Generative Latent Space Dynamics of Electron Density

Yuan Chiang[*]
University of California Berkeley
Lawrence Berkeley National Laboratory
cyrusyc@berkeley.edu

Youngsoo Choi
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
choi15@llnl.edu

Daniel Osei-Kuffuor
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
oseikuffuor1@llnl.edu

Abstract

Modeling the time-dependent evolution of electron density is essential for understanding quantum mechanical behaviors of condensed matter and enabling predictive simulations in spectroscopy, photochemistry, and ultrafast science. Yet, while machine learning methods have advanced static density prediction, modeling its spatiotemporal dynamics remains largely unexplored. In this work, we introduce a generative framework that combines a 3D convolutional autoencoder with a latent diffusion model (LDM) to learn electron density trajectories from *ab-initio* molecular dynamics (AIMD) simulations. Our method encodes electron densities into a compact latent space and predicts their future states by sampling from the learned conditional distribution, enabling stable long-horizon rollouts without drift or collapse. To preserve statistical fidelity, we incorporate a scaled Jensen-Shannon divergence regularization that aligns generated and reference density distributions. On AIMD trajectories of liquid lithium at 800 K, our model accurately captures both the spatial correlations and the log-normal-like statistical structure of the density. The proposed framework has the potential to accelerate the simulation of quantum dynamics and overcome key challenges faced by current spatiotemporal machine learning methods as surrogates of quantum mechanical simulators.

1 Introduction

The theoretical description of electrons and nuclei forms the cornerstone of understanding the physical and chemical properties of matter, yet it remains one of the most challenging frontiers of modern quantum mechanics. Electronic structure calculations, while offering a pathway to predict the ground and excited states of quantum many-body systems, are computationally expensive, with costs scaling steeply with the number of electrons.

In fact, full configuration interaction (FCI), although providing the exact solution to the time-independent Schrödinger equation, scales as $\mathcal{O}(n!)$ with respect to the number of molecular orbitals and basis set size. Coupled-cluster theory, while mitigating this issue through an exponential ansatz, still suffers from steep scaling: for a maximum excitation order r, the cost is $\mathcal{O}(n^{2r+2})$, where n is the number of basis functions. The gold-standard CCSDT method (coupled cluster singles, doubles, and full triples) scales as $\mathcal{O}(n^8)$, restricting its application to systems of at most tens to hundreds of atoms. Even Kohn–Sham density functional theory (KS-DFT), widely regarded as a computationally affordable alternative, typically scales as $\mathcal{O}(n^3)$, which becomes prohibitive for large-scale simulations or for repeated evaluations in dynamical settings.

[*] work started as a student at Lawrence Livermore National Laboratory.

Chiang et al., Generative Latent Space Dynamics of Electron Density (Extended Abstract). Presented at the Fourth Learning on Graphs Conference (LoG 2025), Hybrid Event, December 10–12, 2025.

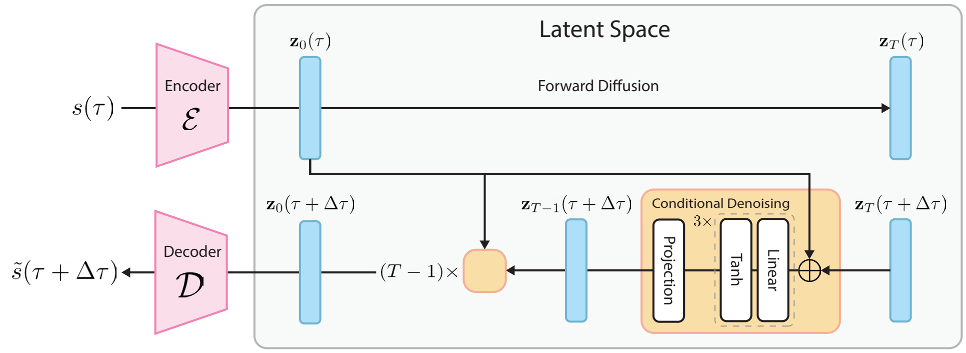

Figure 1: Illustration of latent space generative process to emulate the dynamics of electron density through the combination of autoencoder and latent diffusion model. At each physical time step τ, the model conditions the latent denoiser with the current latent state $\mathbf{z}_0(\tau)$ to predict the next latent state $\mathbf{z}_0(\tau+\Delta\tau)$. The decoder maps the latent state back to physical space $\tilde{s}(\tau+\Delta\tau) = \mathcal{D}(\mathbf{z}_0(\tau+\Delta\tau))$.

A promising strategy to accelerate such calculations is to provide a high-quality initial guess for the electron density or wavefunctions [1–4], which can significantly speed up convergence in self-consistent field (SCF) iterations. While most machine learning approaches to date have focused on predicting static electron densities from molecular geometries [5–8], far fewer address the dynamical evolution of the electron density. However, this dynamical information is essential: the time-dependent electron density encodes rich physical observables, such as the dynamic structure factor, excitation energies, and transition moments, and underpins many applications in spectroscopy, photochemistry, and ultrafast science.

Previous works on electron density modeling often rely on graph-based neural networks, where atoms are treated as nodes and bonds or spatial cutoffs define edges. While powerful, this approach imposes an atomic representation that may be less natural for modeling the volumetric nature of the electron density in a continuous space. By contrast, volumetric representations preserve translational and rotational structure more directly, enable direct operator learning in physical space, and can better capture delocalized electrons, charge density waves or excitations.

In this work, we propose a framework that combines a 3D convolutional autoencoder with a latent diffusion model (LDM) [9] to learn and evolve the electron density in a compressed latent space. The autoencoder first encodes volumetric electron density fields into a compact latent representation, preserving intrinsic spatial and physical structure while reducing dimensionality. The LDM then learns the conditional distribution of the next latent state given the current state, enabling autoregressive probabilistic generation of full electron density trajectories. Our formulation efficiently compresses the high-dimensional observed space into low-dimensional manifold while enabling the robust, long-horizon sampling without commonly seen drifting, collapse, or state stagnation problems.

