Learning Long Timescale in Molecular Dynamics by Nano-GPT

Wenqi ZENG [1] **Yuan YAO** [1 2]

Abstract

Long-term dynamics in biomolecular processes are crucial for understanding the key evolutionary transformations of these systems. However, these long-term events requires extended simulation timescales to appear, often beyond the feasible forecast length of typical models. Consequently, the task is left to shorter but less accurate simulations. Although these simulations are brief, they are initiated with distinct perturbations, allowing them to sample the entire phase space and capture a wide range of behaviors over time. Recently, language models have been employed to learn key long-term dynamics from short simulations. However, existing approaches are limited to systems with low-dimensional reaction coordinates, projecting dynamics with memory effects. Here, we introduce nano-GPT, a novel deep learning model inspired by GPT architecture, specifically designed to manage complex dynamics and long-term dependencies in high-dimensional systems. The model employs a two-pass training structure to gradually replace MD tokens with model comprehension, thereby addressing biases in short simulations. Our findings demonstrate nano-GPT's superior ability to capture intricate dynamical properties and statistical features across extensive timescales, highlighting its potential to advance the understanding of biomolecular processes.

1. Introduction

In molecular dynamics (MD) simulations, long-term dynamics refers to the behavior and properties of a system over extended simulation timescales (Leimkuhler et al., 1996). Understanding these dynamics is crucial for studying processes such as protein folding, conformational changes, and the stability of molecular complexes. However, long-term dynamics occur on timescales ranging from microseconds to milliseconds or longer, while MD simulations require very small time steps (typically on the order of femtoseconds) to accurately integrate the equations of motion (Dullweber et al., 1997). Consequently, capturing long-term dynamics in MD presents several significant challenges due to limitations in computational resources and the inherent complexity of molecular systems (Chodera et al., 2007; Pan & Roux, 2008; Noé & Nuske, 2013).

Language models have recently been utilized as tools to learn the evolution of entire biomolecular processes (Tsai et al., 2020; 2022; Cao et al., 2023; Bai et al., 2022). In such scenarios, long-term dynamics are described as transitions occurring on a scale of 10^3 ps. The task aims to recover these dynamics using short sequences of around 10 ps, simulated along different paths to form a total sampling of 10^2 ns. The integration of short sequences provides sufficient coverage in time and phase space, making it possible to capture rare long-term events. A recent study (Tsai et al., 2020) demonstrates that LSTM networks (Gers et al., 2000) can achieve this goal under low-dimensional reaction coordinate projections. This work projects MD simulation trajectories onto low-dimensional reaction coordinates (e.g., ϕ and ψ) and employs a character-level LSTM model to learn probabilistic models of biophysical trajectories. However, these constrained probes reduce the system's complexity to two dimensions, focusing on specific torsional changes. They offer only limited insights into broader dynamics and are relatively straightforward to investigate.

The subsequent study (Tsai et al., 2022) addresses the challenge of recovering longer dynamics from faster short-term simulation trajectories. Their methodology integrates static and dynamic constraints to train an LSTM model twice, guided by the principle of Maximum Caliber. Results show that the transition time of 2000 ps is recovered with short sequences of 200 fs, with a total simulation of 20 ns. However, the test MD system is still projected onto a single reaction coordinate, which gives a more simplified and localized view. In contrast, the high-dimensional system encompasses the full 3D positions of all atoms in the molecule and provides a detailed global measure of structural changes. This complexity poses challenges in discerning subtle long-term

[1]Department of Mathematics, Hong Kong University of Science and Technology [2]Department of Chemical and Biological Engineering, Hong Kong University of Science and Technology. Correspondence to: Yuan YAO <yuany@ust.hk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

dynamics, as the signal of interest may be obscured by noise and irrelevant information. Therefore, it is valuable to investigate the performance of capturing long-term dynamics in high-dimensional space.

To extend deep learning models to high-dimensional systems, we propose a novel approach termed nano-GPT, which is based on a GPT-like structure (Brown et al., 2020) combined with scheduled sampling (Pang & He, 2020). The transition from LSTM to GPT is driven by the latter's ability to manage longer-term dependencies, which is essential for modeling the complex and evolving dynamics in biomolecular systems. Specifically, our model incorporates a two-pass structure that progressively substitutes the ground truth tokens with the model's predictions during training. Such design allows the model to generate tokens that may exceed the biases inherent in the ground truth, prompting subsequent adjustments to its predictions. Notably, nano-GPT is specifically designed to be a simplified model that can be effectively implemented on a 2080Ti GPU.

We validate the effectiveness of our approach using two distinct systems: a simulated system and the alanine dipeptide on varying levels of complexity and slow dynamics. Our nano-GPT model effectively captures statistical and dynamic features across a broad spectrum of timescales, irrespective of low-dimensional or high-dimensional data. In low-dimensional data, nano-GPT accurately represents long-term dynamics within 30,000 ps using 10 ps sequences, while in high-dimensional data, it achieves the same within 80,000 ps using 20 ps sequences, all within a total simulation duration of 100 ns. This capability is possibly achieved by effectively absorbing critical details from earlier positions in the sequence. Theoretically, we establish a linkage between the embeddings of states and their kinetic distances, which are essential for understanding metastable molecular dynamics.

2. Methods

Fig. 1 illustrates the workflow of the nano-GPT model presented in this paper. Given an input sequence $[x_1,...,x_t]$ of length t, the model outputs a prediction for the $(t+1)^{\text{th}}$ element, denoted as $\hat{x_{t+1}}$. A scheduled sampling scheme is incorporated into the model. This approach progressively adjusts the balance between using the ground truth and the model's own predictions during training, thereby enhancing the model's ability to generalize from training to inference scenarios.

We represent a molecular dynamic sequence as $[x_1,...,x_t] \in \mathbb{R}^t$ and transform it into higher dimension embeddings as $\mathbf{X} := [\mathbf{x}_1,...,\mathbf{x}_t]^T \in \mathbb{R}^{t\times D_x}$, where t represents the length of input sequence, D_x is the embedding dimension. Within the GPT model, these embeddings, $\mathbf{X}$, are transformed into hidden vectors $\mathbf{H}^{(l)}$ at the l-th layer. The model then processes $\mathbf{H}^{(l)}$ to produce the final probability distribution $\mathbb{P}(\hat{x}_{t+1}|x_1,...,x_t)$, which predicts the next element in the sequence.

A pivotal component of the GPT architecture is the self-attention mechanism. This mechanism transforms the input embeddings $\mathbf{X}$ into hidden vectors $\mathbf{H}^{(l)}$ through following steps:

Given query matrix $\mathbf{W}_Q \in \mathbb{R}^{D_q\times D_x}$, key matrix $\mathbf{W}_K \in \mathbb{R}^{D_q\times D_x}$ and value matrix $\mathbf{W}_V \in \mathbb{R}^{D_v\times D_x}$, the $\mathbf{X}$ is projected into $\mathbf{Q},\mathbf{K},\mathbf{V}$ as:

$$\begin{aligned}\mathbf{Q}^{(l)} &= \mathbf{H}^{(l-1)}\mathbf{W}_Q^T,\\ \mathbf{K}^{(l)} &= \mathbf{H}^{(l-1)}\mathbf{W}_K^T,\\ \mathbf{V}^{(l)} &= \mathbf{H}^{(l-1)}\mathbf{W}_V^T\end{aligned}$$

where $\mathbf{H}^{(0)} = \mathbf{X}$ and we denote $\mathbf{Q} := [\mathbf{q}_1,..,\mathbf{q}_t]^T$, $\mathbf{K} := [\mathbf{k}_1,..,\mathbf{k}_t]^T$, $\mathbf{V} := [\mathbf{v}_1,..,\mathbf{v}_t]^T$ for $i = 1,...,t$.

