Received: 30 November 2023 | Revised: 5 March 2024 | Accepted: 8 March 2024

DOI: 10.1002/jcc.27349

RESEARCH ARTICLE

Journal of COMPUTATIONAL CHEMISTRY WILEY

Neural graph distance embedding for molecular geometry generation

Johannes T. Margraf

Bavarian Center for Battery Technology (BayBatt), University of Bayreuth, Bayreuth, Germany

Correspondence
Johannes T. Margraf, Bavarian Center for Battery Technology (BayBatt), University of Bayreuth, Bayreuth, Germany.
Email: johannes.margraf@uni-bayreuth.de

Abstract

This article introduces neural graph distance embedding (nGDE), a method for generating 3D molecular geometries. Leveraging a graph neural network trained on the OE62 dataset of molecular geometries, nGDE predicts interatomic distances based on molecular graphs. These distances are then used in multidimensional scaling to produce 3D geometries, subsequently refined with standard bioorganic forcefields. The machine learning-based graph distance introduced herein is found to be an improvement over the conventional shortest path distances used in graph drawing. Comparative analysis with a state-of-the-art distance geometry method demonstrates nGDE's competitive performance, particularly showcasing robustness in handling polycyclic molecules—a challenge for existing methods.

KEYWORDS
conformers, geometry prediction, graph neural network, machine learning

1 | INTRODUCTION

The generation of the three-dimensional geometries of molecules from connectivity information (e.g., based on molecular graphs or SMILES strings) is a ubiquitous task in computational chemistry.[1,2] This has gained additional relevance in recent years as generative machine learning (ML) models for molecular and materials discovery often work based on graphs and strings.[3–5] To evaluate the properties of thus generated molecules with electronic structure calculations, they obviously must be converted to Cartesian coordinates. Relatedly, state-of-the-art atomistic machine learning models (e.g., equivariant/directional neural networks or neighborhood density representations) also rely on the full geometrical information.[6–9] If they are to be used for predicting the properties of candidate molecules proposed by a chemical language model (e.g., in conditional or guided generation settings), realistic cartesian coordinates must be obtained.

Arguably, the most commonly used approach for the generation of molecular geometries is based on the distance geometry (DG) approach.[10–12] Simply put, DG based conformer generators sample conformational space in a random manner. This entails defining matrices of (smoothened) upper and lower bounds for interatomic distances, sampling random distance matrices from these and finally embedding (and refining) 3D geometries based on these distance matrices. In the most commonly used DG approach, the refinement step includes torsion-angle preferences obtained from experimental small-molecule crystallographic data and additional chemical knowledge, for example, regarding the structure of aromatic rings. This method is known as Experimental Torsion and Knowledge Distance Geometry (ETKDG).[11]

ETKDG and related approaches are thus generally well suited for generating conformer ensembles of small molecules. This focus on small drug-like molecules also means that the method is not applicable to general chemical systems with equal accuracy, however. Furthermore DG embedding may fail in some cases (e.g., for certain polycyclic molecules) or require many attempts to find a useful embedding. Consequently, it is challenging to build robust workflows around ETKDG in highly explorative settings.

To overcome this limitation, there has been significant interest in developing ML models for this task. The most prominent example

here is the AlphaFold2 model, which has become the de facto standard for protein geometry generation.[13] Similar to ETKDG, AlphaFold2 leverages domain knowledge and is limited to a well defined (and highly important) use case, namely protein structure generation. Other models have been reported for small molecule geometry generation, for example, trained on the QM9 database.[14] Overall such ML models show promising accuracy, but lack general applicability in terms of elements and molecule sizes.

In this work, a new approach termed neural graph distance embedding (nGDE) is presented, in order to obtain a robust and generally applicable model for molecular geometry generation. This is achieved by combining traditional concepts from force directed graph drawing with a graph neural network (GNN) trained on a highly diverse database of reference geometries.

