (chain architecture and stoichiometry) as well as to novel (77%) and unique (39%) newly generated polymers. For this model, both reconstructed test set and generated sample set consist of >99.5% chemically valid molecules. The validity of generated molecules is comparable to SOTA models for small molecule design benchmark data sets, such as MOSES [27] and ZINC [28], however, novelty and uniqueness are lower. This is likely due to MOSES and ZINC covering a much broader chemical space, i.e. 1.6M and 250K molecules respectively, in comparison to the data set we use which covers only 690 monomer chemistries.

As shown in Figure 2(a)-(c), the trained latent space is structured mostly according to A-monomer type, while the different stoichiometries and chain architectures can be found throughout the latent space. When sampling 512 molecules around a seed molecule, we observe changes in the monomers, stoichiometry, and chain architecture, as shown in Figure 2(e) and (f). In Appendix B.4, we additionally show an example of stepwise interpolation between two known molecules, revealing how changes in the latent space change the decoded polymers. The main sources of novelty during sampling are novel combinations of monomers, stoichiometry, and chain architecture and novel monomer chemistries. Novel stoichiometries and chain architectures are not yet observed, since the data set only comprises three classes each.

We also trained a property-guided model, based on model configuration 6. In Appendix B.3, we show that the property-guided LS is less organized according to the type of A-monomer but also considers the stoichiometry. Also, we observe property gradients in the LS. However, compared the unsupervised model trained on the augmented data set, the property-guided model shows lower

novelty (31.03%) and uniqueness (13.43%) when sampling, most likely due to the dataset.
The code and data used in this study can be found at https://github.com/GaVogel/G2SVAE/ under MIT license.

Influence of training procedure We find that across different model architectures, the use of the augmented data set, as described in Appendix A.2, increases not only the novelty but also the reconstruction accuracy on the same test set. The increase in novelty is expected since we increased the number of possible A-monomers. We further observe that the use of cyclical KL annealing [25] improves the LS organization (see Appendix B.1) and convergence of the network as well as reconstruction accuracy, novelty, and uniqueness, i.e. compare models 1 and 3 in Table 1.

Table 1: Evaluation metrics (reconstruction, novelty, and uniqueness) reported for six different model configurations. We varied training procedure (β-schedule; use/no use of cyclical KL annealing), latent space feeding ($\boldsymbol{z}$-feeding; with options concat.: our improved latent space feeding and E-D-att.: encoder-decoder attention between latent code $\boldsymbol{z}$ and token embeddings), tokenization strategy (tok; with options SM: SMILES-based tokenization and RT: Regression Transformer-like tokenization), and loss function (loss; with options ce: cross-entropy and wce: weighted cross-entropy).

Model configuration					Metrics in %		
Nr.	β-schedule	$\boldsymbol{z}$-feeding	tok	loss	reconstruction	novelty	uniqueness
1	no	concat.	SM	ce	44.40 ± 16.74	59.17 ± 11.86	14.34 ± 5.03
2	yes	E-D-att.	SM	ce	16.20 ± 12.31	74.40 ± 23.10	18.64 ± 4.67
3	yes	concat.	SM	ce	59.70 ± 26.19	73.53 ± 9.78	29.83 ± 5.49
4	yes	concat.	RT	ce	91.07 ± 3.67	75.97 ± 3.30	39.00 ± 4.31
5	yes	concat.	SM	wce	50.72 ± 11.70	74.51 ± 10.28	28.72 ± 5.58
6	yes	concat.	RT	wce	$\mathbf{92.12 \pm 2.36}$	$\mathbf{76.89 \pm 6.40}$	$\mathbf{39.43 \pm 3.61}$

Influence of model architecture Our architectural change how the latent representation $\boldsymbol{z}$ is fed to the Transformer decoder significantly improves LS organization and the considered evaluation metrics. The model using the default encoder-decoder attention, with the encoder output being a single $\boldsymbol{z}$ vector per sample (configuration 2), performs poorly regarding reconstruction and is also limited in terms of uniqueness, as shown in Table 1. Our modification (see Section 2) improves the performance in terms of reconstruction and uniqueness (see configuration 3). In Appendix B.2, we further show differences in the LS organization and a detailed view on the evaluation metrics including chemical validity in the extended results B.5 in Tables 3 and 4. It becomes apparent that our modification mainly improves the reconstruction accuracy of the monomer chemistries and the uniqueness of newly generated samples, implying that this architectural change could possibly also improve Transformer-based VAEs for small molecules.
Further, we investigated how the tokenization strategy and the reconstruction loss function affect the model performance. Comparing model configurations 3 and 4 as well as 5 and 6, we observe for both loss functions that using the RT tokenization substantially increases the reconstruction accuracy and validity of novel samples, and also improves the robustness as seen in the standard deviation of the metrics in Table 1. The RT tokenization distinguishes between the numerical tokens in the polymer strings, which are part of the stoichiometry and the chain architecture, and the numbers indicating ring structures in the SMILES. In the extended results in Table 3, we see that the RT tokenization improves the reconstruction accuracy of stoichiometry and chain architecture and with it the full reconstruction. Moreover, we find that using the weighted cross-entropy loss slightly improves the evaluation metrics. However, the improvements are small compared to the other architectural changes.
Note, that we see a high standard deviation across different random initializations in the model metrics in Table 1. During the training mode, the reconstruction accuracy does not show this behavior (Model 6: 99.04$\pm$0.10%, Model 5: 98.80$\pm$0.09%). This indicates, that the main source of variation lies in the autoregressive decoding procedure during the inference mode. Extensive hyperparameter tuning, e.g. complexity of the decoder and latent dimensions, and changes in the polymer string representation (less repetitive) could likely decrease the gap in training and inference stability.