The output for next layer $\mathbf{H}^{(l)}$ is then defined as:

$$\mathbf{H}^{(l)} = \mathbf{H}^{(l-1)} + \mathbf{A}^{(l)} + \mathbf{B}^{(l)}, \tag{1}$$

$$\mathbf{A}^{(l)} = softmax(\frac{\mathbf{Q}^{(l)}(\mathbf{K}^{(l)})^T}{\sqrt{D_q}})\mathbf{V}^{(l)}, \tag{2}$$

$$\mathbf{B}^{(l)} = f_\theta(\mathbf{H}^{(l-1)} + \mathbf{A}^{(l)}) \tag{3}$$

From the left to right in Eq. 1, the addition of $\mathbf{H}^{(l-1)}$ stands for the residual connection inspired by ResNet (He et al., 2016). In Eq. 2, $\mathbf{A}$ is also known as attention matrix, which stands for the contextual information learned for every token in input sequence. The softmax function is applied row-wise on $\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_q}}$. For each vector in $\mathbf{A}$, an equivalent form is given as $\mathbf{A} := [\mathbf{a}_1,...,\mathbf{a}_t]$ where:

$$\mathbf{a}_i = \sum_{j=1}^{t} softmax(\mathbf{q}_i^T\mathbf{k}_j/\sqrt{D_q})\mathbf{v}_j$$

The $\mathbf{B}^{(l)}$ in Eq. 3 stands for the MLP network in Fig. 1, which is consisted of a two-layer neural network and normalizing non-linear networks. For the simplicity of notation, we use f_θ to characterize the non-linear transformation. Recall $\mathbf{H}^{(0)}$ is initialized as $\mathbf{X}$, $\mathbf{H}^{(l)}$ can be rewritten as:

$$\mathbf{H}^{(l)} = \mathbf{X} + \sum_{k=0}^{l}(\mathbf{A}^{(k)} + \mathbf{B}^{(k)}) \tag{4}$$

The decoding steps that yield the final probability distribution $\mathbb{P}(\hat{x}_{t+1}|x_1,...,x_t)$ involve a feed-forward neural network,

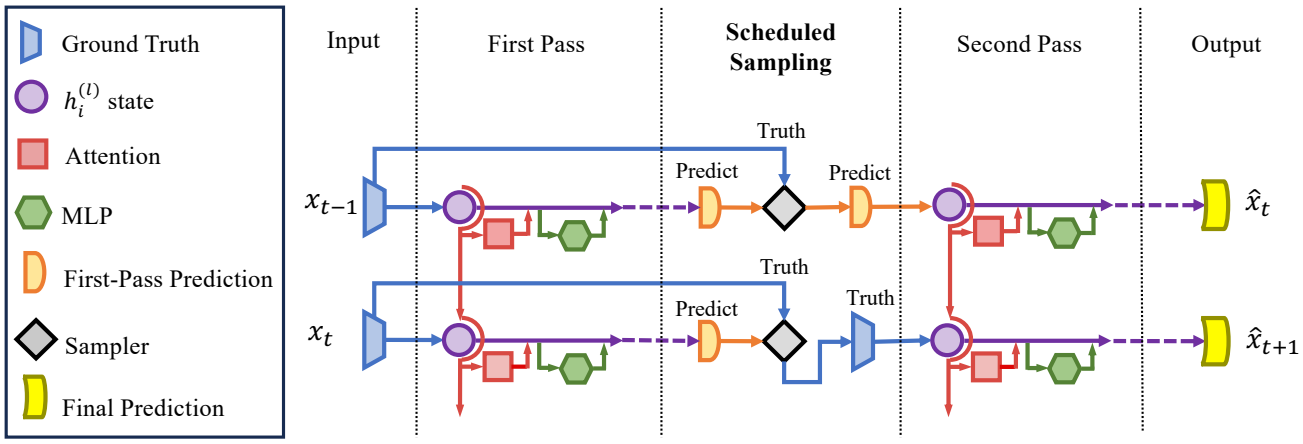

Figure 1: Nano-GPT model in our paper. It consists of a two-pass structure. The first pass operates as a standard decoder, with ground truth tokens (Golden) as the target input. While the second pass uses sampled tokens, chosen from either the golden tokens or the first-pass predictions.

which incorporates layer normalization and a residual connection. To simplify the notation, we represent this process as a nonlinear function f_θ. Following f_θ, the model employs a linear projection, represented by $\mathbf{D} \in \mathbb{R}^{D_{out} \times D_x}$, and concludes with a softmax activation function. This sequence of operations effectively transforms the hidden representations into a probability distribution over potential output tokens.

$$\mathbb{P}(\hat{x}_{t+1}|x_1,...,x_t) = softmax(f_\theta(\mathbf{H}^{(l)})\mathbf{D}^T + \mathbf{b}) \quad (5)$$

Cross-entropy loss equivalent to path entropy: To achieve a token-level pairwise matching between the predicted sequence and the training data, the optimization objective is the cross-entropy loss, as detailed in Eq. 6. This approach ensures that each token in the predicted sequence is as close as possible to its counterpart in the training dataset, thereby enhancing the model's accuracy and predictive performance.

$$Loss = -\sum_{t=0}^{t}\sum_{m} \mathbb{1}(x_{t+1} = m) \cdot \ln \mathbb{P}(\hat{x}_{t+1} = m|x_1,...,x_t) \quad (6)$$

Under the assumptions of first-order Markovianity and ergodicity, Tsai et al. demonstrates that optimizing a model using cross-entropy loss in Eq. 6, is equivalent to learning path entropy under the framework of Maximum Caliber (Pressé et al., 2013). This conclusion is readily extendable to a wide array of deep learning models, encompassing GPT architectures as well. It is crucial to emphasize that within this framework, the loss function is computed over the entire sequence rather than being limited to the last token.

Scheduled Sampling Although the Markovian assumption suggests an equivalence between optimizing path entropy and cross-entropy, such optimization relies on an exact match between predicted and input sequences. During training, models consistently receive the correct previous token as input. However, during generation, models rely on their own previously generated tokens. This fundamental difference between training (using ground truth data) and generation (relying on their own outputs) can result in significant performance variations. Additionally, this approach overlooks the fact that transitions between states often offer multiple choices.