2 | THEORY

2.1 | Force directed graph drawing

The graph drawing (or layout) problem has a long tradition in graph theory and data visualization.[15] Given a graph with a set of nodes v_i connected by a set of edges e_{ij}, the question is how to embed the nodes in a two-dimensional space, so that the relationships between them (as encoded by the edges) are visually best represented. Clearly there is some ambiguity in what should be considered the best representation, so that a large number of graph drawing methods exist, for example, based on spectral, tree or circular layouts. Similarly, the quality of a layout can be measured in different ways, for example, via edge lengths, the number of edge crossing or the angles between edges.

From a chemical perspective, force-directed layouts are particularly appealing.[16] Here, edges are interpreted as harmonic bonds and the graph drawing problem becomes an energy minimization task. In order to avoid clashes, force directed layouts typically define repulsive forces between unconnected nodes, for example, emulating electrostatic interactions. Such a graph layout is thus quite similar to a molecular forcefield, although a forcefield based solely on harmonic bonds and pairwise repulsion would of course not be useful for molecular structure prediction.

The current work is instead based on the force-directed graph drawing method of Kamada and Kawai.[17] Here, harmonic bonds are placed between all nodes (connected or not), with the corresponding equilibrium bond distances defined via a graph distance measure D_{ij}. The nature of this measure will be discussed in more detail below, but it is often defined such that connected nodes have a distance of unity, whereas unconnected nodes have larger distances. Additionally, the spring constant w_{ij} for each bond is chosen to be inversely proportional to D_{ij} (i.e., as $\frac{1}{D_{ij}^2}$), so that close and connected nodes have a stronger influence in the energy function. Note that for dense three dimensional systems, the number of pairs grows cubically with the distance. Here higher exponents in the damping would likely be warranted.

Overall, the Kamada-Kawai layout minimizes the function:

$$\mathcal{L}(\mathbf{r}) = \sum_{i,j} w_{ij} \left(||r_i - r_j|| - D_{ij} \right)^2, \quad (1)$$

where $\mathbf{r}$ is the matrix of node positions in the layout and r_i is the position vector of node i.

Interestingly, this form of the graph layout problem is equivalent to multidimensional scaling (MDS), a popular dimensionality reduction method.[18] As such, MDS is by construction agnostic towards the number of target dimensions (e.g., two for a graph layout or three for a molecule). Furthermore, Equation (1) is typically minimized via a stress majorization algorithm in this context, which is monotonically convergent and therefore highly robust.

In the following a molecular geometry prediction method that uses a graph distance matrix and MDS to obtain 3D coordinates will be referred to as a graph distance embedding (GDE). This leaves the question, whether a useful graph distance can be found, which yields realistic molecular geometries upon minimization of Equation (1).

2.2 | Graph distances

In graph theory, a number of measures exist to quantify the distance between two nodes in a graph. The simplest and most commonly used is the length of the shortest path connecting the nodes ($D_{ij}^{(s)}$), where the shortest path can be found with established methods such as Dijkstra's algorithm.[19] Assigning a length of unity to each edge, the graph distance then measures the minimum number of hops required to get from one node to another. Alternative definitions of graph distance also exist, such as the resistance distance,[20] and the personalized PageRank,[9] which both take into account the number of paths connecting the nodes, as well as their length.

In the context of molecular geometry prediction, it makes sense to define the edge lengths in terms of the sum of covalent radii of the respective elements. Based on this definition, $D_{ij}^{(s)}$ turns out to be a robust upper bound for the true distance between two atoms in a molecule. Specifically, $D_{ij}^{(s)}$ corresponds to the distance between two atoms in a completely linear geometry of all connecting atoms (e.g., in a cumulene). The presence of non-linear bonding geometries will lead to interatomic distances smaller than D_{ij}.

To illustrate this, Figure 1 shows a density plot of $D_{ij}^{(s)}$ versus the true interatomic distance for the OE62 dataset of molecular geometries.[21] The OE62 set contains ca. 62k DFT optimized molecular geometries taken from the Cambridge Crystal Structure Database, covering a wide range of molecular sizes and chemical elements (up to 92 non-hydrogen atoms and 16 different elements). It thus represents an important and challenging benchmark, both in terms of the quality of the geometries and the diversity of the molecules it covers. As can be seen, the true interatomic distances are consistently below the parity line in Figure 1. Additionally, the sum of vdW-Radii (illustrated for H-H) can be used as a lower bound for disconnected atoms.