2 Generative Latent Space Dynamics of Electron Density

Our framework models the dynamics of electron density using a two-stage generative process, as illustrated in Figure 1. The methodology consists of density normalization and state representation, a compression stage using a 3D convolutional autoencoder, and a temporal evolution stage using a conditional latent diffusion model. We provide the detailed methodology in Appendix A and related work in Appendix B.

Dataset and state representation. We generate an *ab-initio* molecular dynamics (AIMD) simulation of 32 liquid lithium atoms at 800 K (appendix A.1). We observe that the electron density values are approximately log-normally distributed (fig. S1). To leverage this property and naturally enforce positivity, we apply a log-transform followed by standardization to the density field $\rho(\mathbf{r},\tau)$, yielding a normalized field $\tilde{\rho}(\mathbf{r},\tau)$ as described in eq. (S6). The full state representation $s(\tau)$ at each physical timestep τ consists of this normalized density $\tilde{\rho}$ and its time derivative $\dot{\tilde{\rho}}$ (eq. (S8)).

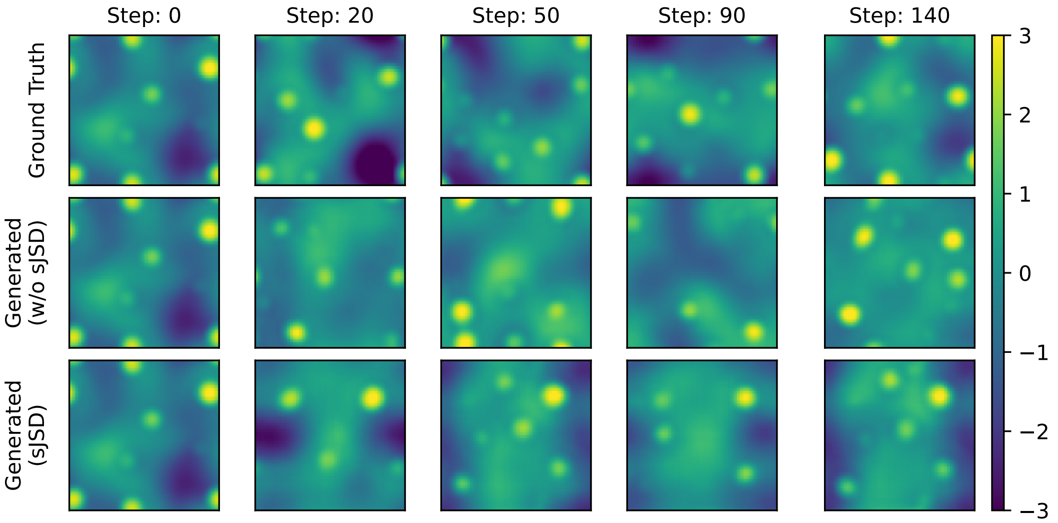

Figure 2: Autoregressive trajectories. LDM generates and evolves electron density qualitatively consistent with unseen test trajectory of Li atoms at 800 K. The model trained with sJSD loss has less delocalized electron distribution similar to the ground truth. 2D slices at the middle plane along x direction are shown. The grid values presented are normalized pseudo charge density $\tilde{\rho}$ (eq. (S6), fig. S1). See Figure 3 for probability density and structure factor comparison.

Model architecture. A 3D convolutional autoencoder (AE) provides a low-dimensional manifold for learning the dynamics. The encoder $\mathcal{E}$ maps the high-dimensional state $s(\tau)$ to a compact latent vector $\mathbf{z}(\tau)$ using a sequence of residual blocks and strided convolutions, with an average pooling layer to enforce translation invariance. The decoder $\mathcal{D}$ then reconstructs the physical state $\tilde{s}(\tau)$ from the latent vector.

The temporal evolution is learned in the latent space by a conditional latent diffusion model (LDM). The LDM is trained to approximate the transition probability $p(\mathbf{z}(\tau+\Delta\tau)|\mathbf{z}(\tau))$. During inference, it autoregressively predicts the next state by sampling from the learned conditional distribution, enabling the generation of long-horizon trajectories. This is achieved by conditioning the denoising network $\boldsymbol{\epsilon}_\theta$ on the current latent state $\mathbf{z}(\tau)$ to predict the noise in the corrupted future state $\mathbf{z}_t(\tau+\Delta\tau)$.

Training and regularization. To ensure the generated electron densities retain the statistical fidelity of the ground truth, we augment the standard reconstruction and denoising losses. We introduce a scaled Jensen-Shannon Divergence ($\mathcal{L}_{\text{sJSD}}$) term, as defined in eq. (S14), which penalizes mismatches between the value distributions of the reconstructed and target densities. This loss is made differentiable by computing probabilities via a soft histogram with Gaussian kernels. The full training objective (eq. (S15)) is a weighted sum of the AE reconstruction, LDM denoising (eq. (S12)), and sJSD (eq. (S14)) losses.

Experiments. We found that for electron density prediction, the ML model could still achieve low reconstruction error with average prediction on most of the grid points, as only a few of them have concentrated electron density. In such cases, the model learns only the fuzzy average of the input distribution and could not preserve the distributional attributes of the electron densities. Therefore, we apply sJSD loss eq. (S14) on the reconstructed electron densities, with $\lambda_1 = 0.1$ and $\lambda_2 = 10$ in eq. (S15) used.

Figure 2 presents the generated autoregressive traejctories on two models, with and without sJSD losses. Generally, the model with sJSD regularization generates electron densities with visual characteristics closer to the ground-truth test trajectory. Figure 3 further compares the distributional similarity between two generated and ground-truth, log-normal like distributions. The model trained with sJSD loss clearly demonstrates more similar distribution to test trajectory than the one without sJSD regularization, which overemphasizes the population around $0.05\,\text{Å}^{-3}$. See Figure S2 for training loss comparison.