In Figure 1, our model utilizes scheduled sampling during training, gradually substituting golden tokens with its own predictions. This approach employs a two-pass decoder: the first pass operates as a standard decoder, outputting weighted sums of target embeddings as probabilities, while the second pass uses sampled tokens, chosen from either the golden tokens or the first-pass predictions. The sampling probability follows a decaying scheme based on the i-th training step and t-th decoding position, as detailed in (Liu et al., 2021).

$$p = \begin{cases} \varepsilon^{t(1-k^i)} & \text{choose golden token} \\ 1 - \varepsilon^{t(1-k^i)} & \text{choose first pass prediction} \end{cases} \quad (7)$$

where ε and k are constants in the range (0, 1). This scheme entails that as the training step and decoding position increase, more model predictions are revealed. While for smaller training steps and decoding positions, more golden tokens are exposed. Denote the modified input as $\tilde{x}$. Conse-

quently, the loss in Eq. 6 is adjusted as follows:

$$Loss = -\sum_{t=0}^{t}\sum_{m} \mathbb{1}(x_{t+1}=m)\cdot \ln \mathbb{P}(\hat{x}_{t+1}=m|\tilde{x}_1,...,\tilde{x}_t) \quad (8)$$

The two decoders are identical and share the same parameters. During inference, only the first decoder is used.

Token Embedding Captures Kinetic Distances Eq. 5 can be rewritten as follow, where $\mathbf{e}_m \in \mathbb{R}^{D_{out}}$ is a one-hot vector with the m-th element non-zero.

$$\mathbb{P}(\hat{x}_{t+1}=m|\tilde{x}_1,...,\tilde{x}_{t-1}) = \frac{\exp((f_\theta(\mathbf{H}^{(l)})\mathbf{D}^T+\mathbf{b})\times \mathbf{e}_m)}{\sum_k \exp((f_\theta(\mathbf{H}^{(l)})\mathbf{D}^T+\mathbf{b})\times \mathbf{e}_k)} \quad (9)$$

By using Taylor's theorem, the $f_\theta(\mathbf{H}^{(l)}))$ can be approximated around a differentiable point $\mathbf{X}=\mathbf{m}$:

$$f_\theta(\mathbf{H}^{(l)}) \approx f_\theta(\mathbf{H}^{(l)})|_{\mathbf{X}=\mathbf{m}} + (\mathbf{X}-\mathbf{m})\mathbf{M}_\theta^T$$

where $\mathbf{M}_\theta$ is defined as $(\mathbf{M}_\theta)_{ij} = \frac{\partial (f_\theta)_i}{\partial \mathbf{x}_j}|_{\mathbf{X}=\mathbf{m}}$.

By Eq. 4, Eq. 9 becomes,

$$\mathbb{P}(\hat{x}_{t+1}=m|x_1,...,x_t) = \frac{\exp(\mathbf{C}_m)\exp(\mathbf{X}\mathbf{M}_\theta^T\mathbf{D}^T\mathbf{e}_m)}{\sum_k \exp(\mathbf{C}_k)\exp(\mathbf{X}\mathbf{M}_\theta^T\mathbf{D}^T\mathbf{e}_k)} \quad (10)$$

where $\mathbf{C}_m = [f_\theta(\mathbf{X}+\sum_{k=0}^{l}(\mathbf{A}^{(k)}+\mathbf{M}^{(k)}))|_{\mathbf{X}=\mathbf{m}} - \mathbf{m}\mathbf{M}_\theta^T)\mathbf{D}^T+\mathbf{b}]\times\mathbf{e}_m$.

In Eq. 10, $\mathbf{M}_\theta^T\mathbf{D}^T\mathbf{e}_m$ can be treated as the output embedding for m-th state with the projection matrix as $\mathbf{M}_\theta^T\mathbf{D}^T$, noted as $\hat{\mathbf{x}}^{(m)} := \mathbf{M}_\theta^T\mathbf{D}^T\mathbf{e}_m$. Similar to (Tsai et al., 2020), $\mathbf{C}_m$ is a correction term for time lag effect. While there is no exact calculation for such correction term, under first order Markovian assumption, the transition probability between two states can be rewritten as a ansatz:

$$\mathbb{P}(\hat{x}_{t+1}=m|x_t=l) = \frac{\exp(\mathbf{x}_t^{(l)}\hat{\mathbf{x}}_{t+1}^{(m)})}{\sum_k \exp(\mathbf{x}_t^{(l)}\hat{\mathbf{x}}_{t+1}^{(k)})} \quad (11)$$

The kinetic distance in Eq. 12, or equivalently average commute time, can be measured as the inverse of interconversion probability, where Q_l stands for the Boltzmann distribution calculated for state l. In other words, the model embeddings hold information for kinetic distances. In the experiments section, we demonstrate that these embeddings contain sufficient information to accurately recover the final prediction, suggesting their importance in capturing dynamical relations.

$$t_{lm} = \frac{1}{Q_l * \mathbb{P}(\hat{x}_{t+1}=m|x_t=l) + Q_m * \mathbb{P}(\hat{x}_{t+1}=l|x_t=m)} \quad (12)$$

Table 1: Summary of experiment datasets. The alanine dipeptide are consist of (a)alanine$_\psi$ and alanine$_\phi$: Preprocessing of the MD simulation trajectories by projecting them onto 2 torsional angles: ϕ and ψ. (b)alanine$_{\text{RMSD}}$: Directly decompose MD simulation trajectories into states using the root mean square displacement (RMSD) distances without any pre-processing of the high-dimensional MD data.

Dataset	Data Size (m)	States Num	MFPT (ps) α_l to C_5	MFPT (ps) C_5 to α_l
4-state	1.6	4	-	-
alanine$_\psi$	1	20	153	89
alanine$_\phi$	1	20	984	30836
alanine$_{\text{RMSD}}$	1	100	5417	84830

3. Experiments

3.1. Datasets, Settings and Evaluation Metrics

Datasets We evaluate the performance of nano-GPT on both a 4-state model and alanine dipeptide systems. The potential for the 4-state model, which represents a system with four discrete states, is derived from (Tsai et al., 2020). The alanine dipeptide, a well-known molecule consisting of 22 atoms and 66 Cartesian coordinates, forms the basis of our more complex test cases. The dataset comprises 100 trajectories of the alanine dipeptide, each trajectory includes the molecule in conjunction with 888 water molecules. The positional data of these molecules are recorded every 0.1 ps over a total duration of 100 ns.

Specifically, three distinct datasets are simulated for the alanine dipeptide: alanine$_\phi$ simulated on phi coordinates; alanine$_\psi$ simulated on psi coordinates, and alanine$_{\text{RMSD}}$ based on RMSD distance. In alanine$_\phi$ and alanine$_\psi$, the original trajectories are projected onto the torsional angles where most degrees of freedom are related with fast dynamics (like vibration of chemical bonds). In alanine$_{\text{RMSD}}$, trajectories are directly decomposed into states using the root mean square displacement (RMSD) distances without any pre-processing of the high-dimensional MD data. These states are derived using a k-center clustering method to split the conformation space (Zhao et al., 2013), which approximates an ε-cover of samples (Sun et al., 2008; Yao et al., 2009; 2013) based on the RMSD distances of heavy (non-hydrogen) atoms.

Fig. 2 shows metastable states on the Ramachandran plot. In the context of alanineψ and alanineϕ, the ψ and ϕ angles serve as reaction coordinates for examining the dynamics projected onto them. Specifically, dynamics along the ϕ coordinate predominantly involve transitions between alpha-R and alpha-R / C7ex states, while the ψ coordinate primarily captures transitions between alpha-R and C7ex states.

