Final project for CMSC 35450 1 (Autumn 2022) Geometric Deep Learning: A Comparison of Generative Modelling Approaches for Conditional Molecular Structure Generation

Mike Jones[1], *Kirill Shmilovich*[1]

[1]Pritzker School of Molecular Engineering, University of Chicago, Chicago, Illinois 60637, United States

mikejones@uchicago.edu, kirills@uchicago.edu

Abstract

In this work, we compare two approaches for conditional molecular structure generation: generative adversarial networks (GANs) and denoising diffusion probabilistic models (DDPMs). These approaches are compared in application to two prototypical biomolecules alanine dipeptide and the Trp-Leu-Ala-Leu-Leu (WLALL) pentapeptide. In our tests, we evaluate the fidelity of the generated structures and find that while DDPMs quantitatively outperform GANs, both models are capable of generating visibly convincing structures indistinguishable from the reference data. These results demonstrate the potential, limitations, possible improvements and future directions for generative modeling approaches in conditional molecular structure generation.

1. Introduction

Molecular dynamics (MD) simulations are a commonplace tool in the computational sciences for studying nanoscale phenomena such as molecules and materials [1, 2]. MD simulations are frequently applied to proteins and other organic biomolecules as these systems represent important components in biological cascades, and a deeper understanding of their structure and function can have implications in designing new interventions, diagnostics, and therapeutics [3]. The inner workings of MD simulations are fairly straightforward: a collection of N bonded atoms (i.e., a molecule) are simulated by treating each atom as a point-particle with induced forces determined by the relative positions of the other atoms. The system can then be dynamically evolved in time by integrating forward Newton's equations of motion (F=ma) to update the atom positions as a function of their instantaneous forces. While MD simulations have proven to be remarkably successful at recapitulating experimental observables, their accuracy is contingent on small integration time steps commensurate with the fastest molecular motions typically on the order of 10^{-12} seconds [4]. However, many interesting phenomena, such as protein folding, can occur on much longer timescales – sometimes on the order of seconds, or even minutes – which can result in long simulation times that may span days or weeks to accurately probe these events and gather sufficient statistics [5, 6].

In order to bridge the gap between MD and experimentally relevant timescales, numerous techniques have been developed to project the high-dimensional $3N$ configuration space into a low-dimensional latent space [7, 8, 9]. The effective simulation time can then be extended by orders of magnitude by training a surrogate model to propagate a simulation within the latent space instead of integrating the equations of motion for each atom at every time step. The premise of this approach is rooted in the manifold hypothesis that postulates there exists a low dimensional subspace with dimension $d << 3N$ of the full $3N$ dimensional configuration space that encapsulates the slowest evolving motions with the remaining degrees of freedom relegated as dynamical noise [10, 11]. Latent Space Simulators (LSS) [9] provide a framework for performing simulations within low-dimensional latent spaces by training an encoder to optimally distill a system's kinetic variance, and then training a decoder to reconstruct all-atom configurations from the simulated set of latent coordinates. Previous work [9] has used generative adversarial networks (GANs) [12] for this task of backmapping or decoding these latent space trajectories, as this provides a non-deterministic approach for molecular structure generation that captures the configurational diversity associated with unique latent codes.

We evaluate in this work different approaches for conditional molecular structure generation by providing a comparison between GANs and Denoising Diffusion Probabilistic Models (DDPM) [13]. While previous work has already established GANs as a useful tool for molecular structure generation, we explore DDPMs as an alternative approach that has shown improvements in generative modeling leveraging more stable training and superior error correction at inference time via the multi-step diffusion process. In this work, we tackle the problem of conditional molecular structure generation applied to two exemplar biomolecular systems: alanine dipeptide (ADP) [14, 15] and the Trp-Leu-Ala-Leu-Leu (WLALL) pentapeptide [16]. The GAN and DDPM models are compared by evaluating the generated structures by analyzing intramolecular bond lengths and the ability to adhere to conditioning variables. Our tests reveal that DDPM models slightly outperform GANs, but both approaches are capable of generating realistic structures and are well-suited for the task of conditional molecular structure generation.