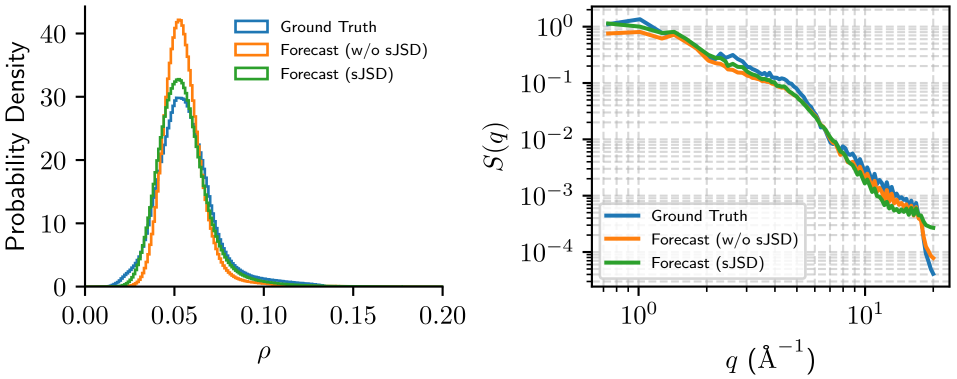

Figure 3: Statistical consistency of generated trajectories. Probability distribution and static structure factor $S(q)$ of test-set (ground-truth) and generated (forecast) electron densities. The LDM model trained with sJSD loss outperforms the model without sJSD. The static structure factor is obtained by averaging over the unrolled trajectory frames. The detailed $S(q)$ calculation can be found in Appendix C.

To ensure the model also generates correct spatial distribution as well, we further compare the structure factor $S(q)$ of the three trajectories in Figure 3. It can be seen that convolutional autoencoder, while highly constrained in design, is able to robustly preserve the correct spatial correlation as the dynamics evolves. The model with sJSD loss aligns better with ground truth in low q, long wavelength regime, demonstrating better ability to capture long-range spatial correlation of electron density. This is, however, at the expense of the high q, short wavelength regime, but the difference is orders of magnitude smaller. The kink near $q = 4$ is consistent with the atomic radius of a lithium atom (≈ 1.52 Å):

$$q = \frac{2\pi}{a} = 4, \quad a \approx 1.57.$$

3 Discussion

Learning in Fourier space. Fourier representation of electron density, while efficient for dimension reduction, struggles to form meaningful latent spaces due to translation symmetry and phase-shift ambiguity. Our preliminary experiments show that multi-layer perceptron (MLP) autoencoders overfit to Fourier features and extrapolate poorly. Although FNOs have shown strong success in weather forecasting and PDEs [10, 11], our test of FNO [10] revealed pronounced drift and instability during rollout. In contrast, the average pooling bottleneck in our encoder enforces translation invariance and stabilizes trajectories without noticeable shift. Future work should explore models equivariant to phase shifts and robust to translation.

Another bitter lesson. In our preliminary tests, direct enforcement of physics-informed neural network (PINN) loss on FNO and MLP AEs fails to learn latent representations that can reliably roll out dynamics. Autoencoders trained with PINN loss on charge conservation and spatial gradients easily overfit, and when entering unseen latent regions, the decoder either drifts or freezes due to inactive neurons. We show that a small CNN with average pooling bottleneck and sJSD loss, despite having fewer parameters than MLP AEs, transformers, and neural operators, generalizes better to unseen AIMD trajectories and reliably rolls out without drift or freezing. This contrasts with recent efforts favoring larger models for PDE solving, likely because the expressive models easily overfit high-dimensional noise ($48^3 = 110,592$ per frame in our case), and the quantum-mechanical system is far more discontinuous than the continuum problems routinely solved.

Limitations and opportunities. Since ions move classically here, our experiments can be bootstrapped by machine-learning interatomic potentials [12] and electron density prediction models in alternating steps. Still, our framework could extend to more demanding calculations such as time-dependent DFT (TD-DFT) [13] and equation-of-motion coupled cluster (EOM-CC) [14], where data is scarcer and time dependence is crucial. The current simulations are based on Born-Oppenheimer molecular dynamics and thus can be replaced with more expensive methods to include nuclear quantum effects, such as path-integral MD (PIMD), multicomponent DFT (MDFT) [15], nuclear-electron orbital (NEO) method [16] and its constrained variant (CNEO) [17]. The ionic positions are not explicitly encoded but learned implicitly in our current approach. Future work to simultaneously learn the ionic and electronic dynamics, potentially through shared embedding, will be interesting. We also aim to generalize to multiple atomic species for practical DFT use—whether for initial guesses or density optimization, akin to orbital-free DFT and variants [18–20]. A complementary path is integrating with the data-driven finite element method (DD-FEM) [21], which could reduce the cost of generating high-fidelity electron density data while retaining interpretability by learning operators on element scales. Such approaches could extend generative latent dynamics models toward practical, multiscale applications in spectroscopy, photochemistry, and ultrafast science.

References

[1] Moritz Gubler, Moritz R Schäfer, Jörg Behler, and Stefan Goedecker. Accuracy of charge densities in electronic structure calculations. *The Journal of Chemical Physics*, 162(9), 2025. 2

[2] Péter Pulay. Convergence acceleration of iterative sequences. the case of scf iteration. *Chemical physics letters*, 73(2):393–398, 1980.

[3] Charles G Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of computation*, 19(92):577–593, 1965.

[4] Sambit Das and Vikram Gavini. Accelerating self-consistent field iterations in kohn-sham density functional theory using a low-rank approximation of the dielectric matrix. *Physical Review B*, 107(12):125133, 2023. 2

[5] Xiang Fu, Andrew Rosen, Kyle Bystrom, Rui Wang, Albert Musaelian, Boris Kozinsky, Tess Smidt, and Tommi Jaakkola. A recipe for charge density prediction. *Advances in Neural Information Processing Systems*, 37:9727–9752, 2024. 2, 11

[6] Teddy Koker, Keegan Quigley, Eric Taw, Kevin Tibbetts, and Lin Li. Higher-order equivariant neural networks for charge density prediction in materials. *npj Computational Materials*, 10(1): 161, 2024. 11

[7] Peter Bjørn Jørgensen and Arghya Bhowmik. Equivariant graph neural networks for fast electron density estimation of molecules, liquids, and solids. *npj Computational Materials*, 8(1):183, 2022.