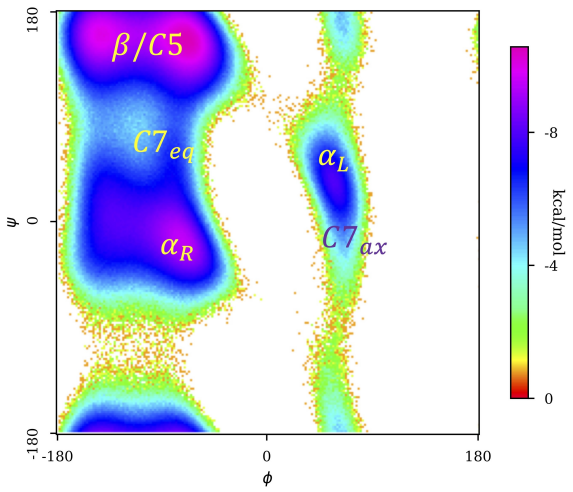

Figure 2: Ramachandran plot for alanine dipeptide (with ϕ on the horizontal axis and ψ on the vertical axis). There four metastable states located as follows: β (C7eq, top-left), alpha helix (α_R, left-center and slightly below), left-hand helix (α_L, right-center and slightly above), and C7ax (bottom-right).

Alanine$_{\text{RMSD}}$ characterizes the entire conformational space. These four slowest modes correspond to the following transitions: the transition from α_R to structures on the left side, the transition between α_L and C7eq, and the transitions between α_R and C7ax.

Table 1 summarizes all the datasets. The challenges escalate from alanine$_\psi$ to alanine$_\phi$, as the dynamics slow down and become more difficult to capture. This is evident in the increasing Mean First-Passage Time (MFPT). Moving from alanine$_\phi$ to alanine$_{\text{RMSD}}$, the dynamics become even slower, with the number of states increasing from 20 to 100. Consequently, the states are finer, resulting in a lower Boltzmann distribution of metastable states and making transitions between these states even rarer events.

Evaluation and Analysis We compare nano-GPT with the LSTM model used in (Tsai et al., 2020) for generation performance. During generation, both models recursively predict future values by appending them to the original sequence and shifting the old values. Two evaluation metrics, Implied Time Scales (ITS) and Mean First-Passage Time (MFPT), are employed. In the context of MSM, ITS is computed from the eigenvalues of the Markov model and serves as an estimate for the Markovian lag time, which reflects the order of magnitude of dynamics. The MFPT denotes the transition time between each pair of states.

3.2. Results

3.2.1. 4-STATE TEST SYSTEM

In this section, we demonstrate the ability of nano-GPT and its counterpart LSTM models to capture Boltzmann statistics. The projection from a high-dimensional space to one-dimensional data may inevitably impact kinetic properties. Nonetheless, our results indicate that nano-GPT can accurately capture kinetics, even in the presence of potential distortions in data quality.

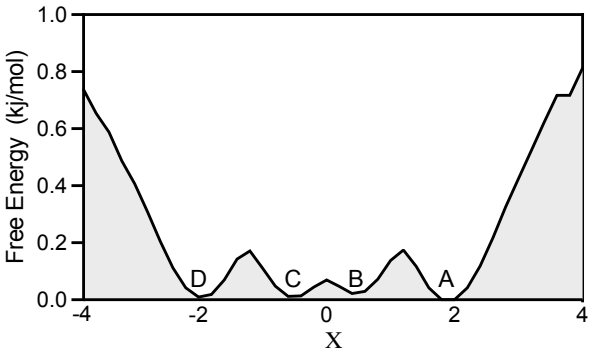

Figure 3: Free energy landscape of the 4-state system.

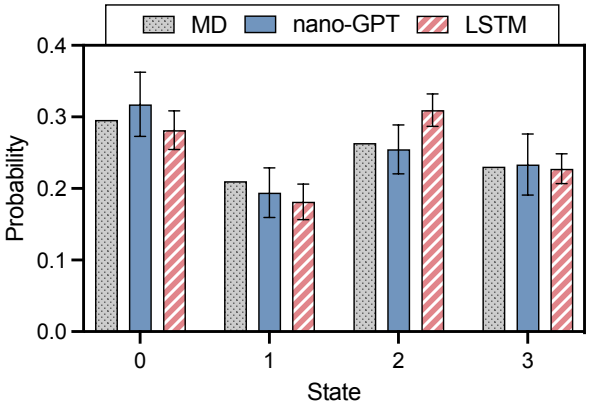

Figure 4: Botlzmann distribution for 4-state and reconstructions from nano-GPT and LSTM.

Fig. 3 illustrate the free energy landscape of the test system, highlighting several key features. Firstly, states B and C, which are kinetically proximal, exhibit relatively small kinetic distances. Secondly, significant energy barriers are observed between state pairs AD and state pairs BC. Our nano-GPT model successfully captures these characteristics, as demonstrated in the following figures. Fig. 4 represents the equilibrium distribution for all four states. This distribution is calculated based on population counts. Both nano-GPT and LSTM exhibit similar behavior with acceptable fluctuations.

3.2.2. ALANINE DIPEPTIDE

Our findings reveal varying degrees of difficulty in capturing dynamics. Dynamics in alanineψ prove to be the easiest to capture. In the case of alanineϕ, nano-GPT outperforms LSTM in terms of both free energy and ITS reconstruction. However, when it comes to alanine$_{\text{RMSD}}$, both models face challenges, with nano-GPT showing better performance than LSTM. Expanding the intervals from 0.1ps to 10ps improves LSTM's performance in this scenario.

Looking further into the scenario of alanine$_{\text{RMSD}}$, it becomes evident that deep learning models can sometimes be limited by the local information available to them, restricting their ability to capture global dynamics. At the end of this section, we provide an inside view of how information flows within nano-GPT to process local details. Results indicate that GPT utilizes information from both near and distant positions, and embedding plays a crucial role for the final prediction accuracy.

3.2.3. alanine$_\psi$: EASY TO CAPTURE

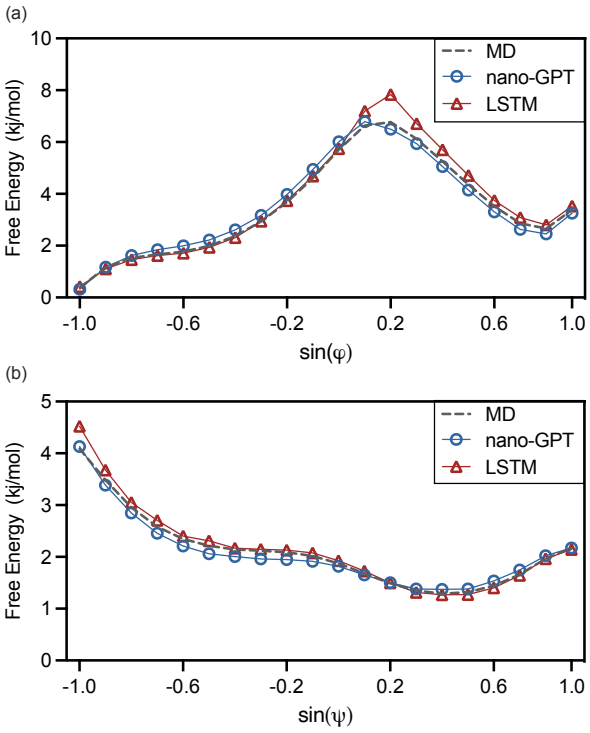

Figure 5: (a) Free energy for dataset alanine$_\psi$. (b) Free energy for dataset alanine$_\phi$.