2. Methods

In this work we compare two generative modeling approaches for the task of molecular structure generation: generative adversarial networks (GANs) and Denoising Diffusion Probabilistic Models (DDPMs). We specifically tackle the problem of *conditional* molecular structure generation, where we are interested in generating configurations that are congruent with some conditioning variables. These conditioning variables serve to define "macro" aspects of the molecular structure, e.g., if a molecule is in the folded or unfolded state, while the fine details like sidechain orientations and hydrogen atom placements can be hallucinated realistically using our GAN or DDPM models. In our comparisons between the two generative modeling approaches,

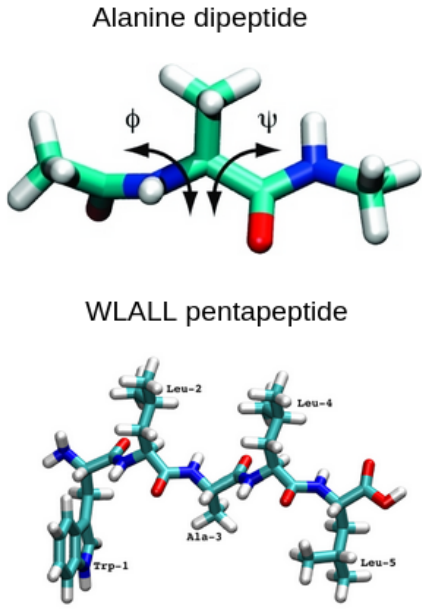

Figure 1: *Rendering of the two molecules alanine dipeptide (top) and the WLALL pentapeptide (bottom) investigated in this work. For alanine dipeptide the backbone ϕ,ψ angles are identified, while for WLALL the five side chains are labeled.*

we quantitatively evaluate the structural validity of hallucinated structures alongside their capability of generating configurations that adhere to the requested conditioning variables.

2.1. Data Curation

Data used for training and evaluation of our models comprises molecular trajectories of two prototypic biomolecules alanine dipeptide and Trp-Leu-Ala-Leu-Leu (WLALL) pentapeptide that are publicly available via mdshare (Fig. 1) [17]. For each system, the available data comprises k separate trajectories with an aggregated n trajectory snapshots each containing the spatial xyz coordinates for all N atoms of the molecule. Conditioning variables are generated for each system using either *a priori* known physical descriptors for alanine dipeptide, or a dimensionality reduction technique to produce low dimensional descriptors in the case of the pentapeptide system. A synopsis of our training and evaluation data setups is provided in Table 1.

2.2. Conditioning Variables

Optimal conditioning variables should be low-dimensional representations of a system's slowest dynamical processes e.g. protein (un)folding or DNA (de)hybridization. For small, well-studied bio-molecular systems these coordinates can be *a priori* known and typically manifest as intramolecular distances or dihedral angles. For alanine dipeptide the ϕ and ψ backbone dihedrals are well-established collective variables that distinguish relevant metastable states, and as such we adopt these same coordinates as our conditioning variables. For larger systems, various dimensionality reduction approaches are employed to reduce a set of roto-translationally invariant features – usually distances and/or angles – into a kinetically meaningful latent space. For our WLALL pentatpeptide system, we employ Time-lagged Independent Component Analysis (TICA) [18] to produce a four-dimensional conditioning variable as a function of intramolecular pairwise distance between either (i) the protein backbone atoms or (ii) any (non-hydrogen) heavy atoms. The models trained with condition (i) are used to generate only the pentapeptide backbone (20 atoms), and models trained with condition (ii) are designed to generate all-atom configurations (94 atoms). The TICA method can be understood as a time-lagged analog to PCA, where the time-lagged Independent Components (tICs) t_i are determined by solving the eigenvalue problem $C_\tau t_i = C_0 \lambda t_i$ where C_0 is a covariance matrix and C_τ is a time-lagged covariance matrix [19]. The leading eigenvectors t_i capture the majority of the system's kinetic variance and represent coordinates identifying the slowest evolving dynamical motions.

2.3. Models

The GAN and DDPM models used in this work share a common data representation that combines the molecular structure and conditioning variables. The molecular structure consisting of N atoms is exposed to our models as a flattened vector of the xyz coordinates with dimensionality $3N$. This coordinate representation can then be concatenated with the conditioning variables to couple the conditioning and coordinate data representations (Fig. 2). At inference time, both the GAN and DDPM models generate synthetic configurations by using as input a randomly sampled noise vector concatenated with the input conditioning variable (Fig. 2). From this synthetic structure we can then recalculate the conditioning variable, which in our application amounts to either calculating the backbone dihedral angles for ADP or tICs as a function of pairwise distances for WLALL. We should expect that our hallucinated structures reflect the input conditioning variables used to generate them, and by recalculating the condition based on these synthetic structures we can evaluate the extent to which our models adhere to the input condition. Details for the GAN and DDPM models are provided in separate sections below, and a complete PyTorch implementation of the models trained throughout this work is provided at: `https://github.com/KirillShmilovich/GAN_DDPM_conditional_molecular_structure_generation`.

2.3.1. GAN

The flavor of GAN trained in this work is specifically a Wasserstein GAN that uses the gradient penalty to enforce the Lipschitz constraint on the discriminator/critic (WGAN-GP) [20, 21]. Both the generator and discriminator networks are simple 3-layer MLPs using Sigmoid Linear Unit (SiLU) [22] activation functions trained using the RMSProp optimizer, which reflects the architecture and training settings used in previous work for a similar conditional molecular structure generation task [9]. All GANs trained throughout this work use this same underlying architecture, with the only difference between models being the hidden dimensionality of the generator and discriminator/critic networks. All MLPs use a hidden dimensionality of either 256 or 512, with the final production model selected by performing a sweep over these hidden dimension sizes.