[8] Peter Bjørn Jørgensen and Arghya Bhowmik. Deepdft: Neural message passing network for accurate charge density prediction. *arXiv preprint arXiv:2011.03346*, 2020. 2, 11

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 10, 11

[10] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020. 4, 12

[11] Thorsten Kurth, Shashank Subramanian, Peter Harrington, Jaideep Pathak, Morteza Mardani, David Hall, Andrea Miele, Karthik Kashinath, and Anima Anandkumar. Fourcastnet: Accelerating global high-resolution weather forecasting using adaptive fourier neural operators. In *Proceedings of the platform for advanced scientific computing conference*, pages 1–11, 2023. 4

[12] Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M. Elena, Dávid P. Kovács, Janosh Riebesell, Xavier R. Advincula, Mark Asta, William J. Baldwin, Noam Bernstein, Arghya Bhowmik, Samuel M. Blau, Vlad Cărare, James P. Darby, Sandip De, Flaviano Della Pia, Volker L. Deringer, Rokas Elijošius, Zakariya El-Machachi, Edvin Fako, Andrea C. Ferrari, Annalena Genreith-Schriever, Janine George, Rhys E. A. Goodall, Clare P. Grey, Shuang Han, Will Handley, Hendrik H. Heenen, Kersti Hermansson, Christian Holm, Jad Jaafar, Stephan Hofmann, Konstantin S. Jakob, Hyunwook Jung, Venkat Kapil, Aaron D. Kaplan, Nima Karimitari, Namu Kroupa, Jolla Kullgren, Matthew C. Kuner, Domantas Kuryla, Guoda Liepuoniute, Johannes T. Margraf, Ioan-Bogdan Magdău, Angelos Michaelides, J. Harry Moore, Aakash A. Naik, Samuel P. Niblett, Sam Walton Norwood, Niamh O'Neill, Christoph Ortner, Kristin A. Persson, Karsten Reuter, Andrew S. Rosen, Lars L. Schaaf, Christoph Schran, Eric Sivonxay,

Tamás K. Stenczel, Viktor Svahn, Christopher Sutton, Cas van der Oord, Eszter Varga-Umbrich, Tejs Vegge, Martin Vondrák, Yangshuai Wang, William C. Witt, Fabian Zills, and Gábor Csányi. A foundation model for atomistic materials chemistry, December 2023. URL http://arxiv.org/abs/2401.00096. arXiv:2401.00096 [cond-mat, physics:physics]. 4

[13] Erich Runge and Eberhard KU Gross. Density-functional theory for time-dependent systems. *Physical review letters*, 52(12):997, 1984. 4

[14] John F Stanton and Rodney J Bartlett. The equation of motion coupled-cluster method. a systematic biorthogonal approach to molecular excitation energies, transition probabilities, and excited state properties. *The Journal of chemical physics*, 98(9):7029–7039, 1993. 4

[15] T Kreibich and EKU Gross. Multicomponent density-functional theory for electrons and nuclei. *Physical Review Letters*, 86(14):2984, 2001. 5

[16] Simon P Webb, Tzvetelin Iordanov, and Sharon Hammes-Schiffer. Multiconfigurational nuclear-electronic orbital approach: Incorporation of nuclear quantum effects in electronic structure calculations. *The Journal of chemical physics*, 117(9):4106–4118, 2002. 5

[17] Xi Xu and Yang Yang. Constrained nuclear-electronic orbital density functional theory: Energy surfaces with nuclear quantum effects. *The Journal of Chemical Physics*, 152(8), 2020. 5

[18] Wenhui Mi, Kai Luo, SB Trickey, and Michele Pavanello. Orbital-free density functional theory: An attractive electronic structure method for large-scale first-principles simulations. *Chemical Reviews*, 123(21):12039–12104, 2023. 5

[19] Vincent L Lignères and Emily A Carter. An introduction to orbital-free density functional theory. In *Handbook of materials modeling: methods*, pages 137–148. Springer, 2005.

[20] Kaili Jiang and Michele Pavanello. Time-dependent orbital-free density functional theory: Background and pauli kernel approximations. *Physical Review B*, 103(24):245102, 2021. 5

[21] Youngsoo Choi, Siu Wun Cheung, Youngkyu Kim, Ping-Hsuan Tsai, Alejandro N Diaz, Ivan Zanardi, Seung Whan Chung, Dylan Matthew Copeland, Coleman Kendrick, William Anderson, et al. Defining foundation models for computational science: A call for clarity and rigor. *arXiv preprint arXiv:2505.22904*, 2025. 5

[22] Jens Jørgen Mortensen, Ask Hjorth Larsen, Mikael Kuisma, Aleksei V Ivanov, Alireza Taghizadeh, Andrew Peterson, Anubhab Haldar, Asmus Ougaard Dohn, Christian Schäfer, Elvar Örn Jónsson, et al. Gpaw: An open python package for electronic structure calculations. *The Journal of Chemical Physics*, 160(9), 2024. 9

[23] Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, et al. The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27):273002, 2017. 9

[24] Peter E Blöchl. Projector augmented-wave method. *Physical review B*, 50(24):17953, 1994. 9

[25] Jens Jørgen Mortensen, Lars Bruno Hansen, and Karsten Wedel Jacobsen. Real-space grid implementation of the projector augmented wave method. *Physical Review B—Condensed Matter and Materials Physics*, 71(3):035109, 2005. 9

[26] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980. 10

[27] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 10

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385. 10

[30] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 4(5):11, 2015. 10

[31] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 11

[32] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 11

[33] Erik Englesson and Hossein Azizpour. Generalized jensen-shannon divergence loss for learning with noisy labels. *Advances in Neural Information Processing Systems*, 34:30284–30297, 2021. 11

[34] Youngkyu Kim, Karen Wang, and Youngsoo Choi. Efficient space–time reduced order model for linear dynamical systems in python using less than 120 lines of code. *Mathematics*, 9(14): 1690, 2021. 12

[35] Sean McBane and Youngsoo Choi. Component-wise reduced order model lattice-type structure design. *Computer methods in applied mechanics and engineering*, 381:113813, 2021.