For alanineψ, which exhibits relatively manageable dynamics across all three datasets, both nano-GPT and LSTM yield satisfactory results in terms of both thermodynamics and dynamics. The thermodynamic aspects, reflected in the free energy landscape in Fig. 5, are well-replicated by both nano-GPT and LSTM, accurately capturing the true curve and trend. Regarding dynamical behavior, both nano-GPT and LSTM exhibit consistency with the baseline, particularly in terms of Mean First-Passage Time (MFPT) in Table. 2. Even for the slowest mode, with a duration of 1265 ps, both models provide reliable predictions.

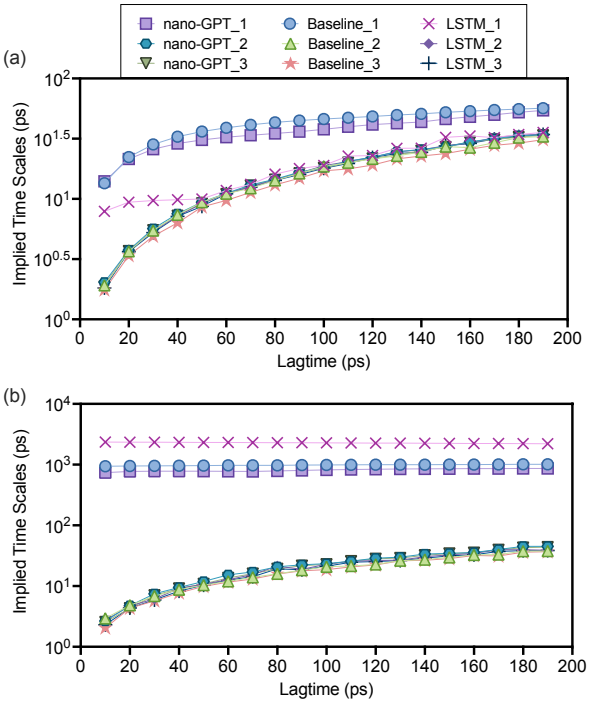

Figure 6: (a) ITS for dataset alanine$_\psi$. (b) ITS for dataset alanine$_\phi$.

3.2.4. alanine$_\phi$: SLOW DYNAMICS POSES CHALLENGES

In the case of alanine$_\phi$, transitions become more intricate, and nano-GPT closely matches the baseline performance in both the free energy landscape (Fig.5) and Mean First-Passage Time (MFPT) values (Table.3). The results on the simulated dataset highlight nano-GPT's superior capability in capturing rare transitions, as evidenced by the large MFPT values presented in the table. Findings from the results on alanine$_\phi$ are consistent with the conclusions drawn from the analysis of the simulated dataset.

Fig. 6 depicts the three slowest motions observed in the alanine dipeptide system, represented by the first ITS (red line), second ITS (blue line), and third ITS (green line). The dominant first ITS is well-replicated by nano-GPT, while LSTM's generation exhibits noticeable fluctuations. In the case of the second and third ITS, nano-GPT exhibits similar trends but with smoother transitions, effectively mitigating

Table 2: Comparison of MFPT on alanine$_\psi$ with 0.1ps interval. For both nano-GPT and LSTM, the first row represents the average value, while the second row represents the standard deviation (std) obtained from 5 different runs.

MFPT (ps)	α_r to α_l	α_l to α_r	α_l to β	β to α_l	α_l to C_5	C_5 to α_l	C7eq to C7ax	C7ax to C7eq
Baseline	378	102	153	89	153	89	1265	349
nano-GPT	335 (46)	106 (10)	152 (10)	92 (9)	152 (10)	92 (9)	1183 (174)	338 (21)
LSTM	369 (45)	89 (4)	158 (7)	84 (4)	158 (7)	84 (4)	1340 (287)	333 (10)

Table 3: Comparison of MFPT on alanine$_\phi$ with 0.1ps interval. For both nano-GPT and LSTM, the first row represents the average value, while the second row represents the standard deviation (std) obtained from 5 different runs.

MFPT (ps)	α_r to α_l	α_l to α_r	α_l to β	β to α_l	α_l to C_5	C_5 to α_l	C7eq to C7ax	C7ax to C7eq
Baseline	984	30836	1086	30837	984	30836	32206	984
nano-GPT	1226 (306)	34696 (1451)	1337 (304)	34696 (1451)	1226 (306)	34696 (1451)	35962 (1410)	1225 (308)
LSTM	1696 (673)	65335 (614)	1792 (683)	65334 (616)	1696 (673)	65335 (614)	66944 (1254)	1695 (674)

minor noise. Likewise, LSTM captures these ITS properties smoothly and closely.

3.2.5. alanine$_{\text{RMSD}}$: LONGER TIMESCALE LEARNED BY NANO-GPT

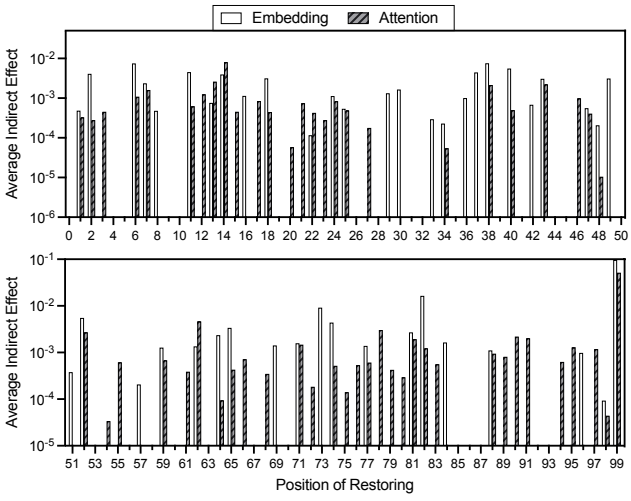

Figure 7: Logarithmized average indirect effect analysis conducted on 989 sequences, which are divided into segments of length 100 and ultimately leading to the final prediction of the α_L state.

In this section, we conducted an experiment using the conformational dynamics of alanine$_{\text{RMSD}}$ to evaluate the performance of LSTM and nano-GPT in reconstructing dynamics within a reduced dimension. The study places a particular focus on long-term behaviors, such as metastable transitions, which involve rare crossings between metastable states typically separated by high energy barriers. These transitions are infrequent and present a challenge in terms of capture, especially within the constraints of relatively short sequence lengths.

The one-dimensional dataset encapsulates all conformational changes in the alanine dynamics but introduces challenges due to information loss during projection. Nonetheless, the results indicate that with saving intervals of 0.1ps and 0.2ps, nano-GPT effectively learns the long-term dynamic behavior of the 100 states. In contrast, LSTM struggles to capture the slowest motion between crucial states, only achieving acceptable performance when the saving interval is extended to 1ps.

Nano-GPT excels in accurately capturing slow dynamics, as evidenced in Table. 4. This is particularly evident in the longer dynamics, such as transitions from α_l to α_r, β to α_l, and C_5 to α_l, which exhibit Mean First-Passage Times (MFPT) in the range of 80ns to 100ns. The dynamic information is effectively captured and can be accurately replicated in predicted trajectories with saving intervals of 0.1ps, 0.2ps, and 1ps.