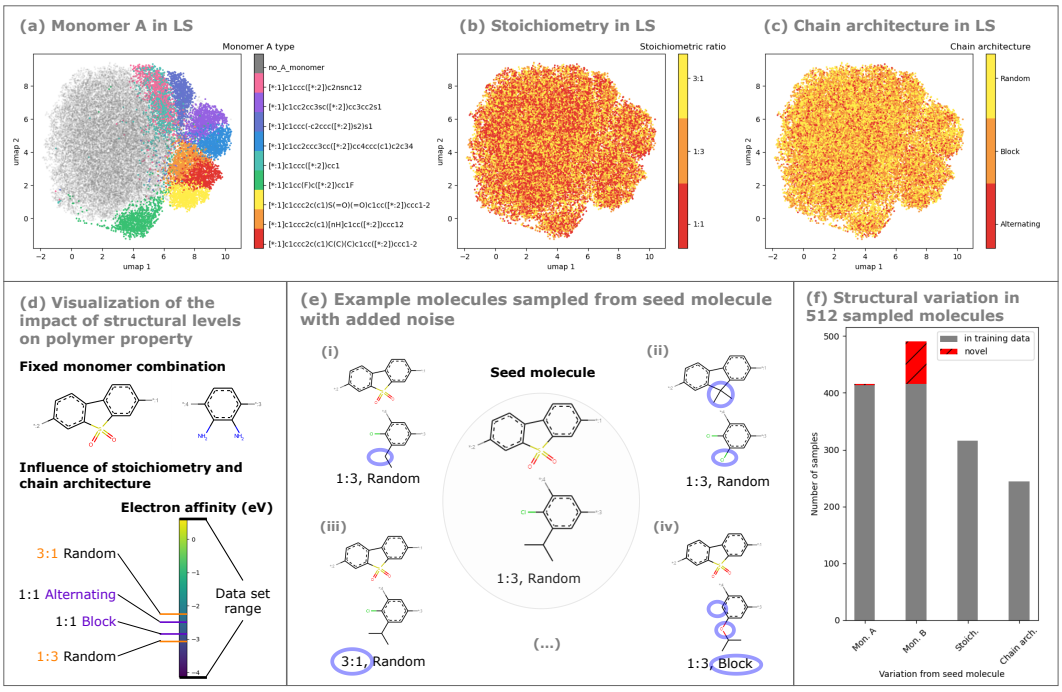

Figure 2: Figures (a)-(c) show a UMAP projection of the trained latent space of model configuration 6 colored by (a) monomer A type, (b) stoichiometry, and (c) chain architecture. Figure (d) shows the influence of structural levels of a polymer on the electron affinity (eV). While the largest influence is attributed to monomer combinations, for a fixed monomer combination stoichiometry and chain architecture have a non-negligible influence on the polymer property. Figure (e) shows example molecules sampled from a seed molecule with changes in (i) B-monomer, (ii) A- and B-Monomer, (iii) only the stoichiometry, and (iv) B-monomer and the chain architecture. Figure (f) shows the frequency of structural changes for 512 sampled molecules, in red indicating novel monomers.

4 Conclusion

We develop a novel Graph-to-string VAE for generative polymer design that takes monomer combinations, stoichiometry, and chain architectures into account. The model is able to generate new polymer structures including higher-order structural information with high chemical validity, high novelty, and acceptable uniqueness.
As discussed, the main limitations of the current approach are the limited diversity of the data set and the need for further hyperparameter tuning. The data set incorporates the information about chain architecture and stoichiometry, yet it covers only a small part of the polymer space and was built up computationally using a combinatorial approach. The limited variability in the data set most likely causes lower uniqueness and novelty scores than in generative models for small molecules. Furthermore, we highlight the need for further hyperparameter tuning as we observe larger variations of the evaluation metrics for different initializations during inference of the model.

Acknowledgments and Disclosure of Funding

We wish to thank Tammo Dukker, Qinghe Gao, and Artur Schweidtmann for their valuable support and discussions regarding the graph neural network architecture. Furthermore, we are grateful for the feedback provided by colleagues in the Delft Bioinformatics Lab and the insightful conversations with Tamim Abdelaal and Max Theisen on VAEs.

References

[1] Kianoosh Sattari, Yunchao Xie, and Jian Lin. Data-driven algorithms for inverse design of polymers. Soft Matter, 17(33):7607–7622, 2021.

