LEONARDO: A Physics-Informed Generative Model for Stochastic Nanoparticle Dynamics in Liquid-Phase TEM

Zain Shabeeb
Georgia Institute of Technology
Atlanta, GA
zshabeeb3@gatech.edu

Vida Jamali
Georgia Institute of Technology
Atlanta, GA
vida@gatech.edu

Abstract

Liquid-phase transmission electron microscopy (LPTEM) enables direct visualization of nanoparticle motion in the native liquid environment with nanometer and millisecond resolution. These combined capabilities open new opportunities for studying nanoscale dynamics, but also create a broad space of experimental choices where automation can play a critical role. Developing such automation requires realistic simulators of particle motion in LPTEM, yet quantitative interpretation and simulation of the resulting complex stochastic motion remain challenging due to the lack of tractable, physics-aware models. To address this, we introduce LEONARDO, a transformer-based variational autoencoder (VAE) with a physics-informed loss, trained on $\sim$38,000 experimental gold nanoparticle trajectories from LPTEM. The model encodes temporal dependencies of nanoparticle motion via self-attention and reconstructs trajectories by matching key statistical descriptors linked to physical phenomena, resulting in latent variables that capture experimental properties in an unsupervised way. To evaluate generative fidelity, we introduce Fréchet Motion Distance (FMD), an analogue of the Fréchet Inception Distance designed for stochastic time-series data. FMD measures the Fréchet distance between feature embeddings from MoNet2.0, a domain-specific CNN trained for anomalous diffusion classification. LEONARDO achieves an FMD of 7.8 with experimental trajectories, compared to 22.6-39.9 achieved by classical stochastic processes, and >95% of its generated trajectories are labeled "LPTEM" by a domain classifier. By functioning as a black-box simulator, LEONARDO generates realistic and diverse trajectories, providing a foundation for autonomous electron microscopy, where physically faithful synthetic data enable the development of data-driven control and analysis methods.

1 Introduction

Liquid-phase transmission electron microscopy (LPTEM) enables direct visualization of nanoparticles, including biomolecules such as proteins, in their native liquid environment [1, 2]. Unlike cryo-EM, which captures biomolecular conformations in a frozen state [3], LPTEM reveals real-time dynamics at nanometer and millisecond resolution [4, 5]. The access to nanoscale motion in liquid holds enormous potential for advancing our understanding of self-assembly [6, 7], catalysis [8], and biomolecular interactions [9, 10].

The opportunities to capture nanoscale dynamics with LPTEM also create a wide space of experimental choices. Expert microscopists adjust parameters such as electron beam dose to balance signal with radiation damage [11] while navigating the liquid cell, making trade-offs between signal quality and sample stability. These decisions are well suited to automation: data-driven control can increase

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI for Science.

throughput, improve reproducibility, and provide real-time feedback to guide experiments. In doing so, automation not only lowers the barrier to entry for new users but also frees researchers to focus more on scientific discovery than on instrument operation [12].

A critical step toward automating LPTEM experiments is the creation of simulation environments that allow data-driven control methods, such as reinforcement learning, to be developed and validated. Realistic simulators are essential in reducing the dependence on expensive and delicate experiments and providing a testbed for algorithm development. However, building such simulators is challenging because nanoparticle motion in LPTEM displays complex, non-Gaussian stochastic behavior that is not captured by classical diffusion processes [13, 14].

To address this challenge, we introduce LEONARDO, a transformer-based variational autoencoder (VAE) with a physics-informed loss, trained on experimental LPTEM trajectories of gold nanoparticles at varying electron beam dose rates and particle sizes. LEONARDO learns latent variables that encode experimental conditions and can generate trajectories faithful to experimental conditions. To evaluate generative fidelity, we introduce the Fréchet Motion Distance (FMD), an analogue of the Fréchet Inception Distance [15], designed for general stochastic time-series data via embeddings from a domain classifier. In this work we apply FMD to stochastic motion using the stochastic motion classifier MoNet2.0, an extension of MoNet [13] to 2D motion with seven diffusion classes. LEONARDO achieves significantly lower FMD (lower is better) than classical stochastic processes, and the classifier identifies >95% of the generated trajectories as "LPTEM".

By serving as a black-box simulator for LPTEM, LEONARDO establishes a foundation for autonomous electron microscopy. With access to realistic synthetic trajectories, researchers can design and train control strategies, optimize experimental parameters in silico, and accelerate discovery by shifting effort from microscope operation to interpretation of nanoscale dynamics.

2 Related Work

Classical stochastic processes such as Brownian motion (BM) [16], fractional Brownian motion (FBM) [17], and continuous-time random walks(CTRW) [18] have provided useful baselines for describing nanoparticle diffusion. However, trajectories observed in LPTEM often combine features of multiple processes and display non-Gaussian statistics that these models cannot capture [13, 19–22]. More general approaches, such as the generalized Langevin equation [23, 24], remain difficult to apply in heterogeneous experimental environments due to the need to assume specific forms of memory kernels and noise terms [25–28].

Machine learning approaches have recently been introduced as alternatives. Supervised networks trained on simulated processes can classify experimental trajectories into canonical diffusion types [13, 29–33], while unsupervised methods such as autoencoders [34] and VAEs [35] have been applied to model ideal stochastic dynamics. Yet these efforts rely primarily on synthetic training data, limiting their ability to reproduce the hybrid, non-Gaussian behavior found in experiments. Generative models trained directly on experimental trajectories, as pursued here, address this gap by producing realistic synthetic motion and enabling simulation environments for automation in electron microscopy [36].

3 LEONARDO Overview

Figure 1 illustrates the LEONARDO framework. The model is trained on experimental LPTEM trajectories, encoding them into a latent space and decoding them back into realistic synthetic trajectories using a physics-informed loss function. After training, LEONARDO can simulate nanoparticle motion under varying experimental conditions. Generated trajectories serve as input to downstream decision-making models, and latent variables can be adjusted to mimic changes in experimental parameters such as electron beam dose. In this way, LEONARDO functions both as a generative model of LPTEM dynamics and as a simulation environment for developing future automation strategies.

3.1 Problem Setup and Data

We first describe the dataset and formalize the problem setting that LEONARDO addresses. We collected a large dataset of single-particle trajectories of various lengths by tracking gold nanorods

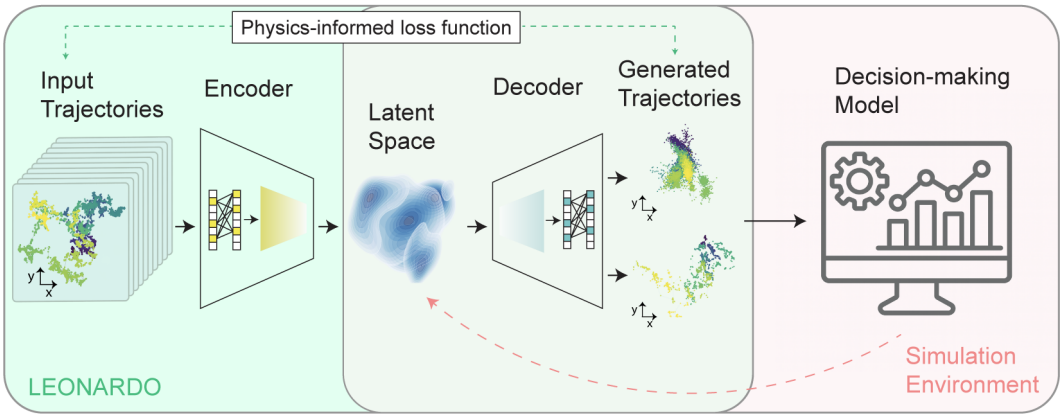

Figure 1: Overview of LEONARDO and its role in automated microscopy. LEONARDO is trained on experimental LPTEM trajectories, mapping input motion sequences through an encoder into a latent space, then decoding them into generated trajectories that reproduce nanoscale dynamics. These synthetic trajectories can be fed into decision-making models, creating a simulation environment for developing autonomous microscopy workflows.

diffusing in water and interacting with the membrane surface of the LPTEM microfluidic chamber. These raw trajectories were segmented into 200-frame sequences, yielding a dataset of 38,279 processed trajectories used for training, with additional held-out sets of 5,934 and 3,202 trajectories for validation and testing, respectively. Each trajectory was normalized prior to training (see Appendix A for more details).

Formally, let a trajectory be a sequence $\mathbf{r} = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_T\}$ of T frames (here $T = 200$), where each $\mathbf{r}_t = (x_t, y_t)$ is the 2D position of a nanoparticle at time step t. Our goal is to learn the data distribution $p(\mathbf{r})$ such that we can generate synthetic trajectories $\hat{\mathbf{r}}$ that resemble the experimentally observed ones.