The conditional component of the GANs is enforced by concatenating the conditioning variables as an input to the generator and discriminator/critic networks. Specifically, the generator takes as input a 128-dimensional random noise vector concatenated with the conditioning variable, this is a 2-dimensional conditioning variable in the ADP application and a 4-dimensional conditioning variable for WLALL. From this

System	Conditioning	Condition dim	Generate	N Atoms	Train Set	Test set
Alanine dipeptide	Backbone dihedrals	2	Backbone atoms	8	2 x 250,000	1 x 250,000
Pentapeptide	Backbone TICs	4	Backbone atoms	20	20 x 5,001	5 x 5,001
Pentapeptide	All Heavy TICs	4	All atoms	94	20 x 5,001	5 x 5,001

Table 1: *Summary of our training and testing data setups*

conditioning and noise, the generator then produces a flattened set of xyz coordinates as the synthetic structure. During training, the discriminator/critic is then tasked with determining the validity of the real and synthetic structures. This is achieved by feeding the discriminator/critic the synthetic or real structures concatenated with the conditioning variable. The discriminator/critic ultimately determines the validity of each fake/real structure jointly with the conditioning variable and as a result enables the generator to create realistic structures that are attentive to the conditioning variables.

2.3.2. *DDPM*

We employed a modified Denoising Diffusion Probabilistic Model [13] with a 1D convolution U-net [23] architecture and a masked conditioning vector. We adapted code from the work of Wang et. al. [24] who used a similar model to generate molecular backbone angles as a function of MD simulation temperatures. The masked conditioning vector is concatenated to the initial noise vector and is held constant at each diffusion time-step in order to ensure the condition is enforced throughout training and inference. We used an initial U-net channel dimension of 32 and multiplied the channel by a factor of 2 for each subsequent layer. A weight-standardized 1D convolution [25] and residual connections were used at each U-net step as implemented in the original model. A cosine noise scheduler [26] was used to add noise as a function of diffusion time-step. We used 1,000 diffusion steps and between 500,000 and 1,000,000 training steps depending on the system.

2.4. Evaluation Metrics

We developed and implemented three evaluation metrics to determine (i) structural similarity, (ii) physical plausibility, and (iii) fidelity to the conditioning variable.

2.4.1. *Root Mean Squared Bond Distance (RMSD)*

$$RMSD = \sqrt{\frac{1}{BF}\sum_{i}^{F}\sum_{j}^{B}(b_{ij}^{t} - b_{ij}^{g})^{2}} \qquad (1)$$

The RMSD is computed from the difference in bond lengths for the test set b_{ij}^{t} and the generated set b_{ij}^{g} for all B bonds in all F frames. Although this serves as a useful comparison, we would not expect this quantity to asymptote to zero as we should expect some structural diversity in the generated structure. This metric nevertheless provides a roto-translationlly invariant metric for the quality of the generated structures.

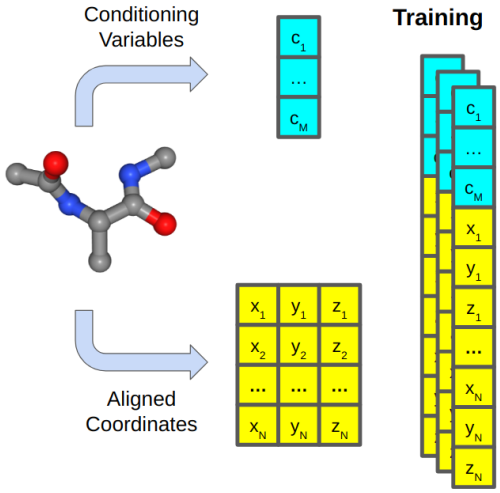

Figure 2: *Training data extracted from molecular dynamics configurations. Trajectories are aligned to a reference structure, and Cartesian coordinates (yellow) are flattened into a $3N_{atoms}$-dimensional vector. Condition vectors (blue) are determined directly from roto-translationally invariant dihedrals, or by dimensionality reduction via TICA. These conditioning vectors are concatenated to the coordinate vectors and aggregated across all training frames*

.