This upper bound property of $D_{ij}^{(s)}$ is useful, but it leads to a bias towards extended molecular geometries in a GDE scheme (see below). While scaling the shortest path distance may help in this regard, one should not expect a single scaling factor to be appropriate for all atom pairs. For example, two oppositely charged functional groups will tend to favor smaller interatomic distances, other things being equal. Similarly, the bonding topology around a given atom (e.g., whether it is part

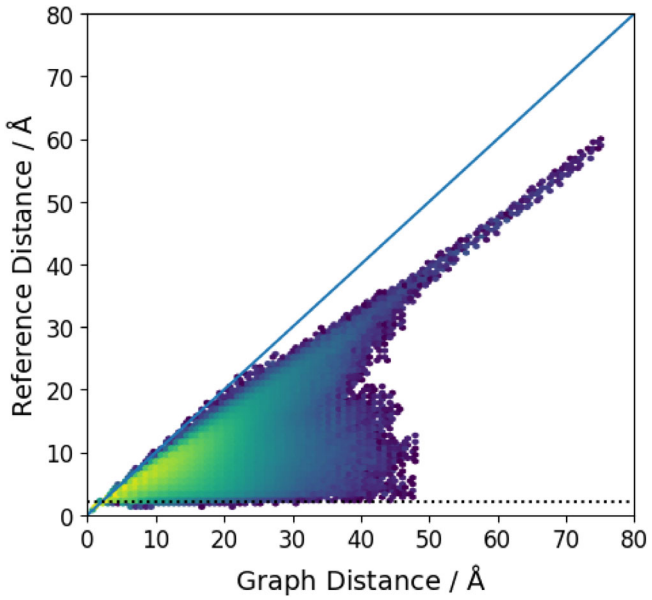

FIGURE 1 Density plot of shortest path graph distances versus reference interatomic distances from the OE62 dataset. Colors indicate logarithmic density from low (purple) to high (yellow). The parity line is shown in blue, indicating that the graph distance is an upper bound on the interatomic distance. The dotted line indicates twice the van-der-Waals radius of hydrogen, as a lower distance bound for non-covalently bound atoms.

of an aromatic ring or in an alkyl chain) contains information about how flexible its environment is. In the next section a learnable graph distance function that takes these factors into account is therefore defined.

2.3 | Neural graph distance embedding

The proposed nGDE method uses a GNN to encode the chemical environment of each atom in a given molecule into a vector.[22] In order to ensure transferability to diverse chemistries, a very basic graph representation of molecules is used. Specifically, each covalent bond is represented by an edge between two atoms and each atom type is represented by its atomic number. Additional information like spin states, partial charges, bond orders and so forth are not required.

In a first step, a learnable embedding is used to assign an H-dimensional vector to each atomic number Z (with H being the number of hidden nodes in each layer of the network). This vector forms the initial node representation $\mathbf{x}_i^{(0)}$ of each atom.

This representation is then updated via T message passing steps, using a simple graph convolutional operator[23]:

$$\mathbf{x}_i^{(t+1)} = \mathrm{ReLU}\left(\mathbf{W}_1\mathbf{x}_i^{(t)} + \mathbf{W}_2 \sum_{j\in\mathcal{N}(i)} \mathbf{x}_j^{(t)}\right). \tag{2}$$

Here, ReLU is the rectified linear unit activation function, $\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable weight matrices, $t < T$ indicates the current message passing step and $\mathcal{N}(i)$ denotes the set of nodes connected to i.

After message passing, the final node representation is obtained by feeding $\mathbf{x}_i^{(T)}$ through a small multi-layer perceptron (MLP) ϕ with one hidden layer:

$$\mathbf{x}_i^{(\mathrm{f})} = \phi(\mathbf{x}_i^{(T)}). \tag{3}$$

At this point a flexible, trainable representation of the chemical environment of each atom in a molecule is defined. In order to obtain the graph distance between two atoms, a second MLP γ is used, which predicts the distance based on the node representations of i and j, as well as the shortest path distance $D_{ij}^{(\mathrm{s})}$:

$$D_{ij}^{(\mathrm{nGDE})} = \frac{1}{2}\left[\gamma\left(\mathbf{x}_i^{(\mathrm{f})}, \mathbf{x}_j^{(\mathrm{f})}, D_{ij}^{(\mathrm{s})}\right) + \gamma\left(\mathbf{x}_j^{(\mathrm{f})}, \mathbf{x}_i^{(\mathrm{f})}, D_{ij}^{(\mathrm{s})}\right)\right]. \tag{4}$$