3.2 Transformer–VAE Model

Here, we introduce the architecture of LEONARDO, which models LPTEM trajectories with a VAE. LEONARDO introduces a continuous latent vector $\mathbf{z} \in \mathbb{R}^d$ (here $d = 12$) and defines the generative process

$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad \mathbf{r} \sim p_\theta(\cdot \mid \mathbf{z}), \tag{1}$$

where $p_\theta(\mathbf{r} \mid \mathbf{z})$ is parameterized by a Transformer-based decoder. At train time, a variational encoder $q_\phi(\mathbf{z} \mid \mathbf{r})$ produces a Gaussian posterior with diagonal covariance,

$$q_\phi(\mathbf{z} \mid \mathbf{r}) = \mathcal{N}\big(\boldsymbol{\mu}, \operatorname{diag} \boldsymbol{\sigma}^2\big), \tag{2}$$

and samples are drawn via the reparameterization trick

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{3}$$

3.3 Model architecture

The encoder first maps raw coordinates into higher-dimensional embeddings with convolutional layers, then applies stacked self-attention blocks to capture temporal correlations across the trajectory. The output sequence is aggregated to obtain the mean and variance vectors that parameterize $q_\phi(\mathbf{z} \mid \mathbf{r})$.

The decoder begins from a latent code $\mathbf{z}$, maps it to a sequence representation, and uses Transformer blocks to enforce temporal coherence before projecting back to two spatial channels. A convolutional

layer at the output stage helps preserve temporal correlations in the generated trajectories. The complete model is detailed in Appendix B.

Once trained, LEONARDO serves as a generative model for LPTEM trajectories. To synthesize new samples, we first draw a latent code $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ from the prior distribution. The decoder then maps $\mathbf{z}$ into a temporally coherent sequence of spatial positions $\hat{\mathbf{r}}$, yielding synthetic trajectories of length $T = 200$ (a balanced choice to capture particle dynamics across varying video frame rates).

3.4 Physics-informed loss

Given the stochastic nature of nanoparticle trajectories, it is not meaningful to enforce exact frame-by-frame reconstruction after passing through a low-dimensional latent space, as previously noted in the literature [35]. Instead, the goal is to reconstruct the statistical properties that capture the underlying physics. To this end, LEONARDO introduces a physics-informed loss that supplements the standard VAE objective by aligning trajectory-level statistics between inputs and reconstructions, while down-weighting the contribution of the conventional MSE term.

Formally, for positions $\{\mathbf{r}_t\}_{t=1}^{T}$ with reconstruction $\{\hat{\mathbf{r}}_t\}_{t=1}^{T}$, let $\Delta\mathbf{r}_t = \mathbf{r}_t - \mathbf{r}_{t-1}$ and $\Delta\hat{\mathbf{r}}_t = \hat{\mathbf{r}}_t - \hat{\mathbf{r}}_{t-1}$. We evaluate a collection of trajectory statistics $\{\Phi_k(\cdot)\}_{k=1}^{K}$ on both sequences that are related to the underlying physics of the system, and penalize their discrepancies:

$$\mathcal{L}_{\text{phys}} = \sum_{k=1}^{K} w_k \left\| \Phi_k(\mathbf{r}) - \Phi_k(\hat{\mathbf{r}}) \right\|_2^2, \tag{4}$$

with per-term weights w_k (initialized from first-epoch magnitudes; see Appendix C). This yields gradients that steer the generator toward the correct physics without requiring pointwise alignment.

Statistical fingerprints of nanoparticle motion The statistics $\Phi_k(\cdot)_{k=1}^{K}$ used in Eq. (4) are defined as follows:

- **Displacement distribution:** match the first four moments of the step-size distribution $|\Delta\mathbf{r}|$ (mean, variance, skewness, kurtosis). This captures typical particle displacement scales and rare large displacements.
- **Velocity autocorrelation:** match the normalized $C_{\mathbf{v}}(\tau) = \frac{\langle \mathbf{v}(t) \cdot \mathbf{v}(t+\tau) \rangle}{\langle \mathbf{v}^2(t) \rangle}$ for small lags τ (weighted toward short lags), enforcing the negative VACF characteristic of viscoelastic "caging."
- **Positional autocorrelation:** match the normalized $C_{\mathbf{r}}(\tau) = \frac{\langle \mathbf{r}(t) \cdot \mathbf{r}(t+\tau) \rangle}{\langle \mathbf{r}^2(t) \rangle}$ over larger lags, capturing confinement and residence.
- **Motion anisotropy:** match the Pearson correlation between x and y components, $\rho_{\Delta x, \Delta y} = \text{corr}(\Delta x_t, \Delta y_t)$ which captures directional coupling of steps.
- **Median step size:** match the median of the step magnitude $\tilde{\boldsymbol{\Delta}}\mathbf{r}$, which provides stability under heavy tails.

Full objective. The physics term complements a lightly weighted reconstruction error and the standard Kulback-Leibler (KL) regularizer:

$$\mathcal{L} \;=\; \mathcal{L}_{\text{phys}} \;+\; \lambda_{\text{MSE}} \frac{1}{N} ||\mathbf{r} - \hat{\mathbf{r}}||_2^2 \;+\; \beta\, D_{\text{KL}}(q(\mathbf{z}|\mathbf{r}) || p(\mathbf{z})), \tag{5}$$

where N is the batch size. We keep λ_{MSE} small to avoid incentivizing pointwise matching for inherently stochastic signals. More details of the loss function and magnitudes of the weights of each loss term are provided in Appendix C.

4 Experiments

4.1 Baselines

We evaluate the fidelity of LEONARDO-generated trajectories against experimental LPTEM trajectories. As baselines, we include classical stochastic diffusion processes that have been widely used to

describe nanoparticle motion in liquids: BM, FBM, CTRW, annealed transient time model (ATTM), scaled Brownian motion (SBM), and Lévy walks (LW), because these processes capture different aspects of anomalous diffusion such as thermal noise, viscoelastic correlations, and trapping-and-escape events [13, 37–39]. For a detailed discussion of these stochastic processes, see [40].

For the baselines, trajectories are generated under standard parameterizations: for FBM, CTRW, ATTM, and SBM we sample the anomalous exponent $\alpha \sim \mathcal{U}(0.1, 1.0)$ (where α is the anomalous diffusion exponent that determines how the mean squared displacement scales with time, with $\alpha = 1$ corresponding to normal diffusion); for LW we sample $\alpha \sim \mathcal{U}(1.0, 2.0)$; and for BM we use the canonical generator without an anomalous exponent. Each baseline produces $N = 3{,}202$ trajectories of length $T = 200$, with preprocessing and normalization matched to the experimental data (see Appendix A). This setup enables direct comparison of how well LEONARDO and classical processes capture the statistical signatures of nanoparticle motion in LPTEM.

4.2 Evaluation Metrics

Fréchet Motion Distance (FMD) To quantify generative fidelity for stochastic time-series, we introduce FMD, an analogue of FID [15] that replaces the Inception embedding with features from a domain classifier. This formulation applies broadly to stochastic sequences where suitable embeddings exist (e.g., neural spike trains [41, 42], turbulent-flow trajectories/fields [43, 44], and molecular dynamics trajectories [45]). In our work, we instantiate FMD for nanoparticle motion by using embeddings from MoNet2.0, an extension of MoNet anomalous diffusion classifier [13] trained on a broad set of diffusion classes including BM, FBM, CTRW, LW, ATTM, SBM, and experimental LPTEM trajectories. This ensures that the embeddings capture dynamics-relevant features across a wide spectrum of anomalous diffusion behaviors. Unlike direct comparisons on hand-crafted statistics, FMD yields a single distribution-level score that integrates higher-order and temporal signatures learned from data, providing a standardized metric for evaluating generative models of time-series stochastic trajectories.

Given two sets of trajectories, experimental ($\mathcal{D}_{\text{exp}}$) and generated ($\mathcal{D}_{\text{gen}}$), we compute their embeddings through the pretrained classifier MoNet2.0. Let $d_i \in \mathcal{D}_{\text{exp}}$ and $\tilde{d}_j \in \mathcal{D}_{\text{gen}}$ denote individual trajectories, with M_{exp} and M_{gen} the numbers of trajectories in the experimental and generated sets, respectively. Applying the embedding map $f(\cdot)$ (MoNet2.0) to each trajectory yields

$$\{f(d_i)\}_{i=1}^{M_{\text{exp}}}, \qquad \{f(\tilde{d}_j)\}_{j=1}^{M_{\text{gen}}}.$$

We denote these embedding sets as $f(\mathcal{D}_{\text{exp}}) = \{f(d_i)\}_{i=1}^{M_{\text{exp}}}$ and $f(\mathcal{D}_{\text{gen}}) = \{f(\tilde{d}_j)\}_{j=1}^{M_{\text{gen}}}$. Each set of embeddings is then modeled as a multivariate Gaussian,

$$f(\mathcal{D}_{\text{exp}}) \sim \mathcal{N}(\mu_{\text{exp}}, \Sigma_{\text{exp}}), \qquad f(\mathcal{D}_{\text{gen}}) \sim \mathcal{N}(\mu_{\text{gen}}, \Sigma_{\text{gen}}),$$

where μ_{exp} and Σ_{exp} denote the empirical mean and covariance of the experimental embeddings, and μ_{gen} and Σ_{gen} denote the empirical mean and covariance of the generated embeddings.