[2] Chiho Kim, Anand Chandrasekaran, Tran Doan Huan, Deya Das, and Rampi Ramprasad. Polymer genome: a data-powered polymer informatics platform for property predictions. The Journal of Physical Chemistry C, 122(31):17575–17585, 2018.

[3] Shingo Otsuka, Isao Kuwajima, Junko Hosoya, Yibin Xu, and Masayoshi Yamazaki. Polyinfo: Polymer database for polymeric materials design. In 2011 International Conference on Emerging Intelligent Data and Web Technologies, pages 22–29. IEEE, 2011.

[4] Yoshifumi Amamoto. Data-driven approaches for structure-property relationships in polymer science for prediction and understanding. Polymer Journal, 54(8):957–967, 2022.

[5] Cheng Yan and Guoqiang Li. The rise of machine learning in polymer discovery. Advanced Intelligent Systems, 5(4):2200243, 2023.

[6] Kan Hatakeyama-Sato. Recent advances and challenges in experiment-oriented polymer informatics. Polymer Journal, 55(2):117–131, 2023.

[7] Guang Chen, Lei Tao, and Ying Li. Predicting polymers' glass transition temperature by a chemical language processing model. Polymers, 13(11):1898, 2021.

[8] Ruimin Ma and Tengfei Luo. Pi1m: a benchmark database for polymer informatics. Journal of Chemical Information and Modeling, 60(10):4684–4690, 2020.

[9] Christopher Kuenneth and Rampi Ramprasad. polybert: a chemical language model to enable fully machine-driven ultrafast polymer informatics. Nature Communications, 14(1):4099, 2023.

[10] Evan R Antoniuk, Peggy Li, Bhavya Kailkhura, and Anna M Hiszpanski. Representing polymers as periodic graphs with learned descriptors for accurate polymer property predictions. Journal of Chemical Information and Modeling, 62(22):5435–5445, 2022.

[11] Jaehong Park, Youngseon Shim, Franklin Lee, Aravind Rammohan, Sushmit Goyal, Munbo Shim, Changwook Jeong, and Dae Sin Kim. Prediction and interpretation of polymer properties using the graph convolutional network. ACS Polymers Au, 2(4):213–222, 2022.

[12] Rishi Gurnani, Christopher Kuenneth, Aubrey Toland, and Rampi Ramprasad. Polymer informatics at scale with multitask graph neural networks. Chemistry of Materials, 35(4):1560–1567, 2023.

[13] Dylan J Walsh, Weizhong Zou, Ludwig Schneider, Reid Mello, Michael E Deagen, Joshua Mysona, Tzyy-Shyang Lin, Juan J de Pablo, Klavs F Jensen, Debra J Audus, et al. Community resource for innovation in polymer technology (cript): A scalable polymer material data structure, 2023.

[14] Matteo Aldeghi and Connor W Coley. A graph representation of molecular ensembles for polymer property prediction. Chemical Science, 13(35):10486–10498, 2022.

[15] Ludwig Schneider, Dylan Walsh, Bradley Olsen, and Juan de Pablo. Generative bigsmiles: An extension for polymer informatics, computer simulations & ml/ai. 2023.

[16] Tzyy-Shyang Lin, Connor W Coley, Hidenobu Mochigase, Haley K Beech, Wencong Wang, Zi Wang, Eliot Woods, Stephen L Craig, Jeremiah A Johnson, Julia A Kalow, et al. Bigsmiles: a structurally-based line notation for describing macromolecules. ACS central science, 5(9): 1523–1531, 2019.

[17] Seonghwan Kim, Charles M Schroeder, and Nicholas E Jackson. Open macromolecular genome: Generative design of synthetically accessible polymers. ACS Polymers Au, 2023.

[18] Minghao Guo, Wan Shou, Liane Makatura, Timothy Erps, Michael Foshey, and Wojciech Matusik. Polygrammar: grammar for digital polymer representation and generation. Advanced Science, 9(23):2101864, 2022.

[19] Orion Dollar, Nisarg Joshi, Jim Pfaendtner, and David AC Beck. Efficient 3d molecular design with an e (3) invariant transformer vae. The Journal of Physical Chemistry A, 2023.

[20] Yang Bai, Liam Wilbraham, Benjamin J Slater, Martijn A Zwijnenburg, Reiner Sebastian Sprick, and Andrew I Cooper. Accelerated discovery of organic polymer photocatalysts for hydrogen evolution from water through the integration of experiment and theory. Journal of the American Chemical Society, 141(22):9063–9071, 2019.

[21] Qinghe Gao, Tammo Dukker, Artur M. Schweidtmann, and Jana M. Weber. Self-supervised graph neural networks for polymer property prediction. [Manuscript in preparation], 2023.

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

[23] Le Fang, Tao Zeng, Chaochun Liu, Liefeng Bo, Wen Dong, and Changyou Chen. Transformer-based conditional variational autoencoder for controllable story generation. arXiv preprint arXiv:2101.00828, 2021.

[24] Jannis Born and Matteo Manica. Regression transformer: Concurrent conditional generation and regression by blending numerical and textual tokens. arXiv preprint arXiv:2202.01338, 2022.