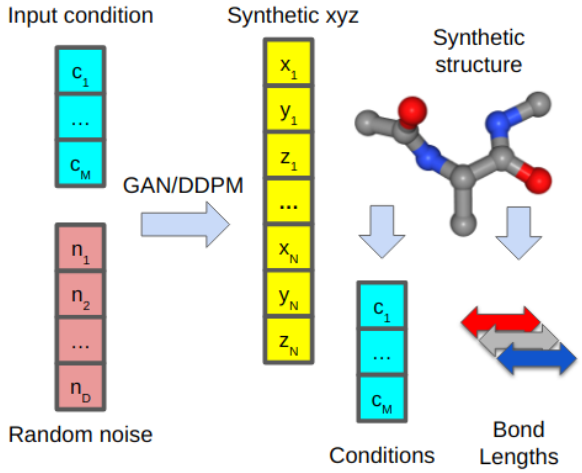

Figure 3: *Deployment and inference of generative models and evaluation of atomic configurations. Each model starts with a d-dimensional noise vector (red) and is conditioned on c to recover a vector of flattened atomic coordinates. This vector is reshaped into a N_{atoms} x 3 array which can be translated into a molecular structure and used to re-calculate bond lengths and conditioning variables for the evaluation of model performance.*

.

2.4.2. Area Under Bond Fraction Curve (AUC)

$$AUC = \frac{1}{BF}\sum_{i}^{F}\sum_{j}^{B} v_{ij} \tag{2}$$

$$v_{ij} = \begin{cases} 1, & \text{if } b_j^{min} < b_{ij} < b_j^{max} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

AUC quantifies the physical viability of a trajectory across frames and bonds. To evaluate a viable range for each bond, we take the minimum b_j^{min} and maximum values b_j^{max} of that bond across all frames in the training data. We compare each bond the generated trajectories to its respective physical limits across all frames and calculate the total fraction of realistic bond lengths. Unlike RMSD, this quantity should continue to improve until it reaches 1, which would indicate that the models are exclusively generating physically plausible structures. This also serves as a more stringent criterion, as achieving the maximum possible value of 1.0 indicates that 100% of the generated frames have 100% physical bond lengths.

2.4.3. Average Conditioning Correlation (ACC)

$$ACC = \frac{1}{M}\sum_{i}^{M} \mathbf{P}(c_i^t, c_i^g) \tag{4}$$

ACC evaluates the extent to which the conditioning variable is reproduced in the generated data. For each system, we recalculate the condition from the generated coordinates – in the case of TICA we used the same encoding parameterized on training data – and compute a Pearson correlation $\mathbf{P}$ between this generated condition c_i^g the original condition from the test set c_i^t. We compute the average across each condition where values near 1 indicate very strong agreement with the condition and 0 indicates no correlation.

3. Results

3.1. Molecular backbone generation

As an initial test of our models, we generated configurations for the eight-atom backbone of ADP. We found that both models converged quickly for this system, producing RMSDs below 0.004 and over 99.98 % correct bond configurations as indicated by AUC in Table 2. The DDPM performed slightly better in both metrics and showed a more notable 0.01 improvement in ACC, indicating that DDPM structures were more faithful to the conditioning variable than GAN structures. Visualizations in Figure 4 show that generated ADP structures are nearly indistinguishable from the respective test set configurations they are conditioned on.

Next, we used both models to generate the WLALL backbone atoms. This system contains 20 total atoms defined by eight dihedral angles along the Carbon-Oxygen-Carbon backbone. This represents substantially more configurational variability than ADP, and we observe a greater diversity of structures in the training data. Select frames in Figure 4 show more variation between the generated and test set, particularly in frame 4275, for both models compared to ADP. High ACC values, however, suggest that this variation occurs in structural regions that are not strongly conditioned by the TICs and are therefore a valid reconstruction of their latent coordinates. We emphasize that the conditioning variables used in this test were

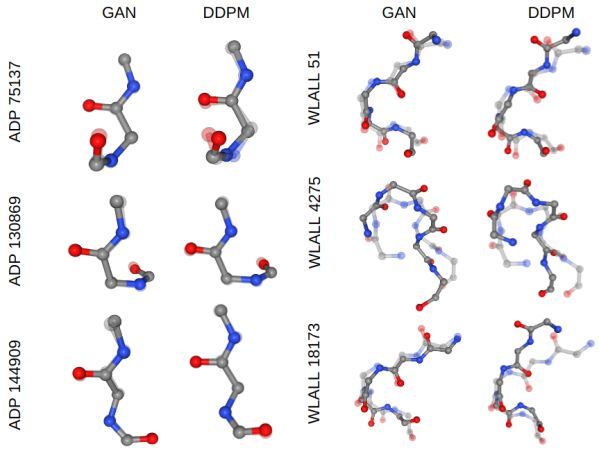

Figure 4: *Generated structures overlaid on corresponding transparent frames from the test set for ADP (left) and the WLALL backbone (right). For each system, three frames with distinct configurations are used as comparisons for structures generated from the GAN and DDPM*

.

a function of only the backbone atoms, and as a result, the sidechains of WLALL are effectively neglected here. Greater variation leads to a larger bond RMSD, with the DDPM remaining slightly better (0.0063) than the GAN (0.0077). We observe a larger discrepancy between AUC for the DDPM (0.964) and GAN (0.936) indicating that there are about twice as many unphysical bonds being generated by the GAN model.