The FMD is then defined as

$$\text{FMD}^2 = \|\mu_{\text{exp}} - \mu_{\text{gen}}\|_2^2 + \text{Tr}\Big(\Sigma_{\text{exp}} + \Sigma_{\text{gen}} - 2(\Sigma_{\text{exp}}^{1/2}\Sigma_{\text{gen}}\Sigma_{\text{exp}}^{1/2})^{1/2}\Big).$$

A lower FMD indicates closer alignment between the distribution of generated and experimental trajectories in the learned feature space, corresponding to better generative fidelity.

Classifier-based Evaluation As a complementary measure, we also examine the final prediction layer of MoNet2.0 to quantify the fraction of LEONARDO-generated trajectories that MoNet2.0 classifies as LPTEM. A higher fraction indicates that generated trajectories more closely resemble experimental LPTEM dynamics, whereas a lower fraction suggests systematic differences between synthetic and real data.

While FMD measures statistical similarity in the feature distribution, the classifier-based evaluation measures distinguishability on a sample-by-sample basis. Taken together, these metrics provide a rigorous and interpretable framework for evaluating generative models of nanoparticle diffusion.

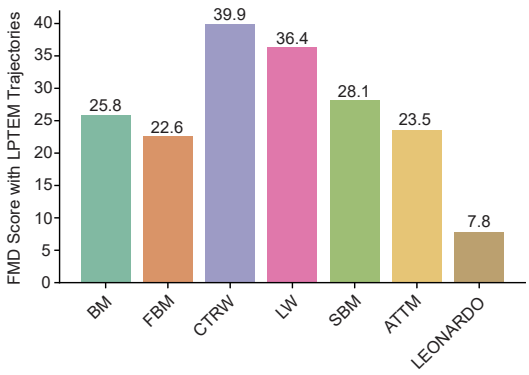

Figure 2: **Fidelity via Fréchet Motion Distance (FMD).** Mean FMD scores of experimental LPTEM trajectories with (1) baseline stochastic processes (BM, FBM, CTRW, ATTM, SBM, LW), and (2) trajectories generated by LEONARDO. Each bar shows the mean across 10 random seeds (lower is better). LEONARDO achieves the lowest FMD compared to all baselines.

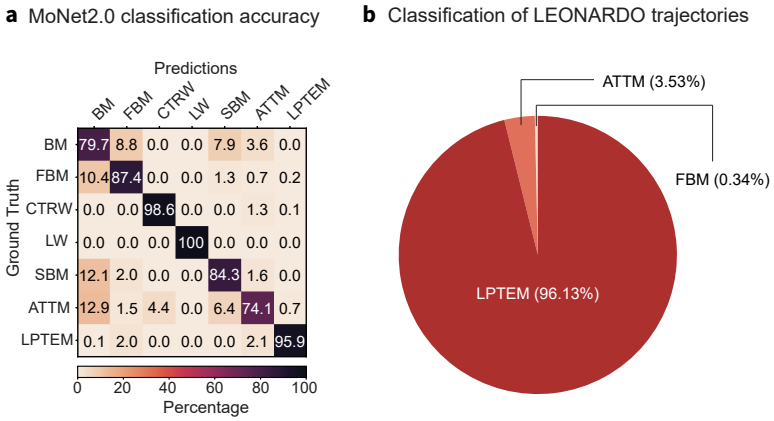

Figure 3: **Classifier-based evaluation of fidelity. (a)** Confusion matrix showing the performance of MoNet2.0 on a held-out test set of stochastic processes and LPTEM trajectories, achieving an overall F1-score of 0.88. **(b)** Classification of LEONARDO-generated trajectories by MoNet2.0. The vast majority (96.13%) are identified as "LPTEM", confirming strong fidelity to LPTEM trajectories.

4.3 FMD and Classifier Consistency

Figure 2 reports FMD scores of experimental LPTEM trajectories with 1) baseline stochastic processes, and 2) trajectories generated by LEONARDO. For each model, we performed 10 runs with different random seeds and report the mean value across runs. Standard deviations were consistently very small, confirming the stability of the estimates. LEONARDO achieves the lowest FMD (7.8), substantially outperforming all baselines. For comparison, baselines yield 25.8 for BM, 22.6 for FBM, 39.9 for CTRW, 23.5 for ATTM, 28.1 for SBM, and 36.4 for LW. For additional context, Appendix D reports a lower-triangular matrix of FMD scores between classes of stochastic processes (including intra-class comparisons). These intra-class values, which range from 0.19 to 1.74, serve as a lower bound when interpreting the scores in Figure 2.

We also assess how trajectories are classified by MoNet2.0 trained on experimental LPTEM trajectories and stochastic processes. Figure 3 summarizes these results. First, the classifier achieves strong performance on a held-out test set of stochastic processes, with an overall F1-score (a metric that balances precision and recall) of 0.88 (Fig. 3a). Next, when applied to LEONARDO-generated trajectories, MoNet2.0 identifies 96.13% of them as "LPTEM" (Fig. 3b). A large fraction of LEONARDO-generated trajectories are classified by MoNet2.0 as LPTEM, indicating close alignment with experimental data.

Table 1: **Ablation of physics-informed candidates.** Δ values are relative to the core configuration. Negative ΔFMD and positive Δ %LPTEM indicate improvement.

Added term (one-at-a-time)	ΔFMD $\downarrow$	Δ %LPTEM $\uparrow$
Median displacement	-0.058	$+4.39$
Positional autocorrelation $C_{\mathbf{r}}(\tau)$	-0.196	$+16.57$
DFT moments of displacements	$+0.027$	-0.61
IPSD moments	$+0.122$	-3.01
DMSD moments	$+0.036$	-21.06
Morlet wavelet moments	-0.134	-10.45

4.4 Ablation of Loss Function Terms

LEONARDO's physics-informed loss is built around a set of trajectory-level statistics, but many more descriptors of anomalous diffusion exist. To determine which terms should be included beyond our core configuration, we systematically evaluated a broad candidate pool and measured their impact on generative fidelity.

Core configuration Our baseline objective includes three main components: (i) the first four moments of the step-size distribution $|\Delta\mathbf{r}|$ (mean, variance, skewness, kurtosis), (ii) single-trajectory and batch velocity autocorrelations $C_{\mathbf{v}}(\tau)$, and (iii) motion anisotropy. A lightly weighted MSE and the standard VAE KL term are also retained but down-weighted relative to the physics-informed terms.

Candidate pool We evaluated additional statistical descriptors drawn from a large feature library for anomalous diffusion [46], including median displacement, lagged displacement moments, ratios of consecutive steps, spectral features from discrete Fourier transforms (DFT) and integrated power spectral density (IPSD), multiscale descriptors such as DMSD and Morlet wavelet coefficients, and global measures like integral autocorrelation and frequency–time asymmetry. Approximate entropy (AppEn) was excluded due to non-differentiability. Before running ablations, we screened which features were already implicitly captured by the core model by tracking their discrepancies over training checkpoints; candidates that consistently decreased without explicit addition to the loss function were not included in further experiments.

Protocol and results Each surviving candidate was added individually to the core objective, with weights initialized from their first-epoch magnitudes for consistency with the main training setup. We then retrained LEONARDO and evaluated performance using FMD (lower is better), and the percentage of generated trajectories classified as LPTEM by MoNet2.0 (higher is better). As shown in Table 1, two descriptors improved both metrics: median displacement, which provides a robust measure of scale under heavy-tailed statistics, and positional autocorrelation, which captures long-lived spatial memory. By contrast, several spectral or multiscale terms worsened performance on at least one metric, suggesting that they emphasize high-frequency content at the expense of the statistics most relevant to LPTEM fidelity.

4.5 LEONARDO latent variable reflects electron beam dose

Because LEONARDO is trained on trajectories acquired under varying experimental conditions, its latent space naturally learns to represent underlying factors of variation in the data. To analyze this structure, we examined individual latent variables and found that one variable, z_4, correlates strongly with the experimental electron beam dose of the microscope (Pearson $r = -0.61$, $p = 2.21 \times 10^{-44}$; Fig. 4a). This indicates that the model captures dose-dependent changes in nanoparticle dynamics without being provided explicit dose labels.