[25] Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclical annealing schedule: A simple approach to mitigating kl vanishing. arXiv preprint arXiv:1903.10145, 2019.

[26] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426, 2018.

[27] Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, et al. Molecular sets (moses): a benchmarking platform for molecular generation models. Frontiers in pharmacology, 11:565644, 2020.

[28] John J Irwin and Brian K Shoichet. Zinc- a free database of commercially available compounds for virtual screening. Journal of chemical information and modeling, 45(1):177–182, 2005.

[29] Camille Bilodeau, Wengong Jin, Tommi Jaakkola, Regina Barzilay, and Klavs F Jensen. Generative models for molecular discovery: Recent advances and challenges. Wiley Interdisciplinary Reviews: Computational Molecular Science, 12(5):e1608, 2022.

A Additional details on method

A.1 Motivation for choice of architecture

The choice of encoder and decoder type aims to leverage the strengths of both molecular graph (2D) and molecular string (1D) representations. The recently published polymer graph representation and wDMPNN architecture [14] have shown promising results in structure-to-property learning for polymers. Decoding this type of graph in a generative setup, however, necessitates a more intricate development of a tailored graph decoder. Contrary, using a transformer-based decoder only requires extending the molecular string to polymers, including stoichiometry and chain architecture. Notably, we use concatenation of the latent embeddings with token embeddings to condition the string decoder on the latent embeddings learned by the graph encoder, which especially improves the information flow of the latent embedding during the autoregressive string generation.

A.2 Data augmentation and data split

We randomly split the original data set in train- (80%), validation- (10%), and test set (10%). Since the data set is an exhaustive combination of monomers, stochiometries, and chain architectures, the different data sets are at present highly similar. For future investigations a datasplit that excludes e.g., certain monomer combinations should additionally be investigated.
We increased the amount and diversity of training data by allowing B-B monomer combinations in the training set. To introduce more variety in Monomer A, we combined all B-monomers with 20 randomly selected B-monomers, stoichiometries, and chain architectures. This increases the training data by approximately a factor of three. The UMAP plot in Figure 2a shows that the augmented polymer data and the original data are separated, most likely due to the still higher frequency of the original A monomers.

A.3 Polymer string tokenization

The two tokenization strategies differ in how numbers in the polymer strings are tokenized. The SM-tokenization uses the same vocabulary for the digits in the SMILES string and the digits in the floating point numbers. The RT-tokenization is adopted from the Regression Transformer [24] which distinguishes digits in SMILES and the digits in floating point numbers. We demonstrate the difference using the number 0.125 and the monomer string [*:1]c1cc2sc3cc([*:2])sc3c2s1. In red we highlighted the tokens that are encoded with the same vocabulary but represent a different meaning.

> SM-tokenization: 0.125 → 0 . 1 2 5
>
> SM-tokenization: [*:1]c1cc2sc3cc([*:2])sc3c2s1 → [* :1] c 1 c c 2 s c 3 c c ([* :2]) s c 3 c 2 s 1

Using the RT-tokenization, the digits in floating point numbers are enriched by the information of their decimal position (..,0,-1,-2, -3, ...) which mitigates this issue of using the same vocabulary:

> RT-tokenization: 0.125 → 0_0 . 1_-1 2_-2 5_-3
>
> RT-tokenization: [*:1]c1cc2sc3cc([*:2])sc3c2s1 → [* :1] c 1 c c 2 s c 3 c c ([* :2]) s c 3 c 2 s 1

A.4 Loss function

The loss function for the VAE training consists of two terms, the reconstruction loss and the Kullback-Leibler (KL) divergence which is weighted with the parameter $\beta \geq 0$.

$$\mathcal{L} = \mathcal{L}_{recon} + \beta \cdot \mathcal{L}_{KL} \tag{1}$$

For the reconstruction loss term $\mathcal{L}_{recon}$, we use cross-entropy based on the polymer string representation, which is typically used to evaluate language models. Moreover, we use weighted cross-entropy which accounts for the frequency of tokens of the vocabulary in the training set by assigning higher weights to less frequent tokens. This means that the model penalizes wrong predictions of less frequent tokens more. For the property-guided model variation, we add an additional loss term

$\mathcal{L}_{MSE_{y1,y2}}$ that is the sum of the mean squared errors for the two properties (ionization potential and electron affinity) scaled by α, in our study set to 0.1 to balance the contribution of the loss terms.

$$\mathcal{L} = \mathcal{L}_{recon} + \beta \cdot \mathcal{L}_{KL} + \alpha \cdot \mathcal{L}_{MSE_{y1,y2}} \tag{2}$$

A.5 Definition of evaluation metrics

Table 2 shows the definition of the metrics that are used to evaluate different models.

Table 2: Metrics used for evaluation of different experiments.