3.2. All-atom generation

The most challenging system we tested was all-atom reconstruction of WLALL. This task required generating 94 atoms, including all peptide side chains and hydrogen atoms which may not be well captured by the conditioning variables. Indeed, predicting hydrogen atoms is often not included in the generation task [9] and may be accomplished by a post-processing tool, however, we found this additional challenge helpful for differentiating our models. Again, we find the DDPM outperforms the GAN across all metrics, particularly in AUC where we observe a 15% higher fraction of physically realistic bonds. We also observe that the DDPM ACC remains nearly as high as that of the backbone, whereas the GAN performance drops to 0.93. Still, we find that both models produce visually realistic structures across a majority of frames as shown in Figure 5. Large variations in the Tryptophan sidechain make it challenging to compare similarities with the test data, however, we observe that the relevant backbone features are well preserved across frames in both models.

3.3. Generated structures are distributed around conditioning coordinates

We investigated the distribution of configurations generated by a single condition for both the GAN and DDPM. For both models, we generated 1000 structures conditioned on TIC coordinates from the first frame of the all-atom WLALL test set. In Figure 6 we show sample configurations generated from each model along with the distribution of generated TICs compared to the conditioning (ground-truth) TIC. The distribution for both models appears roughly Gaussian, where the variance for TIC_0

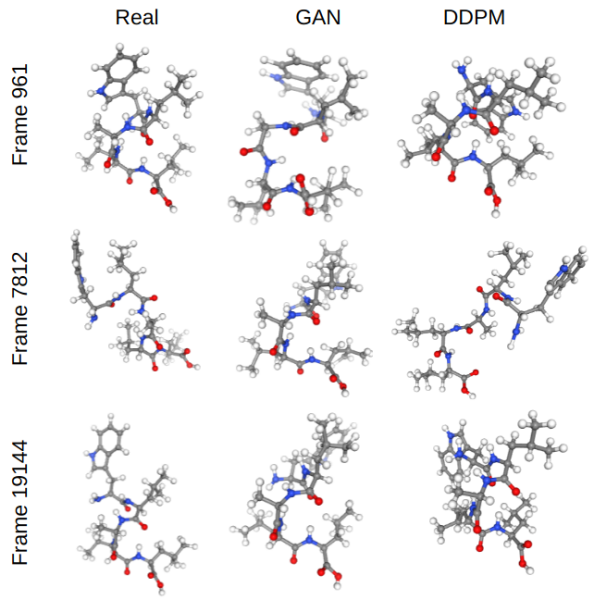

Figure 5: *Structural comparison for WLALL where all 94 atoms are generated. Three frames from the test set are shown in the left column, and generated structures from corresponding TIC conditions are shown for the GAN and DDPM in the center and right columns, respectively.*

.

System	Model	RMSD ↓	AUC ↑	ACC ↑
Alanine Dipeptide	GAN	0.0039	0.99982	0.975
	DDPM	**0.0037**	**0.99984**	**0.986**
Pentapeptide (backbone)	GAN	0.0077	0.936	0.959
	DDPM	**0.0063**	**0.964**	**0.979**
Pentapeptide (all-atom)	GAN	0.0157	0.494	0.930
	DDPM	**0.0101**	**0.648**	**0.977**

Table 2: *Numerical comparison of GAN and DDPM models for the three systems studied in this work.*

and TIC_1 are lower compared to TIC_2 and TIC_3. We observe that the mean of the DDPM distributions are closer to the ground truth TICs, especially TIC_2. Furthermore, the GAN is more susceptible to generating a small population of outlier structures that do no not satisfy the conditioning whereas the DDPM distributions are more tightly bounded. These outliers may account for why we observe systematically lower TIC correlations (Table 2) for the GAN across all systems.

3.4. Visualization of the GAN training progress

To visualize the progression throughout GAN training we saved generated structures produced at intermediate training epochs (Fig. 7). A movie of this progress throughout the GAN training is viewable at `https://github.com/KirillShmilovich/GAN_DDPM_conditional_molecular_structure_generation`. We find that the backbone structure stabilizes and converges more rapidly than the side chains or hydrogen atom placements. This behavior is someone expected because the conditioning TICs better encapsulates the backbone variability as determinants of the slow modes. Other high-frequency motions such as the side chain states or hydrogen atom places take much longer to converge, as these molecular motions are poorly described by the leading TICs representing faster-occurring motions.