The correlation between z_4 and dose means that this latent variable can act as a proxy for simulating trajectories at different electron doses. By fixing all other latent variables and varying z_4, the generated trajectories exhibit systematic changes consistent with dose-dependent dynamics: lower values produce confined motion with strong caging, while higher values yield more frequent escapes

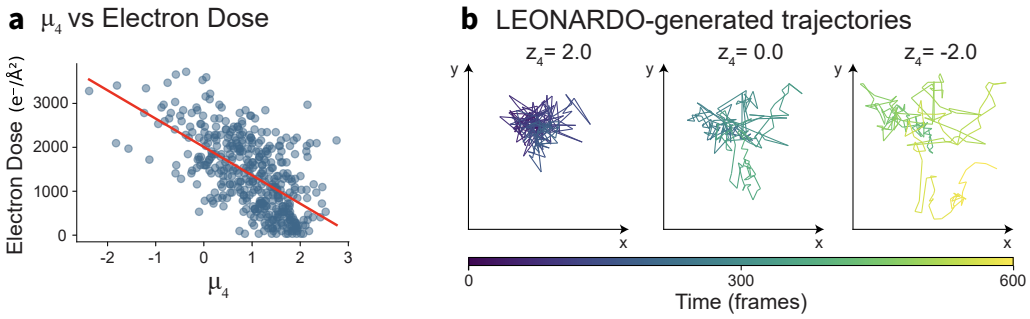

Figure 4: **Latent variable reflects electron beam dose. (a)** Scatter plot showing the correlation between latent μ_4 and experimental electron dose. **(b)** Example LEONARDO-generated trajectories obtained by fixing all other latent variables and varying z_4 from high (left) to low (right). Trajectories shift from confined, caged motion at high z_4 to more frequent escapes at low z_4, reproducing experimental dose-dependent trends.

(Fig. 4b). These trends mirror experimental observations, showing that traversal along z_4 effectively functions as a simulator of nanoparticle motion across electron beam doses.

This controllability makes z_4 an interpretable "dose axis" in the latent space. Beyond offering insight into how electron beam dose influences nanoparticle dynamics, it provides a practical handle for generating synthetic trajectories across various doses. This capability allows models to explore a range of experimental conditions in silico, avoiding the need to perform new, costly LPTEM experiments for each setting.

5 Conclusion and Future Directions

We introduced LEONARDO, a transformer-based variational autoencoder with a physics-informed loss, trained directly on experimental LPTEM trajectories. By combining temporal self-attention with trajectory-level statistical constraints, the model generates synthetic trajectories that faithfully reproduce the stochastic features of experimental data. Evaluation with the FMD and domain classification confirmed that LEONARDO captures experimental dynamics more accurately than classical stochastic processes. Beyond generative fidelity, we showed that its latent space reflects meaningful experimental factors by aligning with electron beam dose, enabling controllable simulation of dose-dependent nanoparticle dynamics.

While electron beam dose is a critical imaging parameter, it is only one of many variables that influence particle motion in liquids. Temperature, liquid chemistry, and particle–particle interactions all shape nanoscale trajectories and present opportunities for deeper modeling. Extending LEONARDO to datasets acquired under systematically varied conditions will allow its latent variables to represent these experimental factors, while incorporating additional physics-informed losses tailored to each mechanism. Such developments would push LEONARDO toward a general-purpose simulator of nanoparticle dynamics in complex environments.

Looking forward, the ability to generate realistic trajectories across a range of conditions offers a pathway to simulation-driven automation. By replacing costly and delicate experiments with in silico exploration, models like LEONARDO can provide the synthetic data needed to train control algorithms, optimize imaging parameters, and accelerate scientific discovery with autonomous electron microscopy.

Acknowledgements

This research was supported by the NSF, Division of Chemical, Bioengineering, Environmental, and Transport Systems under award 2338466.

References

[1] Niels de Jonge and Frances M. Ross. Electron microscopy of specimens in liquid. *Nature Nanotechnology*, 6:695–704, 11 2011. ISSN 1748-3387. doi: 10.1038/nnano.2011.161.

[2] Ivan A Moreno-Hernandez, Michelle F Crook, Vida Jamali, and A Paul Alivisatos. Recent advances in the study of colloidal nanocrystals enabled by in situ liquid-phase transmission electron microscopy. *MRS Bulletin*, 47(3):305–313, 2022.

[3] Cryo-electron microscopy – a primer for the non-microscopist. *The FEBS Journal*, 280:28–45, 1 2013. ISSN 1742-464X. doi: 10.1111/febs.12078.

[4] Frances M. Ross. Opportunities and challenges in liquid cell electron microscopy. *Science*, 350, 12 2015. ISSN 0036-8075. doi: 10.1126/science.aaa9886.

[5] Niels de Jonge, Lothar Houben, Rafal E Dunin-Borkowski, and Frances M Ross. Resolution and aberration correction in liquid cell transmission electron microscopy. *Nature Reviews Materials*, 4(1):61–78, 2019.

[6] Zihao Ou, Ziwei Wang, Binbin Luo, Erik Luijten, and Qian Chen. Kinetic pathways of crystallization at the nanoscale. *Nature materials*, 19(4):450–455, 2020.

[7] Hoduk Cho, Ivan A Moreno-Hernandez, Vida Jamali, Myoung Hwan Oh, and A Paul Alivisatos. In situ quantification of interactions between charged nanorods in a predefined potential energy landscape. *Nano Letters*, 21(1):628–633, 2020.

[8] Jiangshan Qu, Manling Sui, and Rengui Li. Recent advances in in-situ transmission electron microscopy techniques for heterogeneous catalysis. *iScience*, 26:107072, 7 2023. ISSN 25890042. doi: 10.1016/j.isci.2023.107072.

[9] Jia ye Li, He Sun, and Huan Wang. Imaging biomacromolecules in action with liquid-phase electron microscopy. *Trends in Chemistry*, 6:281–284, 6 2024. ISSN 25895974. doi: 10.1016/j.trechm.2024.04.004.

[10] John W Smith, Lauren N Carnevale, Aditi Das, and Qian Chen. Electron videography of a lipid–protein tango. *Science Advances*, 10(16):eadk0217, 2024.

[11] Karthik Gnanasekaran, Nathan D. Rosenmann, Roberto dos Reis, and Nathan C. Gianneschi. Extent of radiolytic damage from liquid cell tem experiments on metal–organic frameworks via post-mortem 4d-stem. *Nano Letters*, 24:10161–10168, 8 2024. ISSN 1530-6984. doi: 10.1021/acs.nanolett.4c02242.

[12] Steven R Spurgeon, Colin Ophus, Lewys Jones, Amanda Petford-Long, Sergei V Kalinin, Matthew J Olszta, Rafal E Dunin-Borkowski, Norman Salmon, Khalid Hattar, Wei-Chang D Yang, et al. Towards data-driven next-generation transmission electron microscopy. *Nature materials*, 20(3):274–279, 2021.

[13] Vida Jamali, Cory Hargus, Assaf Ben-Moshe, Amirali Aghazadeh, Hyun Dong Ha, Kranthi K. Mandadapu, and A. Paul Alivisatos. Anomalous nanoparticle surface diffusion in lctem is revealed by deep learning-assisted analysis. *Proceedings of the National Academy of Sciences*, 118, 3 2021. ISSN 0027-8424. doi: 10.1073/pnas.2017616118.

[14] Isabel Panicker, Zain Shabeeb, Cory Hargus, and Vida Jamali. Modulating nanoparticle-surface interactions at liquid-solid interfaces via ion screening in liquid-phase tem. *ChemRxiv*, 2025. doi: 10.26434/chemrxiv-2025-60c5k.

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. 6 2017.

[16] Albert Einstein. On the movement of small particles suspended in a stationary liquids required by the molecular-kinetic theory of heat. *Annalen der physik*, 17:549–560, 1905.

[17] Benoit B. Mandelbrot and John W. Van Ness. Fractional Brownian Motions, Fractional Noises and Applications. *SIAM Review*, 10(4):422–437, 10 1968. ISSN 0036-1445. doi: 10.1137/1010093.

[18] Harvey Scher and Elliott W. Montroll. Anomalous transit-time dispersion in amorphous solids. *Physical Review B*, 12(6):2455–2477, 9 1975. ISSN 0556-2805. doi: 10.1103/PhysRevB.12.2455.

[19] Aubrey V. Weigel, Blair Simon, Michael M. Tamkun, and Diego Krapf. Ergodic and nonergodic processes coexist in the plasma membrane as observed by single-molecule tracking. *Proceedings of the National Academy of Sciences*, 108(16):6438–6443, 4 2011. ISSN 0027-8424. doi: 10.1073/pnas.1016325108.

[20] Raphaël Sarfati and Daniel K. Schwartz. Temporally Anticorrelated Subdiffusion in Water Nanofilms on Silica Suggests Near-Surface Viscoelasticity. *ACS Nano*, 14(3):3041–3047, 3 2020. ISSN 1936-0851. doi: 10.1021/acsnano.9b07910.

[21] Bo Wang, Stephen M. Anthony, Sung Chul Bae, and Steve Granick. Anomalous yet Brownian. *Proceedings of the National Academy of Sciences*, 106(36):15160–15164, 9 2009. ISSN 0027-8424. doi: 10.1073/pnas.0903554106.

[22] Bo Wang, James Kuo, Sung Chul Bae, and Steve Granick. When Brownian diffusion is not Gaussian. *Nature Materials*, 11(6):481–485, 6 2012. ISSN 1476-1122. doi: 10.1038/nmat3308.

[23] Robert Zwanzig. *Nonequilibrium Statistical Mechanics*. Oxford University Press, 2001.

[24] R Kubo. The fluctuation-dissipation theorem. *Reports on Progress in Physics*, 29(1):306, 1 1966. ISSN 00344885. doi: 10.1088/0034-4885/29/1/306.