Metric	Data	Definition
Reconstruction	test set	Percent of correctly reconstructed molecules
Validity	test set, sampled set	Percent of valid molecules evaluated using RDKit's molecular structure parser [29]
Novelty	sampled set	Percent of molecules not present in training set [29]
Uniqueness	sampled set	Percent of unique molecules [29]

B Additional results

B.1 Cyclical KL annealing vs. normal VAE training

Cyclical KL annealing [25] schedules the parameter β between 0 and 1 during training. We use five cycles over 20 epochs, with 0-ratio=0.2, increase-ratio=0.5, 1-ratio=0.3. From Figure 3, we can draw the conclusion that cyclical KL annealing slightly improves latent space organization, i.e. monomer A types are clustered better in Figure 3b than Figure 3a. Furthermore, in Tables 3 and 4, we see the improvements in reconstruction, novelty and uniqueness metrics. The loss in validity can be mainly attributed to invalid chain architecture and stoichiometry (monomer validity is around 99%) and is mitigated when using the RT tokenization.

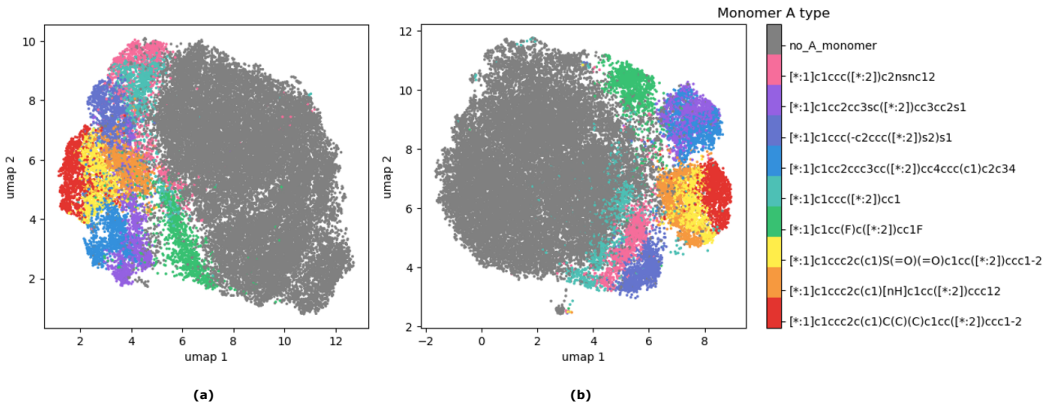

Figure 3: UMAP plots of trained latent space for (a) normal VAE with $\beta = 1$ until convergence and (b) using cyclical KL annealing for 20 epochs and then $\beta = 1$ until convergence.

B.2 Feeding of latent space to Transformer

Figures 4(a),(c), and (e) show that the model using only encoder-decoder attention mostly organizes the latent space according to the chain architecture and stoichiometry and less to the monomer chemistry. Despite chain architecture and stoichiometry influencing polymer properties (see Figure 2) the monomer types are the predominant structural factor for polymer properties and should therefore intuitively be the main focus for LS organization. The model with our improved latent space feeding learns to prioritise the monomer chemistry over the stoichiometry and chain architecture. (see Figure 4(b),(d), and (f)). Moreover, our implementation improves the evaluation metrics as demonstrated in Table 1 and Tables 3 and 4.

B.3 Property-guided model training

Figures 5 show the latent space arrangements for the model with joint property prediction from the latent space, as described in Section 2. Compared to the unsupervised model variation, we observe that the latent space is structured less according to monomer A type but more according to the stoichiometry and property (here: electronic affinity) values. One advantage of this model variation is that the latent space shows gradients for the polymer properties, potentially facilitating property-guided inverse design using gradient-based optimization in the latent space.

B.4 Interpolation between two molecules

We performed stepwise interpolation between two known molecules, i.e. between two latent codes $\boldsymbol{z}_{m1}$ and $\boldsymbol{z}_{m2}$. Here, we created ten latent codes $\boldsymbol{z}_{i,m1\rightarrow m2}$, $i \in \{1, 2, 3, \ldots, 10\}$., by interpolating elementwise in equidistant steps between the latent codes of the start and end molecule. Figure 6 visualizes the interpolation path between two copolymers, with monomer A in the upper part and monomer B in the lower part. We observe that the stoichiometry and chain architecture are changing