Note that by averaging over the outputs of γ with reversed arguments, permutational invariance between atom pairs ij and ji is ensured. The corresponding distance matrix $\mathbf{D}^{(\mathrm{nGDE})}$ can then directly be used to set up an MDS problem according to Equation (1).

A common problem of the current method and other geometry generation approaches (such as DG or generative models) is that small but important structural details (particular regarding angles) are often incorrect, while the broad structure is captured well. In the case of organic molecules, this issue can easily be addressed by using classical forcefields for structural refinement. In the following, all embedded geometries are therefore relaxed in a two step procedure, first using the Merck molecular force field (MMFF) and subsequently using the universal GFN forcefield of Spicher and Grimme.[24,25] For consistency, the same refinement is also used for ETKDG in all comparisons.

With this, the nGDE procedure can now be fully described: For a given molecular graph (including hydrogen atoms), first a distance matrix $\mathbf{D}^{(\mathrm{nGDE})}$ is predicted. Then cartesian atomic positions are randomly initialized and optimized via MDS. Finally, the geometries are refined using classical force fields. For comparison, GDE using the shortest path distance is referred to as sGDE.

3 | METHODS

The nGDE model discussed in the following was trained on 10,000 molecules randomly drawn from the OE62 database. As a loss function, the mean squared difference between predicted and observed interatomic distances was used. Model dimensions were defined as $H = 128$ nodes per hidden layer and $T = 4$ message passing iterations. Weights were optimized using the ADAM minimizer with a learning rate of 1×10^{-3}, a weight decay constant of 5×10^{-4}, and a batch size of 32.

The `numpy/pyTorch/PyG` implementation and the trained nGDE model are available at https://gitlab.com/jmargraf/ngde/. `rdKit`[26] and the Atomic Simulation Environment (`ase`)[27] were used for pre- and post-processing. ETKDG conformers were generated using `rdKit`, with the latest ETKDG variant.[12] No stereochemical information was used for conformer generation.

4 | RESULTS AND DISCUSSION

The nGDE model predicts interatomic distances for an unseen test set of 50,485 OE62 molecules with a root mean square error (RMSE) of 1.4 Å. In relative terms, real and predicted distances differ on average by 16.5%. The fact that some error remains in the predictions is of course not unexpected. For one, the molecular graph simply does not contain all information necessary to reproduce the 3D geometry. A given graph in principle maps to a plethora of conformers and rotamers. The most important question therefore is: does the improved distance prediction of nGDE over sGDE map to better predicted geometries?

To illustrate this, consider the alanine hexamer. This is a simple model peptide that displays a large number conformers in different states of folding.[28] Figure 2 shows an overlay of 500 conformers generated via ETKDG, sGDE, nGDE. Visually, the sGDE ensemble is significantly more elongated than the nGDE or ETKDG ones. The figure also displays GFN-FF energy distributions for all three ensembles. This reveals that the sGDE ensembles contains the highest energy structures. Indeed, the sGDE distribution barely overlaps with the ETKDG distribution, which is both narrower and significantly lower in energy. Meanwhile, the nGDE ensemble yields a broader distribution, covering the main energy range of both other methods. Importantly, however, the spurious high energy conformers produced by the sGDE method are missing. The improved distance estimates of nGDE thus indeed yield a method that is more effective at generating stable geometries.