[25] Huan Lei, Nathan A. Baker, and Xiantao Li. Data-driven parameterization of the generalized Langevin equation. *Proceedings of the National Academy of Sciences*, 113(50):14183–14188, 12 2016. ISSN 0027-8424. doi: 10.1073/pnas.1609587113.

[26] Max Berkowitz, John D. Morgan, Donald J. Kouri, and J. Andrew McCammon. Memory kernels from molecular dynamics. *The Journal of Chemical Physics*, 75(5):2462–2463, 9 1981. ISSN 0021-9606. doi: 10.1063/1.442269.

[27] John Fricks, Lingxing Yao, Timothy C. Elston, and M. Gregory Forest. Time-Domain Methods for Diffusive Transport in Soft Matter. *SIAM Journal on Applied Mathematics*, 69(5):1277–1308, 1 2009. ISSN 0036-1399. doi: 10.1137/070695186.

[28] Minxin Chen, Xiantao Li, and Chun Liu. Computation of the memory functions in the generalized Langevin models for collective dynamics of macromolecules. *The Journal of Chemical Physics*, 141(6), 8 2014. ISSN 0021-9606. doi: 10.1063/1.4892412.

[29] Naor Granik, Lucien E. Weiss, Elias Nehme, Maayan Levin, Michael Chein, Eran Perlson, Yael Roichman, and Yoav Shechtman. Single-Particle Diffusion Characterization by Deep Learning. *Biophysical Journal*, 117(2):185–192, 7 2019. ISSN 00063495. doi: 10.1016/j.bpj.2019.06.015.

[30] Gorka Muñoz-Gil, Miguel Angel Garcia-March, Carlo Manzo, José D Martín-Guerrero, and Maciej Lewenstein. Single trajectory characterization via machine learning. *New Journal of Physics*, 22(1):013010, 1 2020. ISSN 1367-2630. doi: 10.1088/1367-2630/ab6065.

[31] Patrycja Kowalek, Hanna Loch-Olszewska, and Janusz Szwabiński. Classification of diffusion modes in single-particle tracking data: Feature-based versus deep-learning approach. *Physical Review E*, 100(3):032410, 9 2019. ISSN 2470-0045. doi: 10.1103/PhysRevE.100.032410.

[32] Stefano Bo, Falko Schmidt, Ralf Eichhorn, and Giovanni Volpe. Measurement of anomalous diffusion using recurrent neural networks. *Physical Review E*, 100(1):010102, 2019.

[33] Borja Requena, Sergi Masó-Orriols, Joan Bertran, Maciej Lewenstein, Carlo Manzo, and Gorka Muñoz-Gil. Inferring pointwise diffusion properties of single trajectories with deep learning. *Biophysical Journal*, 122(22):4360–4369, 11 2023. ISSN 00063495. doi: 10.1016/j.bpj.2023.10.015.

[34] Gorka Muñoz-Gil, Guillem Guigo i Corominas, and Maciej Lewenstein. Unsupervised learning of anomalous diffusion data: an anomaly detection approach. *Journal of Physics A: Mathematical and Theoretical*, 54(50):504001, 12 2021. ISSN 1751-8113. doi: 10.1088/1751-8121/ac3786.

[35] Gabriel Fernández-Fernández, Carlo Manzo, Maciej Lewenstein, Alexandre Dauphin, and Gorka Muñoz-Gil. Learning minimal representations of stochastic processes with variational autoencoders. *Physical Review E*, 110:L012102, 7 2024. ISSN 2470-0045. doi: 10.1103/PhysRevE.110.L012102.

[36] Steven R. Spurgeon, Colin Ophus, Lewys Jones, Amanda Petford-Long, Sergei V. Kalinin, Matthew J. Olszta, Rafal E. Dunin-Borkowski, Norman Salmon, Khalid Hattar, Wei-Chang D. Yang, Renu Sharma, Yingge Du, Ann Chiaramonti, Haimei Zheng, Edgar C. Buck, Libor Kovarik, R. Lee Penn, Dongsheng Li, Xin Zhang, Mitsuhiro Murayama, and Mitra L. Taheri. Towards data-driven next-generation transmission electron microscopy. *Nature Materials*, 20 (3):274–279, 3 2021. ISSN 1476-1122. doi: 10.1038/s41563-020-00833-z.

[37] See Wee Chee, Utkarsh Anand, Geeta Bisht, Shu Fen Tan, and Utkur Mirsaidov. Direct Observations of the Rotation and Translation of Anisotropic Nanoparticles Adsorbed at a Liquid–Solid Interface. *Nano Letters*, 19(5):2871–2878, 5 2019. ISSN 1530-6984. doi: 10.1021/acs.nanolett.8b04962.

[38] Taylor J. Woehl and Tanya Prozorov. The Mechanisms for Nanoparticle Surface Diffusion and Chain Self-Assembly Determined from Real-Time Nanoscale Kinetics in Liquid. *The Journal of Physical Chemistry C*, 119(36):21261–21269, 9 2015. ISSN 1932-7447. doi: 10.1021/acs.jpcc.5b07164.

[39] Evangelos Bakalis, Lucas R. Parent, Maria Vratsanos, Chiwoo Park, Nathan C. Gianneschi, and Francesco Zerbetto. Complex Nanoparticle Diffusional Motion in Liquid-Cell Transmission Electron Microscopy. *The Journal of Physical Chemistry C*, 124(27):14881–14890, 7 2020. ISSN 1932-7447. doi: 10.1021/acs.jpcc.0c03203.

[40] Zain Shabeeb, Naisargi Goyal, Pagnaa Attah Nantogmah, and Vida Jamali. Learning the diffusion of nanoparticles in liquid phase tem via physics-informed generative ai. *Nature Communications*, 16:6298, 7 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-61632-1.

[41] David Sussillo, Rafal Jozefowicz, L. F. Abbott, and Chethan Pandarinath. Lfads - latent factor analysis via dynamical systems, 2016. URL `https://arxiv.org/abs/1608.06315`.

[42] Manuel Molano-Mazon, Arno Onken, Eugenio Piasini*, and Stefano Panzeri*. Synthesizing realistic neural population activity patterns using generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=r1VVsebAZ`.

[43] Ludovico Nista, Heinz Pitsch, Christoph D. K. Schumann, Mathis Bode, Temistocle Grenga, Jonathan F. MacArt, and Antonio Attili. Influence of adversarial training on super-resolution turbulence reconstruction. *Physical Review Fluids*, 9:064601, 6 2024. ISSN 2469-990X. doi: 10.1103/PhysRevFluids.9.064601.

[44] Mathis Bode, Michael Gauding, Zeyu Lian, Dominik Denker, Marco Davidovic, Konstantin Kleinheinz, Jenia Jitsev, and Heinz Pitsch. Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows. *Proceedings of the Combustion Institute*, 38:2617–2625, 2021. ISSN 15407489. doi: 10.1016/j.proci.2020.06.022.

[45] Bowen Jing, Hannes Stärk, Tommi Jaakkola, and Bonnie Berger. Generative modeling of molecular dynamics trajectories. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 40534–40564. Curran Associates, Inc., 2024. URL `https://proceedings.neurips.cc/paper_files/paper/2024/file/478b06f60662d3cdc1d4f15d4587173a-Paper-Conference.pdf`.

[46] Alessia Gentili and Giorgio Volpe. Characterization of anomalous diffusion classical statistics powered by deep learning (CONDOR). *Journal of Physics A: Mathematical and Theoretical*, 54, 2021.

A Trajectory Pre-processing

The trajectories of particle motion obtained from processing the in situ videos were a time series of the x and y coordinates of the particles in each frame. To formulate the training dataset, these trajectories of various lengths were segmented into shorter 200-frame-long trajectories. The chosen trajectory length of 200 frames provides a balance between capturing particle dynamics and handling trajectories acquired at different video frame rates. Longer experimental trajectories can be divided into fixed-length segments to resolve local dynamics, while subsampling longer trajectories enables analysis of longer-range statistics with LEONARDO. To augment the training set with trajectories that reflect motion at longer time scales, the originally processed trajectories were also sub-sampled at rates from 2 to 60, and then segmented. For example, a sub-sampling rate of 2 means that every second x and y coordinate in the trajectory was extracted to form a new trajectory, which was then segmented into 200-time-point trajectories. The x and y components were treated as a combined 2D trajectory. A total of $38,279$ trajectories from LPTEM videos were collected for training.

The 200-frame experimental trajectories were normalized to lie between 0 and 1 for model training. The normalization process was performed as follows: each trajectory was first centered by subtracting the minimum value of all time frames for the x and y axes. The centered trajectory was then normalized by dividing by the range of values across the x and y axes. Mathematically, the normalization can be expressed as:

$$\mathbf{r}_{\text{normalized}} = \frac{\mathbf{r} - \min_t \mathbf{r}}{\max_{t,i} \mathbf{r} - \min_{t,i} \mathbf{r}} \tag{6}$$

where the subscript t refers to the time frame, the subscript i refers to the axes, $\min_t \mathbf{r}$ denotes the per-axis minima, $\min_t \mathbf{r} = [\min_t x_t \quad \min_t y_t]$, and $\max_{i,t} \mathbf{r}$ and $\min_{i,t} \mathbf{r}$ denote the maximum and minimum values across all entries in $\mathbf{r}$.