3.5. DDPM learning is concentrated in later diffusion time-steps

To better understand the DDPM learning process, we saved the state vector from each diffusion time-step and generated a continuous trajectory in Cartesian space. The full movie is shown at: `https://github.com/KirillShmilovich/GAN_DDPM_conditional_molecular_structure_generation` and select steps are shown in Figure7, where the final structure at t=1000 is represented by the transparent configuration. We found that most atoms remain far from their final positions during the first half of the diffusion process, although there is a high density of atoms in conserved backbone regions. Even by step 900, many hydrogen and side chain atoms remain distant from their final sites. This makes sense given that these regions are least dependent on the conditioning variables, and show the most diversity when generating multiple configurations from the same conditioning as discussed above and shown in Figure6.

4. Discussion

We find that both the GAN and DDPM are well-suited to the task of conditional molecular structure generation, and we hypothesize that one model may be better suited over the other depending on the application. Although the DDPM outperforms the GAN across all systems and evaluation metrics, the improvement is marginal in most cases and both models produce visually and physically realistic configurations. The best performing DDPM model contains $\sim 3\times$ more parameters than the GAN ($\sim$ 900k for the GAN vs. $\sim$3M for the DDPM) and requires 1000x more inferences steps as the DDPM must perform inference via the 1000-step diffusion process rather than the one-shot inference of the GAN model. This does not pose a problem for applications such as Latent Space Simulator [9] which only require one latent decoding step, but other molecular propagators [27] and back-mapping approaches require iterative encoding/decoding and would be better served by a less accurate but faster model. We did find that the DDPM performance remained strong down to 250 time-steps (4x decrease), however decreasing the size of the input channels and U-Net dimensions had a more deleterious effect on performance. The DDPM may be best suited for tasks similar to all-atom WLALL reconstruction, where fine-grained details such as hydrogen bonds must be reproduced despite the fact that they are not well represented in the conditioning variables. Another advantage of DDPM models as compared to GANs is the relative ease of training. Training GAN models requires a delicate balance between the generator and discriminator/critic networks, requiring careful tuning to ensure proper convergence.

In addition to optimizing model hyper-parameters, we also explored more specialized geometric deep learning encoders for the Cartesian coordinate inputs. We implemented a PointNet [28] encoder for the GAN discriminator, but found that it disrupted the balance with the generator and significantly decreased the performance of both models during training. We also find this imbalance to be prevalent for other types of discriminator networks, such as 1D convolution-based discriminators. A contributing factor to this imbalance could be related