When contrasting ETKDG and nGDE, it is notable that the former distribution is rather unsymmetrical. This is likely due to the inclusion of experimental information on torsional angle distributions (i.e., the ET in ETKDG). While both methods sample a similar space of interatomic distances, ETKDG removes unlikely torsional angles and is thus more effective in generating low energy conformations. Meanwhile, the nGDE ensemble is more unbiased. This can be seen as a downside in terms of the relative stability of the conformers. It can also be useful, however, for example, for the generation of training data for ML interatomic potentials. In this case, a fuller exploration of torsional space will lead to more robust potentials.[8]

To further compare ETKDG and nGDE, 5369 randomly drawn drug-like molecules from the ZINC database were considered.[29] For each, five conformers were embedded using ETKDG and nGDE. Figure 3 shows the means and standard deviations of the energy differences between the lowest energy conformers (according to GFN-FF), where a negative value indicates a higher stability of the nGDE conformer, binned according to the number of atoms. This reveals two trends. First, the mean difference is close to zero for the smallest molecules and gradually becomes more negative as the molecules increase. For the largest molecules (70–80 atoms), the most stable nGDE conformer is on average 0.045 eV (≈1 kcal/mol) lower in energy than the most stable ETKDG conformer. Second, the standard deviation of the energy differences increases with the size of the molecules. This is expected, since the conformational space of a molecule (i.e., the number of local minima) becomes larger with more degrees of

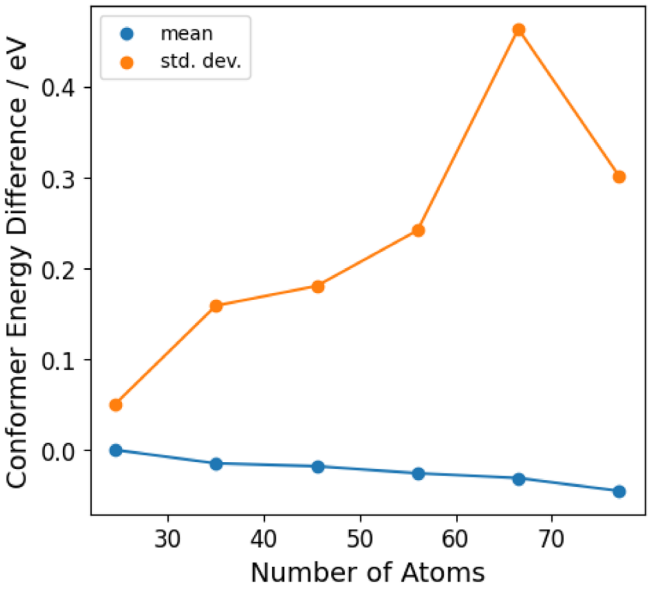

FIGURE 3 Means and standard deviations of energy differences between nGDE and ETKDG conformers of 5369 randomly selected drug-like molecules from the ZINC database, binned according to the number of atoms. Negative values indicate higher stability of the nGDE conformer. Fifteen molecules with particularly large conformer energy differences are displayed in Figure 4.

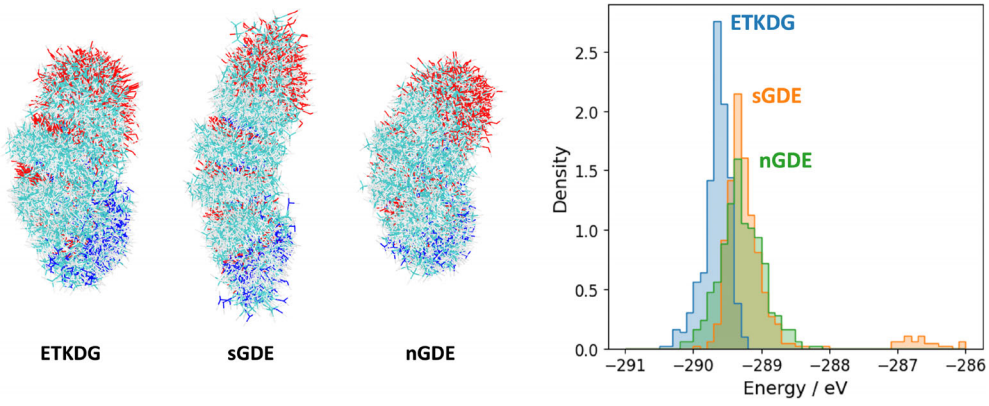

FIGURE 2 Left: Overlay of 500 conformers of hexaalanine in the gas-phase, generated with the ETKDG, sGDE, and nGDE methods. Right: Histograms showing the distribution of GFN-FF forcefield energies for each conformer ensemble.