B Model Architecture and Training

Figure A1 shows the architecture of LEONARDO, which is a variational autoencoder with the encoder and decoder adapted from the Transformer architecture [1] that maps an input trajectory of 200 time frames, $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_{T=200})$, where each $\mathbf{r}_t = (x_t, y_t)$ represents the position vector of the nanoparticle with x and y denoting the x and y coordinates of the particle's position, respectively, to a sequence of continuous representation $\mathbf{z} = (z_1, \cdots, z_{12})$. Given $\mathbf{z}$, the decoder generates an output trajectory $\hat{\mathbf{r}} = (\hat{\mathbf{r}}_1, \hat{\mathbf{r}}_2, \cdots, \hat{\mathbf{r}}_{T=200})$, where each $\hat{\mathbf{r}}_t = (\hat{x}_t, \hat{y}_t)$. Here is the detail of each block:

First, a batch of N input trajectories is passed through a convolutional layer L to increase the embedding dimensions of each trajectory from 2 to 128 and get to $\mathbf{X}$ that is a tensor of size $N \times 128 \times 200$, where N is the batch size. $\mathbf{X}$ is the input to the encoder block depicted in Figure A2.

Encoder The encoder consists of two parts. First, the tensor $\mathbf{X}$ goes through a multi-headed self-attention layer (detailed by Vaswani et al. [1]) with 8 heads to capture the time dependencies within a trajectory, followed by layer normalization and fully connected feed-forward layers. The attention layer maps a query, $\mathbf{Q}$, and a set of key-value pairs, $\mathbf{K}, \mathbf{V}$, to an output. The multi-head attention layer allows the model to jointly attend to information from different representation subspaces at different positions along the length of the trajectory.

$$\begin{aligned} &\text{Multihead}(\mathbf{Q}, \mathbf{V}, \mathbf{K}) = \text{concatenate}(\text{head}_1, \cdots, \text{head}_8)\mathbf{W^O}, \\ &\text{where} \;\; \text{head}_i = \text{Attention}(\mathbf{Q}\mathbf{W^Q}_i, \mathbf{K}\mathbf{W^K}_i, \mathbf{V}\mathbf{W^V}_i), \end{aligned} \tag{7}$$

where $\mathbf{W^O} \in \mathbb{R}^{dh_v \times d_{\text{model}}}$, $\mathbf{W^Q} \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $\mathbf{W^K} \in \mathbb{R}^{d_{\text{model}} \times d_k}$, and $\mathbf{W^V} \in \mathbb{R}^{d_{\text{model}} \times d_v}$ are parameter matrices of the model. The attention matrix used here is the standard scaled Dot-Product attention [1] that is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V. \tag{8}$$

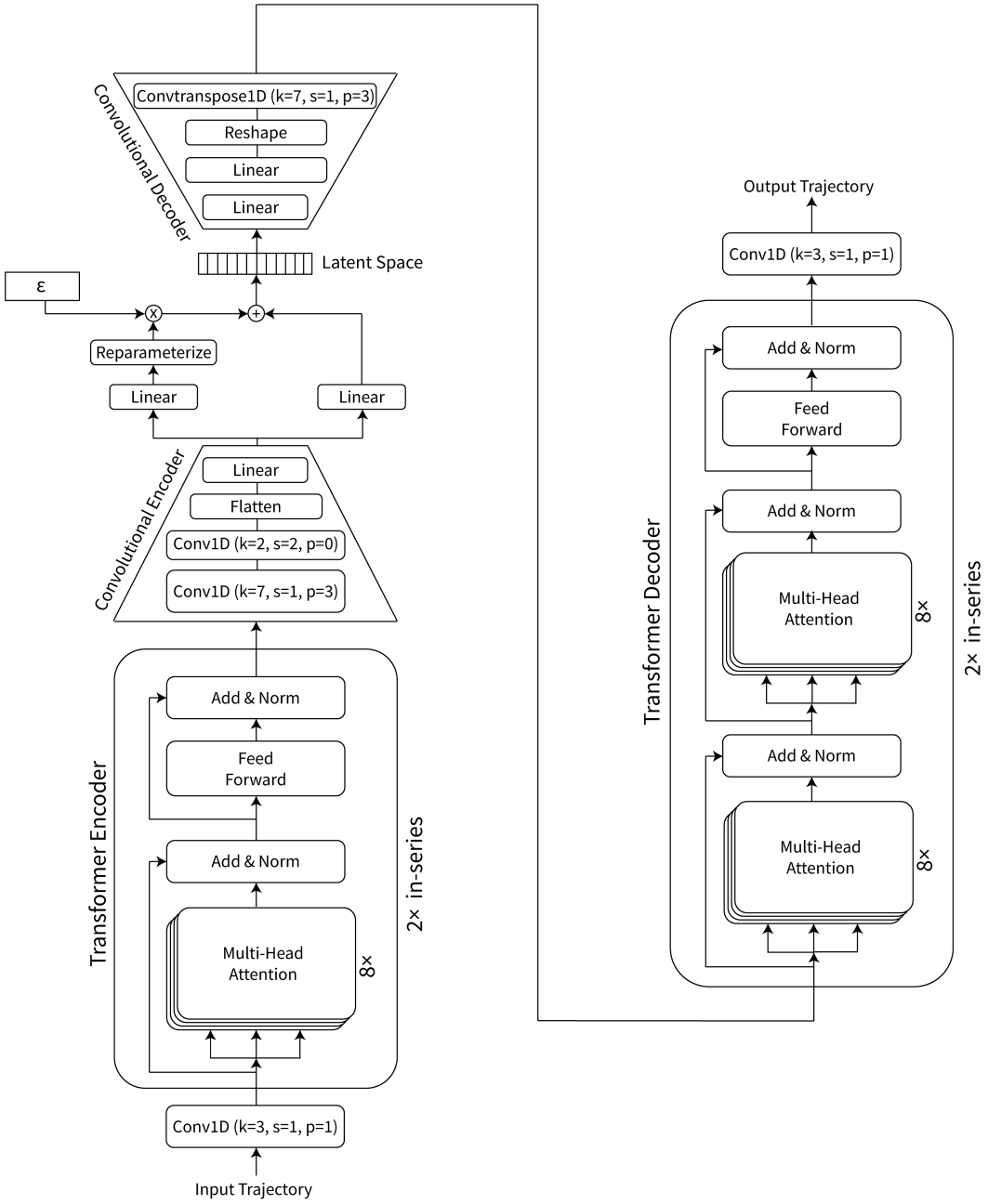

Figure A1: LEONARDO model architecture

Here, we employed $h = 8$ attention heads; therefore, $d_v = d_k = d_{\text{model}}/8 = 32$. The fully connected feed-forward layers deployed in the attention layer of the encoder block consist of two linear, fully connected transformations with a ReLU activation in between:

$$\text{FFN}(\mathbf{x}) = \max(0, \mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \tag{9}$$

where $\mathbf{x}$ is the input, $\mathbf{W}_1$, and $\mathbf{W}_2$ are weight matrices, and $\mathbf{b}_1$ and $\mathbf{b}_2$ are bias vectors. The output of this transformer block goes through another identical transformer block before entering the second part of the encoder block, which is a series of convolutional layers that reduce the size of the tensor to the latent space dimension. The first convolutional layer has a kernel size of 7, a stride of 1, and a padding of 3 to reduce the embedding dimension from 128 to 32. The second convolutional layer has a kernel size of 2 and a stride of 2 to reduce the size of the last tensor dimension from 200 to 100. This tensor is flattened to a size of 3200 before being further reduced in a linear layer to a size of 512.

The next few operations are adapted from the standard variational autoencoder [2]. In this stage, the encoder generates two vectors of size 12, $\boldsymbol{\mu}$ and $\log(\boldsymbol{\sigma}^2)$, representing the mean and log-variance of the latent space distribution. These vectors are used to sample the latent variable $\mathbf{z}$ via the reparameterization trick:

$$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}. \tag{10}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is standard Gaussian noise that allows for backpropagation through the stochastic sampling process. The sampled latent variables vector $\mathbf{z}$ is then passed to the decoder.

Decoder The latent space is up-sampled by two linear layers to sizes of 1024 and 6400, respectively, before being reshaped to dimensions of $N \times 32 \times 200$, where N is the batch size. This reshaped tensor goes into a transpose convolutional layer with a kernel size of 7, stride of 1, and padding of 3 to a shape of $N \times 128 \times 200$. The output from the convolutional decoder layer enters two transformer decoder blocks in series, each of which has two multi-headed self-attention layers in series, layer normalization, and feedforward layers as shown in Figure A1. A convolutional layer at the output of the transformer decoder reduces the size of the tensor to the size of the output trajectory, which is equal to the size of the input trajectory.