[36] Youngsoo Choi, Gabriele Boncoraglio, Spenser Anderson, David Amsallem, and Charbel Farhat. Gradient-based constrained optimization using a database of linear reduced-order models. *Journal of Computational Physics*, 423:109787, 2020.

[37] Youngsoo Choi and Kevin Carlberg. Space–time least-squares petrov–galerkin projection for nonlinear model reduction. *SIAM Journal on Scientific Computing*, 41(1):A26–A58, 2019.

[38] Dylan Matthew Copeland, Siu Wun Cheung, Kevin Huynh, and Youngsoo Choi. Reduced order models for lagrangian hydrodynamics. *Computer Methods in Applied Mechanics and Engineering*, 388:114259, 2022.

[39] Youngsoo Choi, Peter Brown, William Arrighi, Robert Anderson, and Kevin Huynh. Space–time reduced order model for large-scale linear dynamical systems with application to boltzmann transport problems. *Journal of Computational Physics*, 424:109845, 2021.

[40] Youngsoo Choi, Deshawn Coombs, and Robert Anderson. Sns: A solution-based nonlinear subspace method for time-dependent model order reduction. *SIAM Journal on Scientific Computing*, 42(2):A1116–A1146, 2020.

[41] Jessica T Lauzon, Siu Wun Cheung, Yeonjong Shin, Youngsoo Choi, Dylan M Copeland, and Kevin Huynh. S-opt: A points selection algorithm for hyper-reduction in reduced order models. *SIAM Journal on Scientific Computing*, 46(4):B474–B501, 2024.

[42] Siu Wun Cheung, Youngsoo Choi, Dylan Matthew Copeland, and Kevin Huynh. Local lagrangian reduced-order modeling for the rayleigh-taylor instability by solution manifold decomposition. *Journal of Computational Physics*, 472:111655, 2023. 12

[43] Mark L Davison and Stephen G Sireci. Multidimensional scaling. In *Handbook of applied multivariate statistics and mathematical modeling*, pages 323–352. Elsevier, 2000. 12

[44] Youngkyu Kim, Youngsoo Choi, David Widemann, and Tarek Zohdi. A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder. *Journal of Computational Physics*, 451:110841, 2022. 12

[45] Alejandro N Diaz, Youngsoo Choi, and Matthias Heinkenschloss. A fast and accurate domain decomposition nonlinear manifold reduced order model. *Computer Methods in Applied Mechanics and Engineering*, 425:116943, 2024.

[46] William D Fries, Xiaolong He, and Youngsoo Choi. Lasdi: Parametric latent space dynamics identification. *Computer Methods in Applied Mechanics and Engineering*, 399:115436, 2022. 12

[47] Xiaolong He, Youngsoo Choi, William D Fries, Jonathan L Belof, and Jiun-Shyan Chen. glasdi: Parametric physics-informed greedy latent space dynamics identification. *Journal of Computational Physics*, 489:112267, 2023.

[48] Christophe Bonneville, Youngsoo Choi, Debojyoti Ghosh, and Jonathan L Belof. Gplasdi: Gaussian process-based interpretable latent space dynamics identification through deep autoencoder. *Computer Methods in Applied Mechanics and Engineering*, 418:116535, 2024.

[49] April Tran, Xiaolong He, Daniel A Messenger, Youngsoo Choi, and David M Bortz. Weak-form latent space dynamics identification. *Computer Methods in Applied Mechanics and Engineering*, 427:116998, 2024.

[50] Jun Sur Richard Park, Siu Wun Cheung, Youngsoo Choi, and Yeonjong Shin. tlasdi: Thermodynamics-informed latent space dynamics identification. *Computer Methods in Applied Mechanics and Engineering*, 429:117144, 2024.

[51] Xiaolong He, April Tran, David M Bortz, and Youngsoo Choi. Physics-informed active learning with simultaneous weak-form latent space dynamics identification. *International Journal for Numerical Methods in Engineering*, 126(1):e7634, 2025.

[52] Teeratorn Kadeethum, Francesco Ballarin, Youngsoo Choi, Daniel O'Malley, Hongkyu Yoon, and Nikolaos Bouklas. Non-intrusive reduced order modeling of natural convection in porous media using convolutional autoencoders: comparison with linear subspace techniques. *Advances in Water Resources*, 160:104098, 2022. 12

[53] Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000. 12

[54] Marina Meilă and Hanyu Zhang. Manifold learning: What, how, and why. *Annual Review of Statistics and Its Application*, 11(1):393–417, 2024. 12

[55] Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. *arXiv preprint arXiv:2111.13802*, 2021. 12

[56] Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019. 12

[57] Rudy Morel, Jiequn Han, and Edouard Oyallon. Disco: learning to discover an evolution operator for multi-physics-agnostic prediction. *arXiv preprint arXiv:2504.19496*, 2025. 12

[58] Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931. 12

[59] Tim Hsu, Babak Sadigh, Vasily Bulatov, and Fei Zhou. Score dynamics: Scaling molecular dynamics with picoseconds time steps via conditional diffusion model. *Journal of Chemical Theory and Computation*, 20(6):2335–2348, 2024. 12

[60] François Rozet, Ruben Ohana, Michael McCabe, Gilles Louppe, François Lanusse, and Shirley Ho. Lost in latent space: An empirical study of latent diffusion models for physics emulation. *arXiv preprint arXiv:2507.02608*, 2025. 12

[61] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018. 12

[62] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019. 12

[63] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022. 12

A Methodology Details

A.1 *Ab-initio* Molecular Dynamics (AIMD)