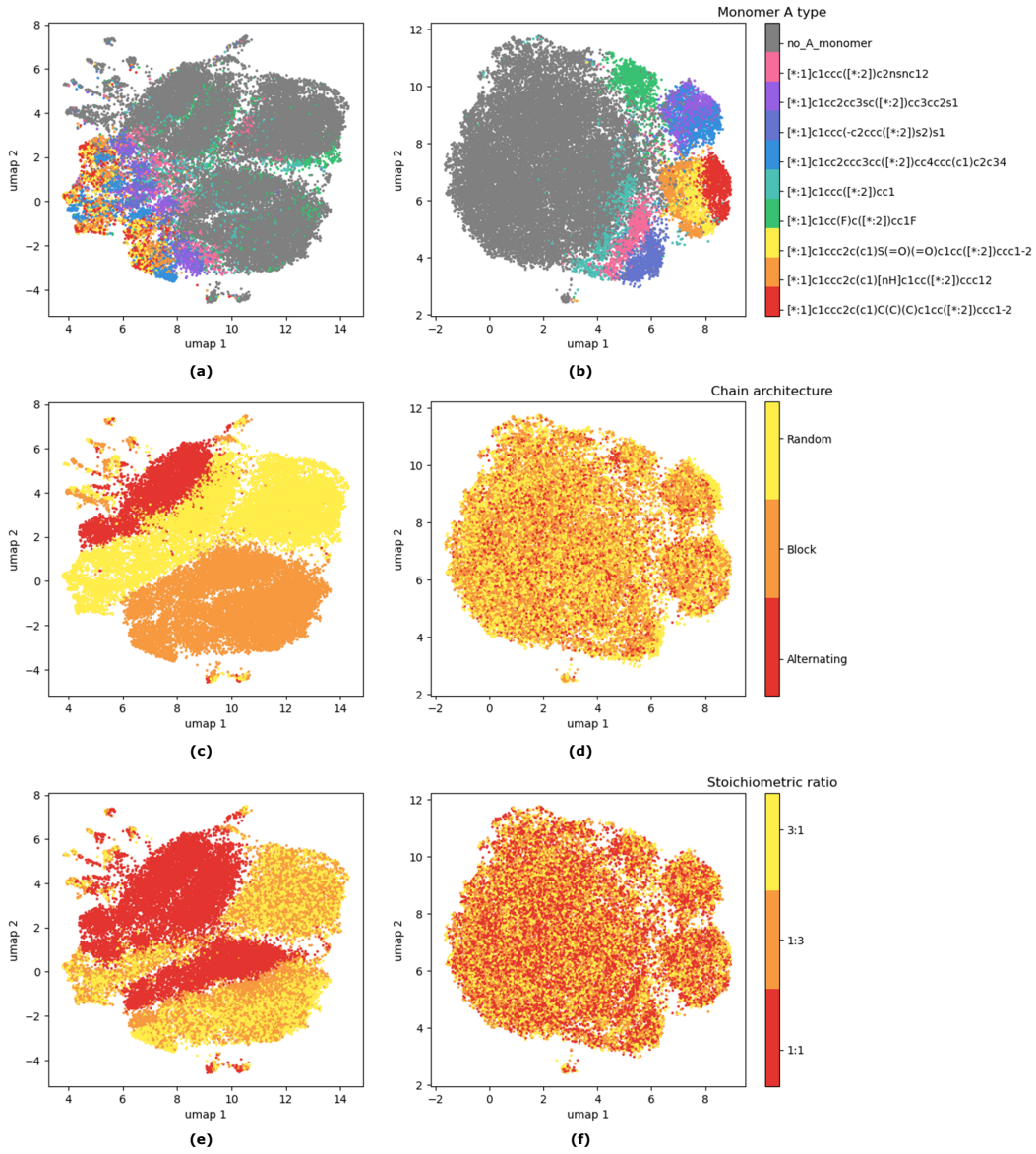

Figure 4: UMAP plots of trained latent space with A-monomer type, chain architecture (random, block, alternating), and stoichiometry (3:1, 1:1, 1:3) as color labels for two different model architectures. (a), (c), (e): Feeding of latent code $\boldsymbol{z}$ is implemented only via encoder-decoder attention (model configuration 2). (b), (d), (f): Improved latent space feeding, additionally concatenating $\boldsymbol{z}$ with the token embeddings after positional encoding (model configuration 3).