Model training All the parameters of the LEONARDO model architecture were trained by back-propagating the derivative of the loss function with respect to the model parameters using the ADAM optimizer with a learning rate of 3×10^{-4}.

Model validation Separate sets of experimental trajectories were segmented and sub-sampled for validation and testing. This resulted in $3,202$ trajectories for validation and $5,934$ trajectories for testing. Validation was performed to tune hyperparameters and reconstruct losses at each epoch during model training to compare against the training losses. These trajectories were not used to update the model parameters during backpropagation. The test set was used to report the final model performance. Figure A2 shows the validation losses per epoch averaged over the 3202 trajectories for each loss component of LEONARDO and their comparison with respect to the training loss at each epoch.

C Physics-informed Loss Function

The loss function consists of a total of 11 terms; all summed together with their respective weights determined based on the first epoch losses as defined below.

$$\mathcal{L} = \sum_{j=1}^{11} w_j \times \mathcal{L}_j, \tag{11}$$

where, w_j and $\mathcal{L}_j$ are the weights and loss components j, respectively. Each loss function component is defined in the following set of equations. The mean squared error (MSE) loss between the input and generated trajectories is defined first as the L2 norm:

$$\mathcal{L}_1 = \frac{1}{N} ||\mathbf{r} - \hat{\mathbf{r}}||_2^2, \tag{12}$$

where $N = 1000$ is the batch size.

The KL-divergence loss, as defined below, ensures that the posterior distribution of latent variables adheres to the prior Gaussian distribution:

$$\mathcal{L}_2 = D_{\mathrm{KL}}(q(\mathbf{z}|\mathbf{r})||p(\mathbf{z})) = \int q(\mathbf{z}|\mathbf{r}) \log \frac{q(\mathbf{z}|\mathbf{r})}{p(\mathbf{z})} \, d\mathbf{z}, \tag{13}$$

where $q(\mathbf{z}|\mathbf{r})$ is the approximate posterior distribution of latent variables $\mathbf{z}$ given the input trajectory $\mathbf{r}$, and $p(\mathbf{z})$ is a standard normal distribution $\mathcal{N}(0, 1)$.

The next four equations describe the loss components of the moments of the distributions of displacements of the trajectories. In each case, a mean squared error was taken between the moments of the input trajectory and the moments of the reconstructed trajectory.

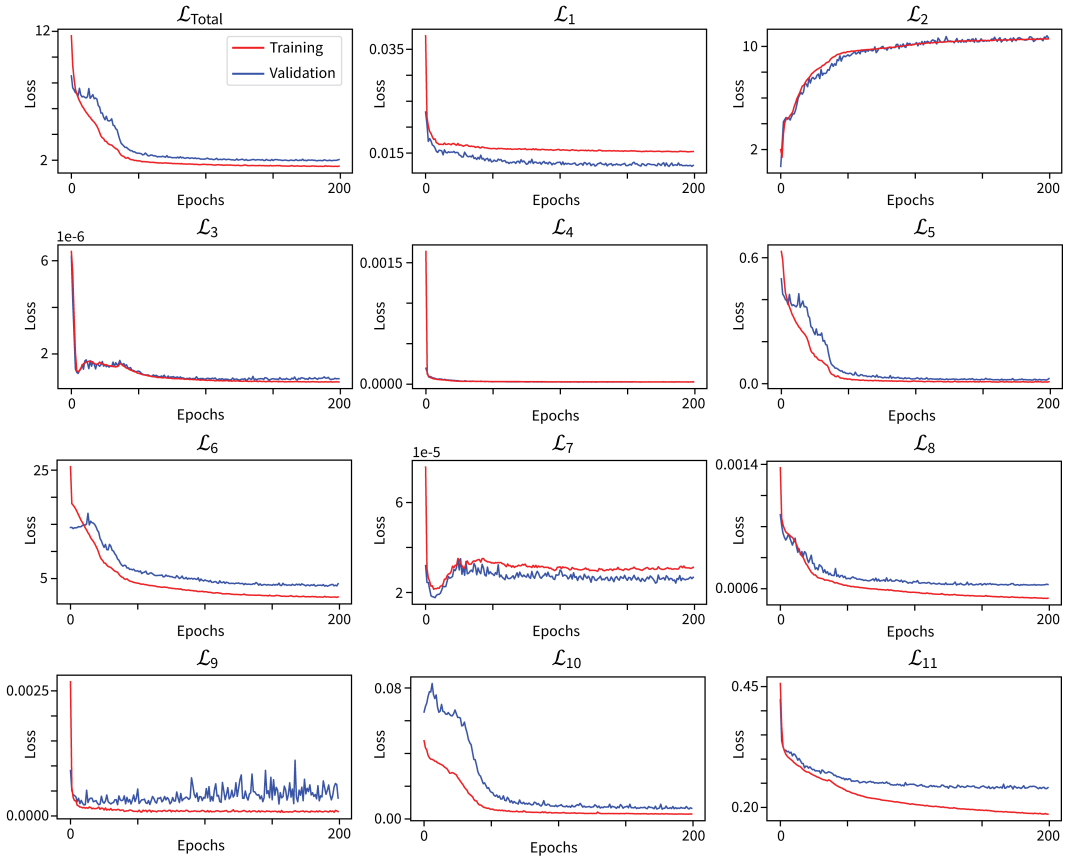

Figure A2: LEONARDO training and validation losses for each loss component at each epoch

$$\mathcal{L}_3 = \frac{1}{N}||\langle \mathbf{\Delta r} \rangle - \langle \mathbf{\Delta \hat{r}} \rangle||_2^2, \tag{14}$$

where $\langle \mathbf{\Delta r} \rangle$ is the mean, μ_{Δ_r}, of the distribution of displacements of the input trajectory, and $\langle \mathbf{\Delta \hat{r}} \rangle$ is the mean, $\mu_{\Delta \hat{r}}$, of the distribution of displacements of the reconstructed trajectory.

$$\mathcal{L}_4 = \frac{1}{N}||\langle (\mathbf{\Delta r} - \mu_{\Delta r})^2 \rangle - \langle (\mathbf{\Delta \hat{r}} - \mu_{\Delta \hat{r}})^2 \rangle||_2^2, \tag{15}$$

$$\mathcal{L}_5 = \frac{1}{N}\left|\left| \frac{\langle (\mathbf{\Delta r} - \mu_{\Delta r})^3 \rangle}{\langle \sigma_{\Delta r} \rangle^3} - \frac{\langle (\mathbf{\Delta \hat{r}} - \mu_{\Delta \hat{r}})^3 \rangle}{\langle \sigma_{\Delta \hat{r}} \rangle^3} \right|\right|_2^2, \tag{16}$$

$$\mathcal{L}_6 = \frac{1}{N}\left|\left| \frac{\langle (\mathbf{\Delta r} - \mu_{\Delta r})^4 \rangle}{\langle \sigma_{\Delta r}^2 \rangle^2} - \frac{\langle (\mathbf{\Delta \hat{r}} - \mu_{\Delta \hat{r}})^4 \rangle}{\langle \sigma_{\Delta \hat{r}}^2 \rangle^2} \right|\right|_2^2, \tag{17}$$

where $\sigma_{\Delta r}$ and $\sigma_{\Delta \hat{r}}$ are the standard deviation of $\mathbf{\Delta r}$ and $\mathbf{\Delta \hat{r}}$ distributions, respectively, and $\langle \cdot \rangle$ denotes an average over the trajectory displacements.

The loss comparing the medians of the distributions of displacements for the input and reconstructed trajectories is defined as:

$$\mathcal{L}_7 = \frac{1}{N}||(\tilde{\mathbf{\Delta}}\mathbf{r}) - (\tilde{\mathbf{\Delta}}\hat{\mathbf{r}})||_2^2, \tag{18}$$

where $(\tilde{\mathbf{\Delta}}\mathbf{r})$ represents the median of the displacement distribution of the input trajectory, and $(\tilde{\mathbf{\Delta}}\hat{\mathbf{r}})$ represents the median of the displacement distribution of the reconstructed trajectory.

The velocity autocorrelation loss component is defined for $\tau = 1$ to $\tau = 50$ (the first 50 time delays of $\mathbf{C_v}$), with each time delay weighted by $1/\tau$:

$$\mathcal{L}_8 = \frac{1}{N}\sum_{\tau=1}^{50}\frac{1}{\tau}||C_{\mathbf{v}}(\tau) - C_{\hat{\mathbf{v}}}(\tau)||_2^2, \tag{19}$$

where $C_{\mathbf{v}}(\tau) = \frac{\langle \mathbf{v}(t)\cdot\mathbf{v}(t+\tau)\rangle}{\langle \mathbf{v}^2(t)\rangle}$ refers to the velocity autocorrelation of the input trajectories, with $C_{\hat{\mathbf{v}}}(\tau) = \frac{\langle \hat{\mathbf{v}}(t)\cdot\hat{\mathbf{v}}(t+\tau)\rangle}{\langle \hat{\mathbf{v}}^2(t)\rangle}$ referring to the velocity autocorrelation of the reconstructed trajectories. The weighting factor of $1/\tau$ emphasizes the importance of short time lags (τ), which are more relevant for understanding the viscoelasticity of the interaction energy landscape investigated in this study. Longer time lags contribute less to the overall loss, as they are less informative for our study and are statistically less reliable due to the finite length of the trajectories. This weighting is particularly appropriate because the velocity autocorrelation function is most meaningful when $\tau \ll T$, where T is the total length of the trajectory.