Theory. The time evolution of a many-electron system is, in principle, governed by the time-dependent many-body Schrödinger equation for all electrons and nuclei. In practice, this is computationally intractable beyond hundreds of atoms, and for weakly correlated systems the adiabatic Born–Oppenheimer (BO) approximation can be employed: the electronic and nuclear degrees of freedom are decoupled, and the nuclei move classically on the potential energy surface (PES) of the electronic ground state. For a fixed configuration of nuclei $\{\mathbf{R}_I\}$, the ground-state PES and *electron number density* $\rho(\mathbf{r})$ are obtained from a self-consistent field (SCF) solution of the Kohn–Sham (KS) equations,

$$\left(-\frac{\hbar^2}{2m_e}\nabla^2 + V_{\text{eff}}(\mathbf{r})\right)\psi_i(\mathbf{r}) = \varepsilon_i\psi_i(\mathbf{r}), \tag{S1}$$

$$\rho(\mathbf{r}) = \sum_i \left|\psi_i(\mathbf{r})\right|^2, \tag{S2}$$

$$E = \min_\rho E\left[\rho\right], \tag{S3}$$

where $V_{\text{eff}}(\mathbf{r})$ is the effective external potential parameterized by $\{\mathbf{R}_I\}$ and $E\left[\rho\right]$ is the energy functional of the electron density. After the SCF calculation is converged, the time evolution of atomic nuclei can be integrated from forces (and additional terms for ensembles other than microcanonical ensemble)

$$M_I\ddot{\mathbf{R}}_I = -\nabla_{\mathbf{R}_I}E, \tag{S4}$$

where $-\nabla_{\mathbf{R}_I}E$ is the Hellmann–Feynman force evaluated by the derivative of ground-state KS Hamiltonian and electron orbitals: $\mathbf{F}_I = -\left\langle \boldsymbol{\psi}_o \,\middle|\, \frac{\partial \hat{H}_{\text{KS}}}{\partial \mathbf{R}_I} \,\middle|\, \boldsymbol{\psi}_o \right\rangle$.

Dataset. We generated a electron density trajectory of 32 Li atoms from isochoric-isothermal NVT AIMD simulation at 800 K. For each ionic step, KS-DFT calculation is performed with generalized gradient approximation to search for the ground state electron density. Perdew-Burke-Ernzerhof (PBE) functional was used to describe exchange-correlation energy. The electron wave functions are expanded in plane-wave bases, with maximum energy cutoff 680 eV. The AIMD trajectory was performed for 10 ps at the timestep of 2 fs, where the first 8 ps was used for training and the last 2 ps as test set. The AIMD and electron density trajectories were performed using GPAW [22] and ASE [23]. We recorded both total and pseudo electron densities for each frame. Nonetheless, only the pseudo electron density is variationally optimized and contains rich bonding information in the projected-augmented wave formalism [24, 25]. We use pseudo electron density as model learning objective and hereafter denote the pseudo electron number density as electron density throughout the work.

We analyzed the distribution of electron density in Figure S1 and found that the value is roughly log-normally distributed for our toy system. We describe the observation below.

Proposition 1 (Log-normal distribution of ρ)**.** *The electron (number) density $\rho(\mathbf{r},\tau)$ from an equilibrated AIMD trajectory is approximately log-normally distributed in the space-time dimensions*

$$\ln\rho(\mathbf{r},\tau) \sim \mathcal{N}\left(\mu,\sigma^2\right) \tag{S5}$$

, where μ and σ^2 is the mean and variance of the Gaussian distribution.

To leverage this nice property, we therefore normalize our electron density data by logarithm and shift-scale transformation

$$\tilde{\rho} = \frac{\ln\rho - \mu}{\sigma}, \tag{S6}$$

where $\mu = -2.911$ and $\sigma = 0.271$ are the mean and standard deviation of $\ln\rho$ in training set. The probability density of $\tilde{\rho}(\mathbf{r},\tau)$ is thereby close to standard normal distribution $\mathcal{N}(0,1)$, as shown in the bottom panel of Figure S1. This transformation allows flexible reconstruction space for the decoder and diffusion models, and naturally enforces positivity with inverse relation $\exp\left(\sigma\tilde{\rho}+\mu\right) > 0$.

State representation. The state space of the classical atomistic system is determined by atomic positions and velocities. Similarly, in order to be complete, the state of the electron density should be at least described by both number density and its time derivative, as the time derivative of number density is linked to charge current density $\mathbf{j}(\mathbf{r},\tau)$ by the continuity equation: $\dot{\rho}(\mathbf{r},\tau) = -\nabla \mathbf{j}(\mathbf{r},\tau)$. We formally specify each frame at the *physical* time τ with the state as:

$$s(\tau) = (\mathbf{A}(\tau), \rho(\mathbf{r},\tau), \dot{\rho}(\mathbf{r},\tau)) \in \mathbb{R}^{3\times 3} \times \mathbb{R}_{\geq 0}^{N_1\times N_2\times N_3} \times \mathbb{R}^{N_1\times N_2\times N_3}, \quad \text{(S7)}$$

where $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]$ is unit cell matrix of three lattice vectors, $\mathbf{R}$, $\dot{\mathbf{R}}$ are the ionic positions and velocities, and N_i are the number of grid points along three dimensions. Since we have lifted the positivity constraint on electron density by eq. (S6) and fixed the cubic cell geometry under NVT ensemble, the state representation reduces to

$$s(\tau) = \big(L, \tilde{\rho}(\mathbf{r},\tau), \dot{\tilde{\rho}}(\mathbf{r},\tau)\big) \in \mathbb{R} \times \mathbb{R}^{N_1\times N_2\times N_3} \times \mathbb{R}^{N_1\times N_2\times N_3}, \quad \text{(S8)}$$

where L is the lattice constant (the length of cubic cell vector). We further leave L out from the state representation in this work as the volume is fixed throughout the AIMD trajectory. In principle, the lattice parameters could be easily placed back by concatenation in the latent space.

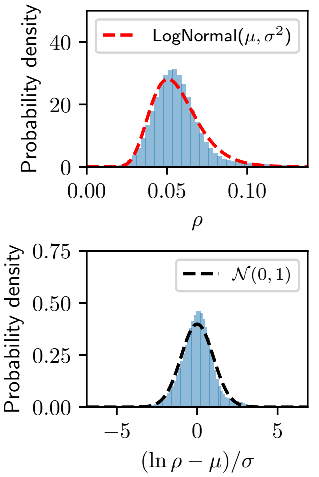

Figure S1: Data distribution. Pseudo electron density distribution of Li32 trajectory approximately follows single log-normal distribution with $\mu = -2.911, \sigma = 0.271$. Only training set is visualized.