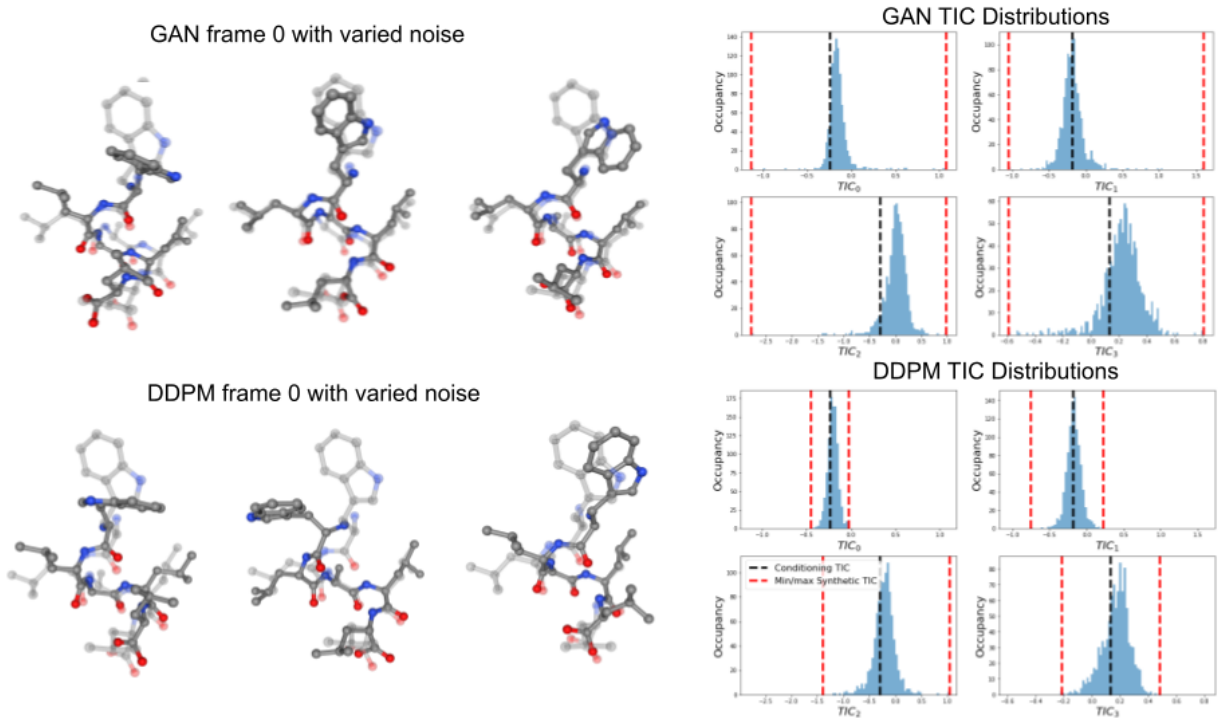

Figure 6: *Sampling generated structures using fixed conditioning variables (first frame of the test set). The top left structures are three samples from the GAN, and the bottom left are three structures sampled from the DDPM. Histograms show distributions of 1000 samples from each model. Dashed black lines correspond to the conditioning TIC, and dashed red lines show the minimum and maximum TICs generated for each synthetic distribution.*

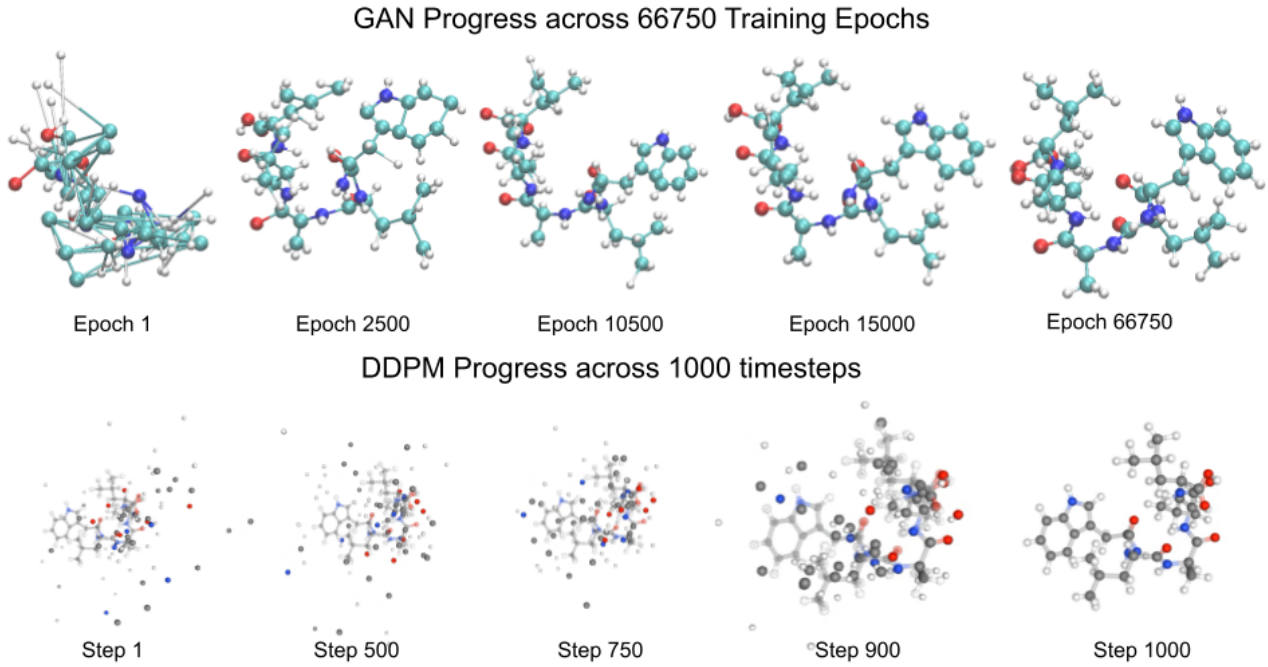

Figure 7: *Top shows generated structure for fixed conditioning during select epochs of GAN training. Bottom shows sampled structures for intermediate time-steps in the DDPM inference process.*

to the limited expressiveness of the generator operating on the flattened xyz data representation. The generator network is unable to leverage good inductive biases when operating on this data modality, making the discriminator's job comparatively easy. For the DDPM, we considered replacing the 1D convolution U-Net with a structure that accounted for a geometric structure such as Point-Unet [29] or Graph U-Nets [30]. However, we found that the DDPM performance was very sensitive to any changes to the U-Net architecture, and the problem of introducing diffusion time conditioning to a novel architecture was a non-trivial one. Therefore we leave the construction of molecule-specific U-Net to future work, and we hypothesize that such an architecture may be required to accurately generate molecules on the order of many hundreds or thousands of atoms. We believe that both the GAN and DDPM models could benefit from an improved data representation of the molecular structure to provide stronger inductive biases for improved, and more efficient, molecular structure generation.

5. References

[1] T. Hansson, C. Oostenbrink, and W. van Gunsteren, "Molecular dynamics simulations," *Current opinion in structural biology*, vol. 12, no. 2, pp. 190–196, 2002.