In a comparative analysis, it becomes evident that under a shorter saving interval, such as 0.2 ps, LSTM struggles to produce accurate results when compared to the baseline simulation and nano-GPT predictions. However, when the saving interval is increased from 0.2ps to 1ps, more lo-

Table 4: Comparison of MFPT on alanine$_{\text{RMSD}}$ with different intervals (0.1ps, 0.2ps & 1ps). For both nano-GPT and LSTM, the first row represents the average value, while the second row represents the standard deviation (std) obtained from 5 different runs.

Interval	MFPT (ps)	α_r to α_l	α_l to α_r	α_l to β	β to α_l	α_l to C_5	C_5 to α_l
0.1ps	Baseline	1504	84921	1302	84836	5417	84830
	nano-GPT	622 (83)	61983 (9220)	910 (344)	62013 (9194)	5951 (52)	62013 (9185)
	LSTM	906 (335)	57100 (17232)	913 (339)	57118 (17239)	10307 (1326)	57121 (17242)
0.2ps	Baseline	1712	127178	1525	127088	9760	127078
	nano-GPT	741 (118)	110899 (28012)	496 (78)	110916 (23762)	4728 (808)	110919 (23762)
	LSTM	867 (161)	48422 (23167)	731 (174)	48438 (23129)	4968 (44)	48438 (23132)
1ps	Baseline	471	84903	482	84888	8722	84882
	nano-GPT	1772 (209)	82653 (5053)	1498 (78)	82615 (5058)	9430 (737)	82613 5058
	LSTM	1624 (236)	78421 (15961)	1489 (217)	78377 (15939)	9944 (242)	78370 (15942)

cal information is incorporated, aiding LSTM in capturing long-term behavior. In particular, for the rare transitions from metastable states to α_l, the learned Mean First-Passage Time (MFPT) improves, converging to values closer to those obtained in the baseline simulations.

Tracing information flow in nano-GPT.

In this section, we look into the internal mechanism of nano-GPT to highlight the informative key information stored within GPT. To pinpoint this essential information, we adopt the causal trace technique as outlined in (Meng et al., 2022). This approach helps identify which embedding/attention have a direct impact on the final results, shedding light on the model's decision-making process. The causal trace technique involves a three-step process where details can be found in the appendix.

The findings presented in Fig. 7 encompass the average indirect causal effect computed across 989 sequences, all culminating in the final prediction of the α_L state. The results suggest that, for the final prediction, embeddings hold significant importance in comparison to the attention mechanism. Notably, nano-GPT attributes nearly equal importance to embeddings in both near and distant positions. While attention is conventionally regarded as the primary mechanism for information retention, this conclusion underscores the crucial role of embeddings, particularly in encoding kinetic distances for metastable states. The noteworthy discovery, that the embedding layer plays a crucial role as the attention layers, aligns with the findings in Eq. 12.

4. Conclusions

Molecular dynamics trajectories are inherently sequential and benefit from autoregressive modeling, where predictions of current states are informed by past states. This characteristic facilitates the application of models such as GPT and LSTM. Previous research primarily utilized LSTM to investigate molecular dynamics on low-dimensional data. We have expanded these studies to higher-dimensional systems with increased noise using our nano-GPT models. Our findings indicate that nano-GPT exhibits superior performance in capturing extended long-term information within constrained sequences. Notably, it effectively utilizes local sequence information to think globally, managing to capture long-term information of 80,000 ps mean first passage time (MFPT) in sequences segmented into 100 ps intervals. This capability highlights nano-GPT's advanced potential in modeling complex molecular dynamics over significant timescales.

Acknowledgements

The authors gratefully acknowledge National Natural Science Foundation of China / Research Grants Council Joint Research Scheme Grant HKUST635/20 and Hong Kong Research Grant Council (HKRGC) Grant 16308321. This

research made use of the computing resources of the X-GPU cluster supported by the HKRGC Collaborative Research Fund C6021-19EF.

References

Bai, Q., Liu, S., Tian, Y., Xu, T., Banegas-Luna, A. J., Pérez-Sánchez, H., Huang, J., Liu, H., and Yao, X. Application advances of deep learning methods for de novo drug design and molecular dynamics simulation. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 12(3):e1581, 2022.

Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Bycroft, B. and Blechschmidt, I. llm viz, 12 2023. URL https://github.com/bbycroft/llm-viz.

Cao, Q., Ge, C., Wang, X., Harvey, P. J., Zhang, Z., Ma, Y., Wang, X., Jia, X., Mobli, M., Craik, D. J., et al. Designing antimicrobial peptides using deep learning and molecular dynamic simulations. *Briefings in Bioinformatics*, 24(2): bbad058, 2023.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Chodera, J. D., Singhal, N., Pande, V. S., Dill, K. A., and Swope, W. C. Automatic discovery of metastable states for the construction of markov models of macromolecular conformational dynamics. *The Journal of chemical physics*, 126(15):04B616, 2007.

Dullweber, A., Leimkuhler, B., and McLachlan, R. Symplectic splitting methods for rigid body molecular dynamics. *The Journal of chemical physics*, 107(15):5840–5851, 1997.

Eslamibidgoli, M. J., Mokhtari, M., and Eikerling, M. H. Recurrent neural network-based model for accelerated trajectory analysis in aimd simulations. *arXiv preprint arXiv:1909.10124*, 2019.

Gers, F. A., Schmidhuber, J., and Cummins, F. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Kadupitiya, J., Fox, G. C., and Jadhao, V. Deep learning based integrators for solving newton's equations with large timesteps. *arXiv preprint arXiv:2004.06493*, 2020.

Leimkuhler, B. J., Reich, S., and Skeel, R. D. Integration methods for molecular dynamics. In *Mathematical Approaches to biomolecular structure and dynamics*, pp. 161–185. Springer, 1996.

Liu, H., Feng, Y., Mao, Y., Zhou, D., Peng, J., and Liu, Q. Action-depedent control variates for policy optimization via stein's identity. *arXiv preprint arXiv:1710.11198*, 2017.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Liu, Y., Meng, F., Chen, Y., Xu, J., and Zhou, J. Scheduled sampling based on decoding steps for neural machine translation. *arXiv preprint arXiv:2108.12963*, 2021.

Lukoševičius, M. and Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

Mihaylova, T. and Martins, A. F. Scheduled sampling for transformers. *arXiv preprint arXiv:1906.07651*, 2019.

Noé, F. and Nuske, F. A variational approach to modeling slow processes in stochastic dynamical systems. *Multiscale Modeling & Simulation*, 11(2):635–655, 2013.

Pan, A. C. and Roux, B. Building markov state models along pathways to determine free energies and rates of transitions. *The Journal of chemical physics*, 129(6): 064107, 2008.

Pang, R. Y. and He, H. Text generation by learning from demonstrations. *arXiv preprint arXiv:2009.07839*, 2020.

Pathak, J., Hunt, B., Girvan, M., Lu, Z., and Ott, E. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical review letters*, 120(2):024102, 2018.