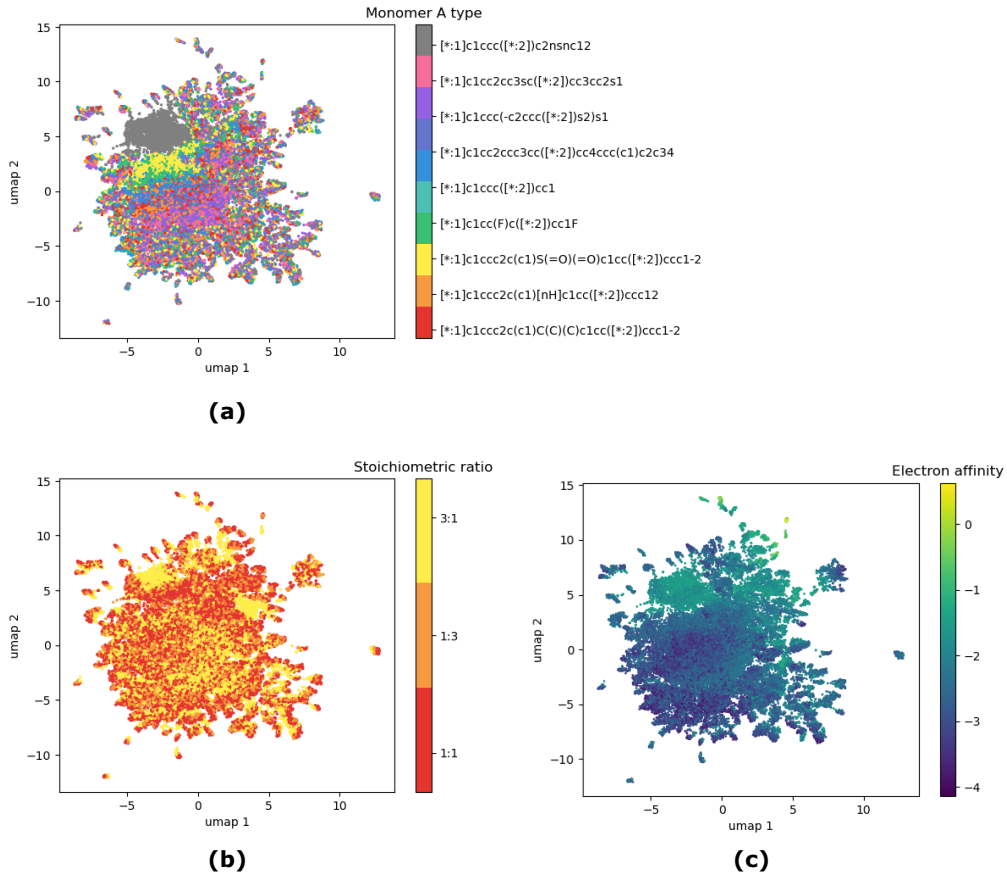

Figure 5: Latent space of property-guided model architecture trained on original data set. UMAP plots are colored by (a) A-monomer type, (b) stoichiometry (3:1, 1:1, 1:3) and (c) Electron affinity.

less frequently. In contrast, the monomer chemistry changes already for smaller variations in the latent vector. None of the intermediates are found in our data set, all are novel and valid copolymers.

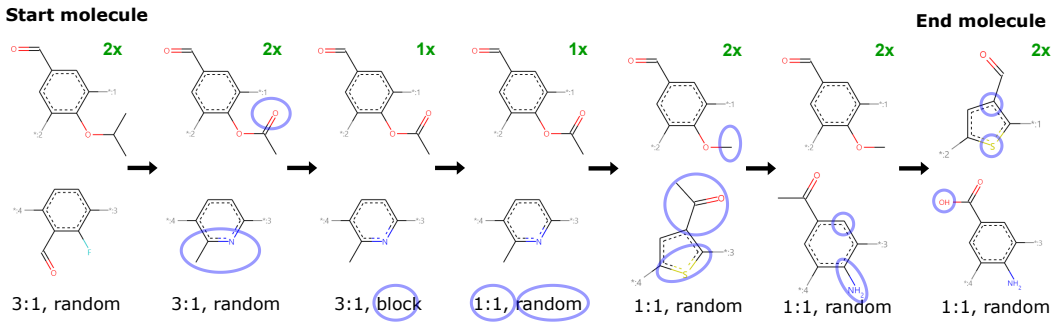

Figure 6: Step-wise interpolation path between two molecules revealing how changes in latent space lead to changes in the decoded polymer (highlighted in blue). The path shows changes in monomer A (upper molecules), monomer B (bottom molecules), stoichiometry and chain architecture. The green number indicates if multiple steps decode to the same polymer.

B.5 Extended results tables

In the extended results, we report more details on the single metrics, e.g. the reconstruction accuracy of the different structural levels (monomers, stoichiometry, chain architecture), and all five runs for each model. Table 3 shows the metrics related to the reconstructed test data and Table 4 shows the metrics related to the novel sampled data as described in Section 2.