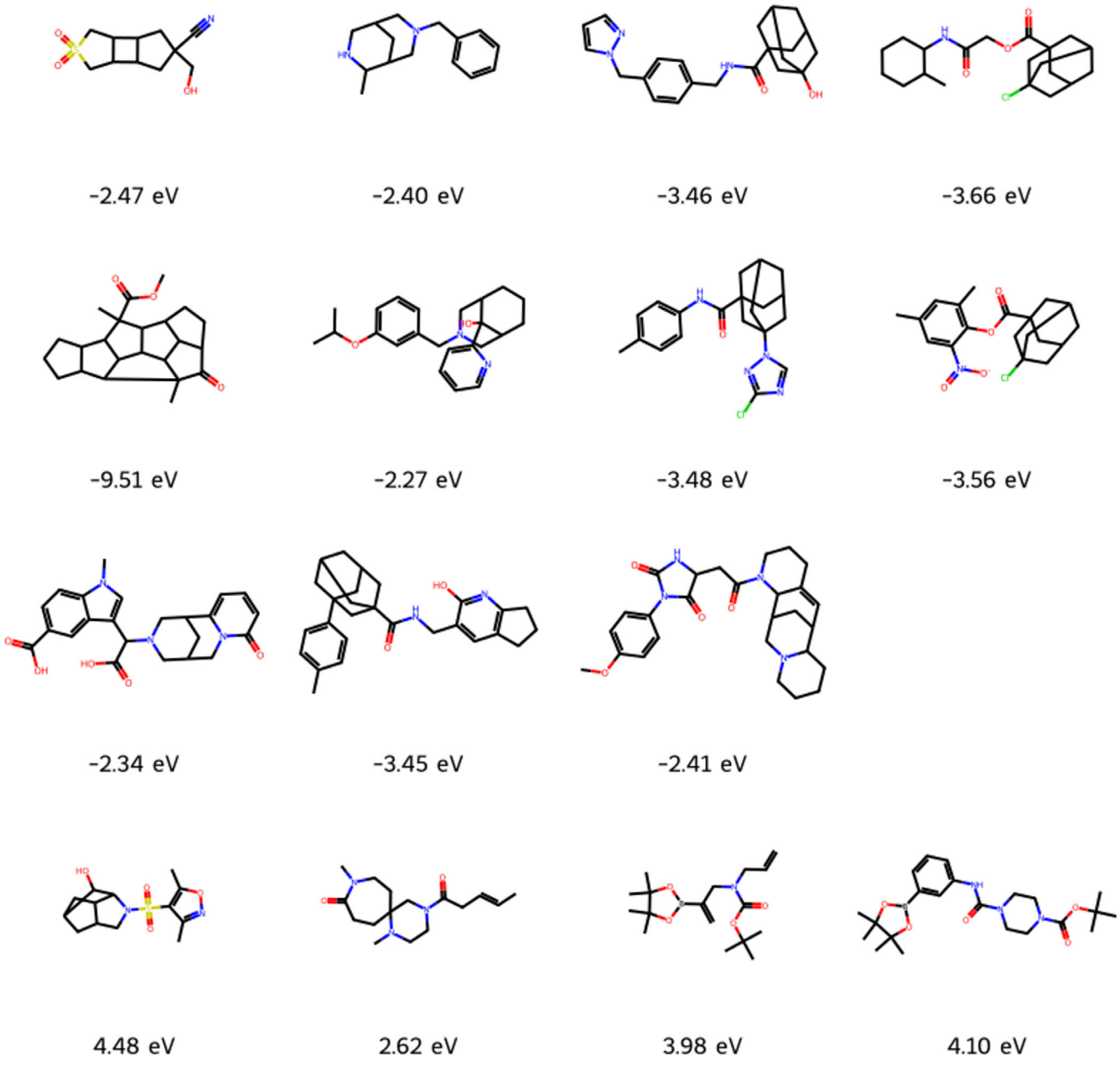

FIGURE 4 Structures of fifteen outlier molecules with conformer energy differences beyond the range of Figure 3. The top three rows show systems that are problematic for ETKDG, the bottom row shows systems that are problematic for nGDE.

freedom. Here, the rather small number of 5 conformers that are generated for each system inevitably leads to a larger spread in energies.