A.2 Latent Diffusion Model for 3D Scalar Field

Model architecture. Our model is conceptually inspired by latent diffusion model (LDM) [9], but has undergone multiple major modifications to suit our spatiotemporal forecasting setting. Whereas the original LDM was designed for 2D image synthesis, inpainting, and related computer vision tasks, we generalize the learning task from 2D pixels to 3D voxel grids and recast the temporal prediction task as conditional probabilistic generation in the latent space (Figure 1).

The encoder $\mathcal{E}$ maps the initial physical state $s \in \mathbb{R}^{2\times N_1\times N_2\times N_3}$ into a compact 1D latent vector $z = \mathcal{E}(s) \in \mathbb{R}^c$ via a sequence of 3D convolutional layers [26–28], residual blocks [29] with ELU activation [30], and circular padding to respect periodic boundaries. Downsampling is performed using strided convolutions, progressively increasing channel depth while reducing spatial resolution, followed by an average global pooling to produce a 1D fixed-size latent representation. The decoder $\mathcal{D}$ starts from a learned projection of the latent vector into a low-resolution 3D feature map, and then successively applies upsampling, residual blocks, and a final convolution to recover the voxel field $\tilde{s} = \mathcal{D}(z)$.

The diffusion process happens in the latent space. To not confuse diffusion timestep t with physical time τ, we denote forward and inverse diffusion steps as subscripts. In the forward diffusion process, Gaussian noise is gradually added to the sampled latents $\mathbf{z}_0$ in T steps with linear variance schedule $\beta_{1:T}$, producing an array of corrupted samples $\mathbf{z}_{0:T}$:

$$q(\mathbf{z}_t|\mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \sqrt{1-\beta_t}\mathbf{z}_{t-1}, \beta_t \mathbf{I}), \quad q(\mathbf{z}_{1:T}|\mathbf{z}_0) = \prod_{t=1}^{T} q(\mathbf{z}_t|\mathbf{z}_{t-1}). \quad \text{(S9)}$$

The corrupted samples gradually lose their distinguishable features as diffusion step increases and approach Gaussian distribution. To recover the samples from the Gaussian, our goal is to learn a denoising predictor p_θ that approximates the conditional probabilities in the reverse diffusion process starting from a Gaussian at $p(\mathbf{z}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$:

$$p_\theta(\mathbf{z}_{0:T}) = p(\mathbf{z}_T) \prod_{t=1}^{T} p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t), \quad p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{z}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{z}_t, t)). \quad \text{(S10)}$$

The learning objective [9, 31] is to train a noise predictor $\boldsymbol{\epsilon}_\theta$ for the corrupted sample $\mathbf{z}_t = \mathcal{E}(\mathbf{s}_t)$ on the variation bound resembling denoising score matching [32]

$$\mathcal{L}_{\text{LDM}} := \mathbb{E}_{t\sim[0,T],\mathcal{E}(\mathbf{s}_0),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\mathbf{z}_t, t\right)\|^2\right] \tag{S11}$$

with step uniformly sampled $t \sim [1, T]$. As the latents have been compressed into 1D vectors, our denoising network $\boldsymbol{\epsilon}_\theta$ adopts a simple multi-layer perceptron (MLP) conditioned by time embeddings with sinusoidal positional encoding. In this study, we use 32 dimensions for the latent embeddings, and 128 dimensions for the diffusion time embeddings. The number of diffusion steps is set as 1,000.

Conditional generation. To recast a generative model into an autoregressive model, we condition the denoising network ϵ_θ on the latent representation of the current state $\mathbf{z}(\tau) = \mathcal{E}(\mathbf{s}(\tau))$ to reconstruct the next state $\mathbf{z}(\tau + \Delta\tau)$. Formally, it is achieved by concatenating the current latent state $\mathbf{z}_0(\tau)$ in the input:

$$\mathcal{L}_{\text{LDM}} := \mathbb{E}_{t,\mathbf{z}_t(\tau),\mathbf{z}_t(\tau+\Delta\tau),\boldsymbol{\epsilon}}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\mathbf{z}_t(\tau+\Delta\tau), t, \mathbf{z}_0(\tau)\right)\|^2\right], \tag{S12}$$

where we denote physical time as τ and diffusion step as t here. At the inference time, the model generates the next state by denoising from Gaussian distribution on the condition of current state $\mathbf{s}(\tau)$.

Loss and regularization. To ensure that the statistical distribution of the generated electron density matches the ground truth, we incorporate a smooth, differentiable version of the Jensen-Shannon Divergence (JSD) as a loss term. Standard histograms are non-differentiable due to their discrete binning process, making them unsuitable for gradient-based optimization. To overcome this, we implement a *soft histogram* where each data point contributes to multiple bins, weighted by a Gaussian kernel based on its proximity to the bin centers. This creates a smooth and differentiable approximation of probability distribution. We compute two such distributions, P for the model's reconstruction and Q for the target data, and then calculate the JSD between them. The JSD is a symmetric and smoothed measure of similarity between two probability distributions, defined as the average of the Kullback-Leibler (KL) divergences from each distribution to their midpoint average, providing a stable and robust loss signal for training.

The process is formalized by first calculating the soft probability distribution P for a set of data points $\{x_i\}$ and bins with centers $\{b_j\}$:

$$\pi_P(j) = \frac{\sum_i \exp\left(-\frac{(\text{clip}(x_i, b_{\min}, b_{\max}) - b_j)^2}{2\sigma^2}\right)}{\sum_k \sum_i \exp\left(-\frac{(\text{clip}(x_i, b_{\min}, b_{\max}) - b_k)^2}{2\sigma^2}\right)}. \tag{S13}$$

Inspired by the previous work on generalized JSD loss [33], our empirical test shows that the scaled JSD by a constant factor $Z = -(1 - \pi_Q)\ln(1 - \pi_Q)$ achieves better alignment between generated and target distributions. Given the distribution P from the reconstructed data and Q from the ground truth data, the scaled JSD loss $\mathcal{L}_{\text{sJSD}}$ is estimated from two probabilistic densities $\pi_P \sim P, \pi_Q \sim Q$:

$$\mathcal{L}_{\text{sJSD}} = \frac{1}{Z} D_{\text{JS}}(P \,\|\, Q) = \frac{1}{2Z}\left[D_{\text{KL}}(P \,\|\, M) + D_{\text{KL}}(Q \,\|\, M)\right], \quad \text{where} \quad M = \frac{P+Q}{2}. \tag{S14}$$

The total training objective adds the additional reconstruction loss from CNN autoencoder $\mathcal{L}_{\text{AE}}$ and the denoising loss from LDM $\mathcal{L}_{\text{LDM}}$:

$$\mathcal{L} = \mathcal{L}_{\text{AE}} + \lambda_1 \mathcal{L}_{\text{LDM}} + \lambda_2 \mathcal{L}_{\text{sJSD}}, \tag{S15}$$

where λ_1 and λ_2 are tunable weights.