Model configuration						Reconstruction in %							
Nr	schedule	seed	z-feeding	tok	loss	Full	Both mons	MonA	MonB	Stoich.	Chain a.	MonA val	Mon B val
1	no	42	concat	SM	ce	60.88	84.20	95.49	84.22	87.20	85.52	100.00	99.98
1	no	43	concat	SM	ce	50.87	82.06	93.88	82.06	87.18	71.05	100.00	100.00
1	no	44	concat	SM	ce	20.60	60.23	87.71	60.74	70.72	47.85	100.00	99.81
1	no	45	concat	SM	ce	28.69	48.01	88.11	48.06	88.39	70.17	100.00	100.00
1	no	46	concat	SM	ce	60.97	69.19	91.90	69.19	88.22	99.95	100.00	99.98
2	yes	42	EDAtt	SM	ce	34.21	36.44	78.75	41.24	94.23	99.93	99.98	99.30
2	yes	43	EDAtt	SM	ce	8.87	14.08	50.87	22.48	87.25	68.40	99.88	94.30
2	yes	44	EDAtt	SM	ce	27.39	36.84	86.87	39.96	77.33	93.90	99.98	98.42
2	yes	45	EDAtt	SM	ce	7.98	43.59	79.52	47.36	66.12	30.53	99.95	98.02
2	yes	46	EDAtt	SM	ce	2.54	14.29	59.09	18.48	79.75	31.42	99.70	96.18
3	yes	42	concat	SM	ce	89.95	95.00	97.56	95.25	94.60	99.95	100.00	100.00
3	yes	43	concat	SM	ce	39.10	89.88	97.30	90.20	86.41	51.18	100.00	99.86
3	yes	44	concat	SM	ce	41.61	85.71	96.14	85.92	85.99	50.55	100.00	99.72
3	yes	45	concat	SM	ce	34.54	89.29	97.39	89.88	85.55	48.71	100.00	99.95
3	yes	46	concat	SM	ce	93.30	93.39	99.09	93.48	99.79	99.95	100.00	100.00
4	yes	42	concat	RT	ce	94.58	94.76	98.56	94.83	99.74	99.93	100.00	99.98
4	yes	43	concat	RT	ce	87.22	87.53	94.04	87.97	99.58	99.91	100.00	99.91
4	yes	44	concat	RT	ce	86.08	89.81	98.28	90.32	95.90	99.93	100.00	99.93
4	yes	45	concat	RT	ce	93.00	93.18	96.51	93.37	99.77	99.95	100.00	100.00
4	yes	46	concat	RT	ce	94.46	94.74	97.44	94.93	99.60	99.93	100.00	99.95
5	yes	42	concat	SM	wce	61.53	94.74	97.56	94.86	85.62	71.19	100.00	99.95
5	yes	43	concat	SM	wce	51.11	81.52	92.95	82.22	88.25	70.42	99.98	99.44
5	yes	44	concat	SM	wce	28.88	66.84	97.74	67.16	85.43	54.11	100.00	99.09
5	yes	45	concat	SM	wce	60.16	89.81	96.58	90.41	94.48	70.91	100.00	99.98
5	yes	46	concat	SM	wce	51.92	90.99	95.83	92.25	85.41	71.72	100.00	99.95
6	yes	42	concat	RT	wce	92.46	93.37	98.42	93.51	99.63	99.19	100.00	100.00
6	yes	43	concat	RT	wce	90.25	92.97	98.28	93.06	96.97	99.88	100.00	99.91
6	yes	44	concat	RT	wce	95.93	96.04	98.81	96.25	99.74	99.95	100.00	99.95
6	yes	45	concat	RT	wce	89.09	89.57	95.23	89.76	99.26	99.95	100.00	99.93
6	yes	46	concat	RT	wce	92.88	93.60	97.16	93.81	99.05	99.88	100.00	99.93

Table 3: Extended results on reconstructed data

Model conf.		Validity (sampled) in %			Novelty (sampled) in %				
Nr	seed	Mon A val.	Mon B val.	val.	nov.	MonA nov.	MonB nov.	nov. (full data set)	Uniqueness in %
1	42	100.00	99.93	99.93	50.55	0.11	1.36	46.88	13.99
1	43	100.00	99.76	99.76	69.39	0.02	1.08	61.69	19.63
1	44	100.00	99.98	90.58	57.69	0.30	2.15	45.72	11.48
1	45	100.00	100.00	92.48	42.94	0.08	1.40	23.73	6.66
1	46	100.00	99.99	99.99	75.26	0.06	1.11	70.75	19.92
2	42	99.99	99.93	99.92	33.99	1.05	7.69	13.58	11.34
2	43	99.99	98.86	95.77	66.34	4.55	14.16	50.21	15.63
2	44	100.00	98.74	98.66	79.38	5.41	27.46	74.20	23.26
2	45	99.86	98.62	18.93	96.01	7.77	12.40	94.49	19.36
2	46	100.00	96.98	39.02	96.27	9.59	17.84	95.94	23.63
3	42	99.99	99.74	99.73	60.64	0.63	7.10	52.53	34.21
3	43	99.99	99.42	71.48	79.73	13.83	8.14	76.56	29.36
3	44	99.84	99.69	87.36	66.61	5.15	15.67	57.64	19.47
3	45	99.99	98.82	31.49	88.49	1.27	7.33	85.23	34.27
3	46	100.00	99.82	99.82	72.18	1.11	11.81	63.76	31.83
4	42	100.00	99.29	99.29	72.11	1.24	7.89	64.99	40.33
4	43	99.98	99.73	99.69	75.61	5.29	8.56	71.46	31.01
4	44	100.00	98.25	98.25	76.78	4.59	10.96	72.04	39.09
4	45	99.99	99.18	99.17	81.73	4.54	10.60	77.87	43.96
4	46	100.00	99.81	99.81	73.64	4.22	6.29	68.53	40.60
5	42	100.00	99.74	99.74	68.94	2.62	10.17	64.21	33.98
5	43	99.99	98.36	70.23	80.34	2.46	24.97	78.72	23.93
5	44	99.97	99.69	31.53	91.49	3.07	9.00	89.24	36.84
5	45	99.92	99.33	99.25	69.46	3.02	10.17	59.88	25.46
5	46	99.88	99.49	99.37	62.31	1.58	16.18	55.51	23.38
6	42	99.91	99.16	99.10	71.66	1.20	12.05	64.61	34.48
6	43	100.00	99.86	99.86	89.14	2.91	7.76	86.44	43.94
6	44	99.98	99.88	99.85	74.05	0.93	7.56	66.81	43.16
6	45	99.96	99.03	98.98	77.07	5.06	17.09	73.09	38.53
6	46	99.99	99.75	99.74	72.54	1.23	10.17	68.05	37.04

Table 4: Extended results on newly generated data