Beyond the mean and standard deviation, it is also instructive to consider the tails of this distribution. Here, several outliers display energy differences of several eV in magnitude. Specifically, there are eleven molecules where the best nGDE conformer is more than 2 eV lower in energy than the best ETKDG one. Conversely, in four cases the best ETKDG molecule was more than 2 eV lower in energy. These structures are shown in Figure 4.

Here, it is interesting to note that the failure modes for these edge cases are quite distinct. ETKDG mainly struggles with saturated polycyclic compounds. Indeed, for these it sometimes fails to generate embeddings at all. On the other hand, the failures of nGDE are not related to particular structural motifs. Instead, in all cases they are caused by misassignments of the GFN forcefield topology. In three of these cases, the initial geometries generated by nGDE contains bad contacts, which are translated to additional bonds by GFN. This leads to highly strained geometries with overcoordinated hydrogen atoms. Conversely, one case features a missing bond. Again, this is due to a poor initial geometry leading to a wrong force field topology.

On a positive note, the fact that these failure modes are clearly understood also opens a pathway towards improving both the nGDE and ETKDG approaches. In nGDE, additional postprocessing would be a viable route, for example by enforcing a

predefined force-field topology during refinement, or by checking for close contacts. In the case of ETKDG, improvements for certain molecule classes (e.g., aromatic systems or macrocycles) have previously been implemented.[12] Similar fixes could also be developed for polycyclic molecules, such as the ones shown in Figure 4.

5 | CONCLUSIONS

Herein a new approach for generating 3D molecular geometries termed nGDE was presented. nGDE uses a GNN trained on the extensive OE62 set to predict interatomic distances based on the molecular graph. The resulting distance matrices are then used in MDS to produce viable 3D geometries, which are subsequently refined with standard bioorganic forcefields. In this context, the ML-based graph distance introduced herein is shown to be a significant improvement over the conventional shortest path distance used in graph drawing. The nGDE approach is found to be competitive with the state-of-the-art ETKDG method for generating geometries of drug-like molecules. In particular, it is highly robust for polycyclic molecules, which are challenging for ETKDG.

One of the main advantages of nGDE is its conceptual simplicity, which allows it to be modified or extended as necessary. In particular, we aim to translate this approach to the prediction of crystal structures in future work.[30,31] Another possible research direction is the development of a fully end-to-end structure prediction model based on nGDE. Currently, the training only optimizes the prediction of interatomic distances, while the structure prediction itself is performed via conventional MDS. A fully trainable workflow would likely allow for more accurate structure generation.

It should be stressed that the molecular structures considered herein are limited to conventional covalently bonded, charge neutral organic molecules. While nGDE makes no strong assumptions about the bonds (e.g., regarding bond order, aromaticity or atom types), some definition of the bond topology is required, as well as characteristic lengths for each bond. This means that non-covalent systems (including host-guest or mechanically interlocked compounds) cannot be treated by the method out of the box. Such systems can however be described in principle, if the graph is expanded by additional edges corresponding to non-covalent contacts (such as hydrogen bonds). The same is true for coordination compounds.

ACKNOWLEDGMENT

Use of computational resources from the HPC keylab at the University of Bayreuth is greatfully acknowledged. Open Access funding enabled and organized by Projekt DEAL.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in gitlab.com at https://gitlab.com/jmargraf/ngde/.