B Related Work

Electron density prediction. Early ML approaches to electron density prediction focused on mapping molecular geometries to ground-state densities using kernel methods or feedforward networks. More recent works leverage equivariant graph neural networks (GNNs) [6–8] or symmetry-preserving neural architectures [5, 6] to capture both local and long-range correlations. However, these methods typically predict static, ground-state densities, with limited exploration of dynamical or time-dependent behavior.

Dimensionality reduction. Compressing high-dimensional electron density data into a lower-dimensional latent space could potentially facilitate efficient learning and simulation. The possible techniques range from classical approaches (*e.g.* principal component analysis, proper orthogonal decomposition [34–42], multi-dimensional scaling [43]) to more modern, non-linear approaches (*e.g.* autoencoder [44–52], kernel PCA, Isomap [53]) to manifold learning [54].

Operator learning. Neural operator frameworks, such as the Fourier Neural Operator (FNO) [10], Factorized Fourier Neural Operators [55], DeepONet [56], and among others, have emerged as powerful tools for learning solution operators to partial differential equations (PDEs). Physics-informed operator learning has also been applied to dynamical fields, with methods like DISCO [57] enabling spatiotemporal prediction from sparse observations. Relatedly, Koopman operator theory [58] offers foundation for obtaining the linear embedding for nonlinear dynamical systems. Recently, diffusion models have been also adopted to learn the dynamics of molecular simulations [59] and PDE solutions [60].

Neural ODE/PDE solvers. Neural Ordinary Differential Equation (Neural ODE) and Partial Differential Equation (Neural PDE) solvers leverage deep learning to model governing equations in function space, bypassing the need for explicit discretization. Neural ODEs, introduced by Chen et al. [61], parameterize the derivative of a system with a neural network and integrate it using numerical ODE solvers, enabling adaptive time-stepping and memory-efficient training via adjoint methods. Neural PDE solvers extend this idea to spatially extended systems, often combining neural operators or physics-informed neural networks (PINNs) with domain knowledge to learn solutions across space and time [46, 62, 63]. These approaches are particularly powerful for modeling high-dimensional, nonlinear, or multiscale systems where traditional solvers may be computationally expensive, offering mesh-free generalization and the ability to learn from sparse or noisy data.

C Structure factor $S(q)$ calculation

We compute the complex structure factor directly from the electron density field $\rho(\mathbf{r})$ defined on a uniform grid with spacings $\Delta x, \Delta y, \Delta z$ and voxel volume $\Delta V = \Delta x\, \Delta y\, \Delta z$. Using the FFT convention, the forward and inverse relations are

$$F(\mathbf{q}) \;=\; \sum_{\mathbf{r}} \rho(\mathbf{r})\, e^{-i\mathbf{q}\cdot\mathbf{r}}\, \Delta V, \qquad \rho(\mathbf{r}) \;=\; \frac{1}{V} \sum_{\mathbf{q}} F(\mathbf{q})\, e^{+i\mathbf{q}\cdot\mathbf{r}}, \tag{S16}$$

where $V = N_x N_y N_z\, \Delta V$ is the total volume. In our implementation we compute $F(\mathbf{q})$ via an FFT of $\rho(\mathbf{r})$ and multiply by ΔV to be consistent with the continuous-transform normalization. We then form the *amplitude* spectrum (not intensity) by taking the complex modulus,

$$|F(\mathbf{q})| \;=\; \big|\mathcal{F}\{\rho(\mathbf{r})\}\big|. \tag{S17}$$

To obtain a rotationally invariant one-dimensional curve, we spherically average this amplitude over shells of constant $q = |\mathbf{q}|$:

$$S(q) \;\equiv\; \langle |F(\mathbf{q})| \rangle_{|\mathbf{q}|=q} \;=\; \frac{1}{N(q)} \sum_{\mathbf{q}\in[q-\Delta q/2,\, q+\Delta q/2]} |F(\mathbf{q})|, \tag{S18}$$

where $N(q)$ is the number of reciprocal-grid points in the shell. In practice, reciprocal vectors are constructed from FFT frequencies as $q_x = 2\pi n_x/L_x,\;\; q_y = 2\pi n_y/L_y,\;\; q_z = 2\pi n_z/L_z$ with $L_\alpha = N_\alpha \Delta\alpha$. The $\mathbf{q} = \mathbf{0}$ mode is set to zero to remove the mean density contribution.

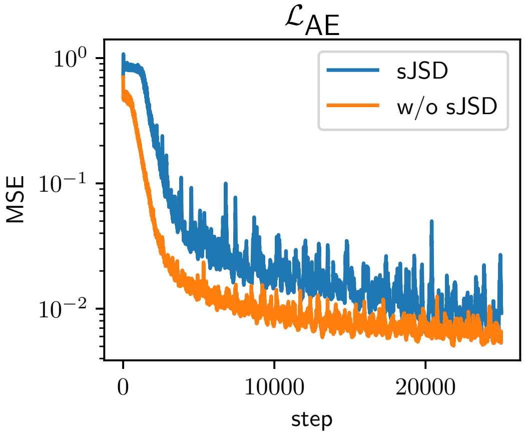

Figure S2: Mean square error (MSE) of autoencoder reconstruction for training set at each optimization step.