[2] S. A. Hollingsworth and R. O. Dror, "Molecular dynamics simulation for all," *Neuron*, vol. 99, no. 6, pp. 1129–1143, 2018.

[3] J. D. Durrant and J. A. McCammon, "Molecular dynamics simulations and drug discovery," *BMC biology*, vol. 9, no. 1, pp. 1–9, 2011.

[4] R. Elber, "Perspective: Computer simulations of long time dynamics," *The Journal of chemical physics*, vol. 144, no. 6, p. 060901, 2016.

[5] J. L. Klepeis, K. Lindorff-Larsen, R. O. Dror, and D. E. Shaw, "Long-timescale molecular dynamics simulations of protein structure and function," *Current opinion in structural biology*, vol. 19, no. 2, pp. 120–127, 2009.

[6] B. E. Husic and V. S. Pande, "Markov state models: From an art to a science," *Journal of the American Chemical Society*, vol. 140, no. 7, pp. 2386–2396, 2018.

[7] W. Chen, H. Sidky, and A. L. Ferguson, "Nonlinear discovery of slow molecular modes using state-free reversible vampnets," *The Journal of chemical physics*, vol. 150, no. 21, p. 214114, 2019.

[8] L. Bonati, G. Piccini, and M. Parrinello, "Deep learning the slow modes for rare events sampling," *Proceedings of the National Academy of Sciences*, vol. 118, no. 44, p. e2113533118, 2021.

[9] H. Sidky, W. Chen, and A. L. Ferguson, "Molecular latent space simulators," *Chemical Science*, vol. 11, no. 35, pp. 9459–9467, 2020.

[10] M. A. Rohrdanz, W. Zheng, and C. Clementi, "Discovering mountain passes via torchlight: Methods for the definition of reaction coordinates and pathways in complex macromolecular reactions," *Annual review of physical chemistry*, vol. 64, pp. 295–316, 2013.

[11] R. Hegger, A. Altis, P. H. Nguyen, and G. Stock, "How complex is the dynamics of peptide folding?" *Physical review letters*, vol. 98, no. 2, p. 028102, 2007.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[13] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[14] F. Nuske, B. G. Keller, G. Pérez-Hernández, A. S. Mey, and F. Noé, "Variational approach to molecular kinetics," *Journal of chemical theory and computation*, vol. 10, no. 4, pp. 1739–1752, 2014.

[15] F. Vitalini, A. S. Mey, F. Noé, and B. G. Keller, "Dynamic properties of force fields," *The Journal of Chemical Physics*, vol. 142, no. 8, p. 02B611_1, 2015.

[16] C. Wehmeyer, M. K. Scherer, T. Hempel, B. E. Husic, S. Olsson, and F. Noé, "Introduction to markov state modeling with the pyemma software—v0. 3."

[17] M. K. Scherer, B. Trendelkamp-Schroer, F. Paul, G. Pérez-Hernández, M. Hoffmann, N. Plattner, C. Wehmeyer, J.-H. Prinz, and F. Noé, "Pyemma 2: A software package for estimation, validation, and analysis of markov models," *Journal of chemical theory and computation*, vol. 11, no. 11, pp. 5525–5542, 2015.

[18] Y. Naritomi and S. Fuchigami, "Slow dynamics of a protein backbone in molecular dynamics simulation revealed by time-structure based independent component analysis," *The Journal of Chemical Physics*, vol. 139, no. 21, p. 12B605_1, 2013.

[19] F. Noé and C. Clementi, "Kinetic distance and kinetic maps from molecular dynamics simulation," *Journal of chemical theory and computation*, vol. 11, no. 10, pp. 5002–5011, 2015.

[20] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.

[21] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[22] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, vol. 107, pp. 3–11, 2018.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[24] Y. Wang, L. Herron, and P. Tiwary, "From data to noise to data for mixing physics across temperatures with generative artificial intelligence," *Proceedings of the National Academy of Sciences*, vol. 119, no. 32, p. e2203656119, 2022.

[25] S. Qiao, H. Wang, C. Liu, W. Shen, and A. Yuille, "Micro-batch training with batch-channel normalization and weight standardization," *arXiv preprint arXiv:1903.10520*, 2019.

[26] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8162–8171.

[27] P. R. Vlachas, J. Zavadlav, M. Praprotnik, and P. Koumoutsakos, "Accelerated simulations of molecular systems through learning of effective dynamics," *Journal of Chemical Theory and Computation*, vol. 18, no. 1, pp. 538–549, 2021.

[28] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[29] N.-V. Ho, T. Nguyen, G.-H. Diep, N. Le, and B.-S. Hua, "Point-unet: A context-aware point-based neural network for volumetric segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2021, pp. 644–655.

[30] H. Gao and S. Ji, "Graph u-nets," in *international conference on machine learning*. PMLR, 2019, pp. 2083–2092.