ORCID

Johannes T. Margraf https://orcid.org/0000-0002-0862-5289

REFERENCES

[1] J.-P. Ebejer, G. M. Morris, C. M. Deane, *J. Chem. Inf. Model.* **2012**, *52*, 1146.
[2] P. C. D. Hawkins, *J. Chem. Inf. Model.* **2017**, *57*, 1747.
[3] P. Schwaller, T. Laino, T. Gaudin, P. Bolgar, C. A. Hunter, C. Bekas, A. A. Lee, *ACS Cent. Sci.* **2019**, *5*, 1572.
[4] T. Blaschke, J. Arús-Pous, H. Chen, C. Margreitter, C. Tyrchan, O. Engkvist, K. Papadopoulos, A. Patronov, *J. Chem. Inf. Model.* **2020**, *60*, 5918.
[5] J. Jiménez-luna, F. Grisoni, G. Schneider, *Nat. Mach. Intell.* **2020**, *2*, 573.
[6] I. Batatia, D. P. Kovacs, G. Simm, C. Ortner, G. Csanyi, in *Advances in Neural Information Processing Systems*, Vol. 35 (Eds: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh), Curran Associates, Inc., New York **2022**, p. 11423.
[7] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, B. Kozinsky, *Nat. Commun.* **2022**, *13*, 2453.
[8] S. Stocker, J. Gasteiger, F. Becker, S. Günnemann, J. T. Margraf, *Mach. Learn.: Sci. Technol.* **2022**, *3*, 045010.
[9] J. Gasteiger, C. Yeshwanth, S. Günnemann, in *Advances in Neural Information Processing Systems* (Eds: A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan), Curran Associates, New York **2021**.
[10] D. C. Spellmeyer, A. K. Wong, M. J. Bower, J. M. Blaney, *J. Mol. Graphics Modell.* **1997**, *15*, 18.
[11] S. Riniker, G. A. Landrum, *J. Chem. Inf. Model.* **2015**, *55*, 2562.
[12] S. Wang, J. Witek, G. A. Landrum, S. Riniker, *J. Chem. Inf. Model.* **2020**, *60*, 2044.
[13] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, D. Hassabis, *Nature* **2021**, *596*, 583.
[14] D. Lemm, G. F. Von Rudorff, O. A. Von Lilienfeld, *Nat. Commun.* **2021**, *12*, 9973.
[15] G. D. Battista, P. Eades, R. Tamassia, I. G. Tollis, *Comput. Geom.* **1994**, *4*, 235.
[16] S. G. Kobourov, arXiv preprint arXiv:1201.3011v1, 2012.
[17] T. Kamada, S. Kawai, *Inf. Process. Lett.* **1989**, *31*, 7.
[18] J. De leeuw, *J. Classif.* **1988**, *5*, 163.
[19] E. W. Dijkstra, *Numer. Math.* **1959**, *1*, 269.
[20] D. J. Klein, M. Randić, *J. Math. Chem.* **1993**, *12*, 81.
[21] A. Stuke, C. Kunkel, D. Golze, M. Todorović, J. T. Margraf, K. Reuter, P. Rinke, H. Oberhofer, *Sci. Data* **2020**, *7*, 241722.
[22] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, G. E. Dahl, in *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70 (Eds: D. Precup, Y. W. Teh), PMLR, New York **2017**, p. 1263.
[23] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, M. Grohe, *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, *AAAI'19/IAAI'19/EAAI'19*, AAAI Press, Washington, DC **2019**.
[24] T. A. Halgren, *J. Comput. Chem.* **1996**, *17*, 490.
[25] S. Spicher, S. Grimme, *Angew. Chem., Int. Ed.* **2020**, *59*, 15665.
[26] rdkit/rdkit: 2023_09_2 (q3 2023) release, 2023.

[27] A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, K. W. Jacobsen, *J. Phys.: Condens. Matter* **2017**, *29*, 273002.

[28] S. Chmiela, V. Vassilev-Galindo, O. T. Unke, A. Kabylda, H. E. Sauceda, A. Tkatchenko, K.-R. Müller, *Sci. Adv.* **2023**, *9*, eadf0873.

[29] J. J. Irwin, K. G. Tang, J. Young, C. Dandarchuluun, B. R. Wong, M. Khurelbaatar, Y. S. Moroz, J. Mayfield, R. A. Sayle, *J. Chem. Inf. Model.* **2020**, *60*, 6065.

[30] S. Wengert, G. Csányi, K. Reuter, J. T. Margraf, *Chem. Sci.* **2021**, *12*, 4536.

[31] S. L. Price, *Chem. Soc. Rev.* **2014**, *43*, 2098.

How to cite this article: J. T. Margraf, *J. Comput. Chem.* **2024**, *45*(21), 1784. https://doi.org/10.1002/jcc.27349