The point-wise ensemble-averaged velocity autocorrelation is another important statistical measure that measures the correlations of an ensemble of particle trajectories at each time delay τ. For example, particle trajectories from LPTEM usually have a negative value for the velocity autocorrelation at short-time delays τ, which can be seen in the point-wise ensemble-averaged velocity autocorrelation of trajectories at short-time delays. These trajectories also have zero correlations at longer time delays, which ensures the stochasticity of the trajectories, *i.e.*, there are no predictable correlations within particle trajectories after the initial correlations at shorter time delays. To define the point-wise ensemble-averaged velocity autocorrelation, for each batch of trajectories in model training, we calculated the mean squared error between the velocity autocorrelation of the input batch, ensemble-averaged over the input batch, and the velocity autocorrelation of the generated batch, ensemble-averaged over the generated batch. This was then averaged across all time delays to obtain a singular value for the error as defined below:

$$\mathcal{L}_9 = \overline{\left(\langle \mathbf{C_v}\rangle - \langle \mathbf{C}_{\hat{\mathbf{v}}}\rangle\right)^2}. \tag{20}$$

where $\langle\cdot\rangle$ denotes average over a batch of N trajectories in the training dataset, and $\overline{(\cdot)}$ denotes the average over time delay windows of size $\tau = 1$ to $\tau = T - 2$ with $T = 200$ in this case.

The correlation between the x and y components of the 2-D trajectories is accounted for by a loss term that measures the deviation in the correlation coefficient between the input and reconstructed trajectories. This term ensures that the model captures any anisotropy or coupling between orthogonal motion components, which is particularly important for 2-D trajectories.

$$\mathcal{L}_{10} = \frac{1}{N}||\rho_{\mathbf{\Delta x},\mathbf{\Delta y}} - \rho_{\mathbf{\Delta \hat{x}},\mathbf{\Delta \hat{y}}}||_2^2, \tag{21}$$

where $\rho_{\mathbf{\Delta x},\mathbf{\Delta y}}$ is the correlation coefficient between the x and y displacements of the input trajectories, and $\rho_{\mathbf{\Delta \hat{x}},\mathbf{\Delta \hat{y}}}$ is the corresponding correlation coefficient for the reconstructed trajectories. The correlation coefficient for each trajectory is computed as:

$$\rho_{\mathbf{\Delta x},\mathbf{\Delta y}} = \frac{\mathrm{Cov}(\mathbf{\Delta x},\mathbf{\Delta y})}{\sqrt{\sigma^2_{\mathbf{\Delta x}}\cdot\sigma^2_{\mathbf{\Delta y}}}}, \tag{22}$$

where $\mathrm{Cov}(\mathbf{\Delta x},\mathbf{\Delta y}) = \frac{1}{N}\sum_{i=1}^{N}(\Delta x_i - \langle\Delta x\rangle)(\Delta y_i - \langle\Delta y\rangle)$ represents the covariance between the x and y displacements, and $\sigma^2_{\mathbf{\Delta x}}$ and $\sigma^2_{\mathbf{\Delta y}}$ represent the variances of the x and y displacements, respectively, and $\langle\cdot\rangle$ denotes the average over the trajectory displacements.

By including this term, the model is encouraged to reproduce the same level of anisotropy or coupling between x and y components as observed in the input trajectories. This is critical for accurately modeling complex systems where the motion in orthogonal directions may not be independent or isotropic.

The positional autocorrelation loss component compares the spatial correlation between particle positions at different time lags for the input and reconstructed trajectories. The positional autocorrelation

function is defined as $C_{\mathbf{r}}(\tau) = \frac{\langle \mathbf{r}(t) \cdot \mathbf{r}(t+\tau) \rangle}{\langle \mathbf{r}^2(t) \rangle}$, where $\mathbf{r}(t)$ and $\mathbf{r}(t+\tau)$ represent the position vectors at times t and $t+\tau$, respectively. The loss term is defined as:

$$\mathcal{L}_{11} = \frac{1}{N} \sum_{\tau=1}^{T-1} ||\mathbf{C}_{\mathbf{r}}(\tau) - \mathbf{C}_{\hat{\mathbf{r}}}(\tau)||_2^2, \tag{23}$$

where $\mathbf{C}_{\mathbf{r}}(\tau)$ represents the positional autocorrelation of the input trajectories, and $\mathbf{C}_{\hat{\mathbf{r}}}(\tau)$ represents the positional autocorrelation of the reconstructed trajectories. The inclusion of this term was motivated by the distinct behaviors observed in our experimental trajectories, where particles often transition abruptly between positions and remain localized in the new positions for extended periods. These dynamics introduce long-term spatial correlations that are not fully captured by displacement-based metrics.

The weights chosen for each loss term component based on the magnitude of the first epoch losses were $w_1 = 0.001, w_2 = 0.05, w_3 = 50,000, w_4 = 500, w_5 = 10, w_6 = 0.06, w_7 = 1, w_8 = 1000, w_9 = 100, w_{10} = 10, w_{11} = 1$. The very low weight assigned to w_1 (corresponding to the MSE-based reconstruction loss) emphasizes that the contribution of the MSE loss term is small compared to the physics-informed loss terms. This choice of weight reflects our focus on reproducing the statistical distribution of the trajectories rather than achieving an exact point-wise reconstruction.

D FMD Context

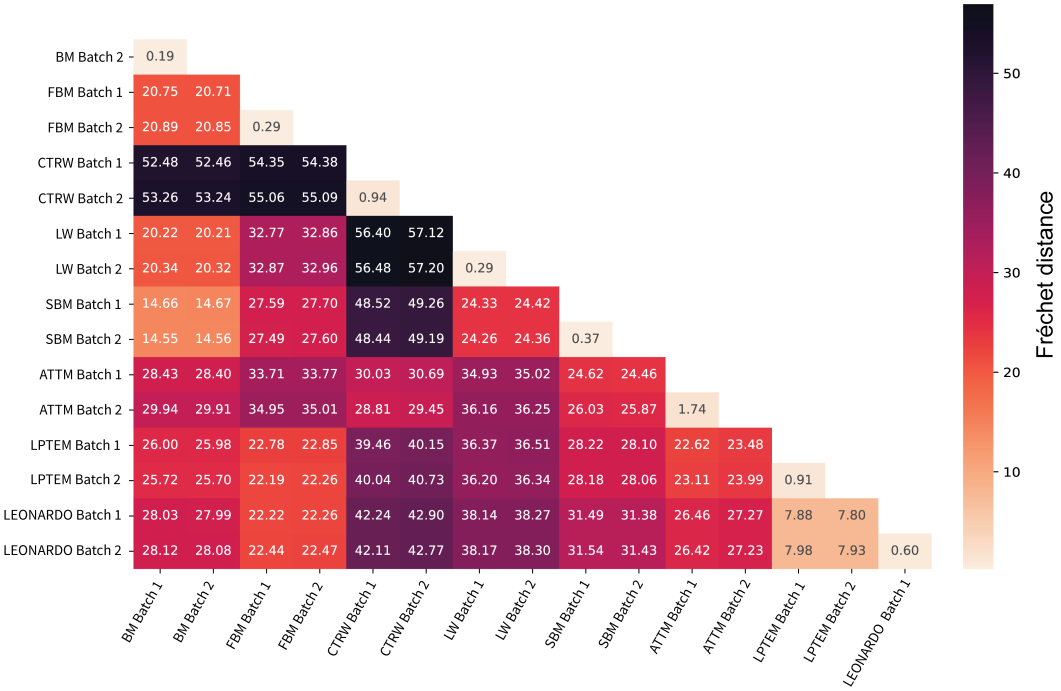

Figure A3: **FMD lower triangular matrix showing scores between classes of stochastic processes including intra-class scores.** The second-last layer output of MoNet2.0 is used to compute the FMD scores between different classes and between batches of the same class (intra-class scores). The matrix shows that the FMD scores between LPTEM and LEONARDO-generated trajectories are significantly lower than the scores between other classes, while intra-class scores, ranging from 0.19 to 1.74, provide a lower bound for contextual comparison.

E Availability

Data. The LPTEM trajectory datasets used for training, validation, and testing of LEONARDO are available at `https://huggingface.co/datasets/anon-user-5828/LEONARDO_train_val_test_datasets`. The trained LEONARDO model is available at `https://huggingface.co/anon-user-5828/LEONARDO`.

Code. Source code for LEONARDO is available at `https://anonymous.4open.science/r/LEONARDO-C33A`.

References (Appendix)

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[2] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.