Pressé, S., Ghosh, K., Lee, J., and Dill, K. A. Principles of maximum entropy and maximum caliber in statistical physics. *Reviews of Modern Physics*, 85(3):1115, 2013.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Sun, J., Yao, Y., Huang, X., Pande, V., Carlsson, G., and Guibas, L. J. A well-controlled fast geometric clustering method on conformation space of biomolecules. In *Biomedical Computation at Stanford (BCATS)*, 2008.

Tsai, S.-T., Kuo, E.-J., and Tiwary, P. Learning molecular dynamics with simple language model built upon long short-term memory neural network. *Nature communications*, 11(1):1–11, 2020.

Tsai, S.-T., Fields, E., Xu, Y., Kuo, E.-J., and Tiwary, P. Path sampling of recurrent neural networks by incorporating known physics. *Nature Communications*, 13(1):7231, 2022.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

Yao, Y., Sun, J., Huang, X., Bowman, G., Singh, G., Lesnick, M., Pande, V., Guibas, L. J., and Carlsson, G. Topological methods for exploring low-density states in biomolecular folding pathways. *J. Chem. Phys.*, 130(14):144115, 2009.

Yao, Y., Cui, R. Z., Bowman, G. R., Silva, D. A., Sun, J., and Huang, X. Hierarchical nyström methods for constructing markov state models for conformational dynamics. *J. Chem. Phys.*, 138:174106, 2013.

Zhang, W., Feng, Y., Meng, F., You, D., and Liu, Q. Bridging the gap between training and inference for neural machine translation. *arXiv preprint arXiv:1906.02448*, 2019.

Zhao, Y., Sheong, F. K., Sun, J., Sander, P., and Huang, X. A fast parallel clustering algorithm for molecular simulation trajectories. *Journal of computational chemistry*, 34(2): 95–104, 2013.

A. Appendix

A.1. Literature Review

Natural Language Generation Natural human language is composed of sequential states that conform to a certain logic or rule, which may also be similar to predict molecular dynamics. In recent years, deep learning recurrent neural network methods such as gated recurrent unit (GRU) (Cho et al., 2014), long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) and their variants have shown great potential in processing sequential data (Lukoševičius & Jaeger, 2009), and there are now studies on using them to analyse trajectories from simulation systems (Eslamibidgoli et al., 2019) (Pathak et al., 2018).

Back to the application of recurrent neural network to molecular dynamics, related researches are still limited. A conservative approach is to incorporate LSTM into the numerical integrator that solves Newton's equations in molecular dynamics simulations (Kadupitiya et al., 2020). Another applies the recurrent neural network directly onto the low dimensional trajectories and predicts the next token in the sequential data (Tsai et al., 2020). They proved the training under cross-entropy loss is equivalent to learning a path entropy and captured both Boltzmann statistics and kinetics. In this work, the authors project their MD simulation trajectories onto a one-dimension reaction coordinate and further discretized the MD conformations by equal distance binning. Pre-processing of MD simulation trajectories to low dimension has been shown to render the LSTM model effective to learn the rare events. However, the applicability of the LSTM and other language models directly on high-dimensional data haven't been extensively examined.

GPT models. GPT models, as exemplified in studies by Radford et al. and Brown et al., have demonstrated significant advancements in various natural language processing (NLP) tasks, including reading comprehension, question answering, and textual entailment (Raffel et al., 2020; Liu et al., 2019; Yang et al., 2019). GPT models are explicitly designed to handle longer-term dependencies within sequences, a crucial attribute for modeling the intricate and time-evolving dynamics inherent to biomolecular systems. Also, GPT-based approaches have consistently achieved state-of-the-art results in numerous tasks, suggesting their potential applicability and success in the realm of biomolecular dynamics. To offer a visualization to the inside mechanism of GPT model, the github project (Bycroft & Blechschmidt, 2023) designs a nano-GPT model that is consisted of 3 transformer decoder stacks.

Exposure Bias. Exposure bias describes the situation that context words are selected from the ground truth sentence during the training phase, but from the last predicted sequence during inference. This inconsistency is known as the exposure bias problem as described by (Ranzato et al., 2015). Consider a scenario where a state A can transition to either state (B, B) or (C, C), resulting in two acceptable sequences: (A, B, B) and (A, C, C). Given (A) as the raw input, the model might predict 'C' as the second token since it is also a reasonable prediction. However, if (A, B, B) is the reference sequence in training, (A, C) will be corrected to (A, B) for the third token prediction. Given (A, B) as the raw input for the third token, the model will output 'B'. During the training phase, the model actually produces (A, C, B), representing an over-correction. Alternatively, during inference, (A, C) will not be corrected, allowing the model to produce a reasonable output such as (A, C, C).

Training Input: (A, B, B)

First Prediction starting with (A): $A \rightarrow C$

During Training: $A \rightarrow C(B) \rightarrow B$

During Inference: $A \rightarrow C \rightarrow C$

Approaches to mitigate exposure bias generally fall into two categories: sentence-level training and sampling-based methods. Sentence-level methods aim to directly maximize the reward of the generated sentence using a reinforcement learning framework. However, these techniques often encounter challenges, including unstable training and optimization issues. These difficulties arise due to the vast space of possible sentences and the high variance in policy gradients, as noted by Liu et al. and Pang & He. On the other hand, sampling-based methods seek to align the distribution of the training data more closely with generated sequence distribution. Bengio et al. introduced a strategy to substitute training tokens from model predictions, with a decaying probability of using ground truth data. Subsequent research has expanded upon this idea. For instance, Zhang et al. enhanced the pool of sampling candidates by integrating techniques like beam search and selection based on higher BLEU scores. Additionally, Mihaylova & Martins and Liu et al. adapted this sampling framework for use with the Transformer architecture, demonstrating its versatility across different model structures.

A.2. Causal Trace

We provide an analysis on nano-GPT of its interior mechanism in comprehending long MD trajectories. The challenge of long sequences stems from the dynamics staying in one state for extended duration, where LSTM may encounter memory leakage issues despite the presence of gating mechanisms. To address this enhanced ability of nano-GPT in dealing with long sequences, we employ causal trace technique (Meng et al., 2022) to investigate the information flow in nano-GPT.

(i) Initially, the model is run normally to obtain the predicted result, denoted as $\hat{x}_{t+1}$, with input X processed as depicted in Fig. 1.

(ii) In the next step, the embeddings $\mathbf{H}_i^{(0)}$ are corrupted by adding noise for all index i, specifically $\mathbf{H}_i^{(0)} + \varepsilon$, where ε follows a normal distribution $N(0, \sigma)$. This results in a corrupted prediction, denoted as $\hat{x}^*_{t+1}$.

(iii) To assess the causal effect, in the third step, at a given token i and layer l, the embeddings $\mathbf{H}_i^{(l)}$ are restored to their clean, noise-free state without ε. An important state should have the capability to recover $\hat{x}_{t+1}^{clean\mathbf{H}_i^{(l)}}$ instead of $\hat{x}^*_{t+1}$.

To quantify the causal effect, we define the Total Effect (TE) as $TE = \mathbb{P}(\hat{x}_{t+1}) - \mathbb{P}(\hat{x}^*_{t+1})$. Additionally, we use the concept of Indirect Effect (IE) to measure how $\mathbf{H}_i^{(l)}$ influences the prediction when compared to the corrupted version: $IE = \mathbb{P}(\hat{x}_{t+1}^{clean\mathbf{H}_i^{(l)}}) - \mathbb{P}(\hat{x}^*_{t+1})$. These measures allow us to assess the extent of influence and mediation that different components have on the final prediction.

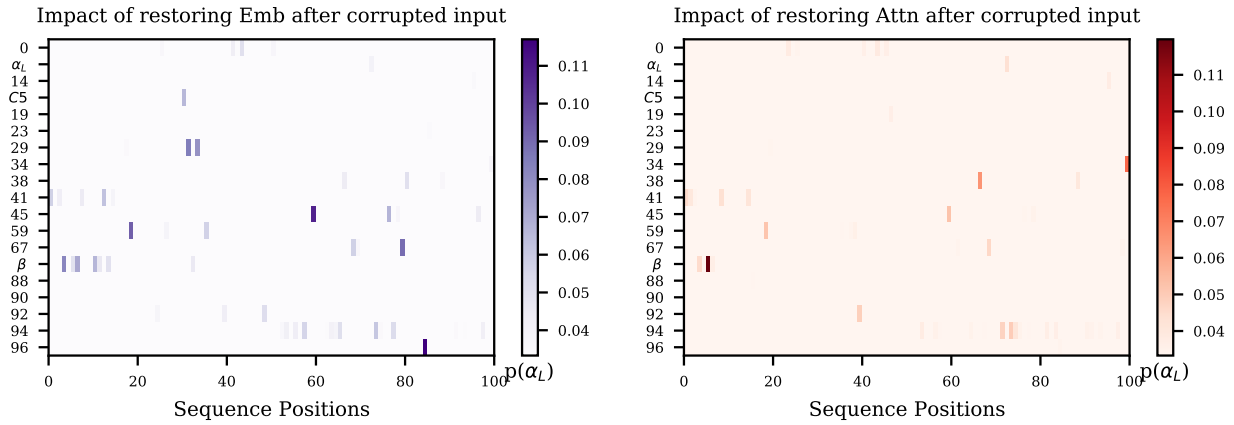

Figure 8: The causal effect analysis on single sequence example. (a) Impact of restoring **embeddings** after corruption in input data. (b) Impact of restoring **attentions** after corruption in input data. Both (a) and (b) experiments are conducted on a single sequence to predict α_L, where the x-axis represents the position in the sequence, and the y-axis denotes the input states, with important metastable states highlighted.

The results presented in Fig. 8 provide insights into the direct probabilities when restoring embeddings or attention layers on a single sample sequence. In this case, the target prediction is the α_l state, and it's expected that states in positions preceding it will have a significant impact on the final prediction. This is reasonable considering that the attention mechanism considers every position without suffering from memory loss. Interestingly, metastable state 87 (β state) is found to contribute significantly to enhancing the probability for accurate prediction. This observation underscores nano-GPT's ability to capture intrinsic transitions within the entire dynamic.

The results presented in Fig. 8 provide insights into the direct probabilities when restoring embeddings or attention layers on a single sample sequence. In this case, the target prediction is the α_l state, and it's expected that states in positions preceding it will have a significant impact on the final prediction. This is reasonable considering that the attention mechanism considers every position without suffering from memory loss. Interestingly, metastable state 87 (β state) is found to contribute significantly to enhancing the probability for accurate prediction. This observation underscores nano-GPT's ability to capture intrinsic transitions within the entire dynamic.

A.3. Further Experiments

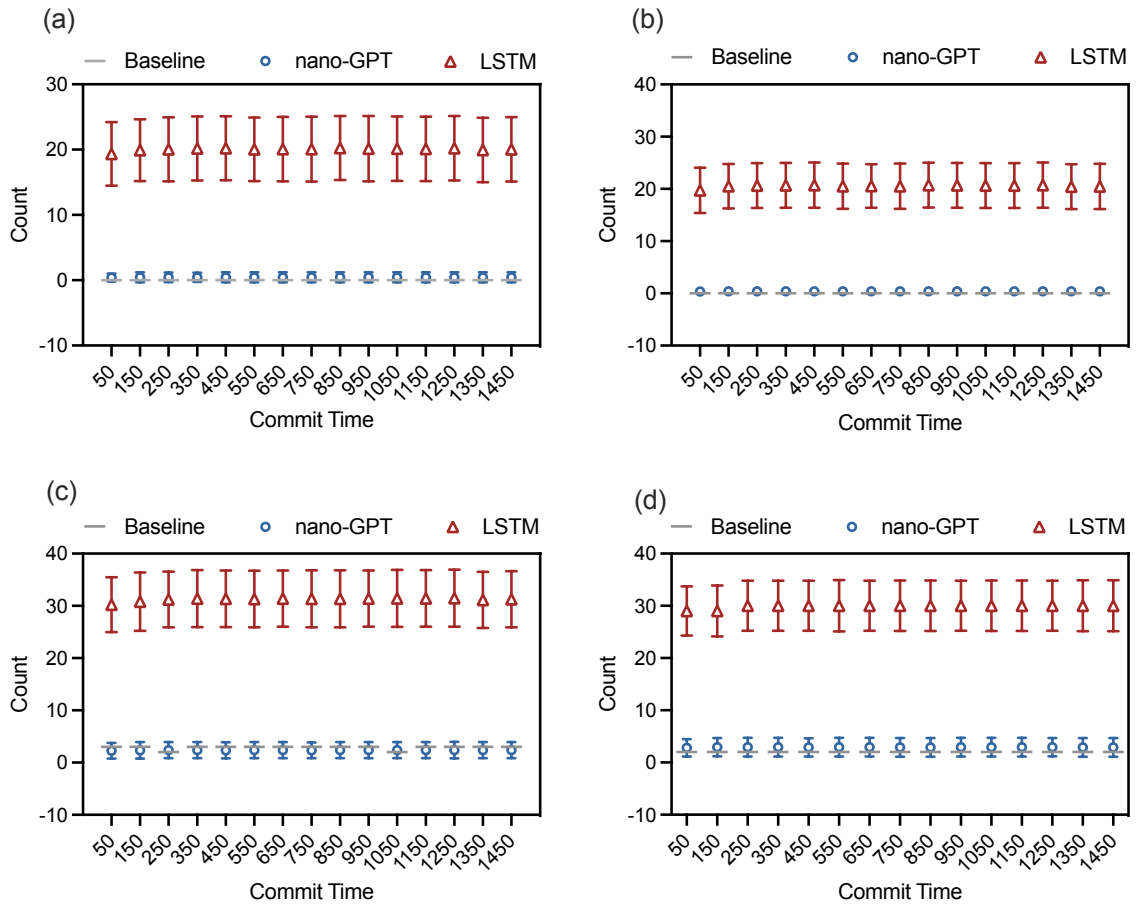

Figure 9: Transition counts between state AD and BC. The results are averaged on 50 runs. (a) Transition count from state A to state D. (b)Transition count from state D to state A. (c)Transition count from state B to state C. (d)Transition count from state C to state B.

To further illustrate the energy barrier between the AD and BC pair, Fig. 9 analyzes the transition count as a function of commit time, as referenced in (Tsai et al., 2020). States with significant energy barriers often involve rare events that can be overlooked by the model. Despite their infrequent occurrence, transitions across these barriers are of great interest. nano-GPT demonstrates a strong agreement between its predictions and the input data in this regard.